UNIVERSITY OF CALIFORNIA
Los Angeles

Machine Learning Frameworks for Data-Driven Personalized Clinical Decision Support and the Clinical Impact

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical and Computer Engineering

by

Changhee Lee

2021

ABSTRACT OF THE DISSERTATION

Machine Learning Frameworks
for Data-Driven Personalized Clinical Decision Support
and the Clinical Impact

by

Changhee Lee
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2021
Professor Mihaela van der Schaar, Chair

Disease progression manifests through a broad spectrum of statically and longitudinally linked clinical features and outcomes. This leads to heterogeneous progression patterns that may vary greatly across individual patients and makes the survival and quality of a patient's life substantially different. Recently, the rapid increase of healthcare databases, such as electronic health records (EHRs) and disease registries, has opened new opportunities for "data-driven" approaches to clinical decision support systems. This dissertation addresses the question of how machine learning (ML) techniques can capitalize on these data resources and provide actionable intelligence to move away from a rules-based clinical care toward a more data-driven and personalized model of care.

To this end, we develop a set of data-driven ML frameworks that can better predict and understand disease progression under two broad clinical setups: (I) the *static setup* where patients' observations are collected at a particular point of time and (II) the *longitudinal setup* where observations of each patient are repeatedly collected over a period of time. In

these setups, we focus on building ML methods that are (i) *accurate* by providing better performance in predicting disease-related outcomes, (ii) *automated* by freeing clinicians from the concern of choosing one particular model for a given dataset at hand, and (iii) *actionable* in a sense that the model is capable of answering “what if” questions and discovering subgroups of patients with similar progression patterns and outcomes.

We highlight the following technical contributions. In the static setting, we present a set of novel ML algorithms for survival analysis, a framework that informs the relationships between the clinical features and the events of interest (such as death, onset of a certain disease, etc.), and predicts what type of event will occur and when it will occur. We start off by developing a deep learning (DL) method that makes no modeling assumptions about the underlying survival process and that flexibly allows for competing events. Then, we propose an automated ML for survival analysis that combines the collective intelligence of different survival models to produce a valid survival function that is both discriminative and well-calibrated. Lastly, we develop a DL model that can accurately estimate heterogeneous treatment effects in survival analysis by adjusting for covariate shifts from multiple sources which makes the problem unique and challenging. In the longitudinal setting, we first develop a DL model for dynamic survival analysis which provides personalized and event-specific survival predictions based on a patient’s heterogeneous and historical context. Then, we provide a novel temporal clustering method that can transform the raw information in the complex longitudinal observations into clinically relevant and interpretable information to recognize future outcomes as well as life-changing disease manifestations which may cause a patient to transit between clusters.

To show the utilities of the proposed models, we evaluate the performance on various real-world medical datasets on breast cancer, prostate cancer, and cystic fibrosis patient cohorts. We demonstrate that the proposed models consistently outperform clinical scores and state-of-the-art ML methods in predicting disease progression, estimating the heterogeneous treatment effects, and providing insights into underlying disease mechanisms.

The dissertation of Changhee Lee is approved.

Gregory J. Pottie

Stanley Osher

William R. Zame

Mihaela van der Schaar, Committee Chair

University of California, Los Angeles

2021

To my parents and my love . . .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I write my acknowledgments to the people who have made my Ph.D. journey possible.

First, I sincerely thank my advisor, Professor Mihaela van der Schaar. She introduced me to the field of machine learning and its applications in healthcare, taught me how to think as a researcher with her unwavering enthusiasm and vision, and supported me with her devoted guidance and insightful research directions. This dissertation would not have been possible without her invaluable support and help. I also thank the other members of my dissertation committee, Professor Greg Pottie, Professor Stan Osher, and Professor William Zame, for their valuable perspective and thoughtful feedback.

A great many persons have helped with the inspiring research discussions, and have furthered my progress by suggestions and criticism. I would like to express gratitude to all of my co-authors and collaborators; Jinsung Yoon, Ahmed Alaa, William Zame, Akash Shah, and Sai Devana at UCLA; Andres Floto, Vincent Gnanapragasam, and Alex Light at the University of Cambridge; Jem Rashbass at Public Health England; Nick Mastronarde at the University at Buffalo. It has been a great privilege to be able to work with such brilliant people. I would also like to thank all of my lab mates; Kartik Ahuja, Onur Atan, Kyeong Ho Moon, Trent Kyono, and Fergus Imrie at UCLA; Ioana Bica at the University of Oxford; Alexis Bellot, James Jordon, Dan Jarrett, Yao Zhang, Zhaozhi Qian, Alihan Hüyük, and Alicia Curth at the University of Cambridge, with whom I had many fruitful discussions.

Finally, I am deeply grateful to my parents, Sang Eun and Young Sim, for the unconditional love and support. I owe the person who I am today to all of the opportunities and experiences they have provided me with. I would like to give my last and most important thanks to my love, Juna, for loving and believing in me throughout all of the ups and downs of this journey. No words can express what your encouragement and support have meant to me.

VITA

2016–2021 Graduate Student Researcher in Electrical and Computer Engineering, University of California, Los Angeles, United States.

2017–2018 Visiting Graduate Student, Engineering Science Department, University of Oxford, United Kingdom.

2019–2019 Vising Researcher, Public Health England, Cambridge, United Kingdom.

PUBLICATIONS

C. Lee, J. Rashbass, M. van der Schaar, "Outcome-Oriented Deep Temporal Phenotyping of Disease Progression," *IEEE Transactions on Biomedical Engineering*, 2021.

C. Lee*, A. Light*, A. Alaa, D. Thurtle, M. van der Schaar, V. J. Gnanapragasam, "Application of a Novel Machine Learning Framework for Predicting Cancer-Specific Mortality: Analysis of 171,942 Men with Non-Metastatic Prostate Cancer from the Surveillance, Epidemiology, and End Results Dataset," *The Lancet Digital Health*, 2021.

C. Lee, J. Yoon and M. van der Schaar, "Dynamic-DeepHit: A Deep Learning Approach for Dynamic Survival Analysis with Competing Risks based on Longitudinal Data," *IEEE Transactions on Biomedical Engineering*, 2019.

C. Lee*, A. Curth*, M. van der Schaar, "SurvITE: Learning Heterogeneous Treatment Effects from Time-to-Event Data," submitted, 2021.

C. Lee, F. Imrie, M. van der Schaar, "Self-Supervision Enhanced Feature Selection with Correlated Gates," submitted, 2021.

C. Lee, M. van der Schaar, "A Variational Information Bottleneck Approach to Multi-Omics Data Integration," *AISTATS 2021.* (oral)

C. Lee, M. van der Schaar, "Temporal Phenotyping using Deep Predictive Clustering of Disease Progression," *ICML 2020.*

C. Lee, W. R. Zame, A. M. Alaa, M. van der Schaar, "Temporal Quilting for Survival Analysis," *AISTATS 2019.*

C. Lee, W. R. Zame, J. Yoon, M. van der Schaar, "DeepHit: A Deep Learning Approach to Survival Analysis with Competing Risks," *AAAI 2018.* (spotlight)

C. Lee, N. Mastronarde, M. van der Schaar, "Estimation of Individual Treatment Effect in Latent Confounder Models via Adversarial Learning," *NeurIPS ML4H 2018.* (spotlight)

S. Devana, A. Shah, **C. Lee**, V. Gudapati, A. Jensen, E. Cheung, C. Solorzano, M. van der Schaar, N. SooHoo, "Development of a Machine Learning Algorithm for Prediction of Complications and Unplanned Readmission following Reverse Total Shoulder Arthroplasty," *Journal of Shoulder and Elbow Arthroplasty*, 2021.

S. Devana, A. Shah, **C. Lee**, A. Roney, M. van der Schaar, N. SooHoo, "A Novel, Potentially Universal Machine- Learning Algorithm to Predict Complications in Total Knee Arthroplasty," *Arthroplasty Today*, 2021.

A. Shah, S. Devana, **C. Lee**, A. Bugarin, E. L. Lord A. N.Shamie D. Y. Park, M. van der Schaar, N. SooHoo, "Prediction of Major Complications and Readmission after Lumbar Spinal Fusion: A Machine Learning-Driven Approach," *World Neurosurgery*, 2021.

A. Shah, S. Devana, **C. Lee**, R. Kianian, M. van der Schaar, N. SooHoo, "Development of a Novel Machine Learning Algorithm for Prediction of Complications After Total Hip Arthroplasty," *The Journal of Arthroplasty*, 2020.

CHAPTER 1

Introduction

Due to the rapid digitization of healthcare, modern clinical data – including electronic health records (EHRs) and disease registries – has become increasingly available [1–3]. This provides extensive opportunities for building clinical decision support systems that move away from a rules-based model of care toward a more data-driven and personalized model of care, allowing clinicians to better understand disease progression, anticipate future health-related outcomes, and design treatment guidelines. However, the availability of large-scale data by itself is not sufficient to build such clinical decision support systems: modern clinical data is often heterogeneous and complex as the progression of diseases – thus, the health-related outcomes – manifests through a broad spectrum of statically- and longitudinally-linked clinical features that may vary greatly across individual patients.

To overcome this, we need to build data-driven machine learning (ML) models that can unravel the heterogeneous progression patterns to better predict health-related outcomes based on patients' traits and to better transform the raw information into more clinically relevant and interpretable information. In this dissertation, we develop novel ML frameworks to achieve such models that can assist clinicians to provide patients a better personalized care and have significant clinical impact.

In the rest of this Chapter, we provide an overview of the type of problems and clinical setups that we focus on throughout the dissertation. Then, we summarize the contributions presented in each of the following chapters. In Chapters 2, 3 and 4, we focus on the static setup for developing ML frameworks for personalized care, whereas in Chapter 5 and 6, we address the longitudinal setup to further unravel the disease progression patterns. In Chapters 7 and 8, we demonstrate the clinical impact of the ML models developed in earlier chapters by applying these models to real-world clinical datasets comprising cohorts of stage 3 breast cancer patients and non-metastatic prostate cancer patients, respectively.

1.1 Machine Learning Frameworks for Personalized Care

We focus on developing ML frameworks towards data-driven and personalized care under the following two broad clinical setups: the *static* setup where patients' observations are collected at a specific point of time and the *longitudinal* setup where observations of a patient are repeatedly collected over a period of time. In these setups, we want the ML models to provide high performance in predicting disease-related outcomes, to free clinicians from the concern of choosing one particular model for a given dataset at hand, and to provide actionable intelligence in the sense that the proposed model is capable of giving answers to "what if" questions or is capable of discovering patient subgroups with similar progression patterns and outcomes.

1.1.1 Modeling Disease Progression for Static Data

In the statistic – also called as cross-sectional – setup, patients' clinical features and health-related outcomes of many diseases are typically encoded in the form of survival data [4]. That is, some patients experience an event of interest (e.g., death due to a particular disease) providing the information about the time-to-event outcomes (i.e., what type of event was occurred and when it occurred), while some patients are right-censored (e.g., lost to follow-up) providing partial information about the event of interest (i.e., the patient had not experienced the event up to the censored time). Survival analysis, which is also known as time-to-event analysis, informs our understanding of the relationships between the (distribution of) time-to-event outcomes and features, and enables us to issue corresponding risk assessments in terms of the probability of an event occurring as a function of time. Due to the right-censoring and the additional dimension (i.e., time) to consider, building ML models for survival analysis is inherently more challenging when compared to conventional classification or regression problems [5].

In this context, we develop novel ML frameworks that address the three criteria mentioned above: high performance, automation, and actionable intelligence. In Chapter 2, we develop a deep learning model that boosts the performance of predicting the time-to-event outcomes. This is achieved by making no modeling assumptions about the underlying time-to-event process and by flexibly allowing for right-censoring and competing events. In Chapter 3, we propose an automated ML model for survival analysis that can free clinicians from the concern of choosing one particular model for a given survival dataset at hand by combining the collective intelligence of different survival models. In Chapter 4, we develop a deep learning model that can accurately estimate heterogeneous treatment effects and give answers to "what if" questions under survival analysis by adjusting for covariate shifts from multiple sources.

1.1.2 Modeling Disease Progression for Longitudinal Data

In the longitudinal setup, patients are followed up over a period of time (e.g., the span of years for patients with chronic diseases) with repeated observations on clinical features (e.g., biomarkers and risk factors) and/or health-related outcomes. This creates the disease progression patterns and related outcomes remarkably varied across individual patients [6, 7]. Information contained in such longitudinal data is of significant importance: capitalizing these observations in their heterogeneous and historical context can improve the risk assessments on the health-related outcomes [8, 9] and can offer an explanation for how the underlying disease progresses [10, 11].

Existing (longitudinal) survival models typically utilize only a small fraction of the available longitudinal (repeated) observations of clinical features. Moreover, these models often make relatively strong assumptions about the underlying stochastic models for the time-to-event process [12] or on both the longitudinal and time-to-event processes [13]. This discards valuable information that has been accrued over time, significantly limiting the utilization of high performing ML models. In Chapter 5, we tackle the problem of improving the performance of predicting health-related outcomes in the longitudinal setup. To this goal, we develop a deep learning model for dynamic survival analysis using a recurrent neural network that can flexibly incorporate longitudinal observations without discarding any information accrued over time, thereby allowing us to make better individualized risk assessments on the time-to-event outcomes.

Transforming the learned information from ML models or the raw information from complex longitudinal observations into clinically relevant and interpretable information is crucial. However, due to the "black-box" nature of ML models and complicated progression patterns of longitudinal observations, clinicians may fail to gain "actionable information" even from a very sophisticated well-performing ML model [10, 14]. In Chapter 6, we address such a challenge by applying temporal clustering which aims at grouping "clinically similar" patients based on how the underlying disease progresses. To do so, we introduce a new notion – i.e., outcome-oriented temporal clustering – to characterize temporal clusters of the underlying disease progression in relevance to the health-related outcomes and to flexibly update cluster assignments as we collect more observations about the underlying disease progression for a patient. By doing so, we ensure that clinicians can leverage temporal clusters as an *actionable tool* to recognize similar past patients (for whom an entire trajectory with an endpoint was already collected) for reasoning about future outcomes as well as life-changing disease manifestations which may cause a patient to transit between clusters.

1.2 Summary of Technical Contributions

In this section, we summarize the technical contributions of each chapter.

1.2.1 Contribution of Chapter 2

In Chapter 2, we develop a deep learning approach to survival analysis that learns the distribution of time-to-event outcomes directly from the data without making any assumption about the form of the underlying stochastic process. This is very different from much of the previous work which has approached the problem by assuming a specific form of the underling survival process and therefore limiting the flexibility of ML models to learn complex interactions between clinical features and the survival outcomes that may present in the available data. In addition, an important aspect of our method is that it smoothly handles situations in which there are multiple competing risks, i.e., settings where there is more than one possible events of interest and observing one event hinders observation of the other event. We employ a network architecture that consists of a single shared sub-network and a family of cause-specific sub-networks that are jointly trained by using novel loss function specifically designed to handle right-censoring and competing risks.

1.2.2 Contribution of Chapter 3

In Chapter 3, we develop an automated ML method that combines the collective intelligence of different underlying survival models to produce a valid survival function that is both discriminative and well-calibrated. This is challenging because the survival models produced by various approaches – ranging from (semi-)parametric to non-parametric – offer different strengths and weaknesses in terms of both discriminative performance and calibration, and their relative performance varies across different datasets and at different time horizons within a single dataset. The core part of our method is an algorithm for configuring the weights sequentially over a grid of time intervals. To render the problem tractable, we apply constrained Bayesian Optimization (BO), which to yields good discriminative performance at different time horizons while providing a valid and well-calibrated survival function.

1.2.3 Contribution of Chapter 4

In Chapter 4, we study the problem of inferring heterogeneous treatment effects from survival data. While both the related problems of (i) estimating treatment effects for *binary or continuous* outcomes and (ii) *predicting survival* outcomes have been well studied in the recent ML literature, their combination – albeit of high practical relevance in healthcare – has received considerably less attention. With the ultimate goal of reliably estimating the effects of treatments on instantaneous risk and survival probabilities, we focus on the problem of learning (discrete-time) treatment-specific conditional hazard functions. We find that unique challenges arise in this context due to a variety of covariate shift issues that go beyond a mere combination of well-studied confounding and censoring biases. We theoretically analyze their effects by adapting recent generalization bounds from domain adaptation and treatment effect estimation to our setting and discuss implications for model design. We use the resulting insights to propose a novel deep learning method for treatment-specific hazard estimation based on balancing representations.

1.2.4 Contribution of Chapter 5

In Chapter 5, we develop a novel deep learning architecture using a recurrent neural network structure that flexibly incorporates the available longitudinal data comprising various repeated measurements (rather than only the last available measurements) to issue dynamically updated survival predictions for one or multiple competing risk(s). Our method learns the time-to-event distributions without the need to make any assumptions about the underlying stochastic models of the longitudinal or the survival processes. Thus, unlike existing works in statistics such as landmarking and joint modeling, our method is able to learn data-driven associations between the longitudinal data and the various associated risks.

1.2.5 Contribution of Chapter 6

In Chapter 6, we develop a deep learning approach for clustering time-series data, where each cluster comprises patients who share similar future outcomes of interest (e.g., adverse events, the onset of comorbidities). Such an outcome-oriented clustering offers actionable information from heterogeneous electronic health records stored in the form of time-series because it can provide (i) patient phenotyping, (ii) anticipating patients' prognoses by identifying "similar" patients, and (iii) designing treatment guidelines that are tailored to homogeneous patient subgroups. To encourage each cluster to have homogeneous future outcomes, the clustering is carried out by learning discrete representations that best describe the future outcome distribution based on novel loss functions.

1.2.6 Contribution of Chapter 7 and Chapter 8

In these Chapters, we present the clinical impact of applying our novel methods in real-world cancer registries. Chapter 7 demonstrates the power of applying the proposed automated ML method for survival analysis to a US population-based cohort of 171,942 men diagnosed with non-metastatic prostate cancer from the prospectively maintained Surveillance, Epidemiology, and End Results Program. By automatically combining optimal attributes from different survival models, it provides better discriminative and calibration performance when compared to commonly used clinical scores and conventional survival models. Chapter 8 shows the impact of applying dynamic survival analysis and the outcome-oriented clustering to a heterogeneous cohort of 11,779 stage III breast cancer patients from the UK National Cancer Registration and Analysis Service. Based on the discovered temporal clusters, we identify the key driving factors that lead to transitions between clusters which can be translated into actionable information to support better clinical decision-making.

Part I

Modeling Disease Progression for Static Data

CHAPTER 2

A Deep Learning Approach to Survival Analysis with Competing Risks

2.1 Introduction

Survival analysis – also called time-to-event analysis – is fundamental in many areas, including economics and finance, engineering and medicine. A long and diverse literature approaches survival analysis by viewing the event of interest as the first hitting time of an underlying stochastic process; i.e. the first time at which the stochastic process reaches a prescribed boundary. Depending on the context, the first hitting time may represent the time until a stock option can profitably be exercised, the time to failure of a mechanical system or the length of time a patient survives following treatment (or non-treatment); see [15] for many other examples. A fundamental problem of survival analysis in all of these areas is to understand the relationship between the (distribution of) hitting times and the covariates, such as the characteristics of the stock on which the option is written, the physical environment in which the mechanical system must operate, and the features of the individual patient. Especially in medical setting, the survival analysis is further applied to discovering risk factors affecting the survival [16], comparison among risks of different subjects at a certain time of interest [17], decision of a cost-efficient sensing period (e.g. screening for cancer) [18].

Most of the previous work in this area has approached the problem by assuming a specific form for the underlying stochastic process, using available data to learn the relationship between the covariates and the parameters of the model, and then deducing the relationship between covariates and the distribution of first hitting times – the *risk* of the event. (In the medical setting, this is typically the risk of death or onset of a certain disease.) The Cox proportional hazards model [19] is the most widely-used model in the medical setting but it makes many strong assumptions about the underlying stochastic process and about the relationship between the covariates and the parameters of that process. Other models allow for various other specific forms of the underlying stochastic process and for more general relationships between covariates and the parameters, but still maintain strong parametric assumptions (especially that the relationship between covariates and parameters of the stochastic process are time-invariant).

This work proposes a very different approach to survival analysis: we construct and use a deep neural network that learns the distribution of first hitting times *directly*. An important aspect of our method, which we call *DeepHit*, is that it smoothly handles situations in which there is a single underlying risk (cause) and situations in which there are multiple *competing risks* (causes). DeepHit employs a network architecture that consists of a single shared sub-network and a family of cause-specific sub-networks. We train the network by using a loss function that exploits both survival times and relative risks. DeepHit makes *no assumptions* about the form of the underlying stochastic process; it therefore allows for the possibility that, even for a fixed cause or causes (e.g. a disease or diseases), both the parameters and the form of the stochastic process *depend on the covariates*.

Although our approach is quite general and applies to all the settings mentioned above, and many others, we focus here on the medical setting (and so we will use medical language, and speak of patients rather than instances, etc.). In the medical context, competing risks are extremely common. (For example, patients suffering from a particular disease, such as cancer, frequently have co-morbidities, such as cardiovascular disease.) With the exception of the Fine-Gray model [20], existing work on survival analysis either cannot be applied or is inadequate in the presence of competing risks except under the assumption that the risks are independent, which is very seldom the case. (To refer to the same example: studies [16] have shown that various treatments for breast cancer increase the risk of a cardiovascular event; the risks are not at all independent.) Survival analysis with competing risks is a challenging problem, and made all the more important because the choice of treatment must take account of these competing risks. We note that right-censoring of data is extremely common in the medical setting: patients are frequently lost to follow-up (often for unknown reasons).[1]

We are not the first to apply neural networks to time-to-event analysis; for example, [21–23] have employed neural networks for modeling non-linear representations for the relation between covariates and the risk of a clinical event. However, these studies have maintained the basic assumptions of the Cox model, weakening only the assumption of the form of the relationship between covariates and the hazard rate. In particular, the time-dependent influence of covariates on time-to-event cannot be addressed by these models.

To demonstrate the usefulness of our approach, we compare its predictive performance with that of competing approaches using three medical datasets and one synthetic dataset. For all these datasets, we compare the performance of DeepHit with previous state-of-the art competing methods, using as the metric of performance the time-dependent concordance index C^{td} [24]. (C^{td} measures the extent to which the ordering of actual survival times of pairs agrees with the ordering of their predicted risk; it is the most-widely-used metric for evaluating the performance of survival models [25].) DeepHit provides large and statistically significant performance improvements over previous state-of-the-art methods. (Detailed descriptions of these datasets, the competing methods, and the performance comparisons are presented in the following sections.)

[1] Throughout this chapter, we follow the literature and assume that right-censoring occurs completely at random.

2.2 Related Work

The survival model most widely used in the statistical and medical research literature is the Kaplan-Meier estimator [26], which has the advantage of being able to learn very flexible survival curves, but the disadvantage of not incorporating patients' covariates. Hence it is useful at the population level but not useful at the individual level. As we have noted already the Cox proportional hazard model [19] (CPH) is capable of incorporating patients' covariates, but assumes that the hazard rate is constant and that the log of the hazard rate is a linear function of covariates. Other models make different assumptions about the underlying stochastic processes and about the relationship between the covariates and the parameters of the assumed process. For instance [27, 28] assume a Wiener process, while [29] assumes a Markov Chain; see [27] for other examples and discussion of the literature. An advantage of these models is that, because they formulate survival analysis as the problem of determining the distribution of the first time at which the prescribed stochastic process hits a prescribed boundary, they are able to incorporate competing risks. The disadvantage of these models is that they are tied to the specific form of stochastic process that they assume. Put differently: the models are of limited use unless we have already learned the underlying stochastic process. In the medical setting this means learning the underlying disease process, which would seem to be an even more complicated problem than survival analysis itself – especially since the states of the disease or diseases are typically hidden and not directly observable. An alternative to this family of models is the one offered by [20], which modifies the traditional proportional hazard model by direct transformation of the cumulative incidence function, but the Fine-Gray model is also severely limited by strong assumptions on the form of the hazard rates and on the way in which the parameters depend on covariates.

The problem of survival analysis has also received substantial recent attention in the machine learning literature. Recently developed survival models include random survival forests [30], deep exponential families [31], dependent logistic regressors [32], and semi-parametric Bayesian models based on Gaussian processes [33]. All of these methods are capable of incorporating the individual patient's covariates, but none of them has considered the problem of competing risks, and none of them seems readily adaptable to this problem. (In principle, these models could be applied to the problem of competing risks by fixing a single event and simply treating all other events right-censoring, but this approach is inadequate unless the competing risks are independent, which is frequently not the case.). Recently, deep multi-task Gaussian process was used to develop a nonparametric Bayesian model for survival analysis with competing risks [34] while still relying on assumption that the latent stochastic process follows Gaussian process.

[21] represents the first application of neural networks to survival analysis. In contrast to the standard CPH model, this work uses a feed-forward network to *learn* the relationship of the covariates to the hazard function. More recently, [22] and [23] have followed the same general approach, although using more sophisticated network architectures and loss functions. These works have improved on the CPH model by relaxing the specific functional relationship between covariates and the hazard function in the standard CPH model while maintaining the other central assumption– that the hazard rate is constant over time. As a result, these works do not fully exploit the potential capacity of deep neural networks to learn complex representations of risk and in particular to capture the time-dependent influence of covariates on survival.

DeepHit improves on existing models because it suffers from none of the difficulties identified above. Because DeepHit learns the (joint) distribution of survival times and events directly, it avoids the problems inherent in assuming a particular form for the underlying stochastic process or a particular form for the relationship of covariates to the underlying stochastic process or any kind of time-invariance. As we shall see, the performance of DeepHit improves dramatically on the performance of previous models in the setting of competing risks and significantly even in the (simpler) setting of a single risk.

2.2.1 Survival Data

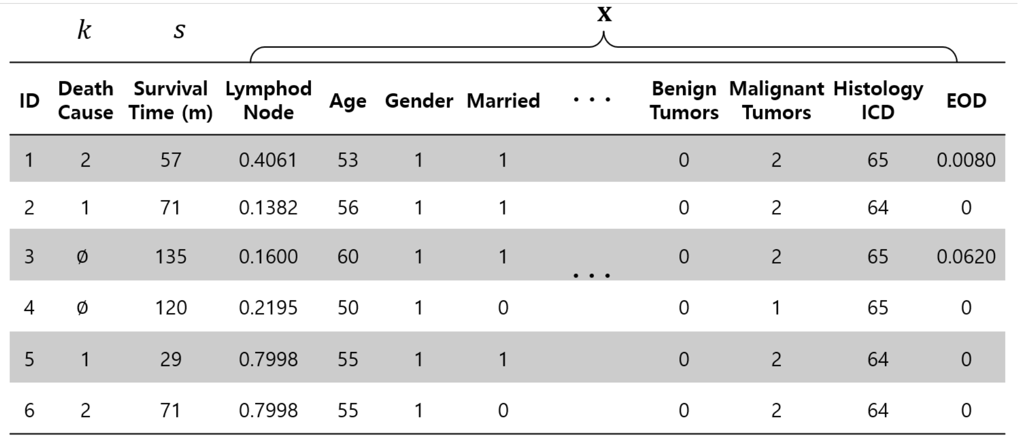
Survival data provides three pieces of information for each instance/patient: 1) observed covariates, 2) time elapsed since covariates were first collected, and 3) a label indicating the type of event (e.g. adverse clinical event or death) that occurred.[2] We treat survival time as discrete and the time horizon as finite (e.g. no patients lived longer than 100 years) so the time set is $\mathcal{T} = \{0, \ldots, T_{max}\}$ for a predefined maximum time horizon T_{max}. We consider $K \geq 1$ possible events of interest; we assume that at exactly one event eventually occurs for each instance/patient (e.g. a patient eventually dies, but can die from only one cause [35]).[3] Because events of interest are not always observed (e.g. patients may be lost to follow-up), survival data are frequently right-censored; handling this difficulty will be a crucial aspect of the analysis. We indicate right-censoring as the "event" $\varnothing$ and therefore represent the set of possible events – including right-censoring – as $\mathcal{K} = \{\varnothing, 1, \cdots, K\}$. Each data point/instance (e.g. patient history) is therefore a triple $(\mathbf{x}, s, k)$ where $\mathbf{x} \in X$ is a D-dimensional vector of covariates, $s \in \mathcal{T}$ is the time at which the (unique) event or censoring occurred, and $k \in \mathcal{K}$ is the event or censoring that occurred at time s. Note that s is either the time at which an event (death) occurred or the time at which the patient was censored (disappeared from follow-up), but in either case the patient was known to be alive at times prior to s. We are given a dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, s^{(i)}, k^{(i)})\}_{i=1}^{N}$ that describe a finite set of observed instances/patients.

Figure 2.1 illustrates survival data of the SEER dataset (see Experiment section for more details) for 6 patients and two possible events (causes of death); patient 2 and 5 died from cause 1, patient 1 and 6 died from cause 2; patient 3 and 4 were lost to follow-up (right-censored).

For each tuple $(\mathbf{x}^*, s^*, k^*)$ with $k^* \neq \varnothing$, we are interested in the true probability $P(s =$

[2] We use medical terms for convenience but we emphasize that our framework and results are quite general.

[3] We leave for later work the more complicated setting in which several events – e.g. the onsets of various diseases – might occur.



	k	s	$\mathbf{x}$								
ID	Death Cause	Survival Time (m)	Lymphod Node	Age	Gender	Married	· · ·	Benign Tumors	Malignant Tumors	Histology ICD	EOD
1	2	57	0.4061	53	1	1		0	2	65	0.0080
2	1	71	0.1382	56	1	1		0	2	64	0
3	∅	135	0.1600	60	1	1	· · ·	0	2	65	0.0620
4	∅	120	0.2195	50	1	0		0	1	65	0
5	1	29	0.7998	55	1	1		0	2	64	0
6	2	71	0.7998	55	1	0		0	2	64	0

Figure 2.1: An illustration of survival data (the SEER dataset).

$s^*, k = k^* | \mathbf{x} = \mathbf{x}^*$); i.e. the true *ex-ante* probability that a (new) patient with covariates $\mathbf{x}^*$ will experience the event k^* at time s^*. Of course the true probability cannot be known on the basis of any finite dataset, so our task is to find *estimates* $\hat{P}$ of the true probabilities.

2.3 Model: DeepHit

In this Section we describe our formal model.

2.3.1 Model Description

Our goal is to train the network to learn $\hat{P}$, the estimate of the joint distribution of the first hitting time and competing events. As illustrated in Figure 2.2, DeepHit is a multi-task network [36] which consists of a shared sub-network and K cause-specific sub-networks. Our architecture, differs from that of conventional multi-task network in two ways. First, we utilize a single softmax layer as the output layer of DeepHit in order to ensure that the network learns the *joint distribution* of K competing events not the *marginal distributions* of each event. Second, we maintain a residual connection [37] from the input covariates into the input of each cause-specific sub-network.

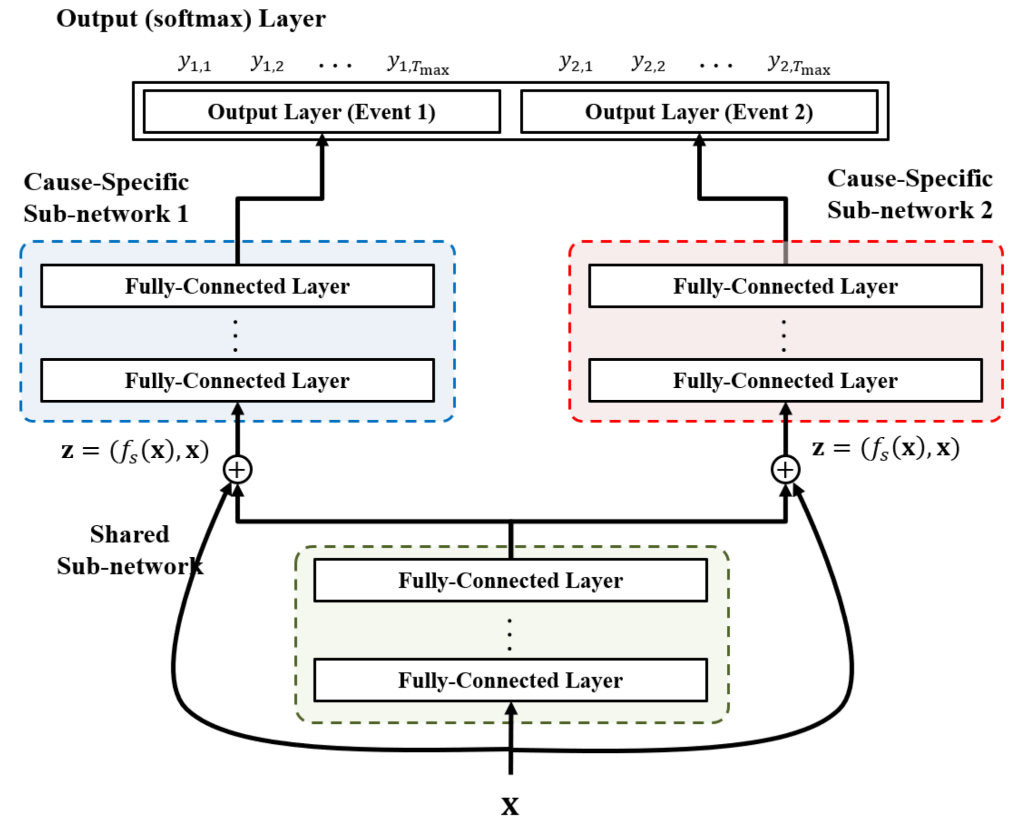


Figure 2.2: The architecture of DeepHit with two competing events.

The shared sub-network and the k-th cause-specific sub-network for $k = 1, \cdots, K$ are comprised of L_S and $L_{C,k}$ fully-connected layers, respectively. The shared sub-network takes as inputs the clinical covariates $\mathbf{x}$ and produces as output a vector $f_s(\mathbf{x})$ that captures the (latent) representation that is common to the K competing events.

Each cause-specific sub-network takes as inputs the pairs $\mathbf{z} = (f_s(\mathbf{x}), \mathbf{x})$ and produces as output a vector $f_{c_k}(\mathbf{z})$, which corresponds to the probability of the first hitting time of a specific cause k. More specifically, the inputs to the sub-networks include *both* the output of the shared network *and* the original covariates; this gives the sub-networks access to the learned common representation $f_s(\mathbf{x})$ while still allowing them to learn non-common part of the representation as well. (If only the learned common representation were used as an input to the sub-networks, the non-common part of the representation would be lost.) The totality of these outputs is a joint probability distribution on the first hitting time and event so the cause-specific sub-networks are learning the distribution for the first hitting time for each cause in parallel. The output of the softmax layer is a probability distribution $\mathbf{y} = [y_{1,1}, \cdots, y_{1,T_{\max}}, \cdots, y_{K,1}, \cdots, y_{K,T_{\max}}]$: given a patient with covariates $\mathbf{x}$, an output element $y_{k,s}$ is the (estimated) probability $\hat{P}(s, k|\mathbf{x})$ that the patient will experience the event k at time s. This architecture drives the network to learn potentially non-linear, even non-proportional, relationships between covariates and risks.

The *(cause-specific) cumulative incidence function* (CIF) expresses the probability that a particular event $k^* \in \mathcal{K}$ occurs on or before time t^* conditional on covariates $\mathbf{x}^*$; as in the Fine-Gray model [20], understanding the CIF is key to the analysis of survival under competing risks. By definition, the CIF for the event k^* is:

$$F_{k^*}(t^*|\mathbf{x}^*) = P(s \leq t^*, k = k^*|\mathbf{x} = \mathbf{x}^*) = \sum_{s^*=0}^{t^*} P(s = s^*, k = k^*|\mathbf{x} = \mathbf{x}^*). \tag{2.1}$$

However, since the *true* CIF, $F_{k^*}(s^*|\mathbf{x}^*)$, is not known, we utilize the *estimated* CIF, $\hat{F}_{k^*}(s^*|\mathbf{x}^*) = \sum_{m=0}^{s^*} y^*_{k,m}$, in order to compare the risk of event occurring and to assess how models discriminate across cause-specific risks among patients.

2.3.2 Loss Function

To train DeepHit, we minimize a total loss function $\mathcal{L}_{\text{Total}}$ that is specifically designed to handle censored data. This loss function is the sum of two terms $\mathcal{L}_{\text{Total}} = \mathcal{L}_1 + \mathcal{L}_2$; $\mathcal{L}_1$ is the log-likelihood of the joint distribution of the first hitting time and event; $\mathcal{L}_2$ incorporates a combination of cause-specific ranking loss functions.

$\mathcal{L}_1$ is the log-likelihood of the joint distribution of the first hitting time and corresponding event, modified to take account of the right-censoring of the data [15] considering K competing risks. For patients who are not censored, it captures both the event that has occurred and the time at which the event has occurred; for patients who are censored, it captures the time at which the patient is censored (lost to follow-up) which provides the information that the

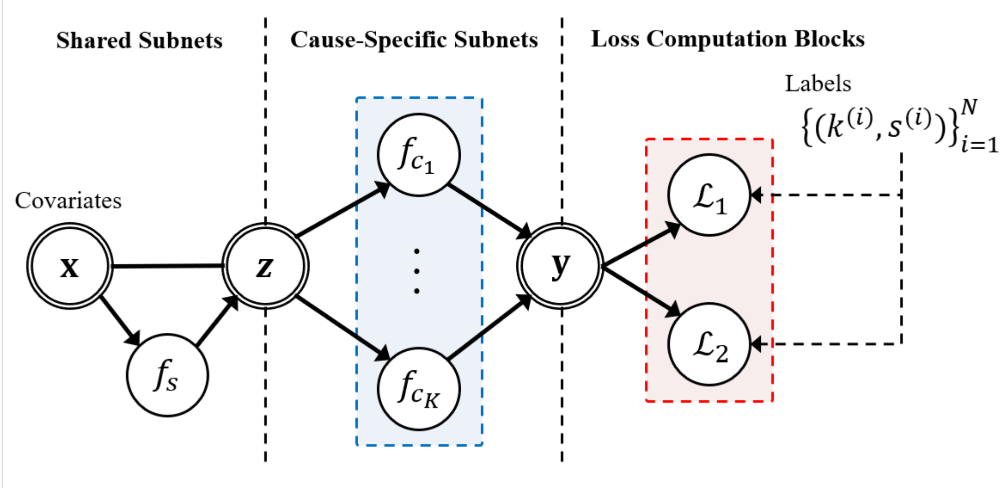


Figure 2.3: An illustration of a computational graph to compute the training loss of DeepHit.

patient was alive up to that time. We define $\mathcal{L}_1$ by

$$\mathcal{L}_1 = -\sum_{i=1}^{N}\left[\mathbb{1}(k^{(i)} \neq \varnothing)\cdot\log\left(y^{(i)}_{k^{(i)},s^{(i)}}\right) + \mathbb{1}(k^{(i)} = \varnothing)\cdot\log\left(1-\sum_{k=1}^{K}\hat{F}_k(s^{(i)}|\mathbf{x}^{(i)})\right)\right], \qquad (2.2)$$

where $\mathbb{1}(\cdot)$ is an indicator function. The first term captures the information provided by uncensored patients; the second term captures the censoring bias by exploiting the knowledge that they are alive at the censoring time, so that that the first hitting event will occur among one of the K causes *after* the given censoring time); see [38].

$\mathcal{L}_1$ drives DeepHit to learn the general representation for the joint distribution of the first hitting time and events; $\mathcal{L}_2$ incorporates estimated CIFs calculated at different times (i.e. the time at which an event actualy occurs) in order to fine-tune the network to each cause-specific estimated CIF. To do so, we utilize a ranking loss function which adapts the idea of concordance [25]: a patient who dies at time s should have a higher risk at time s than a patient who survived longer than s. Write $A_{k,i,j} \triangleq \mathbb{1}(k^{(i)} = k, s^{(i)} < s^{(j)})$ for the indicator function of pairs (i, j) who experience risk k at different time, and whose risks for event k can therefore be directly compared; we call these pairs *acceptable for event k*. Now define

$$\mathcal{L}_2 = \sum_{k=1}^{K}\alpha_k \cdot \sum_{i\neq j} A_{k,i,j}\cdot\eta\left(\hat{F}_k(s^{(i)}|\mathbf{x}^{(i)}), \hat{F}_k(s^{(i)}|\mathbf{x}^{(j)})\right) \qquad (2.3)$$

where the coefficients α_k are chosen to trade off ranking losses of the k-th competing event, and $\eta(x, y)$ is a convex loss function. For convenience, we assume here that the coefficients α_k are all equal (i.e. $\alpha_k = \alpha$ for $k = 1, \cdots, K$ and some α to be chosen), and we use the loss function $\eta(x, y) = \exp(\frac{-(x-y)}{\sigma})$. Incorporating $\mathcal{L}_2$ into the total loss function penalizes incorrect ordering of pairs (with respect to each event) and so minimizing the total loss encourages *correct ordering* of pairs (with respect to each event).

In Figure 2.3, we illustrate a computational graph to compute the training loss of the proposed network: the inputs are the covariates $\mathbf{x}$ and the output is the vector $\mathbf{y}$. Double-circled nodes imply inputs or outputs of DeepHit or those of sub-networks, and single-circled nodes indicate calculation blocks (e.g. sub-networks or loss functions). In training stage, the network exploits $\{k^{(i)}, s^{(i)}\}_{i=1}^{N}$ in order to calculate the indicator functions, to find acceptable pairs, and, hence, to compute the loss function corresponding to input covariates. Based on this computational graph, we can obtain the gradient on the nodes (including hidden nodes of all the sub-networks) and parameters for training the proposed network.

2.4 Experiments

The prognostic performance of DeepHit was evaluated by comparing it with the performance of conventional benchmarks in analyzing three real-world clinical datasets and one synthetic dataset. We give brief descriptions of the datasets below; we take 30 days = 1 month as the basic time interval.

2.4.1 Dataset Description

UNOS. The United Network for Organ Sharing (UNOS) database[4] consists of patients who underwent heart transplantation in the period 1985-2015. Of the total of 60,400 patients who

[4] https://www.unos.org/data/

received heart transplants, 29,436 patients (48.7%) were followed until death; the remaining 30,964 patients (51.3%) were right-censored. We used a total of 50 features (30 recipient-relevant, 9 donor-relevant and 11 donor-recipient compatibility). For details on selected features and pre-processing methods, see to [39].

METABRIC. The Molecular Taxonomy of Breast Cancer International Consortium dataset contains gene expression profiles and clinical features used to determine breast cancer subgroups. Of the total of 1,981 patients in the dataset, 888 patients (44.8%) were followed until death; the remaining 1,093 patients (55.2%) were right-censored. We restricted attention to 21 publicly available clinical features including tumor size, number of positive lymph nodes, etc.; for details see [40]. Missing values were replaced by the mean value for real-valued features and by the mode for categorical features. One-hot encoding was applied for categorical features.

SEER. The Surveillance, Epidemiology, and End Results Program (SEER)[5] dataset provides information on breast cancer patients during the years 1992-2007. Among the 72,809 patients, we focused on 68,325 patients who died due to breast cancer or cardiovascular disease (CVD), or who were right-censored. (So we have two competing risks.) We have 23 patient features, including age, race, gender, morphology information, diagnostic information, therapy information, tumor size, tumor type, etc. Missing values were replaced by mean value for real-valued features and by the mode for categorical features.

SYNTHETIC. We also created a synthetic dataset with two competing risks, in the spirit of [34]. To do this we constructed two stochastic processes with parameters and the hitting times described as follows:

$$\begin{aligned}
&\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \mathbf{x}_3^{(i)} \sim \mathcal{N}(0, \mathbf{I}) \\
&T_1^{(i)} \sim \exp\left((\gamma_3^T \mathbf{x}_3^{(i)})^2 + \gamma_1^T \mathbf{x}_1^{(i)}\right) \\
&T_2^{(i)} \sim \exp\left((\gamma_3^T \mathbf{x}_3^{(i)})^2 + \gamma_2^T \mathbf{x}_2^{(i)}\right)
\end{aligned} \tag{2.4}$$

[5] https://seer.cancer.gov/causespecific/

where $\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \mathbf{x}_3^{(i)})$ is the vector of clinical covariates for patient i and consists of three 4-dimensional variables: for $k = 1, 2$, the covariates $\mathbf{x}_k$ only have an effect on the hitting time for event k while $\mathbf{x}_3$ has an effect on the hitting times of both events. Note that we assume hitting times are exponentially distributed with a mean parameter depending on both linear and non-linear (quadratic) function of covariates. For convenience, we set $\gamma_1 = \gamma_2 = \gamma_3 = 10$. Given the parameters, we first produced $30,000$ patients; among those, we randomly selected $15,000$ patients (50%) to be right-censored at a time $s_c^{(i)}$ randomly drawn from the uniform distribution on the interval $[0, \min\{T_1^{(i)}, T_2^{(i)}\}]$. (This censoring fraction was chosen to be roughly the same censoring fraction as in the real datasets, and hence to present the same difficulty as found in those datasets.) The data for each patient i is therefore $(\mathbf{x}^{(i)}, s^{(i)}, k^{(i)})$ where $s^{(i)} = \min\{T_1^{(i)}, T_2^{(i)}\}$ and $k^{(i)} = \arg\min T_k^{(i)}$ for patients who were not censored and $s^{(i)} = s_c^{(i)}$ and $k^{(i)} = \varnothing$ for patients who were censored.

2.4.2 Benchmarks

Single Risk. The performance of DeepHit was compared with two families of other survival models[6]. The first of these families consists of conventional survival regression models: including Cox Proportional Hazards (**Cox**) [41], Threshold Regression (**ThresReg**) [15], and Random Survival Forests (**RSF**) with # of trees = 100 [42]. The second consists of survival models which are derived from mortality prediction performed by machine learning algorithms: Random Forest (**MP-RForest**), Logistic Regression (**MP-LogitR**), and AdaBoost (**MP-AdaBoost**). We train these conventional ML algorithms to predict new labels which indiates whether a patient is dead or alive over different time horizon m where $m = 0, \cdots, T_{\max}$. The last consists of the cutting-edge deep neural network (**DeepSurv**), which is developed upon Cox proportional assumption [22][7].

[6] We did not compare with [23] because that paper did not provide detailed information to permit implementation.

[7] https://github.com/jaredleekatzman/DeepSurv

Competing Risks. The performance of DeepHit was compared with survival models under competing risks: the Fine-Gray proportional sub-distribution hazards model (**Fine-Gray**) [20], deep multi-task Gaussian process (**DMGP**) [34], and with a cause-specific versions of the single risk survival models – e.g., the cause-specific Cox Proportional Hazards Model (*cs*-**Cox**) – that were created by fixing an event (e.g. death from CVD) and treating the other event (e.g. death from breast cancer) simply as a form of censoring; see [43].

2.4.3 Experimental Setting

For evaluation, we applied 5-fold cross validation: we randomly separated the data into training set (80%) and testing set (20%). We reserved 20% of the training set as a validation set. (In all of these sets, we maintained a constant ratio of patients who experienced each event and patients who were censored.) The hyper-parameters for $\mathcal{L}_{\text{Total}}$, including α and σ, were selected based on the discriminative performance on the validation set. Early stopping was performed based on the total loss. DeepHit is a 4-layer network consisting of 1 fully-connected layer for the shared sub-network and 2 fully-connected layers for each cause-specific sub-network and a softmax layer as the output layer. (Note that if there is a single event, this reduces to 3 fully-connected layers and a softmax layer as the output layer.) For hidden layers, the number of nodes were set as 3, 5, and 3 times of the covariate dimension for the layer 1, 2, and 3, respectively, with ReLu activation function. The network was trained by back-propagation via Adam optimizer with a batch size of 50 and a learning rate of 10^{-4}. Dropout probability of 0.6 and Xavier initialization was applied for all the layers (DeepHit was implemented in a Tensorflow environment).

2.4.4 Discriminative Performance

2.4.4.1 Performance Metric

As our metric of performance, we use the *time-dependent concordance index* (C^{td}-index) [24]. (Recall that the ordinary concordance index (C-index) [25] is a widely used discriminative index based on the assumption that patients who lived longer should have been assigned a lower risk than patients who lived less long. However the ordinary C-index is computed only at the initial time of observation and hence cannot reflect the possible change in risk over time. The time-dependent concordance index takes time into account.) Given the estimated CIF in Eq. (2.1), the C^{td}-index for event k is defined as

$$C^{td} = P\left(\hat{F}_k(s^{(i)}|\mathbf{x}^{(i)}) > \hat{F}_k(s^{(i)}|\mathbf{x}^{(j)})|s^{(i)} < s^{(j)}\right) \approx \frac{\sum_{i\neq j} A_{k,i,j} \cdot \mathbb{1}\left(\hat{F}_k(s^{(i)}|\mathbf{x}^{(i)}) > \hat{F}_k(s^{(i)}|\mathbf{x}^{(j)})\right)}{\sum_{i\neq j} A_{k,i,j}} \tag{2.5}$$

where, as before, $A_{k,i,j}$ is the indicator function for a pair (i, j) to be acceptable for an event k and the approximation comes from the empirical definition. Thus, the C^{td}-index for event k is derived from comparison of pairs in which one patient has experienced event k at a particular time while the other has not experienced any event nor been censored by that time. Because this discriminative index does not depend on a single fixed time, it provides an appropriate assessment for situations in which the influence of covariates on survival varies over time (in other words, risks are non-proportional over time). (Note that the C^{td}-index is equivalent to the usual C-index of [25] in the case of a single event and a survival model for which the proportional hazards assumption holds.)

2.4.4.2 Competing Events/Competing Risks

Comparisons of the performance of DeepHit with other models for the SEER and the SYNTHETIC datasets are shown in Table 2.1 and 2.2, respectively. In the SEER dataset, there are two events – competing risks: death from cardiovascular disease (CVD) and from

Breast Cancer. As can be seen, DeepHit provides performance improvements over other models; with the exception of *cs*-Cox for death by CVD, the performance improvements were all statistically significant ($p < 0.05$ and often $p < 0.001$). The comparisons for death by breast cancer are particularly striking. Fine-Gray and *cs*-Cox both perform poorly with respect to the risk of breast cancer, while DeepHit performs much better. Because Fine-Gray and *cs*-Cox assume linear proportional hazards and DMGP model assumes the underlying stochastic process to follow Gaussian process, while DeepHit makes no such assumption, the performance comparison strongly suggests that non-proportional and/or non-linear relationships between covariates and survival times is crucial for assessing the risk of breast cancer.

Table 2.1: Comparison of cause-specific C^{td}-index performance (mean ± 95% confidence interval) for the SEER dataset.

Algorithms	**CVD**	**Breast Cancer**
cs-Cox	0.672±0.008	0.639±0.006*
cs-RSF	0.280±0.018*	0.584±0.010*
cs-ThresReg	0.664±0.007‡	0.645±0.017*
cs-MP-RForest	0.281±0.018*	0.584±0.010*
cs-MP-AdaBoost	0.671±0.006	0.741±0.006‡
cs-MP-LogitR	0.665±0.020	0.657±0.009*
Fine-Gray	0.663±0.007‡	0.639±0.007*
DMGP	0.657±0.025	0.742±0.004‡
DeepHit ($\alpha = 0$)	0.674±0.013	0.736±0.003
DeepHit	**0.684±0.010**	**0.752±0.004**

$*, \ddagger$ indicate p-value < 0.001, < 0.05

We also compared the discriminative performance of DeepHit with that of Fine-Gray and *cs*-Cox on the SYNTHETIC dataset where there are again two events/competing risks: death from Event 1 and from Event 2. As can be seen in Table 2.2, DeepHit outperformed all the benchmarks and the performance improvements were all statistically significant ($p < 0.001$). This is expected since the *cs*-Cox and Fine-Gray restrict the relationship between covariates and risks to be linear. Thus, they are not able to capture the quadratic relationship introduced when generating the synthetic data. However, DeepHit allows the network to learn the representation of the non-linear relation of covariates.

Table 2.2: Comparison of cause-specific C^{td}-index performance (mean ± 95% confidence interval) for the SYNTHETIC dataset

Algorithms	**Event 1**	**Event 2**
cs-Cox	0.578±0.008*	0.588±0.004*
cs-RSF	0.669±0.005*	0.657±0.005*
cs-ThresReg	0.579±0.005*	0.588±0.003*
cs-MP-RForest	0.620±0.009*	0.610±0.007*
cs-MP-AdaBoost	0.607±0.007*	0.607±0.006*
cs-MP-LogitR	0.579±0.007*	0.586±0.003*
Fine-Gray	0.579±0.007*	0.589±0.004*
DMGP	0.663±0.005*	0.666±0.006*
DeepHit ($\alpha = 0$)	0.739±0.004	0.737±0.005
DeepHit	**0.755±0.006**	**0.755±0.007**

$*$ indicates p-value < 0.001

2.4.4.3 Single Event/Single Risk

As we have noted in the Introduction, an important aspect of DeepHit is that it smoothly handles competing risks. However, it also provides improved performance when there is only a single risk. To show this, we compared the performance of DeepHit with other models for the UNOS and METABRIC (single event) datasets in Table 2.3. As can be seen, DeepHit consistently provided the best performance for both the UNOS and METABRIC datasets. For the UNOS dataset, the improvement of DeepHit over all the competing methods other than AdaBoost was highly statistically significant ($p < 0.01$ and often $p < 0.001$). For the METABRIC data set, the improvement of DeepHit over all the competing methods other than RSF was statistically significant ($p < 0.001$, $p < 0.05$, and often $p < 0.01$).

We suspect that for the single risk setting, the performance improvement of DeepHit comes from its capacity to capture the complicated relationship between covariates and risk, especially in the presence of many covariates. Because the other models make restrictive parametric assumptions, they are unable to capture this complicated relationship. In particular, when compared with DeepSurv, we suspect the performance improvement comes from not relying on the proportional assumption.

2.5 Conclusion

This chapter presents a novel approach, DeepHit, to the analysis of survival data. DeepHit trains a neural network to learn the estimated joint distribution of of survival time and event, while capturing the right-censored nature inherent in survival data. We train the network by using a loss function that exploits both survival times and relative risks. As a test, we compared the performance of DeepHit with the performance of previous models. In settings with competing risks, the performance of DeepHit is much better than that of previous models; even in settings with a single risk the performance of DeepHit is significantly better than that of previous models.

Table 2.3: Comparison of cause-specific C^{td}-index performance (mean ± 95% confidence interval) for single event datasets.

Algorithms	**Datasets**	
	UNOS	METABRIC
Cox	$0.566{\pm}0.003^{*}$	$0.648{\pm}0.014^{\dagger}$
RSF	$0.575{\pm}0.004^{\dagger}$	0.672±0.017
ThresReg	$0.571{\pm}0.003^{*}$	$0.649{\pm}0.016^{\dagger}$
MP-RForest	$0.552{\pm}0.004^{*}$	$0.650{\pm}0.020^{\dagger}$
MP-AdaBoost	0.582±0.004	$0.633{\pm}0.016^{*}$
MP-LogitR	$0.571{\pm}0.004^{*}$	$0.661{\pm}0.016^{\ddagger}$
DeepSurv	$0.563{\pm}0.008^{*}$	$0.648{\pm}0.012^{\dagger}$
DeepHit ($\alpha = 0$)	0.573±0.002	0.646±0.012
DeepHit	**0.589±0.003**	**0.691±0.012**

$*, \dagger, \ddagger$ indicate p-value < 0.001, < 0.01, < 0.05

CHAPTER 3

Automated Machine Learning Approach to Survival Analysis

3.1 Introduction

Survival analysis (time-to-event analysis) plays an important role in many disciplines and especially in medicine. The importance of survival analysis has prompted the development of a variety of approaches to model the survival function (the probability of surviving past a given time as a function of the covariates). Parametric and semi-parametric approaches construct models that rely on specific assumptions about the true underlying distribution; non-parametric approaches take a more agnostic point of view to construct models that rely on (variants of) familiar machine learning methods. The models produced by these various approaches offer different strengths and weaknesses in terms of both discriminative performance and calibration, and their relative performance varies across different datasets and at different time horizons within a single dataset. In particular, no single model is best across all datasets, and frequently no single model is best across all time horizons within a single dataset. This presents a challenge to familiar methods of model selection or ensemble creation. An additional challenge is that survival analysis needs to yield good performance at different time horizons while providing a valid and well-calibrated survival function; this makes the conventional model selection or ensemble methods actually inapplicable.

The usefulness of a survival model should be assessed both by how well the model *discriminates* among predicted risks and by how well the model is *calibrated*. The necessity of

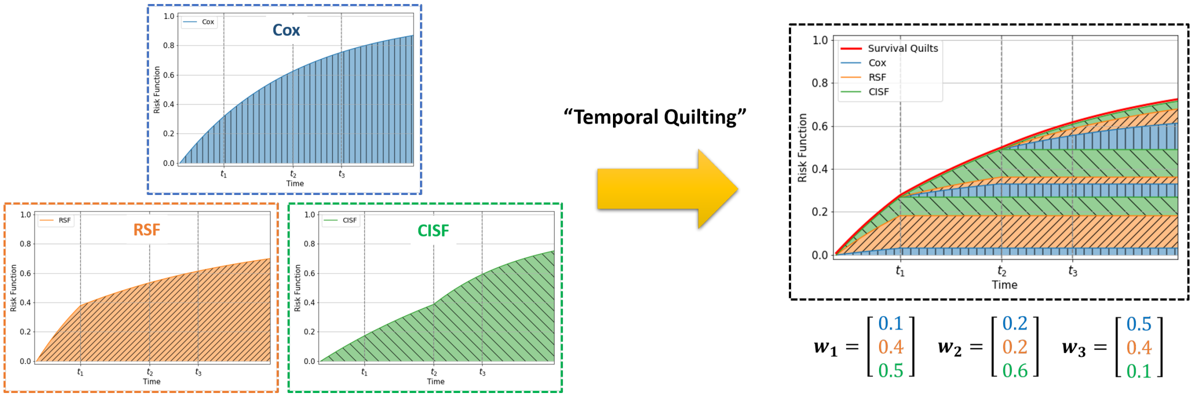


Figure 3.1: A toy example of temporal quilting with prescribed weights for survival models in $\mathcal{M} = \{\text{Cox}, \text{RSF}, \text{CISF}\}$ at t_1, t_2, and t_3. A risk function is constructed by stitching together the weighted increment functions of each survival model between two adjacent time horizons.

keeping both criteria in mind is illustrated by the case of heart transplantation, which is the treatment of last resort for patients with end-stage heart failure. Successful transplantation can mean many additional years of life for such patients, but there are many more patients in need of transplants than there are available donor hearts. So, it is important to correctly discriminate/prioritize recipients on the basis of risk. However, if the risk predictions of a given model are not well calibrated to the truth – i.e. if there is poor agreement between predicted and observed outcomes – then the model will have little prognostic value for clinicians.

This work offers a novel approach that addresses these challenges. Our approach combines the collective intelligence of different underlying survival models to produce a valid survival function that is both discriminative and well-calibrated. Because we piece together these underlying models according to (endogenously determined) weights that vary over time, we refer to our construction as *temporal quilting*, and to the resultant model as a *Survival Quilt*. An illustration of temporal quilting (for given weights) is provided in Figure 3.1. The core part of our method is an algorithm for configuring the weights sequentially over a (perhaps very fine) grid of time intervals. To render the problem tractable, we apply constrained Bayesian

Optimization (BO) [44], which models the discrimination and calibration performance metrics as black-box functions, whose input is an array of weights (over different time horizons) and whose output is the corresponding performance achieved. Based on the constructed array of weights, our method makes a single predictive model – a Survival Quilt.

Our empirical results demonstrate that Survival Quilts provide significant performance gains over the underlying models (which we take as benchmarks) on a variety of real-world survival datasets. Because our approach automatically finds (an approximation to) the best temporal quilting of the underlying survival models, it provides a way to free clinicians from the concern of choosing one particular survival model for each dataset and for each time horizon of interest.

3.2 Related Work

Different approaches, ranging from statistical methods to machine learning based methods, have been proposed for survival analysis. One approach employs (semi-)parametric models that are constructed on the basis of assumptions on the true underlying distribution. This includes i) survival models based on the Cox proportional hazard (Cox-PH) assumption [19], and a variety of extensions [22, 45, 46] and ii) the accelerated failure time (AFT) model based on the Weibull distribution, and extensions [34, 47]. Other approaches employ nonparametric models, including i) ensembles of survival trees constructed via bagging [30, 48] or boosting [49], and ii) deep learning methods [50]. In general, nonparametric models provide better survival predictions than do (semi-)parametric models when the true underlying distribution is unknown or is mis-specified. However, nonparametric models often yield inaccurate predictions at time horizons for which the number of subjects in the dataset who are "at risk" is small [5].

As we have noted in the Introduction, methods based on model selection and ensemble creation that are familiar for classification problems (including the Auto-ML framework

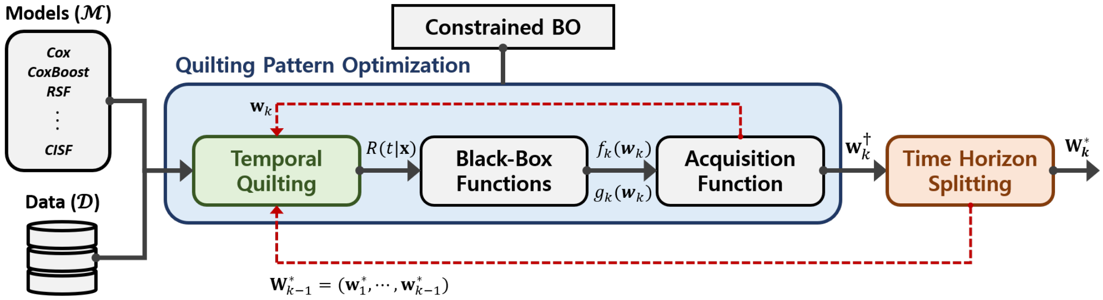


Figure 3.2: A schematic depiction of Survival Quilts and its pattern optimization at step k. Survival Quilts provide risk functions that are constructed on the basis of the final quilting pattern $\mathbf{W}_K^*$. Here, colored boxes are the three main components of our method and dotted lines imply feedback loops for sequential computations.

[51, 52]) do not extend to the survival setting because we need to construct a valid survival function that provides good discriminative performance at different time horizons and is also well-calibrated. Our work is most closely related to a model based on stacking [53], which estimates an optimally weighted combination of different survival models on the basis of calibration performance. However, in order to produce a valid survival function, that model requires the weights to be independent of time. By contrast, our approach exploits weights that depend on time to provide a valid survival function that is well-calibrated and achieves superior discriminative performance at different time horizons. To the best of our knowledge, this work is the first that combines different survival models in a time-dependent manner to provide both discriminative and prognostic power.

3.3 Problem Formulation

For convenience, we couch our description in the medical setting, although our approach is entirely applicable to any time-to-event problem. In our setting, some patients experience the event of interest (e.g. death) and some are censored (lost to follow-up). The data for an individual patient i therefore consists of a vector of covariates $\mathbf{x}_i \in \mathcal{X}$ ($\mathcal{X}$ is the space of all covariates), either a time-to-event, $T_i \in \mathbb{R}^+$, or a time-to-censoring, $C_i \in \mathbb{R}^+$ (both from the initial moment of observation), and an indicator $\Delta_i = I(T_i < C_i)$; $\Delta = 1$ if the patient experienced the event of interest and $\Delta = 0$ if the patient was right-censored. Note that censoring provides the information that the patient had *not* experienced the event (e.g. was alive) up to time C_i. We are given data for N patients so the entire time-to-event dataset is $\mathcal{D} = \{(\mathbf{x}_i, \Delta_i T_i + (1 - \Delta_i) C_i, \Delta_i)\}_{i=1}^N$.

Our goal is to estimate the *risk function* $R : \mathcal{X} \times \mathbb{R}^+ \to [0, 1]$

$$R(t|\mathbf{x}) = \mathbb{P}(T \leq t|\mathbf{x}), \tag{3.1}$$

which is the probability of the event occurring at or before time t given the covariates $\mathbf{x}$. (Equivalently, we could estimate the *survival function* $S : \mathcal{X} \times \mathbb{R}^+ \to [0, 1]$; $S(t|\mathbf{x}) = \mathbb{P}(T > t|\mathbf{x}) = 1 - R(t|\mathbf{x})$ is the probability of the event occurring after time t, given covariates $\mathbf{x}$.)

Since we aim at finding the best predictive model among the set of all models that provide well-calibrated risk functions, it is natural to formulate the optimization problem as maximizing discriminative performance subject to a constraint on calibration. If we write $\mathcal{R}$ for the set of all risk functions, $f(\cdot)$ for a metric of discriminative performance, and $g(\cdot)$ as a metric of calibration, then our problem is to find the risk function $R^* \in \mathcal{R}$ that solves the following maximization problem:

$$\begin{aligned} \max_{R \in \mathcal{R}} \quad & f(R) \\ \text{s.t.} \quad & g(R) \leq c, \end{aligned} \tag{3.2}$$

where $c > 0$ is some prescribed tolerance of predictive error. (In the experiments reported below, we take f and g to be the time-dependent C-index and Brier Score, respectively, but other metrics could be used.)

3.4 Method: Survival Quilts

As noted in the introduction, the existing survival models may fail to capture the true survival behavior in different settings and over different time horizons. (See also the discussion in Section 3.5.) Survival Quilts address both these failings by forming *time-varying* ensembles of different survival models.

Table 3.1: List of survival models used in Survival Quilts

Cox-PH model	**AFT model**	**Survival Forest**
Cox	Weibull	RSF
CoxRidge	LogNormal	CISF
CoxBoost	Exponential	

Given a time-to-event dataset $\mathcal{D}$ and a set of survival models $\mathcal{M}$ e.g., Cox, Weibull, RSF, and etc., (a full list of survival models used in this work is provided in Table 3.1), our method outputs a predictive model – a *Survival Quilt* – that provides a valid risk function. A Survival Quilt is constructed endogenously from the data following three steps. The first step is *temporal quilting* which constructs valid risk functions for a *given* array of weights (a *quilting pattern*) for survival models in $\mathcal{M}$ over time horizons. The second step models the performance of these risk functions as black-box functions and applies constrained BO to (approximately) optimize the quilting pattern. The final step splits the time horizons in order to insure robustness of the (approximately optimized) quilting pattern. A schematic overview of our method is illustrated in Figure 3.2; details of each of these steps are described in the following subsections.

3.4.1 Temporal Quilting: Constructing a New Risk Function

Constructing a survival model entails learning a risk function that spans a continuum of time horizons. We do not treat predictions at each time horizon as separate problems, but rather provide a natural construct for the entire risk function; risk predictions at past time horizons are carried forward to future time horizons to provide a consistent risk function. More specifically, given an increasing sequence of time horizons $\mathcal{T} = \{t_0 = 0, t_1, \cdots, t_K\}$, we first break down the risk functions provided by each survival model in $\mathcal{M}$ into *pieces* by focusing on the increment between two adjacent time horizons, t_{k-1} and t_k for $k = 1, \cdots, K$. We then assemble the pieces in a *quilting pattern* that, on each time interval, assigns weights to each of the increment functions of the underlying survival models and then sums the weighted combination of the increment functions over time.

We define the *increment function* of model $m \in \mathcal{M}$ on the interval $[a, b]$, given covariate $\mathbf{x}$, to be

$$i_m(a, b|\mathbf{x}) = R_m(b|\mathbf{x}) - R_m(a|\mathbf{x}), \tag{3.3}$$

where R_m is the risk function issued by model $m \in \mathcal{M}$. Because R_m is non-decreasing on the interval $[a, b]$, i_m is non-decreasing and non-negative on the interval $[a, b]$. Let $\mathbf{w}$ be a $|\mathcal{M}|$-dimensional *weight vector*, where $\mathbf{w}[m] \in [0, 1]$ indicates the weight for model m and $\sum_{m\in\mathcal{M}} \mathbf{w}[m] = 1$. Given $\mathbf{w}$, the *weighted increment function* on the interval $[a, b]$ is defined to be

$$I_{\mathbf{w}}(a, b|\mathbf{x}) = \sum_{m\in\mathcal{M}} \mathbf{w}[m] \cdot i_m(a, b|\mathbf{x}). \tag{3.4}$$

Then, given weights $\mathbf{w}_1, \ldots, \mathbf{w}_k$ and a time $t \in [t_{k-1}, t_k]$, we set

$$R_0(t|\mathbf{x}) = \sum_{\ell=1}^{k-1} I_{\mathbf{w}_\ell}(t_{\ell-1}, t_\ell|\mathbf{x}) + I_{\mathbf{w}_k}(t_{k-1}, t|\mathbf{x}), \tag{3.5}$$

where the first term is the aggregate risk up to time t_{k-1} and the second term is the *incremental* from time t_{k-1} to time $t \in [t_{k-1}, t_k]$. Now, we define the *risk function* at time t to be

$$R(t|\mathbf{x}) = \min\big\{1, R_0(t|\mathbf{x})\big\}. \tag{3.6}$$

A few words of explanation may be useful.

- Note that $R_m(0|\mathbf{x}) = 0$ (patients are alive at the beginning of the observation period) so that $i_m(0, t|\mathbf{x}) = R_m(t|\mathbf{x})$. Hence, if $t \in [0, t_1]$, then $R(t|\mathbf{x}) = \sum_{m \in \mathcal{M}} \mathbf{w}_1[m] \cdot R_m(t|\mathbf{x})$.
- R_0 might exceed 1, in which case it could not be a valid risk function. (The probability that the event has occurred cannot exceed 1.) Hence, we truncate by setting $R = \min\{1, R_0\}$.
- Because the weighted increment functions are non-decreasing and non-negative, the functions R_0, R are also non-decreasing – hence R is a valid risk function.

We frequently refer to the array of weights $\mathbf{W}_K = (\mathbf{w}_1, \cdots, \mathbf{w}_K)$ as a *quilting pattern*; we refer to the construction above as *temporal quilting*. Figure 1 illustrates a quilting pattern and the resulting risk function constructed via temporal quilting.

3.4.2 Quilting Pattern Optimization via BO

Let $\mathbf{W}_K = (\mathbf{w}_1, \cdots, \mathbf{w}_K)$ be a quilting patterns (configuration of weights); write $\mathbf{W}_k = (\mathbf{w}_1, \cdots, \mathbf{w}_k)$ for the configuration up to time t. Our approach is to find the best risk function R that can be formed as in (3.6). Because R is completely defined by the configuration of weights, this amounts to finding the best quilting pattern $\mathbf{W}_K^*$ – i.e., the quilting pattern that solves the following maximization problem:

$$\begin{aligned} \max_{\mathbf{W}_K} \quad & f(\mathbf{W}_K) \\ \text{s.t.} \quad & g(\mathbf{W}_K) \leq c, \end{aligned} \tag{3.7}$$

where $c > 0$ is the prescribed tolerance of predictive error. In (3.7), we take the function f to be the average of functions f_k that are the metric of time-dependent discriminative performance at t_k (see the definition below in (3.11)); similarly we take the function g to be the average of functions g_k that are the metric of time-dependent calibration performance at t_k (see the definition below in (3.12)). Formally, $f(\mathbf{W}_K) = \frac{1}{K}\sum_{k=1}^{K} f_k(\mathbf{W}_k)$ and

$g(\mathbf{W}_K) = \frac{1}{K}\sum_{k=1}^{K} g_k(\mathbf{W}_k)$. Since the objective and constraint functions in (3.7) have no analytic form, we treat them as black-box functions $f, g : [0,1]^{K\times|\mathcal{M}|} \to \mathbb{R}$. That is, given a quilting pattern $\mathbf{W}_K$, we can only evaluate the noisy versions of f and g which are given by $\frac{1}{J}\sum_{j=1}^{J} \mathcal{L}_f(\mathbf{W}_K; \mathcal{D}_{\text{tr}}^{(j)}, \mathcal{D}_{\text{va}}^{(j)})$ and $\frac{1}{J}\sum_{j=1}^{J} \mathcal{L}_g(\mathbf{W}_K; \mathcal{D}_{\text{tr}}^{(j)}, \mathcal{D}_{\text{va}}^{(j)})$, respectively. Here, $\mathcal{L}_f$ and $\mathcal{L}_g$ are the empirical values for the given performance metrics f and g, respectively, and $\mathcal{D}_{\text{tr}}^{(j)}$ and $\mathcal{D}_{\text{va}}^{(j)}$ denote training and validation splits of $\mathcal{D}$ in the j-th fold of J-fold cross-validation.

To search for the optimal quilting pattern $\mathbf{W}_K^*$, we use Bayesian optimization (BO) and solve a black-box optimization problem under a black-box constraint [44]. The BO algorithm specifies a Gaussian process (GP) prior on f and g as

$$\begin{aligned} f &\sim \mathcal{GP}(\mu_f(\mathbf{W}_K), \kappa_f(\mathbf{W}_K, \mathbf{W}'_K)) \\ g &\sim \mathcal{GP}(\mu_g(\mathbf{W}_K), \kappa_g(\mathbf{W}_K, \mathbf{W}'_K)) \end{aligned} \tag{3.8}$$

where $\mu_f(\mathbf{W}_K)$ and $\mu_g(\mathbf{W}_K)$ are the mean functions, encoding the expected performance of different quilting patterns, and $\kappa_f(\mathbf{W}_K, \mathbf{W}'_K)$ and $\kappa_g(\mathbf{W}_K, \mathbf{W}'_K)$ are the covariance kernels [54], encoding the similarity between different quilting patterns for f and g, respectively. We refer to the optimization problem in (3.7) as the Quilting Pattern Composition Problem (QPCP).

3.4.3 Sequential BO for QPCP

The functions f, g are defined over a space of dimension $D = K \times |\mathcal{M}|$. Note that D is large even for relatively small sets $\mathcal{M}$ of underlying survival models and a relatively coarse grid of time horizons; e.g. $D = 80$ if $|\mathcal{M}| = 8$ (as in our experiments) and $K = 10$. (In practice, it seems desirable to allow the grid of time horizons to be much finer than this; e.g. if the most distant horizon is 10 years we might want the grid to consist of 40 quarters or 120 months or perhaps even something finer. Moreover, although here we use only eight underlying models, it might well be desirable to use many more models – and one of the virtues of our approach is that this is possible.) This high-dimensionality renders standard

GP-based BO infeasible because both the sample complexity of nonparametric estimation of the functions f, g and the computational complexity of maximizing the acquisition function are exponential in D [52, 55]. For these reasons, we propose instead a sequential greedy algorithm that incrementally selects a time horizon and performs constrained BO on that time horizon.

Let $\mathbf{W}^*_{k-1} = (\mathbf{w}^*_1, \cdots, \mathbf{w}^*_{k-1})$ be the configuration of weights found through step $k-1$ (i.e., the time horizon t_{k-1}). Following the greedy approach, we find the weights at step k (i.e., the time horizon t_k) by solving the following BO:

$$\begin{aligned} \max_{\mathbf{w}_k} \quad & f_k(\mathbf{w}_k; \mathbf{W}^*_{k-1}) \\ \text{s.t.} \quad & g_k(\mathbf{w}_k; \mathbf{W}^*_{k-1}) \leq c, \end{aligned} \tag{3.9}$$

where we have written $\mathbf{w}_k; \mathbf{W}^*_{k-1}$ as shorthand for $(\mathbf{w}^*_1, \cdots, \mathbf{w}^*_{k-1}, \mathbf{w}_k)$. We have chosen this notation to emphasize that $\mathbf{W}^*_{k-1}$ is fixed so f_k, g_k depend only on $\mathbf{w}_k$. BO specifies GP priors on f_k and g_k as $f_k \sim \mathcal{GP}(\mu_{f_k}(\mathbf{w}_k; \mathbf{W}^*_{k-1}), \kappa_{f_k}(\mathbf{w}_k, \mathbf{w}'_k; \mathbf{W}^*_{k-1}))$ and $g_k \sim \mathcal{GP}(\mu_{g_k}(\mathbf{w}_k; \mathbf{W}^*_{k-1}), \kappa_{g_k}(\mathbf{w}_k, \mathbf{w}'_k; \mathbf{W}^*_{k-1}))$. From this point forward we simplify notation by omitting the dependence on $\mathbf{W}^*_{k-1}$.

3.4.3.1 Black-box Constrained BO

At step k, to solve the black-box constrained BO in (3.9), we approximate the problem by an augmented Lagrangian framework as proposed in [56]. In particular, (3.9) can be relaxed to minimizing the augmented Lagrangian problem given by

$$L(\mathbf{w}_k; \lambda, \rho) = -f_k(\mathbf{w}_k) + \lambda \cdot (g_k(\mathbf{w}_k) - c) + \frac{1}{\rho} \max\left(0, g_k(\mathbf{w}_k) - c\right)^2, \tag{3.10}$$

where $\rho > 0$ and $\lambda \geq 0$ indicate a penalty parameter and a Lagrange multiplier, respectively.

An efficient algorithm in [57] transforms the original constrained problem into a sequence of subproblems: at the n-th subproblem, we find a weight vector at t_k, which is denoted as $\mathbf{w}_k^{(n)}$, by solving (3.10) given $\rho^{(n-1)}$ and $\lambda^{(n-1)}$. After finding a candidate solution $\mathbf{w}_k^{*(n)}$, the

Algorithm 1 Augmented Lagrangian optimization

```
**Initialize:** $\lambda^{(0)} \geq 0$, $\rho^{(0)} > 0$, and $\mathbf{w}_k^{(0)}$
**for** $n = 1, 2, \cdots, n_{\max}$ **do**
    Find $\mathbf{w}_k^{*(n)}$ that approximately solve (3.10)
    Update $\lambda^{(n)} \leftarrow \max\Big(0, \lambda^{(n-1)} + \frac{1}{\rho^{(n-1)}}(g_k(\mathbf{w}_k^{*(n)}) - c)\Big)$
    Update $\mathbf{w}_k^{\dagger} \leftarrow \mathbf{w}_k^{*(n)}$
    **if** $g_k(\mathbf{w}_k^{\dagger}) \leq c$ **then**
        Update $\rho^{(n)} \leftarrow \rho^{(n-1)}$
    **else**
        Update $\rho^{(n)} \leftarrow \frac{1}{2}\rho^{(n-1)}$
    **end if**
**end for**
```

penalty parameter and approximate Lagrange multipliers are updated and the process repeats until termination conditions are satisfied. We denote the final output of the constrained BO at step k by $\mathbf{w}_k^{\dagger}$. (Throughout the experiments, we set the terminal condition to be satisfaction of the constraint by $\mathbf{w}_k^{\dagger}$ or n reaching the maximum number of subproblems $n_{\max}$.) Algorithm 1 gives the specific updates utilized in this work.

3.4.3.2 Endogenous Time Horizon Splitting

In principle, we could always use $\mathbf{w}_k^{\dagger}$ to extend the sequence of weights. However, doing so would make the construction fragile because the optimal weights might become over-fitted. In order to make the construction more robust, we introduce a required margin of improvement $\delta > 0$; if using $\mathbf{w}_k^{\dagger}$ to extend the sequence of weights leads to an improvement in discriminative performance of at least δ, we set $\mathbf{w}_k^* = \mathbf{w}_k^{\dagger}$; otherwise we set $\mathbf{w}_k^* = \mathbf{w}_{k-1}^*$. In the former case, t_k represents an endogenously learned split in the time horizon – a time when the quilting pattern changes. The overall process of our method is illustrated in Algorithm 2.

Computational Complexity. By following the greedy sequential approach, we have

Algorithm 2 Sequential BO for QPCP

```
**Initialize:** $\mathbf{W}_0^* = \varnothing$, $\delta > 0$, and $\Delta t > 0$
**for** $k = 1, 2, \cdots, K$ **do**
    Set $t_k \leftarrow t_{k-1} + \Delta t$
    Obtain $\mathbf{w}_k^\dagger$ from **Algorithm 1** with $\mathbf{W}_{k-1}^*$ and $t_k$
    **if** $f_k(\mathbf{w}_k^\dagger) - f_k(\mathbf{w}_{k-1}^*) > \delta$ **then**
        Update $\mathbf{w}_k^* \leftarrow \mathbf{w}_k^\dagger$
    **else**
        Update $\mathbf{w}_k^* \leftarrow \mathbf{w}_{k-1}^*$
    **end if**
    Set $\mathbf{W}_k^* \leftarrow (\mathbf{w}_1^*, \mathbf{w}_2^*, \cdots, \mathbf{w}_k^*)$
**end for**
```

side-stepped the main challenge of scaling BO to the high dimensionality [55] by reducing the number of computations to maximize the acquisition function from $\mathcal{O}(n^{K\times|\mathcal{M}|})$ to $\mathcal{O}(K \times n^{|\mathcal{M}|})$. To quantify the computational complexity of training Survival Quilts which can be carried out *off-line*, we first denote the computational complexity of the overall quilting pattern optimization and that of training the m-th baseline survival model as $\mathcal{C}_{\text{BO}}$ and $\mathcal{C}_m$ where $m \in \mathcal{M}$, respectively. Then, the computational complexity of training Survival Quilts can be given as $\mathcal{C}_{\text{BO}} + J\sum_{m\in\mathcal{M}}\mathcal{C}_m$. (Recall that J is the number of cross-validations.) Albeit the increased complexity in the training due to the optimization of quilting pattern, the computational complexity of Survival Quilts for prediction – which must be carried out *on-line* – is bounded by the sum of the computational complexity of the baseline survival models in $\mathcal{M}$ for predicting the risk.

3.5 Experiments

In this section, we present discriminative performance results in comparison to competitive baseline algorithms on six real-world time-to-event datasets. We set $K = 50$ and $\Delta t = \frac{T_{\max}}{K}$

Table 3.2: Descriptive statistics on the six real-world datasets. Mean (standard deviation) times in days are provided for the time-to-event/censoring.

Statistics	**Datasets**					
	MAGGIC	**SUPPORT**	**METABRIC**	**UNOS-I**	**UNOS-II**	**BPD**
No. Patients	5000	9105	1981	792	5000	2510
Events	1827 (36.5%)	6201 (68.1%)	888 (44.8%)	363 (45.8%)	2395 (47.9%)	1999 (79.6%)
Censored	3173 (63.5%)	2904 (31.9%)	1093 (55.2%)	429 (54.2%)	2605 (52.1%)	511 (20.4%)
Time-to-Event	885.3 (957.0)	206.0 (321.9)	2318.5 (1613.8)	141.0 (213.9)	2161.3 (2084.0)	613.5 (853.0)
Time-to-Censoring	927.7 (1032.4)	1060.3 (516.1)	3464.8 (1773.7)	327.6 (380.5)	2733.1 (2151.8)	1331.8 (1407.9)
No. Features	33	42	21	16	50	48

where $T_{\max}$ indicates the maximum of time-to-event and time-to-censoring in each dataset. Throughout the evaluation, we report results using the average of 5 random 80/20 train/test splits.

3.5.1 Experimental Setup

3.5.1.1 Survival Models

The survival models that are used for constructing Survival Quilts and for the comparisons are listed below along with description of the implementations used to compute them: the standard Cox-PH model (**Cox**) [19] and the modification with ridge regression (taking $\alpha = 1$) (**CoxRidge**) are implemented with Python package `scikit-surv`; the survival regression models using the Weibull (**Weibull**), Log-normal (**LogNormal**) and Exponential (**Exponential**) distributions are implemented with R package `survival`; the Cox-PH model with the component-wise likelihood-based boosting algorithm [46] (**CoxBoost**) is implemented with R package `CoxBoost` with 500 iterations; the bagging-based Random Survival Forest [30] (**RSF**) is implemented with the R package `RandomForestSRC` with 1000 trees; and the Conditional Inference Survival Forest [48] (**CISF**) is implemented with the R package `pec` with 1000 trees.

3.5.1.2 Performance Metrics

As discussed above, we assess the predictions of all the survival models with respect to how well the predictions discriminate among individual risks and how accurate the predictions are. As the metric of discriminative power, we use the time-dependent concordance index (C-index) [58], defined by

$$C(t) = \mathbb{P}(R(t|\mathbf{x}_i) > R(t|\mathbf{x}_j) | \Delta_i = 1, T_i \leq t, T_i < T_j). \tag{3.11}$$

As the metric of calibration, we use the Brier Score (BS) [59] which is the mean square error adjusted for the survival setting:

$$BS(t) = \mathbb{E}\left[(\mathbb{1}(T_i \leq t) - R(t|\mathbf{x}_i))^2 \right]. \tag{3.12}$$

These metrics can be evaluated over different time horizons and are adjusted for censoring as defined in [58] and [59].

3.5.2 Datasets

We conducted experiments to investigate the performance of Survival Quilts on six real-world medical datasets from a variety of clinical settings: a preventive care database on chronic heart failure (**MAGGIC**) [60], a study to understand seriously ill hospitalized adults (**SUPPORT**) [61], a study on breast cancer subgroups (**METABRIC**) [62], databases on heart transplant management for patients (**UNOS-I**) wait-listed for transplantation and on patients who underwent a heart transplant (**UNOS-II**)[1], and preventative care records on bipolar disorder (**BPD**) [63]. In Table 3.2, we provide a summary of these time-to-event datasets.

[1] Available at https://www.unos.org/data/

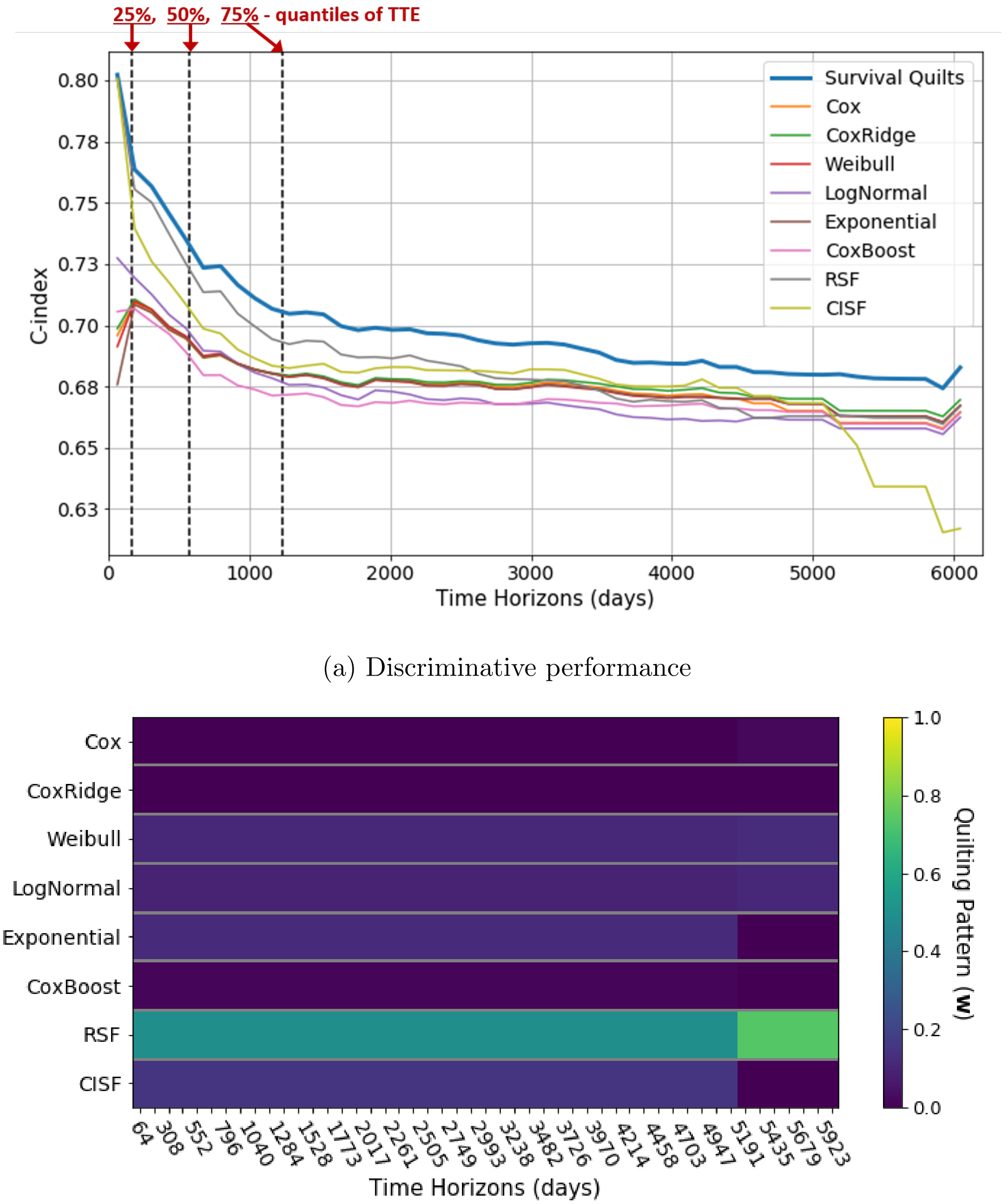
3.5.3 Performance Evaluation

In Tables 3.3 - 3.5, we report the discriminative performance of the various survival models for the MAGGIC, SUPPORT, and METABRIC datasets at three different time horizons, representing the 25%, 50%, and 75%-quantiles of time-to-event. We emphasize that the time horizons used for testing are different from the time horizons that are used in the construction of Survival Quilts, so we are not prejudicing the evaluations in our favor.

Overall, several things are important to note: i) the best performing benchmarks are *different* across the datasets and time horizons, ii) not all of the benchmarks satisfy the Brier Score constraints; i.e., they are not sufficiently well-calibrated, iii) in most cases the performance of Survival Quilts is better than that of the best benchmark, and the improvement is statistically significant over most of the benchmarks, and iv) in some cases (i.e., the UNOS-II and BPD datasets), the performance of Survival Quilts coincides with the best benchmark because it gives full weight to that benchmark.

3.5.3.1 Endogenous Time-Horizon Splits

To illustrate the impact of choosing the quilting patterns endogenously, we call attention to Figure 3.4. The discriminative performance of RSF and CISF usually decreases at longer time horizons. In large part this is because RSF and CISF are nonparametric models and do less well over time horizons in which the number of patients at risk and the number of events are smaller. In contrast, the discriminative performances of the (semi-)parametric models decrease less over longer time horizons. Because our method constructs quilting patterns that change over time, it is able to give greater weight to models whose increments of risk predictions provide good discriminative performance in different time horizons. For example, in the SUPPORT dataset, the weights on RSF and CISF decrease and the weights on the Cox, CoxRidge and LogNormal increase at around $t = 100$ because the performance of RSF and CISF degrade earlier and more abruptly compared to that of Cox, CoxRidge,



(a) Discriminative performance

(b) Quilting Pattern

Figure 3.3: Discriminative performance and quilting patterns over time for the MAGGIC dataset. The dotted black lines depict the 25%, 50%, and 75%-quantiles of time-to-event.

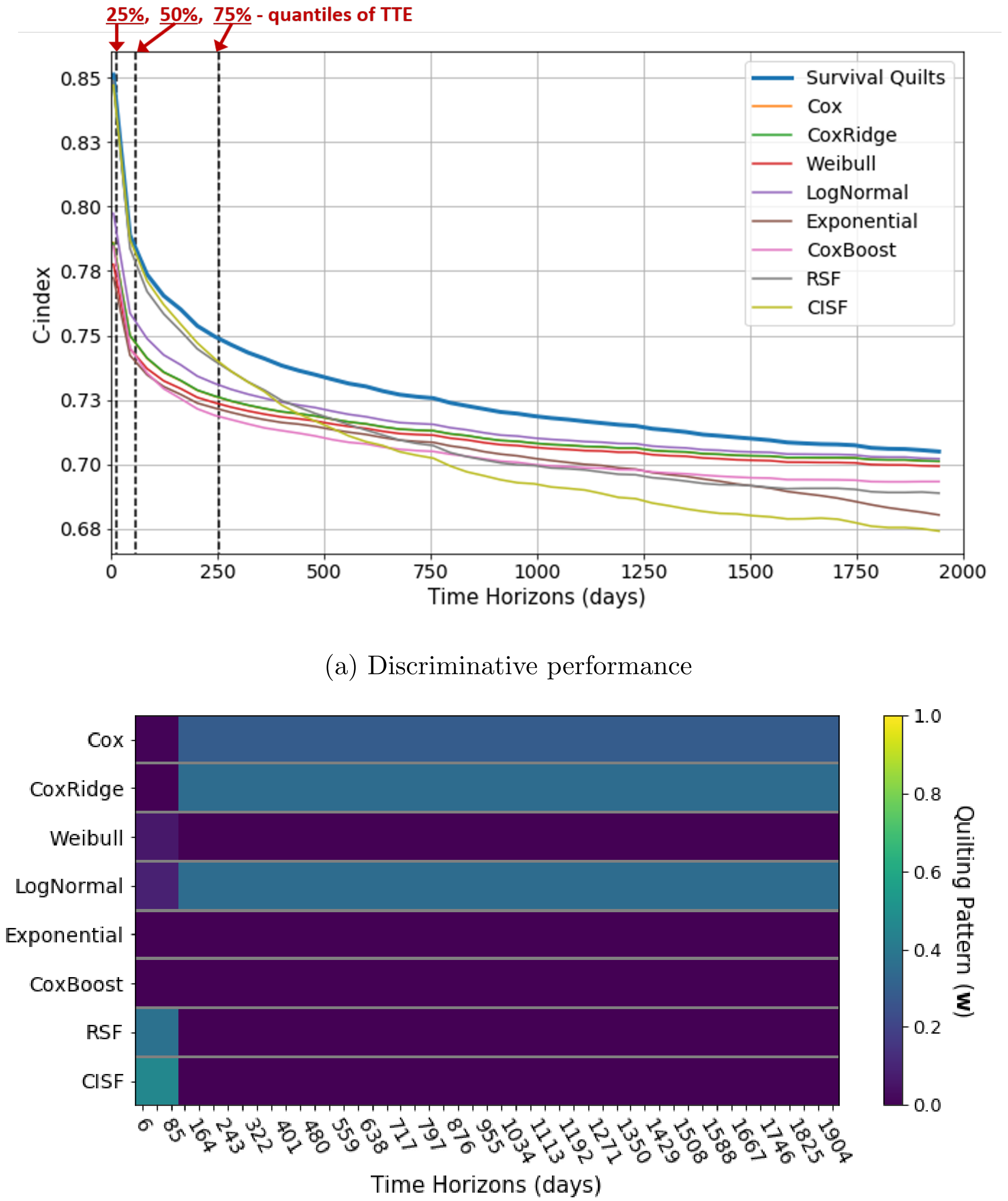
Table 3.3: C-index (mean±std) for the MAGGIC dataset at different time horizons. Blue highlighting indicates that the Brier Score constraints are satisfied.

Models	**Time-Horizons (quantiles)**		
	25%	**50%**	**75%**
Best benchmark	RSF	RSF	RSF
Cox	0.709±0.01*	0.694±0.02*	0.679±0.01*
CoxRidge	0.711±0.01*	0.695±0.02*	0.679±0.01*
Weibull	0.710±0.01*	0.695±0.02*	0.679±0.01*
LogNormal	0.719±0.02*	0.699±0.01*	0.676±0.01*
Exponential	0.708±0.02*	0.695±0.02*	0.679±0.01*
CoxBoost	0.707±0.02*	0.689±0.02*	0.672±0.01*
RSF	0.755±0.02	0.725±0.01	0.692±0.01†
CISF	0.740±0.02	0.708±0.01*	0.683±0.01*
Survival Quilts			
exog. $K = 1$	0.761±0.02	0.730±0.01	0.701±0.00
exog. $K = 2$	0.759±0.02	0.731±0.01	0.702±0.00
exog. $K = 3$	0.758±0.02	0.731±0.01	0.702±0.00
endogenous	**0.764±0.02**	**0.735±0.01**	**0.705±0.00**

∗, † indicate p-value < 0.01, < 0.05

and LogNormal.

Tables 3.3 - 3.5 compare the performance of Survival Quilts against the benchmarks at three time horizons. To highlight the gain achieved by our endogenous construction, we also provide the performance of Survival Quilts constructed using *exogenous* time horizons. When $K = 1$, we are using weights that do not vary with time as an alternative of the time-independent stacking [53]; for $K = 2, 3$, we have chosen exogenous time horizons with



(a) Discriminative performance

(b) Quilting Pattern

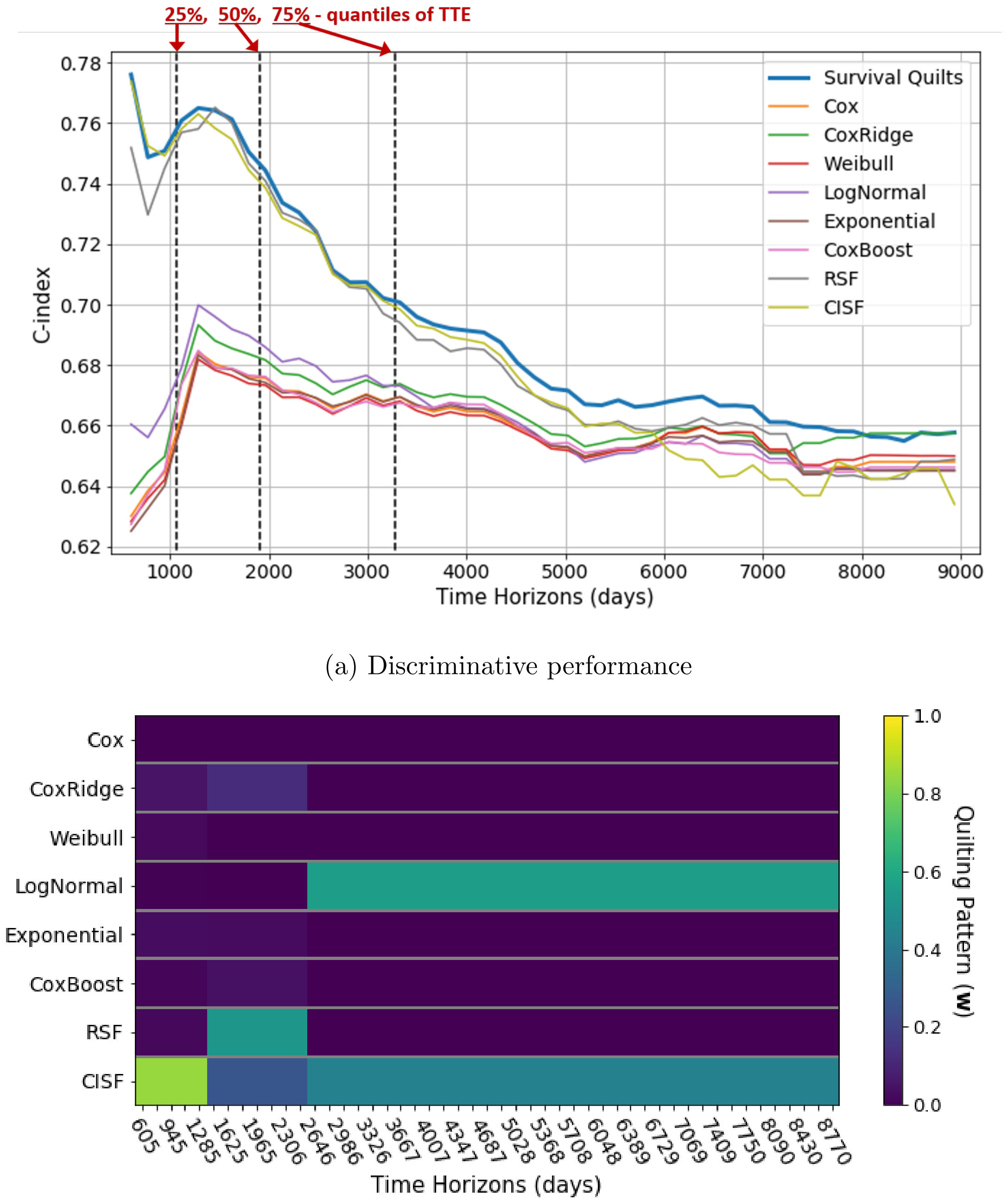
Figure 3.4: Discriminative performance and quilting patterns over time for the SUPPORT dataset. The dotted black lines depict the 25%, 50%, and 75%-quantiles of time-to-event.

Table 3.4: C-index (mean±std) for the SUPPORT dataset at different time horizons. Blue highlighting indicates that the Brier Score constraints are satisfied.

Models	**Time-Horizons (quantiles)**		
	25%	**50%**	**75%**
Best benchmark	RSF	CISF	CISF
Cox	0.786±0.01*	0.750±0.01*	0.726±0.01*
CoxRidge	0.786±0.01*	0.750±0.01*	0.727±0.01*
Weibull	0.778±0.01*	0.745±0.01*	0.724±0.01*
LogNormal	0.797±0.01*	0.759±0.01*	0.731±0.01†
Exponential	0.772±0.01*	0.742±0.01*	0.722±0.01*
CoxBoost	0.785±0.01*	0.745±0.01*	0.719±0.01*
RSF	0.849±0.02	0.784±0.01	0.740±0.01
CISF	0.847±0.02	0.787±0.01	0.741±0.01
Survival Quilts			
exog. $K=1$	0.842±0.02	0.782±0.01	0.743±0.01
exog. $K=2$	0.843±0.02	0.781±0.01	0.742±0.01
exog. $K=3$	0.846±0.01	0.784±0.01	0.743±0.01
endogenous	**0.851±0.02**	**0.789±0.01**	**0.750±0.01**

$*$, † indicate p-value < 0.01, < 0.05

very coarse grids. As seen in the tables, the endogenous construction of Survival Quilts provides the best performance because it chooses the time intervals endogenously and allows for different weights in different time intervals. In the tables, we highlight in blue the results for models and time horizons in which the Brier Score constraints are satisfied; note that satisfaction of the constraints changes over different horizons. Asterisks and daggers indicate that the performance improvements of Survival Quilts are statistically significant at the 0.01



(a) Discriminative performance

(b) Quilting Pattern

Figure 3.5: Discriminative performance and quilting patterns over time for the METABRIC dataset. The dotted black lines depict the 25%, 50%, and 75%-quantiles of time-to-event.

Table 3.5: C-index (mean±std) for the METABRIC dataset at different time horizons. Blue highlighting indicates that the Brier Score constraints are satisfied.

Models	**Time-Horizons (quantiles)**		
	25%	**50%**	**75%**
Best benchmark	CISF	RSF	CISF
Cox	0.663±0.02*	0.676±0.01*	0.669±0.01†
CoxRidge	0.674±0.03*	0.682±0.01*	0.674±0.01†
Weibull	0.660±0.02*	0.673±0.01*	0.668±0.01*
LogNormal	0.679±0.02*	0.686±0.01*	0.673±0.01†
Exponential	0.661±0.02*	0.674±0.01*	0.670±0.01†
CoxBoost	0.674±0.03*	0.676±0.01*	0.668±0.01*
RSF	0.757±0.04	0.741±0.03	0.694±0.02
CISF	0.758±0.02	0.739±0.01	0.698±0.01
Survival Quilts			
exog. $K=1$	0.753±0.03	0.739±0.02	0.698±0.02
exog. $K=2$	0.752±0.03	0.740±0.02	0.698±0.02
exog. $K=3$	0.752±0.04	0.739±0.02	0.693±0.02
endogenous	**0.761±0.03**	**0.744±0.02**	**0.701±0.02**

$*$, $\dagger$ indicates p-value < 0.01, < 0.05

and 0.05 levels, respectively.

3.5.4 Effect of Constrained BO

In this subsection, we address the effect of using constrained BO and how the optimal weight vector, $\mathbf{w}_k^\dagger$, changes as the number of BO iterations increases. Figure 3.6 illustrates the change in augmented Lagrangian objective in (3.10) and the change in time-dependent Brier-Score,

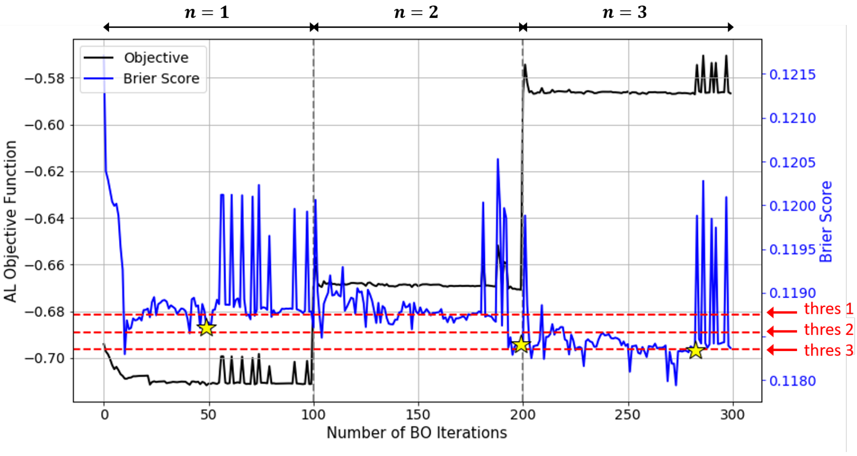


Figure 3.6: An illustration of change in the augmented Lagrangian objective (3.10) and Brier Score (g) with respect to the number of BO optimization iterations. The stars mark the minimal point of the objective for each subproblem. We set the maximum number of subproblems, n_{max}, and the number of BO steps to 3 and 100, respectively. The constrained BO is solved at the time horizon t_1 for the MAGGIC dataset.

g_k, with setting $k = 1$ for the MAGGIC dataset. As seen in the figure, if a strict constraint c is chosen (e.g., $c = thres$ 3 in the figure), the optimal weights for the first two subproblems of (3.10) do not satisfy the Brier Score constraint. Thus, our BO solves the next subproblem with updated λ and ρ, which in turn gives more weight to the calibration performance than in the previous subproblems. In this example, the optimal weight of the third subproblem satisfies the Brier Score constraint and, thus, is selected as $\mathbf{w}_1^\dagger$.

Table 3.6 shows the optimal weight vector $\mathbf{w}_k^\dagger$ that is chosen as we set a stricter constraint as illustrated in Figure 3.6. As seen in the table, Survival Quilts puts more weights on the Cox-PH based methods (Cox, CoxRidge, and CoxBoost), and Exponential when the constraint in (3.10) is less strict. However, with stricter constraints, our method reduces weights on the Cox-PH based methods and Exponential, and instead assigns higher weights on Weibull and LogNormal.

Table 3.6: Optimal weights, $\mathbf{w}_1^\dagger$ with varying Brier Score constraints in Figure 3.6.

Models	**Constraint**		
	thres 1	*thres* 2	*thres* 3
Cox	0.07	0	0
CoxRidge	0.06	0.04	0
Weibull	0	0.14	0.15
LogNormal	0	0.21	0.20
Exponential	0.19	0	0
CoxBoost	0.02	0	0
RSF	0.35	0.42	0.44
CISF	0.31	0.19	0.21

3.6 Conclusion

This work offers a novel approach to survival analysis that creates time-varying ensembles of existing survival models that we call Survival Quilts. Survival Quilts exploit existing models by giving them greater weight in time intervals where these models provide better incremental performance and lesser weight in time intervals where these models provide less good incremental performance. The superiority of Survival Quilts over previous survival models is demonstrated over six real-world datasets. One of the virtues of our approach is that we can adapt to use other survival models as those become available and prove their value.

CHAPTER 4

Learning Heterogeneous Treatment Effects from Time-to-Event Data

4.1 Introduction

The demand for methods evaluating the effect of treatments, policies and interventions on *individuals* is rising as interest moves from estimating population effects to understanding effect heterogeneity in fields ranging from economics to medicine. Motivated by this, the literature proposing machine learning (ML) methods for estimating the effects of treatments on continuous (or binary) end-points has grown rapidly, most prominently using tree-based methods [64–68], Gaussian processes [69, 70], and, in particular, neural networks (NNs) [71–78]. In comparison, the ML literature on heterogeneous treatment effect (HTE) estimation with time-to-event outcomes is rather sparse. This is despite the immense practical relevance of this problem – e.g. many clinical studies consider time-to-event outcomes; this could be the time to onset or progression of disease, the time to occurrence of an adverse event such as a stroke or heart attack, or the time until death of a patient.

In part, the scarcity of HTE methods may be due to time-to-event outcomes being inherently more challenging to model, which is attributable to two factors [79]: (i) time-to-event outcomes differ from standard regression targets as the main objects of interest are usually not only expected survival times but the *dynamics of the underlying stochastic process*, captured by hazard and survival functions, and (ii) the presence of *censoring*. This has led to the development of a rich literature on survival analysis particularly in (bio)statistics, see e.g.

[79, 80]. Classically, the effects of treatments in clinical studies with time-to-event outcomes are assessed by examining the coefficient of a treatment indicator in a (semi-)parametric model, e.g. Cox proportional hazards model [19], which relies on the often unrealistic assumption that models are correctly specified. Instead, we therefore adopt the nonparametric viewpoint of van der Laan and colleagues [81–84] who have developed tools to incorporate ML methods into the estimation of treatment-specific *population average* parameters. Nonparametrically investigating treatment effect *heterogeneity*, however, has been studied in much less detail in the survival context. While a number of tree-based methods have been proposed recently [85–88], NN-based methods lack extensions to the time-to-event setting despite their successful adoption for estimating the effects of treatments on other outcomes – the only exception being [89], who directly model event times under different treatments with generative models.

Instead of modeling event times as regression targets like [89], we consider adapting machine learning methods, with special focus on NNs, for estimation of (discrete-time) treatment-specific hazard functions. We do so because many target parameters of interest in studies with time-to-event outcomes are functions of the underlying temporal dynamics; that is, hazard functions can be used to directly compute (differences in) survival functions, (restricted) mean survival time, and hazard ratios. We begin by exploring and characterising the unique features of the survival treatment effect problem within the context of empirical risk minimization (ERM); to the best of our knowledge, such an investigation is lacking in previous work. In particular, we show that learning treatment-specific hazard functions is a challenging problem due to the potential presence of *multiple* sources of *covariate shift*: (i) non-randomized treatment assignment (confounding), (ii) informative censoring and (iii) a form of shift we term *event-induced* covariate shift, all of which can impact the quality of hazard function estimates. We then theoretically analyze the effects of said shifts on ERM, and use our insights to propose a new NN-based model for treatment effect estimation in the survival context.

Contributions (i) We identify and formalize key challenges of heterogeneous treatment effect estimation in time-to-event data within the framework of ERM. In particular, as discussed above, we show that when estimating treatment-specific hazard functions, *multiple* sources of covariate shift arise. (ii) We theoretically analyse their effects by adapting recent generalization bounds from domain adaptation and treatment effect estimation to our setting and discuss implications for model design. This analysis provides new insights that are of independent interest also in the context of hazard estimation in the absence of treatments. (iii) Based on these insights, we propose a new model (SurvITE) relying on balanced representations that allows for estimation of treatment-specific target parameters (hazard and survival functions) in the survival context, as well as a sister model (SurvIHE), which can be used for individualized hazard estimation in standard survival settings (without treatments). We investigate performance across a range of experimental settings and empirically confirm that SurvITE outperforms a range of natural baselines by addressing covariate shifts from various sources.

4.2 Problem Definition

Assume we observe a time-to-event dataset $\mathcal{D} = \{(a_i, x_i, \tilde{\tau}_i, \delta_i)\}_{i=1}^n$ comprising realizations of the tuple $(A, X, \tilde{T}, \Delta) \sim \mathbb{P}$ for n patients. Here, $X \in \mathcal{X}$ and $A \in \{0, 1\}$ are random variables for a covariate vector and an indicator whether treatment was administered at baseline. Let $T \in \mathcal{T}$ and $C \in \mathcal{T}$ denote random variables for the time-to-event and the time-to-censoring. Then, the time-to-event outcomes of each individual patient are described by $\tilde{T} = \min(T, C)$ and $\Delta = \mathbb{1}(T \leq C)$, which indicate the time elapsed until either an event or censoring occurs and whether the event was observed or not, respectively. Throughout, we treat survival time as discrete[1] and the time horizon as finite with pre-defined maximum $t_{\max}$, so that the set of possible survival times is $\mathcal{T} = \{1, \cdots, t_{\max}\}$.

[1] Where necessary, discretization can be performed by transforming continuous-valued times into a set of contiguous time intervals, i.e., $T = \tau$ implies $T \in [t_\tau, t_\tau + \delta t)$ where δt implies the temporal resolution.

4.2.1 Long and Short Data Structures.

We transform the *short* data structure outlined above to a so-called *long* data structure which can be used to *directly* estimate conditional hazard functions using standard machine learning methods [82]. We define two counting processes $N_T(t)$ and $N_C(t)$ which track events and censoring, i.e. $N_T(t) = \mathbb{1}(\tilde{T} \leq t, \Delta = 1)$ and $N_C(t) = \mathbb{1}(\tilde{T} \leq t, \Delta = 0)$ for $t \in \mathcal{T}$; both are zero until either an event or censoring occurs. By convention, we let $N_T(0) = N_C(0) = 0$. Further, let $Y(t) = \mathbb{1}(N_T(t) = 1 \cap N_T(t-1) = 0)$ be the indicator for an event occuring at time t; thus, for an individual with $\tilde{T} = \tau$ and $\Delta = 1$, $Y(t) = 0$ for all $t \neq \tau$, and $Y(t) = 1$ at the event time $t = \tau$. The conditional hazard is the probability that an event occurs at time τ given that it does not occur before time τ, hence it can be defined as [84]

$$\begin{aligned} \lambda(\tau|a,x) &= \mathbb{P}(T = \tau | T \geq \tau, A = a, X = x) = \mathbb{P}(\tilde{T} = \tau, \Delta = 1 | \tilde{T} \geq \tau, A = a, X = x) \\ &= \mathbb{P}(Y(\tau) = 1 | N_T(\tau-1) = 0, N_C(\tau-1) = 0, A = a, X = x) \end{aligned} \tag{4.1}$$

It is easy to see from (4.1) that given data in long format, $\lambda(\tau|a,x)$ can be estimated for any τ by solving a standard classification problem with $Y(\tau)$ as target variable, considering only the samples *at-risk* at time τ in each treatment arm (individuals for which neither event nor censoring has occurred until that time point; i.e. the set $\mathcal{I}(\tau, a) \overset{\text{def}}{=} \{i \in [n] : N_T(\tau-1)_i = N_C(\tau-1)_i = 0 \cap A_i = a\}$). Finally, given the hazard, the associated survival function $S(\tau|a,x) = \mathbb{P}(T > \tau | A = a, X = x)$ can then be computed as $S(\tau|a,x) = \prod_{t \leq \tau} \big(1 - \lambda(t|a,x)\big)$. The censoring hazard $\lambda_C(t|a,x)$ and survival function $S_C(t|a,x)$ can be defined analogously.

4.2.2 Target Parameters

While the main interest in the standard treatment effect estimation setup with continuous outcomes usually lies in estimating only the (difference between) conditional outcome means under different treatments, there is a broader range of target parameters of interest in the time-to-event context, including both treatment-specific target functions and *contrasts* that represent some form of heterogeneous treatment effect (HTE). We define the treatment-specific (conditional) hazard and survival functions as

$$\begin{aligned}
\lambda^a(\tau|x) &= \mathbb{P}(T = \tau | T \geq \tau, do(A = a), X = x) \\
S^a(\tau|x) &= \mathbb{P}(T > \tau | do(A = a), X = x) = \prod\nolimits_{t \leq \tau} \big(1 - \lambda^a(t|x)\big)
\end{aligned} \tag{4.2}$$

Here, $do(A = a)$ denotes [90]'s do-operator which indicates an intervention in which every individual is assigned treatment a; below we discuss assumptions that are necessary to identify such interventional quantities from observational datasets.

Given $\lambda^a(\tau|x)$ and $S^a(\tau|x)$, possible HTEs of interest[2] include the difference in treatment-specific survival times at time τ, i.e. $\text{HTE}_{surv}(\tau|x) = S^1(\tau|x) - S^0(\tau|x)$, the difference in restricted mean survival time (RMST) up to time L, i.e. $\text{HTE}_{rmst}(x) = \sum_{t_k \leq L} \big(S^1(t_k|x) - S^0(t_k|x)\big) \cdot (t_k - t_{k-1})$, and hazard ratios. In the following, we will focus on estimation of the treatment specific hazard functions $\{\lambda^a(t|x)\}_{a \in \{0,1\}, t \in \mathcal{T}}$ as this can be used to compute survival functions and causal contrasts.

4.2.3 Assumptions

As in [82–84], we assume the fairly general causal structure encoded in the DAG in Figure 4.1. By assuming that observed data was generated from this DAG, the classical identifying assumptions (No Hidden Confounders, Censoring At Random, and Consistency) are implicitly formalized [82]. Equivalently, we can restate the assumptions using potential outcomes [91] notation. As in e.g. [92], we let T_a denote the potential event time that would have been observed had treatment a been assigned, and $C = t_{\max}$ been externally set. Then, the following assumptions are implied by the DAG:

Assumption 1 (1.a No hidden confounders (unconfoundedness))**.** *Treatment assignment is random conditional on covariates, i.e.* $T_a \perp\!\!\!\perp A | X$*.*

[2] *Note:* All parameters of interest to us are *heterogeneous* (also sometimes referred to as *individualized*), i.e. a function of the covariates X, while the majority of existing literature in (bio)statistics considers *population average* parameters that are functions of quantities such as $\mathbb{P}(T > \tau | do(A = a))$, which average over all X.

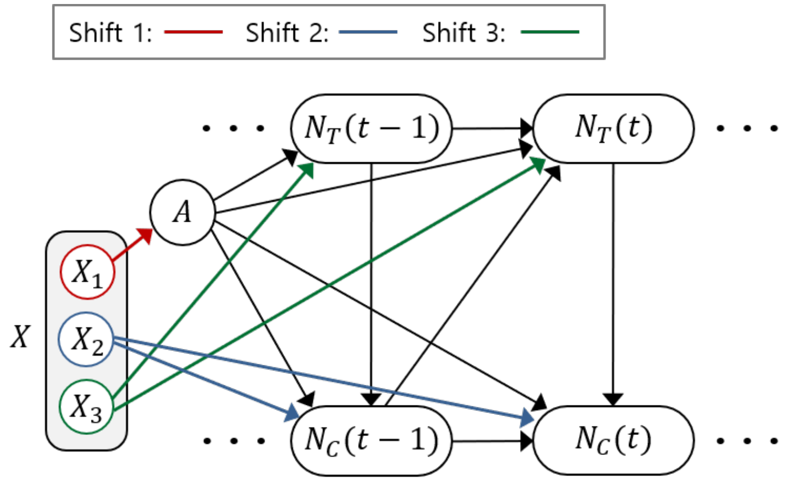


Figure 4.1: The assumed underlying DAG. Covariates X can be split into (possibly overlapping) subsets X_1, X_2 and X_3, determining treatment selection, informative censoring, and event times, respectively.

Assumption 2 (1.b Censoring at random)**.** *Censoring and outcome are conditionally independent, i.e.* $T_a \perp\!\!\!\perp C | X, A$.

Assumption 3 (1.c Consistency)**.** *The observed outcomes are the potential outcomes under the observed intervention, i.e. if* $A = a$ *then* $T = T_a$.

Then, we can write

$$\begin{aligned}\lambda^a(\tau|x) &= \mathbb{P}(T = \tau | T \geq \tau, do(A = a), X = x)\\ &= \mathbb{P}(T_a = \tau | T_a \geq \tau, A = a, X = x)\\ &= \mathbb{P}(T_a = \tau | T_a \geq \tau, C = t_{\max}, A = a, X = x)\\ &= \mathbb{P}(\tilde{T} = \tau, \Delta = 1 | \tilde{T} \geq \tau, C = t_{\max}, A = a, X = x) = \lambda(\tau|a, x)\end{aligned}$$

Here, the equalities in line one and two follow by definition, line three follows by assumption 1.a, line four follows by assumption 1.b, the equality in line five follows by assumption 1.c, and the final line follows by definition.

To enable nonparametric estimation of $\lambda^a(\tau|x)$ for some fixed $\tau \in \mathcal{T}$, we additionally consider a number of conditions on the likelihood of observing certain events.

Assumption 4 (2.a Overlap/positivity (treatment assignment)). *Treatment assignment is non-deterministic, i.e. for some* $\epsilon_1 > 0$*, we have that* $\epsilon_1 < \mathbb{P}(A = a|X = x) < 1 - \epsilon_1$

Assumption 5 (2.b Positivity (censoring)). *Censoring is non-deterministic, i.e. for some* $\epsilon_2 > 0$*, we have that* $\mathbb{P}(N_C(t) = 0|A = a, X = x) = \mathbb{P}(C > t|A = a, X = x) => \epsilon_2 \;\; \forall t < \tau$*.*

Assumption 6 (2.c Positivity (events)). *Not all events deterministically occur before time* τ*, i.e.* $\mathbb{P}(N_T(\tau{-}1) = 0|A = a, X = x) > \mathbb{P}(T > \tau{-}1|A = a, X = x)\epsilon_3 > 0$

Assumptions 1.a, 1.c and 2.a are standard within the treatment effect estimation literature [70, 72]; assumptions 1.b and 2.b are standard within the literature with survival outcomes [88, 92]. Assumption 2.c is needed only if we aim to estimate $\lambda^a(t|x)$ for all t, otherwise it would suffice to follow a convention such as setting $\lambda^a(t|x) = 1$ whenever $\mathbb{P}(N_T(\tau{-}1) = 0|A = a, X = x) = 0$.

4.3 Challenges in Learning Treatment-Specific Hazard Functions using ERM

Preliminaries: ERM under Covariate Shift Recall that in problems with covariate shift, the training distribution $X, Y \sim \mathbb{Q}_0(\cdot)$ used for ERM and the target distribution $X, Y \sim \mathbb{Q}_1(\cdot)$ are mismatched: One assumes that the marginals do not match, i.e. $\mathbb{Q}_0(X) \neq \mathbb{Q}_1(X)$, while the conditionals remain the same, i.e. $\mathbb{Q}_0(Y|X) = \mathbb{Q}_1(Y|X)$ [93]. If the hypothesis class $\mathcal{H}$ used in ERM does not contain the truth (or in the presence of heavy regularization), this can lead to suboptimal hypothesis choice as $\arg\min_{h \in \mathcal{H}} \mathbb{E}_{X,Y \sim \mathbb{Q}_1(\cdot)}[\ell(Y, h(X))] \neq \arg\min_{h \in \mathcal{H}} \mathbb{E}_{X,Y \sim \mathbb{Q}_0(\cdot)}[\ell(Y, h(X))]$ in general.

4.3.1 Sources of Covariate Shift

We now consider how to learn a treatment-specific hazard function $\lambda^a(\tau|x)$ from observational data using ERM. As detailed in Section 4.2, we exploit the long data format by realizing that $\lambda^a(\tau|x)$ can be estimated by solving a standard classification problem with $Y(\tau)$ as dependent variable and X as covariates, using only the samples at risk with treatment status a, i.e. $\mathcal{I}(\tau, a)$, which corresponds to solving the empirical analogue of the problem

$$\hat{\lambda}^a(\tau|x) \in \arg \min_{h_{a,\tau} \in \mathcal{H}} \mathbb{E}_{X,Y(\tau)\sim\mathbb{P}_{a,\tau}(\cdot)}[\ell(Y(\tau), h_{a,\tau}(X)] \tag{4.3}$$

where we use $\mathbb{P}_{a,\tau}$ to refer to the observational (at-risk) distribution $\mathbb{P}_{a,\tau}(X, Y(\tau)) = \lambda^a_T(\tau|X)\mathbb{P}_{a,\tau}(X)$ with $\mathbb{P}_{a,\tau}(X) = \mathbb{P}(X|N_T(\tau-1) = N_C(\tau-1) = 0, A = a) = \mathbb{P}(X|\tilde{T} \geq \tau, A = a)$. If the loss function ℓ is chosen to be the log-loss, this corresponds to optimizing the likelihood of the hazard.

The observational (at-risk) covariate distribution $\mathbb{P}_{a,\tau}(X)$, however, is *not* our target distribution: instead, to obtain reliable treatment effect estimates for the whole population, we wish to optimize the fit over the population at baseline, i.e. the marginal distribution $X \sim \mathbb{P}(X)$ which we will refer to as $\mathbb{P}_0(X)$ below to emphasize it being the baseline at-risk distribution.[3]. Here, differences between $\mathbb{P}_0(X)$ and the population at-risk $\mathbb{P}_{a,\tau}(X)$ can arise due to three distinct sources of covariate shift:

- *(Shift 1) Confounding/treatment selection bias*: if treatment is not assigned completely at random, then $\mathbb{P}(X|A = a) \neq \mathbb{P}_0(X)$ and the distribution of characteristics across the treatment arms differs already at baseline, thus $\mathbb{P}_{a,\tau}(X) \neq \mathbb{P}_0(X)$ in general.
- *(Shift 2) Censoring bias*: regardless of the presence of confounding, if the censoring hazard is not independent of covariates, i.e. $\lambda_C(\tau|a, x) \neq \lambda_C(\tau|a)$, then the population at-risk changes over time such that $\mathbb{P}_{a,\tau_1}(X) \neq \mathbb{P}_{a,\tau_2}(X) \neq \mathbb{P}_0(X)$ in general. If, in addition, there are differences between the treatment-specific censoring hazards, then the at-risk distribution will also differ across treatment arms at any given time-point, i.e. $\mathbb{P}_{a,\tau}(X) \neq \mathbb{P}_{1-a,\tau}(X)$ for $\tau > 1$ in general.

[3] With slight abuse of notation, we will use $\mathbb{P}_0$ and $\mathbb{P}_{a,\tau}$ also to refer to densities of continuous x

- *(Shift 3) Event-induced shifts*: Counterintuitively, even in the absence of both confounding and censoring, there will be covariate shift in the at-risk population if the event-hazard depends on covariates, i.e. if $\lambda(\tau|a,x) \neq \lambda(\tau|a)$ then $\mathbb{P}_{a,\tau_1}(X) \neq \mathbb{P}_{a,\tau_2}(X) \neq \mathbb{P}_0(X)$ in general. Further, if there are heterogenous treatment effects, then $\mathbb{P}_{a,\tau}(X) \neq \mathbb{P}_{1-a,\tau}(X)$ for $\tau > 1$ in general.

4.3.2 What makes the survival treatment effect estimation problem unique?

While *Shift 1* arises also in the standard treatment effect estimation setting, *Shift 2* and *Shift 3* arise uniquely due to the nature of time-to-event data. Thus, estimating treatment effects from time-to-event data is inherently more involved than estimating treatment effects in the standard static setup, as covariate shift at time horizon $\tau > 1$ can arise *even in a randomized control trial (RCT).* Thus, in addition to the overall at-risk population changing over time, both treatment effect heterogeneity and treatment-dependent censoring can lead to differences in the composition of the population at-risk in each treatment arm. Further, Shifts 1, 2 and 3 can also interact to create more extreme shifts; e.g. if treatment selection is based on the same covariates as the event process (i.e. $X_1 = X_3$ in Fig. 4.1) then event-induced shift can amplify the selection effect over time.

Interestingly, changes of the at-risk population over time arise also in standard survival problems (without treatments); yet in the context of *prediction* these do not matter: as the at-risk population at any time-step is also the population that will be encountered at test-time, this shift in population over time is not problematic, unless it is caused by censoring. If, however, our goal is *estimation* of a target parameter over the whole population, this corresponds to a setting where the ideal evaluation is performed on a 'counterfactual' population (i.e. the population resulting if all individuals had survived until time τ) which is never encountered in test sets – and hence requires careful consideration of the consequences of the covariate shifts discussed above. To see why fixing one target population is necessary, note that when the goal is estimation of the difference in survival curves, i.e. $\text{HTE}_{surv}(\tau|x)$, this requires estimation of $2 \times \tau$ hazard functions; if each of them was optimized for a different target population, this would make the final survival curves and their differences hard to interpret.

Finally, we note that Shifts 2 and (particularly) 3 *seemingly* appear only because we chose to represent the data in long format. However, many ML-based discrete-time models targeting hazard or survival function directly *implicitly* rely on the long data-format (or similar transformations), making these shifts problematic for them too. Thus, representation in long format and the use of the classification approach only helps to make these shifts *explicit*. Survival models which model (log) time as a regression target do not suffer from Shift 3; however, as we show in the experiments using the model of [89], their performance on estimating survival functions can be poor.

4.3.3 Possible Remedies and Theoretical Analysis

A natural solution to tackle bias in ERM caused by covariate shift is to use importance weighting [94]; i.e. to reweight the empirical risk by the density ratio of target $\mathbb{P}_0(X)$ and observed distribution $\mathbb{P}_{a,\tau}(X)$. In our context, for any (τ, a), optimal importance weights are given by

$$w^*_{a,\tau}(x) = \frac{\mathbb{P}_0(x)}{\mathbb{P}_{a,\tau}(x)} = \frac{p_{\tau,a}}{e_a(x) r_a(x,\tau)} \tag{4.4}$$

with $p_{\tau,a} = \mathbb{P}(\tilde{T} \geq \tau, A = a)$, $e_a(x) = \mathbb{P}(A = a | X = x)$ the propensity score, and $r^a(x,\tau) = \mathbb{P}(\tilde{T} \geq \tau | A = a, X = x)$ the probability to be at risk, i.e. the probability that neither event nor censoring occurred before time τ. These weights are well-defined due to the overlap assumptions detailed in Sec. 4.2; however, they are in general unknown as they *depend on the unknown target parameters* $\lambda^a(\tau|x)$ through $r^a(x,\tau)$. Further, especially for large τ, these weights might be very extreme even if known, which can lead to highly unstable results [95] – making biased yet stabilized weighting schemes, e.g. truncation, a good alternative. Therefore, we only assume access to some (possibly imperfect) weights $w_{a,\tau}(x)$ s.t. $\mathbb{E}_{X \sim \mathbb{P}_{a,\tau}}[w_{a,\tau}(x)] = 1$, so that we can create a weighted distribution $\mathbb{P}^w_{a,\tau} = w_{a,\tau}(x)\mathbb{P}^a_\tau(x)$. (Note: $\mathbb{P}^a_\tau(x)$ can be recovered by using $w_{a,\tau}(x) = 1$.)

Either instead of [71, 72] or in addition to weighting [73, 75, 77, 96], the literature on learning balanced representations for static treatment effect estimation has focused on finding a different remedy for distributional differences between treatment arms: creating representations $\Phi : \mathcal{X} \to \mathcal{R}$ which have similar (weighted) distributions across arms as measured by an integral probability metric (IPM), motivated by generalization bounds. As we show below, we can exploit a similar feature in our context by finding a representation that minimizes the IPM term not between treatment arms, but between covariate distribution at baseline $\mathbb{P}_0$ and $\mathbb{P}^w_{a,\tau}$. The proposition below bounds the target risk of a hazard estimator $\hat{\lambda}^a_T(\tau|x) = h(\Phi(x))$ relying on any representation. The proof extends the concept of excess target information loss, proposed recently to analyze domain-adversarial training [97], and the standard IPM arguments made in e.g. [96].

Proposition 1. *For fixed a, τ and representation $\Phi : \mathcal{X} \to \mathcal{R}$, let $\mathbb{P}^\Phi_0$, $\mathbb{P}^\Phi_{a,\tau}$ and $\mathbb{P}^{w,\Phi}_{a,\tau}$ denote the target, observational, and weighted observational distribution of the representation Φ. Define the pointwise losses*

$$\begin{aligned} \ell_{h,\mathbb{Q}}(x; a, \tau) &\overset{\text{def}}{=} \mathbb{E}_{Y(\tau)|x,a\sim\mathbb{Q}}[\ell(Y(\tau), h(\Phi(X)))|X = x, A = a] \\ \ell_{h,\mathbb{Q}^\Phi}(\phi; a, \tau) &\overset{\text{def}}{=} \mathbb{E}_{Y(\tau)|\phi,a\sim\mathbb{Q}^\Phi}[\ell(Y(\tau), h(\Phi))|\Phi = \phi, A = a] \end{aligned} \tag{4.5}$$

of (hazard) hypothesis $h \equiv h_{a,\tau} : \mathcal{R} \to [0, 1]$ w.r.t. distributions in covariate and representation space, respectively. Assume there exists a constant $C_\Phi > 0$ s.t. ${C_\Phi}^{-1}\ell_{h,\mathbb{P}^{w,\Phi}_{a,\tau}}(\phi, a, \tau) \in \mathcal{G}$ for some family of functions $\mathcal{G}$. Then we have that

$$\underbrace{\mathbb{E}_{X\sim\mathbb{P}_0}[\ell_{h,\mathbb{P}}(X; a, \tau)]}_{\textit{Target Risk}} \leq \underbrace{\mathbb{E}_{X\sim\mathbb{P}_{a,\tau}}[w_{a,\tau}(X)\ell_{h,\mathbb{P}}(X; a, \tau)]}_{\textit{Weighted observational risk}} + C_\Phi \underbrace{IPM_G(\mathbb{P}^\Phi_0, \mathbb{P}^{w,\Phi}_{a,\tau})}_{\textit{Distance in }\Phi\textit{-space}} + \underbrace{\eta^l_\Phi(h)}_{\textit{Info loss}} \tag{4.6}$$

where $IPM_\mathcal{G}(\mathbb{P}, \mathbb{Q}) = \sup_{g\in\mathcal{G}} \left|\int g(x)(\mathbb{P}(x) - \mathbb{Q}(x))dx\right|$ and we define the excess target information loss $\eta^\ell_\Phi(h)$ analogously to [97] as $\eta^\ell_\Phi(h) \overset{\text{def}}{=} \mathbb{E}_{X\sim\mathbb{P}}[\xi_{\mathbb{P}^\Phi_0,\mathbb{P}}(X) - \xi_{\mathbb{P}^{w,\Phi}_{a,\tau},\mathbb{P}}(X)]$ with $\xi_{\mathbb{Q}^\Phi,\mathbb{Q}}(x) \overset{\text{def}}{=} \ell_{h,\mathbb{Q}^\Phi}(\phi; a, \tau) - \ell_{h,\mathbb{Q}}(x; a, \tau)$. For invertible Φ, $\eta^\ell_\Phi(h) = \xi_{\mathbb{Q}^\Phi,\mathbb{Q}}(x) = 0$.

Unlike the bounds provided in [72, 73, 77, 89, 96], this bound does not rely on representations to be invertible; we consider this feature important as none of the works listed actually enforced invertibility in their proposed algorithms. Given bound (4.6), it is easy to see why non-invertibilty can be useful: for any (possibly non-invertible) representation for which it holds that $Y(\tau) \perp\!\!\!\perp X | \Phi(X), A$, it also holds that $\eta^{\ell}_{\Phi}(h) = \xi_{\mathbb{P}^{\Phi},\mathbb{P}}(x) = \xi_{\mathbb{P}^{w,\Phi}_{a,\tau},\mathbb{P}}(x) = 0$ and the causally identifying restrictions continue to hold. A simple representation for which this property holds is a selection mechanism that chooses only the causal parents of $Y(\tau)$ from within X; if X can be partitioned into variables affecting the instantaneous risk (X_3 in Fig. 4.1), and variables affecting *only* treatment assignment ($X_1 \setminus X_3$) and/or censoring mechanism ($X_2 \setminus X_3$), then the IPM term can be reduced by a representation which drops the latter sets of variables – or irrelevant variables correlated with any such variables – without affecting $\eta^{\ell}_{\Phi}(h)$. As a consequence, event-induced covariate shift can generally not be *fully* corrected for using non-invertible representations (unless the variables affecting event time are different at every time-step). Further, given perfect importance weights w^*, both $\eta^{\ell}_{\Phi}(h)$ and IPM term are zero.

Except for the dependence on $\eta^{\ell}_{\Phi}(h)$, this bound differs from the regression-based bound for survival treatment effects stated in [89] (which is identical to the original treatment effect bound in [72]) in that we have dependence on τ in the IPM term, which, among other things, explicitly captures the effect of censoring. Our bound motivates that, instead of finding representations that balance treatment- and control group at baseline (or at each time step) we should find representations that balance $\mathbb{P}^{\Phi}_{a,\tau}$ towards the *baseline distribution* $\mathbb{P}^{\Phi}_{0}$ for each time step, which motivates our method detailed below. Note that this bound motivates the use of balanced representations for modeling time-to-event outcomes in the presence of informative censoring even in the standard prediction setting, which is a finding that could be of independent interest for the ML survival analysis literature.

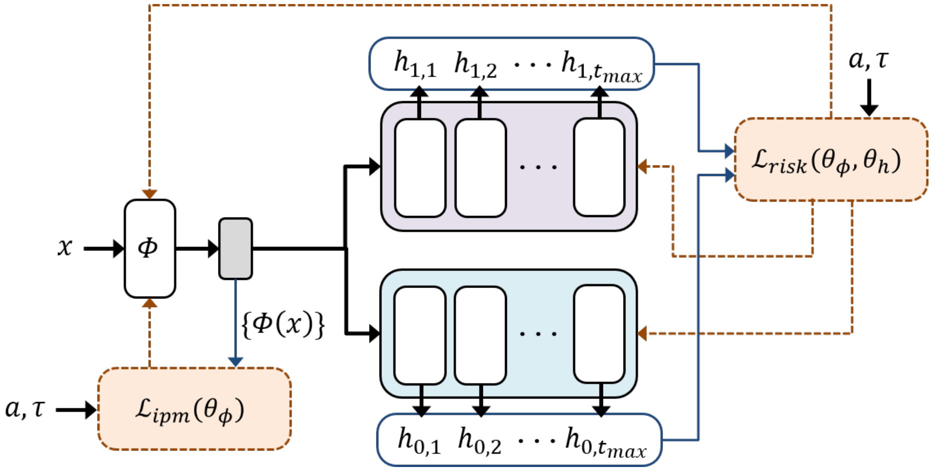

$h_{1,1}$ $h_{1,2}$ $\cdots$ $h_{1,t_{max}}$
a,τ
$\mathcal{L}_{risk}(\theta_\phi, \theta_h)$
x
Φ
$\{\Phi(x)\}$
a,τ
$\mathcal{L}_{ipm}(\theta_\phi)$
$h_{0,1}$ $h_{0,2}$ $\cdots$ $h_{0,t_{max}}$

Figure 4.2: The architecture of SurvITE.

4.4 Method: SurvITE

Based on the theoretical analysis above, we propose a novel deep learning approach to HTE estimation from observed time-to-event data, which we refer to as SurvITE (Individualized Treatment Effect estimator for Survival analysis). The network architecture is illustrated in Figure 4.2. Note that even in the absence of treatments we can use this architecture for estimation of hazards and survival functions by using only one treatment $a = 0$. As we show in the experiments, this version of our method – SurvIHE (Individualized Hazard estimator for Survival analysis) – is of independent interest in the standard survival setting, as it corrects for Shifts 1 & 2. Below, we describe the empirical loss functions we use to find representation Φ and hypotheses $h_{a,\tau}$.

Let $\Phi : \mathcal{X} \to \mathcal{R}$ denote the *representation* (parameterized by θ_ϕ) and $h_{a,\tau} : \mathcal{R} \to [0, 1]$ the *hazard estimator* for treatment a and time τ (parameterized by $\theta_{h_{a,\tau}}$), each implemented as a fully-connected neural network. While the output heads are thus unique to each treatment-group time-step combination, we allow hazard estimators to share information by using *one* shared representation for all hazard functions. This allows for both borrowing of information across different a, τ and significantly reduces the number of parameters of the network.

Then, given the time-to-event data $\mathcal{D}$, we use the following empirical loss functions for the observational risk and the IPM term:

$$\mathcal{L}_{risk}(\theta_\phi, \theta_h) = \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} \sum_{i:\tilde{\tau}_i \geq t} n_{1,t}^{-1} a_i \ell\big(y_i(t), h_{1,t}(\Phi(x_i))\big) + n_{0,t}^{-1}(1-a_i)\ell\big(y_i(t), h_{0,t}(\Phi(x_i))\big),$$

$$\mathcal{L}_{ipm}(\theta_\phi) = \sum_{a\in\{0,1\}} \sum_{t=1}^{t_{\max}} Wass\big(\{\Phi(x_i)\}_{i=1}^{n}, \{\Phi(x_i)\}_{i:\tilde{\tau}_i \geq t, a_i = a}\big),$$

where $Wass(\cdot,\cdot)$ is the finite-sample Wasserstein distance [98]. Further, $n_{a,t} = |\mathcal{I}(\tau, a)|$ is the number of samples at-risk in each treatment arm, its presence ensures that each a, τ-combination contributes equally to the loss. Overall, we can find Φ and $h_{a,\tau}$'s that optimally trade off balance and predictive power as suggested by the generalization bound (4.6) by minimizing the following loss:

$$\mathcal{L}_{target}(\theta_\phi, \theta_h) = \mathcal{L}_{risk}(\theta_\phi, \theta_h) + \beta \mathcal{L}_{ipm}(\theta_\phi) \tag{4.7}$$

where $\theta_h = \{\theta_{h_{a,\tau}}\}_{a\in\{0,1\},\tau\in\mathcal{T}}$, and $\beta > 0$ is a hyper-parameter.

Uniform vs. non-uniform weighting. In (4.7), all samples are weighted uniformly (within each a, τ combination). We tested non-uniform, estimated importance weights $\hat{w}^*_{a,\tau}(x)$, and, in synthetic experiments, even considered 'oracle' weights. Across both strategies for weighting and different truncation thresholds, we found that non-uniform weighting did not improve the performance of SurvITE. This is in line with recent empirical [99] and theoretical [100] findings indicating that weighting may have little impact in deep learning – overparametrized NNs have sufficient capacity to not have to trade-off between classifying different training points [99], which is the problem of low-capacity misspecified models (e.g. linear models) in this context. We conjecture that the IPM-term, on the other hand, does help as it fulfills a slightly different purpose than weighting; it forces Θ to act similarly to a variable selection mechanism (making the subsequent learning problem easier) and encourages 'shift-invariant' representations that generalize better in the presence of different shifts.

4.5 Related work

Heterogeneous treatment effect estimation (non-survival) has been studied in great detail in the recent ML literature. While early work built mainly on tree-based methods [64–67], many other methods, such as Gaussian processes [69, 70] and GANS [101], have been adapted to estimate HTEs. Arguably the largest stream of work [71–78] built on NNs, due to their flexibility and ease of manipulating loss functions, which allows for easy incorporation of balanced representation learning as proposed in [71, 72] and motivated also the approach taken in this work. Another popular approach has been to consider model-agnostic (or 'meta-learner' [102]) strategies, which provide a 'recipe' for estimating HTEs using *any* predictive ML method [78, 102–104]. Because of their simplicity, the *single model* (S-learner) – which uses the treatment indicator as an additional covariate in otherwise standard model-fitting – and *two model* (T-learner) – which splits the sample by treatment status and fit two separate models – strategies [102], can be directly applied to the survival setting by relying on a standard survival (prediction) method as base-learner.

ML methods for survival prediction continue to multiply; here we focus on the most related class of methods – namely on those nonparametrically modeling conditional hazard or survival functions – and *not* on those relying on flexible implementations of the Cox proportional hazards model (e.g. [22, 23, 105]) or modeling (log-)time as a regression problem (e.g. [48, 106–108]). One popular nonparametric estimator of survival functions is [30]'s random survival forest, which relies on the Nelson-Aalen estimator to nonparametrically estimate the cumulative hazard within tree-leaves. The idea of modeling discrete-time hazards directly using *any arbitrary classifier* and long data-structures goes back to at least [109], with implementations using NN-based methods presented in e.g. [110–113]. [114] models the probability mass function instead of the hazard, and [115] use labels $\mathbb{1}\{T > t\}_{t \in \mathcal{T}}$ to estimate the survival function directly using multi-task logistic regression.

Estimating HTEs from time-to-event data has been studied in much less detail.

[85, 87] use tree-based nearest-neighbor estimates to estimate expected differences in survival time directly, and [86] use a BART-based S-learner to output expected differences in log-survival time. [116] performed a simulation study using different survival prediction models as base-learners for a two-model approach to estimating the difference in median survival time. Based on ideas from the semi-parametric efficiency literature, [88] and [92] propose estimators that target the (restricted) mean survival time *directly* and consequently *do not* output estimates of the treatment-specific hazard or survival functions. We consider the ability to output treatment-specific predictions an important feature of a model if the goal is to use model output to give decision support, given that it allows the decision-maker to trade-off relative improvement with the baseline risk of a patient. Finally, [89] recently proposed a generative model for treatment-specific event times which relies on balancing representations to balance only the treatment groups at baseline. This model does not output hazard- or survival functions, but can provide approximations by performing Monte-Carlo sampling.

4.6 Experiments

Unfortunately, when the goal is *estimating* (differences of) survival functions (instead of *predicting* survival), evaluation on real data will not reflect performance w.r.t. the intended baseline population. Therefore, we conduct a range of synthetic experiments with *known* ground truth. We evaluate the effects of different shifts separately by starting with survival estimation *without* treatments, and then introduce treatments. Finally, we use the real-world dataset TWINS [117] which has uncensored survival outcomes for twins (where the treatment is 'being born heavier'), and is hence free of Shifts 1 & 2.

4.6.1 Baselines

We compared SurvITE with baselines ranging from commonly used survival methods to the state-of-the-art HTE methods based on deep neural networks. The details of how we implemented the benchmarks are described as the following:

- **Cox**[4] [19] and **RSF**[4] [30]: When there are treatments, we use these models in a two-model (T-learner) approach by training a separate model using samples in the treated ($A = 1$) and controlled ($A = 0$) groups, respectively. For Cox, we set the coefficient for ridge regression penalty as $\alpha = 0.001$. For RSF, we use the default hyper-parameter setting (i.e., $n\_estimators = 100$ using a survival tree as the baseline estimator and $min\_samples\_leaf = 3$ without maximum depth restriction).

- **LR-sep**: We utilize the long data format as described in Section 2 of the manuscript and train a separate logistic regression model[5] at each time step $t \in \mathcal{T}$ to solve the hazard classification problem utilizing only "at-risk" samples whose time-to-event/censoring is at or after t. Formally, the logistic regression models are trained based on the log-loss. When there are treatments, we use LR-sep in a two-model (T-learner) approach by training a separate model using samples in the treated ($A = 1$) and controlled ($A = 0$) groups, respectively.

- **CSA**[6] [89]: We use the CSA-INFO model of [89], where we use its generative capabilities to approximate target quantities via monte-carlo sampling. We use the code and specifications provided by the authors, in particular we use a hidden dimension of 100, set the imbalance penalty $\alpha = 100$ and train for 300 epochs. To create monte carlo approximations, we sample 1000 times from the model for each observation in the test set.

[4] Python package `scikit-survival` [118]

[5] Python package `scikit-learn`

[6] https://github.com/paidamoyo/counterfactual_survival_analysis

- **SurvITE (CFR-1)** and **SurvITE (CFR-2)**: We consider two variants of SurvITE by replacing our $\mathcal{L}_{ipm}(\theta_\phi)$ with a balancing term based on the CFRNet[7] proposed in [72]:
  - **SurvITE (CFR-1)** creates a representation balancing treatment groups at baseline only which is formally given as:

$$\mathcal{L}_{ipm}(\theta_\phi) = Wass\big(\{\Phi(x_i)\}_{i:a_i=1}, \{\Phi(x_i)\}_{i:a_i=0}\big) \tag{4.8}$$

  - **SurvITE (CFR-2)** creates a representation optimizing for balance of treatment groups *at each time step*

$$\mathcal{L}_{ipm}(\theta_\phi) = \sum_{t=1}^{t_{\max}} Wass\big(\{\Phi(x_i)\}_{i:\tilde{\tau}_i \geq t, a_i=1}, \{\Phi(x_i)\}_{i:\tilde{\tau}_i \geq t, a_i=0}\big) \tag{4.9}$$

  Note that, in both variants, there is no balancing towards $\mathbb{P}_0$. We implement SurvITE (CFR-1) and SurvITE (CFR-2) with the same network architecture and hyper-parameters with those of SurvITE.

4.6.2 Performance Metrics

Once SurvITE (or SurvIHE) is trained, we can simply estimate the (treatment-specific) survival function based on the estimated hazard functions as the following:

$$\hat{S}^a(\tau|x) = \prod_{t \leq \tau} \big(1 - h_{a,t}(\Phi(x))\big) \qquad \text{for} \;\; a \in \{0,1\}. \tag{4.10}$$

Heterogeneous Treatment Effects. For synthetic experiments where we have the ground-truth treatment-specific survival functions i.e., $S^1(\tau|x)$ and $S^0(\tau|x)$, we evaluate $HTE_{surv}(\tau|x) = S^1(\tau|x) - S^0(\tau|x)$ and $HTE_{rmst}(x; L) = \sum_{t_k \leq L} \big(S^1(t_k|x) - S^0(t_k|x)\big) \cdot (t_k -$

[7] https://github.com/clinicalml/cfrnet

t_{k-1}) in terms of the averaged root mean squared error (RMSE) of the estimation:

$$\epsilon_{HTE_{surv}}(t) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(HTE_{surv}(t|x_i) - \widehat{HTE}_{surv}(t|x_i)\big)^2}, \tag{4.11}$$

$$\epsilon_{HTE_{rmst}}(L) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(HTE_{rmst}(x_i;L) - \widehat{HTE}_{rmst}(x_i;L)\big)^2}. \tag{4.12}$$

Here $\widehat{HTE}_{surv}(t|x) = \hat{S}^1(\tau|x) - \hat{S}^0(\tau|x)$ and $\widehat{HTE}_{rmst}(x;L) = \sum_{t_k \leq L}\big(\hat{S}^1(t_k|x) - \hat{S}^0(t_k|x)\big) \cdot (t_k - t_{k-1})$ where $(t_k - t_{k-1})$ may vary depending on how the continuous time is discretized (e.g., non-uniform time intervals for the Twins dataset).

For semi-synthetic experiments where we have the ground-truth treatment-specific time-to-event outcomes but not the treatment-specific survival functions, we only report $\epsilon_{HTE_{rmst}}(L)$ in (4.12) where the ground-truth $HTE_{rmst}(x;L)$ is defined in terms of the ground-truth time-to-event outcomes, i.e., $HTE_{rmst}(x;L) = (\min(T(1),L) - \min(T(0),L))$ where $T(1)$ and $T(0)$ are the time-to-event given $a = 1$ and $a = 0$, respectively.

(Treatment-Specific) Survival Functions. For evaluating the estimation performance of the (treatment-specific) survival functions, we evaluate the averaged RMSE of these estimations as the following:

$$\epsilon_{S^a}(t) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(S^a(t|x_i) - \hat{S}^a(t|x_i)\big)^2}. \tag{4.13}$$

Discriminative Performance. For assessing the survival predictions of all the survival models with respect to how well the predictions discriminate among individual risks, we use the concordance index (C-Index) [119]:

$$C(t) = \mathbb{P}\big(\hat{S}(t|x_i) < \hat{S}(t|x_j)\big|\tilde{\tau}_i < \tilde{\tau}_j, \tilde{\tau}_i \leq t, \delta_i = 1\big) \tag{4.14}$$

where $\hat{S}(t|x) = a \cdot \hat{S}^1(t|x) + (1-a) \cdot \hat{S}^0(t|x)$ is the survival prediction given treatment a. The resulting C-Index in (4.13) tells us how well the given survival model discriminates the individual risks among the events that occur before or at time t.

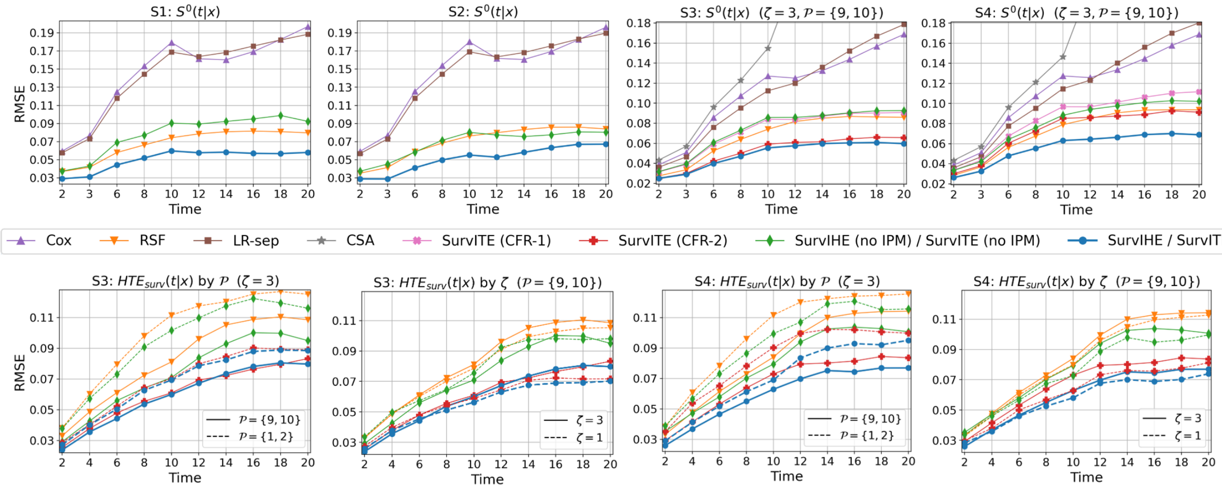


Figure 4.3: RMSE of estimating the survival function $S^0(t|x)$ (top) and the treatment effect $HTE_{surv}(t|x)$ (bottom) for different time steps across synthetic settings. Averaged across 5 runs.

4.6.3 Synthetic Experiments

We consider a range of synthetic simulation setups (S1-S4) to highlight and isolate the effects of the different types of covariate shift. As event and censoring processes, we use

$$\lambda^a(t|x) = \begin{cases} 0.1\sigma(-5x_1^2 - a \cdot (\mathbb{1}\{x_3 \geq 0\} + 0.5)) & \text{for } t \leq 10 \\ 0.1\sigma(10x_2 - a \cdot (\mathbb{1}\{x_3 \geq 0\} + 0.5))) & \text{for } t > 10 \end{cases}, \qquad \lambda_C(t|x) = 0.01\sigma(10x_4^2)$$

with treatment assignment mechanism $a \sim \texttt{Bern}(\xi \cdot \sigma(\sum_{p \in \mathcal{P}} x_p))$, with σ the sigmoid function. Additionally, we assume administrative censoring at $t = 30$ throughout, i.e., $\lambda_C(30|x) = 1$, marking e.g. the end of a hypothetical clinical study. Covariates are generated from a 10-dimensional multivariate normal distribution with correlations, i.e. $X \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ where $\mathbf{\Sigma} = (1 - \rho)\mathbf{I} + \rho \mathbf{1}\mathbf{1}^\top$ with $\rho = 0.2$. We use 5000 independently generated samples each for training and testing.

In S1, we begin with the simplest case – *no* treatments and *no* censoring – using only $\lambda^0(t|x)$ to generate events, considering only event-induced shift (Shift 3). In S2, we introduce informative censoring using $\lambda_C(t|x)$ (Shift 2+3). In S3, we use treatments and consider biased treatment assignment (without censoring) (Shift 1+3). In S4, we consider the most difficult case with all three types of shift (Shift 1+2+3). In the latter two settings, we vary treatment selection by changing (i) whether the covariate set overlaps with the event-inducing covariates ($\mathcal{P}$=$\{1, 2\}$) or not ($\mathcal{P}$=$\{9, 10\}$) and (ii) the selection strength $\xi \in \{1, 3\}$.

Fig. 4.3 (top) shows performance on estimating $S^0(t|x) = \prod_{k\leq t}\big(1 - \lambda^0(k|x)\big)$ for all scenarios and methods, while Fig. 4.3 (bottom) shows performance on estimating the difference in survival functions ($HTE_{surv}(t|x)$) for a selection of methods. In Table 4.1, we further evaluate the estimation of differences in RMST ($\text{HTE}_{rmst}(x)$). We observe that SurvITE (/SurvIHE) performs best throughout, and that introduction of the IPM term leads to substantial improvements across all scenarios. In S1 with only event-induced covariate shift and in S3/4 when treatment selection and event-inducing covariates overlap ($\mathcal{P}$=$\{1, 2\}$), balancing cannot remove all shift as the shift-inducing covariates are predictive of outcome; however, even here the IPM-term helps as it encourages dropping other covariates (which appear imbalanced due to correlations in X). As expected, both Cox and LR-sep do not perform well as they are misspecified, while the nonparametric RSF is sufficiently flexible to capture the underlying DGP and usually performs similarly to SurvITE (architecture only), but is outperformed once the IPM term is added.

A comparison with ablated versions highlights the effect of using the appropriate baseline population to define balance; naive balancing across treatment arms (either at baseline – SurvITE(CFR-1), or over time – SurvITE(CFR-2)) is not as effective as using the baseline population as a target, especially at the later time steps where the effects of bias worsen. While SurvITE(CFR-2) almost matches the performance of the full SurvITE in S3, it performs considerably worse in S4, indicating that this form of balancing suffers mainly due to its ignorance of censoring. Finally, a comparison with CSA highlights the value of modeling hazard functions directly: we found that Monte-Carlo approximation of the survival function using the generated event times gives very badly calibrated survival curves as event times generated by CSA were concentrated in a very narrow interval, leading to survival estimates of

Table 4.1: RMSE on estimation of $\text{HTE}_{rmst}(x)$ (mean ± 95%-CI) for different times for the SYNTHETIC dataset (L values are selected as the 25th and 75th percentiles of event times).

Methods	**S3** ($\zeta = 3$, no overlap)		**S4** ($\zeta = 3$, no overlap)	
	$L = 10$	$L = 20$	$L = 10$	$L = 20$
Cox	0.434±0.03	1.073±0.05	0.424±0.02	1.047±0.04
RSF	0.328±0.02	1.027±0.03	0.332±0.02	1.058±0.03
LR-sep	0.412±0.02	1.111±0.07	0.418±0.02	1.149±0.04
CSA	0.421±0.01	2.098±0.26	0.406±0.01	1.932±0.12
SurvITE (no IPM)	0.275±0.04	0.843±0.11	0.310±0.05	0.930±0.11
SurvITE (CFR-1)	0.269±0.04	0.825±0.09	0.341±0.02	1.016±0.10
SurvITE (CFR-2)	0.236±0.04	0.691±0.08	0.294±0.07	0.815±0.15
SurvITE	**0.225±0.03**	**0.687±0.08**	**0.237±0.03**	**0.703±0.06**

0 and 1 elsewhere. Its performance on estimation of RMST was likewise poor; we conjecture that this is due to (i) CSA modeling continuous time, while the outcomes were generated using a coarse discrete time model, and (ii) the significant presence of administrative censoring.

4.6.4 Real-World Dataset: TWINS

Finally, we consider the TWINS benchmark dataset, containing survival times (in days, administratively censored at t=365) of 11,400 pairs of twins, which is used in [101, 117] to measure HTEs of birthweight on infant mortality. We split the data 50/50 for training and testing (by twin pairs), and similar to [101], use a covariate-based sampling mechanism to select only one twin for training to emulate selection bias. Further, we consider a second setting where we additionally introduce covariate-dependent censoring. For all discrete-time models, we use a non-uniform discretization to construct classification tasks because

Table 4.2: RMSE on estimation of $\text{HTE}_{rmst}(x)$ (mean ± 95%-CI) for different times for the TWINS dataset (L values are selected as the 75th and 95th percentiles of event times).

Methods	TWINS (no censoring)		TWINS (censoring)	
	$L = 30$	$L = 180$	$L = 30$	$L = 180$
Cox	2.85±0.10	20.33±0.50	2.88±0.09	20.60±0.50
RSF	3.15±0.07	22.42±0.36	3.18±0.08	22.62±0.46
LR-sep	2.94±0.10	20.60±0.53	2.94±0.10	20.66±0.52
CSA	3.42±0.12	26.20±1.21	4.41±0.54	47.79±1.55
SurvITE (no IPM)	2.80±0.10	19.80±1.01	2.85±0.22	20.00±1.07
SurvITE (CFR-1)	2.68±0.06	19.16±0.37	2.67±0.15	19.10±0.85
SurvITE (CFR-2)	2.61±0.12	18.69±0.64	2.69±0.22	19.20±1.44
SurvITE	**2.53±0.09**	**18.34±0.70**	**2.63±0.10**	**18.76±0.56**

most events are concentrated in the first weeks. To create an observational time-to-event dataset, we selectively observed one of the two twins **(no censoring)** with selection bias and **(censoring)** with both selection bias and censoring bias as follows: the treatment assignment is given by $a|x \sim \texttt{Bern}(\sigma(w_1^\top x + e))$ where $w \sim \texttt{Uniform}(-0.1, 0.1)^{39\times 1})$ and $e \sim \mathcal{N}(0, 1^2)$, and the time-to-censoring is given by $C \sim \texttt{Exp}(100 \cdot \sigma(w_2^\top x))$ where $w_2 \sim \mathcal{N}(0, 1^2)$. As the data is real and ground truth probabilities are unknown, $\text{HTE}_{rmst}(x)$ is suited best to evaluate performance on estimating effect heterogeneity. The results presented in Table 4.2 largely confirm our findings on relative performance in the synthetic experiments; only RSF performs relatively worse on this dataset.

4.7 Conclusion

We studied the problem of inferring heterogeneous treatment effects from time-to-event data by focusing on the challenges inherent to treatment-specific hazard estimation. We found that a variety of covariate shifts play a role in this context, theoretically analysed their impact, and demonstrated across a range of experiments that our proposed method SurvITE successfully mitigates them.

Limitations. Like all methods for inferring causal effects from observational data, SurvITE relies on a set of strong assumptions which should be evaluated by a domain expert prior to deployment in practice. Here, the time-to-event nature of our problem adds an additional assumption ('random censoring') to the standard 'no hidden confounders' assumption in classical treatment effect estimation. If such assumptions are not properly assessed in practice, any causal conclusions may be misleading.

Part II

Modeling Disease Progression for Longitudinal Data

CHAPTER 5

A Deep Learning Approach for Dynamic Survival Analysis based on Longitudinal Data

5.1 Introduction

Survival analysis informs our understanding of the relationships between the (distribution of) first hitting times of events of interest (such as death, onset of a certain disease, etc.) and the covariates, and enables us to issue corresponding risk assessments for such events. Clinicians use survival analysis to make screening decisions or to prescribe treatments, while patients use the information about their clinical risks to adjust their lifestyles in order to mitigate such risks. Since the Cox proportional hazard model [19] was first introduced, a variety of methods have been developed for survival analysis, ranging from statistical models to deep learning techniques [20, 22, 23, 30, 34, 50, 120].

A key limitation of existing survival models is that they utilize only a small fraction of the available longitudinal (repeated) measurements of biomarkers and other risk factors. In particular, even though biomarkers and other risk factors are measured repeatedly over time, survival analysis is typically based on the last available measurement. This represents a severe limitation, since the evolution of biomarkers and risk factors has been shown to be informative in predicting the onset of disease and various risks. For example, Cystic Fibrosis (CF), which is the most common genetic disease in Caucasian populations [121], gives rise to different forms of dysfunction involving the respiratory and gastrointestinal systems, which primarily lead to progressive respiratory failure [6, 7]. Forced expiratory volume (FEV$_1$), and its development, is a crucial biomarker in assessing the severity of CF as it allows clinicians to describe the progression of the disease and to anticipate the occurrence of respiratory failures [7, 8]. Therefore, to provide a better understanding of disease progression, it is essential to incorporate longitudinal measurements of biomarkers and risk factors into a model. Rather than discarding valuable information recorded over time, this allows us to make better risk assessments on the clinical events.

This work presents a deep neural network, which we call *Dynamic-DeepHit*, that extends our previous work in [50] to dynamic survival analysis. Dynamic-DeepHit learns, on the basis of the available longitudinal measurements, a data-driven distribution of first hitting times of competing events. Thus, the proposed method completely removes the need for explicit model specifications (i.e., no assumption about the form of the underlying stochastic processes are made) and learns the complex relationships between trajectories and survival probabilities. An important aspect of our method is that it naturally handles situations in which there are multiple competing risks where more than one type of event plays a role in the survival setting. (Competing risks are not independent and must be treated jointly; for example, [16] has shown that various treatments for breast cancer increase the risk of a cardiovascular event. See [20, 50] for details of existing survival models that address competing risks.)

To enable dynamic survival analysis with longitudinal time-to-event data, Dynamic-DeepHit employs a shared subnetwork and a family of cause-specific subnetworks. The shared subnetwork encodes the information in longitudinal measurements into a fixed-length vector (i.e., a context vector) using a recurrent neural network (RNN), which has achieved a great success in various applications handling time-series data (e.g, machine translation [122], image caption generation [123], and speech recognition [124]). We employ a temporal attention mechanism [125] in the hidden states of the RNN structure when constructing the context vector. This renders Dynamic-DeepHit to access the necessary information, which has progressed along with the trajectory of the past longitudinal measurements, by paying attention to relevant hidden states across different time stamps. Then, the cause-specific subnetworks take the context vector and the last measurements as an input and estimate the joint distribution of the first hitting time and competing events that is further used for risk predictions.

To demonstrate the usefulness of our method, we compare its performance with that of competing approaches using a longitudinal data which was collected by the UK Cystic Fibrosis Registry. This data contains a cohort of 5,883 adult patients (from age 18 onwards) suffering from CF, who had annual follow-ups between 2009-2015. Throughout the evaluation, we define two competing events: death from respiratory failures and that from other causes. It is essential to jointly account for competing risks to take preventative steps for CF patients: CF is a systemic disease which gives rise to different forms of dysfunctions in multiple systems and organs – CF-associated liver disease has been reported as the third most frequent cause of death [126]. We show that our method achieves significant improvements on the discriminative performance over the state-of-the-art methods and provides the calibration performance that was comparable to the best performing benchmarks. Particularly, Dynamic-DeepHit achieved improvements of 4.36% and 9.67% over the best benchmark (6.26% and 14.97% over the joint model) on average in terms of discriminative performance for death from respiratory failure and death from other causes, respectively. In addition, while the vast majority of clinical literature has focused on spirometric biomarkers, e.g., $\text{FEV}_1\%$ predicted[1], as the main CF risk factors, Dynamic-DeepHit confirmed the importance of the history of intravenous antibiotic treatments and nutritional status in the risk assessment of CF patients.

5.2 Related Work

We start by noting that in this work we focus on dynamic survival analysis with competing risks outside the hospital, where the measurements are sparse and irregular, and a disease

[1] $\text{FEV}_1\%$ predicted is a ratio of the maximum volume of air blown out during lung function test to the predicted value for a 'normal' person of the similar age, sex, and body composition in percentage.

develops or progresses over the duration of months or even years. Hence, our work differs from existing work on predicting risks in the hospital setting, where numerous measurements are available and a patient is recovering or deteriorating over the course of a few hours or possibly days. For instance, with chronic diseases such as CF, patients are followed up over the span of years, usually as part of regular physical examinations. The clinical status of the patient also evolves slowly, allowing for the development of related comorbidities (e.g. CF-induced diabetes), which in turn affect key biomarkers that reflect a patient's clinical status and rate of deterioration, such as lung function scores (e.g., $\text{FEV}_1\%$ predicted) in CF. Thus, we examine related work on dynamic survival analysis that utilizes measurements collected repeatedly, but infrequently, outside the hospital.

The most widely used dynamic survival methods in this setting are joint models which jointly describe both longitudinal and survival processes [13, 127–132]. In particular, a joint model comprises two sub-models – one for repeated measurements of the longitudinal process and the other for the time-to-event data (e.g., typically, a linear mixed model and a Cox model) – linking them using a function of shared random effects. Overall, joint models find to learn a full representation of the joint distribution of the longitudinal time-to-event data. From a dynamic prediction perspective, the full representation of joint models leads to a reduced bias in estimation [127] providing flexibility to make predictions at any time points of interest. However, learning such full representation requires an optimization of the joint likelihood and relies on fixed model specifications for both processes. Thus, model mis-specifications (e.g., the assumption on longitudinal process and proportional hazard assumption on time-to-event) will limit the overall performance and the optimization of the joint likelihood requires severe computational challenges when applied to high-dimensional datasets [130]. Nonparametric specification of the longitudinal process was previously explored in [128] and [129], which models the longitudinal process via individual-level penalized splines and cubic B-splines, respectively, at the cost of higher computational complexity. Joint models integrating latent classes [131, 132] have been recently developed to account for heterogeneous population.

However, these approaches still maintain a proportional hazard assumption which we refrain from doing by adopting deep learning.

Landmarking is another approach for dynamic survival analysis on the basis of longitudinal data [12, 133–135]. The basic idea behind landmarking is to build a survival model (e.g., a Cox model), fitted to the subjects from the original dataset who are still at risk at the landmarking time (usually, the prediction time of the interest). Thus, landmarking is "partially conditional" since each survival model is conditioned on the available information accrued by the corresponding landmarking time, rather than incorporating the entire longitudinal history, and predictions on survival probabilities are issued using the last measurements as an estimate of biomarkers at the landmarking time. Even though longitudinal measurements are not fully explored, it is shown that, in practice, landmarking is competitive with joint models and significantly easier to implement [135]. However, landmarking is not fully dynamic; survival predictions are only available at the predefined landmarking times, not at times at which new measurements are obtained. Moreover, it makes assumptions about the underlying stochastic process for the survival model, which may not be true in practice, thereby limiting the model in terms of learning the relationships between the covariates and events of interest. Lastly, it only incorporates a subset of the longitudinal history up to the landmarking time, which may result in information loss when making predictions.

Deep networks have been shown to achieve significantly improved performance in survival analysis [22, 23, 34, 50, 120] owing to the ability to represent complicated associations between features and outcomes. Authors in [22, 23] have employed deep neural networks for modeling non-linear representations of the relationships between covariates and the risk of a single clinical event. However, these networks are limited to the conventional Cox proportional hazard assumption without addressing time-dependent influences of covariates on the time-to-event. Recently, deep networks have been utilized to develop a nonparametric Bayesian model using the Gaussian process [34], to construct the tree-based Bayesian mixture model [120], and to directly learn the distribution of survival times [50] for survival analysis with competing risks. However, all of these methods provide only static survival analysis: they use only the current information to perform the survival predictions and most of the works focus on a single risk rather than multiple risks. To our best knowledge, this work is the first to investigate a deep learning approach for dynamic survival analysis with competing risks on the basis of repeated measurements (longitudinal data).

5.3 Problem Formulation

5.3.1 Time-to-Event Data

Time-to-event (survival) data provides three pieces of information for each subject: i) observed covariates, ii) time-to-event(s), and iii) a label indicating the type of event (e.g., death or adverse clinical event) including right-censoring. Observed covariates include static (time-invariant) and time-varying covariates that are recorded for a period of time. We suppose that the longitudinal measurement times, event times, and censoring times are aligned based on a synchronization event, such as the entry to a clinical trial, the date of an intervention, and the onset of a condition.

Formally, for each subject i, a sequence of longitudinal observations until time t is described as a d_x-dimensional multivariate time-series $\mathcal{X}^i(t) = \{\mathbf{x}^i(t^i_j) : 0 \leq t^i_j \leq t \text{ for } j = 1, \cdots, J^i\}$, where $\mathbf{x}^i(t_j)$ can be simplified as $\mathbf{x}^i_j = [x^i_{j,1}, \cdots, x^i_{j,d_x}]$ which includes both static and time-varying covariates recorded at time t_j. Covariates are not necessarily measured at regular time intervals and not every covariate is observed at each measurement (i.e., partially missing). Thus, we i) distinguish notations between time stamps $j = 1, \cdots, J^i$ and the corresponding actual times $t^i_j = t^i_1, \cdots, t^i_{J^i}$ and ii) set $x^i_{j,d} = *$ to denote that the d-th element of $\mathbf{x}^i_j$ was not measured (otherwise, $\mathbf{x}^i_j \in \mathbb{R}$). For notational simplicity, we use $\mathcal{X}^i = \mathcal{X}^i(t^i_{J^i})$ to denote a whole set of longitudinal observations available for subject i until the last measurement time $t^i_{J^i}$ of that subject.

We treat survival time as discrete (e.g., a temporal resolution of one month) and the time horizon as finite (e.g., no patients lived longer than 100 years). Thus, a set of possible survival times is denoted as $\mathcal{T} = \{0, 1, \cdots, T_{\max}\}$ where $T_{\max}$ is a predefined maximum time horizon. Discretization is performed by transforming continuous-valued times into a set of contiguous time intervals, i.e., $T = \tau$ implies $T \in [\tau, \tau + \delta t)$ where δt implies the temporal resolution. We assume that every subject experiences exactly one event among $K \geq 1$ possible events of interest within $\mathcal{T}$. (We cannot observe the occurrence of the other events once one event is observed.) For instance, a patient eventually dies, but can die from only one cause [35]. This includes cause-specific deaths due to CF, where deaths from other causes are competing risks for death due to respiratory failure. Survival data is frequently right-censored because events of interest are not always observed (i.e., subjects are lost to follow-up). The set of possible events is $\mathcal{K} = \{\varnothing, 1, 2, \cdots, K\}$, with $\varnothing$ denoting right-censoring. Throughout this work, we assume that censoring is *uninformative*. This assumption is common in the survival literature and implies that whether a subject withdraws from the study depends only on the observed history but not on the clinical outcomes [12, 13, 128, 133, 136].

We consider a dataset $\mathcal{D} = \{(\mathcal{X}^i, \tau^i, k^i)\}_{i=1}^{N}$ comprising survival data for N subjects who have been followed up for a certain amount of time. Here, $\tau^i = \min(T^i, C^i)$ is the time-to-event with $T^i \in \mathcal{T}$ and $C^i \in \mathcal{T}$ indicating the event and the censoring times, respectively, and $k^i \in \mathcal{K}$ being the event or censoring that occurred at time τ^i. Note that τ is either the time at which an event (e.g., death) occurred or the time at which the subject was censored (e.g., disappeared from follow-up); in either case, the subject was known to experience no event at times prior to τ. Figure 5.1 depicts a survival dataset comprising histories of longitudinal measurements with different numbers of measurements at irregular time intervals, where each subject experiences either event type 1 or type 2, or has its endpoint censored.

5.3.2 Cumulative Incidence Function

Our goal is to analyze the cause-specific risk given the history of observations over time and to issue dynamic risk predictions when new measurements are available. To do so, we use the

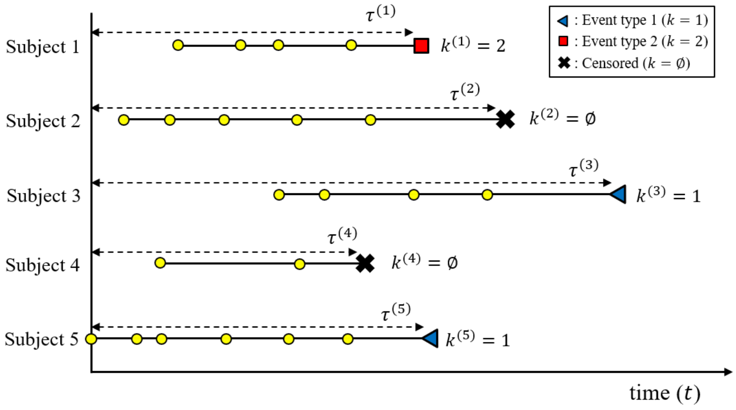
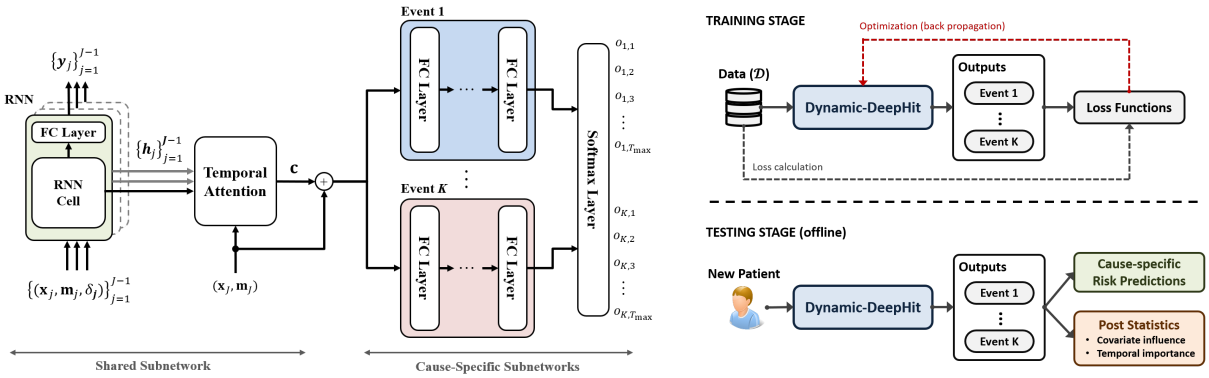


Figure 5.1: An illustration of survival data with longitudinal measurements where subjects are aligned based on the synchronization event. Colored dots indicate the times at which longitudinal measurements are observed.



(a) The network architecture with K competing risks.

(b) A schematic depiction

Figure 5.2: An illustration of (a) the network architecture of Dynamic-DeepHit with K competing risks and (b) a schematic depiction of the network at training/testing stages.

cause-specific cumulative incidence function (CIF) which is key to survival analysis under the presence of competing risks. As defined in [20], the CIF expresses the probability that a particular event $k^* \in \mathcal{K}$ occurs on or before time τ^* conditioned on the history of longitudinal measurements $\mathcal{X}^*$. The fact that longitudinal measurements have been recorded up to $t^*_{J^*}$

implies survival of the subject up to this time point. Thus, the CIF is defined as follows:

$$F_{k^*}(\tau^*|\mathcal{X}^*) \triangleq P(T \leq \tau^*, k = k^*|\mathcal{X}^*, T > t^*_{J^*}) = \sum_{\tau \leq \tau^*} P(T = \tau, k = k^*|\mathcal{X}^*, T > t^*_{J^*}). \quad (5.1)$$

Whenever a new measurement is recorded for this subject at time $t > t^*_{J^*}$, we can update (5.1) accounting for that information in a dynamic fashion.

Similarly, the survival probability of a subject at time τ^* given $\mathcal{X}^*$ can be derived by

$$S(\tau^*|\mathcal{X}^*) \triangleq P(T > \tau^*|\mathcal{X}^*, T > t^*_{J^*}) = 1 - \sum_{k \neq \varnothing} F_k(\tau^*|\mathcal{X}^*). \quad (5.2)$$

However, the *true* CIF, $F_{k^*}(\tau^*|\mathcal{X}^*)$, is not known; we utilize the *estimated* CIF, $\hat{F}_{k^*}(\tau^*|\mathcal{X}^*)$, in order to perform dynamic risk prediction of event occurrences and to assess how models discriminate between cause-specific risks among subjects. The estimated CIF will be described in the next section.

5.4 Method: Dynamic-DeepHit

In this section, we describe our novel Dynamic-DeepHit architecture for survival analysis with competing risks on the basis of longitudinal measurements. We seek to train the network to learn an estimate of the joint distribution of the first hitting time and competing events given the longitudinal observations. This representation is then used to estimate the cause-specific CIFs (5.1) and survival probability (5.2).

Before describing the network architecture in detail, we redefine the history of longitudinal measurements in order to provide the information on measurement times and missing observations to the network as described in the previous section. Let $\mathcal{X}^i = (\mathbf{X}^i, \mathbf{M}^i, \Delta^i)$ where $\mathbf{X}^i = \{\mathbf{x}^i_1, \cdots, \mathbf{x}^i_{J^i}\}$, $\mathbf{M}^i = \{\mathbf{m}^i_1, \cdots, \mathbf{m}^i_{J^i}\}$ which is a sequence of mask vectors that indicate which covariates are missing, and $\Delta^i = \{\delta^i_1, \delta^i_2 \cdots, \delta^i_{J^i}\}$ which is a sequence of time intervals between two adjacent measurements. Here, $\mathbf{m}^i_j = [m^i_{j,1}, \cdots, m^i_{j,d_x}]$ with $m^i_{j,d} = 1$ if $x^i_{j,d} = *$ and $m^i_{j,d} = 0$ otherwise, and δ^i_j implies the actual amount of time that has elapsed until the next measurements are collected, i.e., $\delta^i_j = t^i_{j+1} - t^i_j$ for $1 \leq j < J^i$, and $\delta^i_{J^i} = 0$. Then, the entire training set can be given as a set of tuples $\mathcal{D} = \{(\mathbf{X}^i, \mathbf{M}^i, \Delta^i, \tau^i, k^i)\}_{i=1}^N$.

5.4.1 Network Architecture

Dynamic-DeepHit is a multi-task network, which consists of two types of subnetworks: a shared subnetwork that handles the history of longitudinal measurements and predicts the next measurements of time-varying covariates, and a set of cause-specific subnetworks which estimates the joint distribution of the first hitting time and competing events. As the multi-task learning has been successful across different applications [36, 137–139], we jointly optimize the two subnetworks to help the overall network capture associations between the time-to-event under competing risks and i) the static covariates and ii) the progression of underlying process that governs the time-varying covariates. Figure 5.2 illustrates (a) the overall architecture of Dynamic-DeepHit which comprises a shared subnetwork and K cause-specific subnetworks and (b) the conceptual framework of the proposed network at training/testing stages. Throughout this subsection, we omit the dependence on i for ease of notation.

5.4.1.1 Shared Subnetwork

The shared subnetwork consists of two components: i) a **RNN structure** to flexibly handle the longitudinal data with each subject having different numbers of measurements, that are captured at irregular time intervals and are partially missing and ii) an **attention mechanism** to unravel the temporal importance of the history of measurements in making risk predictions. For each time stamp $j = 1, \cdots J - 1$, the RNN structure takes a tuple of $(\mathbf{x}_j, \mathbf{m}_j, \delta_j)$ as an input and outputs $(\mathbf{y}_j, \mathbf{h}_j)$, where $\mathbf{y}_j$ is the estimate of time-varying covariates after time δ_j has elapsed, i.e., $\mathbf{x}_{j+1}$[2] and $\mathbf{h}_j$ is the hidden state at time stamp j. Utilizing the Gated Recurrent Unit (GRU) RNN [140], $\mathbf{h}_j$ can be derived as follows:

$$\begin{aligned}\mathbf{z}_j &= \sigma(W_z\mathbf{h}_{j-1} + U_z[\mathbf{x}_j\ \mathbf{m}_j\ \delta_j] + \mathbf{b}_z),\\ \mathbf{r}_j &= \sigma(W_r\mathbf{h}_{j-1} + U_r[\mathbf{x}_j\ \mathbf{m}_j\ \delta_j] + \mathbf{b}_r),\\ \tilde{\mathbf{h}}_j &= \tanh(W_h(\mathbf{r}_j \odot \mathbf{h}_{j-1}) + U_h[\mathbf{x}_j\ \mathbf{m}_j\ \delta_j] + \mathbf{b}_h),\\ \mathbf{h}_j &= (1-\mathbf{z}_j) \odot \mathbf{h}_{j-1} + \mathbf{z}_j \odot \tilde{\mathbf{h}}_j,\end{aligned} \tag{5.3}$$

where W, U, and $\mathbf{b}$ are weight matrices and vectors which parameterize the shared subnetwork, $\odot$ is element-wise multiplication, and $\sigma(\cdot)$ is the sigmoid function. Note that we illustrate the subnetwork with GRUs but other RNNs, such as vanilla RNNs, LSTMs [141], and bidirectional RNNs [142], can be also utilized.

The temporal attention mechanism [125] on the hidden states helps our network decide which parts of the previous longitudinal measurements to pay attention to. Formally, it outputs a context vector, $\mathbf{c}$, as an weighted sum of the previous hidden states as follows:

$$\mathbf{c} = \sum_{j=1}^{J-1} a_j \mathbf{h}_j, \tag{5.4}$$

where $a_j = \frac{\exp(e_j)}{\sum_{\ell=1}^{J-1}\exp(e_\ell)}$ represents the importance of the j-th measurements. Here, $e_j = f_a(\mathbf{h}_j, \mathbf{x}_J, \mathbf{m}_J)$ is used to score the importance of the j-th measurement by referencing on the last measurement, $(\mathbf{x}_J, \mathbf{m}_J)$. We set $f_a(\cdot)$ as a two-layer feed-forward network that takes the hidden state at time stamp j, $\mathbf{h}_j$, and the tuple of $(\mathbf{x}_J, \mathbf{m}_J)$ as the input and outputs a scalar e_j for $j = 1, \cdots, J-1$. The temporal mechanism is jointly trained with all the other components of our network.

[2] The time elapsed until the next-time measurements is available since the shared subnetwork only takes the past measurements as inputs.

5.4.1.2 Cause-specific Subnetworks

Each cause-specific subnetwork utilizes a feed-forward network composed of fully-connected layers to capture relations between the cause-specific risk and the history of measurements. The inputs to these subnetworks is the context vector of the shared subnetwork. This gives the subnetworks access to the learned common representation of the longitudinal history, which has progressed along with the trajectory of the past longitudinal measurements, by paying attention to relevant hidden states across the time stamps. Overall, each cause-specific subnetwork captures the latent patterns that are distinct to each competing event. Formally, the k-th cause-specific subnetwork takes as input the vector $\mathbf{c}$ and the last measurement $(\mathbf{x}_J, \mathbf{m}_J)$ and outputs a vector, $f_{c_k}(\mathbf{c}, \mathbf{x}_J, \mathbf{m}_J)$.

5.4.1.3 Output Layer

Dynamic-DeepHit employs a soft-max layer in order to summarize the outcomes of each cause-specific subnetwork, $f_{c_1}(\cdot), \cdots, f_{c_K}(\cdot)$, and to map into a proper probability measure. Overall, the network produces an estimated joint distribution of the first hitting time and competing events. In particular, given a subject with $\mathcal{X}^*$, each output node represents the probability of having event k at time τ, i.e., $o^*_{k,\tau} = \hat{P}(T = \tau, k = k|\mathcal{X}^*)$. Therefore, we can define the estimated CIF for cause k^* at time τ^* as follows:

$$\hat{F}_{k^*}(\tau^*|\mathcal{X}^*) = \frac{\sum_{t^*_{J^*} < \tau \leq \tau^*} o^*_{k^*,\tau}}{1 - \sum_{k \neq \varnothing} \sum_{n \leq t^*_{J^*}} o^*_{k,n}}. \tag{5.5}$$

Note that (5.5) is built upon the condition that this subject has survived up to the last measurement time.

5.4.2 Training Dynamic-DeepHit

To train Dynamic-DeepHit, we minimize a total loss function $\mathcal{L}_{\text{total}}$ that is specifically designed to handle longitudinal measurements and right-censoring. The total loss function is the sum of three terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3, \tag{5.6}$$

where $\mathcal{L}_1$ is the negative log-likelihood of the joint distribution of the first hitting time and events, which is necessary to capture the first hitting time in the right-censored data, and $\mathcal{L}_2$ and $\mathcal{L}_3$ are utilized to enhance the overall network. More specifically, $\mathcal{L}_2$ combines cause-specific ranking loss functions to concentrate on discriminating estimated individual risks for each cause, and $\mathcal{L}_3$ incorporates the prediction error on trajectories of time-varying covariates to capture the hidden representations of the longitudinal history and to regularize the network.

5.4.2.1 Log-likelihood Loss

The first loss function is the negative log-likelihood of the joint distribution of the first hitting time and corresponding event considering the right-censoring [15], which is extended to the survival setting where the history of longitudinal measurements and K competing risks are available. More specifically, for a subject who *is not censored*, it captures both the event that occurs and the time at which the event occurs; for a subject who *is censored*, it captures the time at which the subject is censored (lost to follow-up) in both cases conditioned on the longitudinal measurements recorded until the last observation. We define $\mathcal{L}_1$ as follows:

$$\mathcal{L}_1 = -\sum_{i=1}^{N} \left[\mathbb{1}(k^i \neq \varnothing) \cdot \log\left(\frac{o^i_{k^i,\tau^i}}{1 - \sum_{k\neq\varnothing} \sum_{n \leq t^i_{J^i}} o^i_{k,n}} \right) + \mathbb{1}(k^i = \varnothing) \cdot \log\left(1 - \sum_{k\neq\varnothing} \hat{F}_k(\tau^i | \mathcal{X}^i) \right) \right], \tag{5.7}$$

where $\mathbb{1}(\cdot)$ is the indicator function. The first term captures the information provided by uncensored subjects. The second term follows from the knowledge that they are alive at the censoring time, and so the first hitting time of each event $k \in \mathcal{K}$ occurs after the given censoring time; see [38].

5.4.2.2 Ranking Loss

The second loss function incorporates estimated CIFs calculated at different times (i.e., the time at which an event actually occurs) in order to fine-tune the network to each cause-specific estimated CIF. To do so, we utilize a ranking loss function which adapts the idea of concordance [25]: a subject who dies at time τ should have a higher risk at time τ than a subject who survived longer than τ. However, the longitudinal measurements of subjects can begin at any point in their lifetime or disease progression [31], and this makes direct comparison of the risks at different time points difficult to assess. Thus, we compare the risks of subjects at times elapsed since their last measurements, that is, for subject i, we focus on $s^i = \tau^i - t^i_{J^i}$ instead of τ^i. Define a pair (i, j) an *acceptable pair* for event k if subject i experiences event k at time s^i while the other subject j does not experience any event until s^i (i.e., $s^j > s^i$).[3]

Then, the estimated CIF satisfies the concordance if $\hat{F}_k(s^i + t^i_{J^i}|\mathcal{X}^i) > \hat{F}_k(s^i + t^j_{J^j}|\mathcal{X}^j)$. We define the ranking loss among acceptable pairs of subjects having different histories of measurements as follows:

$$\mathcal{L}_2 = \sum_{k=1}^{K} \alpha_k \sum_{i \neq j} A_{kij} \cdot \eta\Big(\hat{F}_k(s^i + t^i_{J^i}|\mathcal{X}^i), \hat{F}_k(s^i + t^j_{J^j}|\mathcal{X}^j)\Big), \tag{5.8}$$

where $A_{kij} \triangleq \mathbb{1}(k^i = k, s^i < s^j)$ is an indicator for acceptable pairs (i, j) for event k, $\alpha_k \geq 0$ is a hyper-parameter chosen to trade off ranking losses of the k-th competing event, and $\eta(\cdot)$ is a differentiable loss function. For convenience, we choose here that the coefficients α_k are all equal (i.e., $\alpha_k = \alpha$ for $k = 1, \cdots, K$), and the loss function $\eta(a, b) = \exp(-\frac{a-b}{\sigma})$. Incorporating $\mathcal{L}_2$ into the total loss function penalizes incorrect ordering of pairs and encourages correct ordering of pairs with respect to each event.

[3] An acceptable pair (i, j) naturally captures the right-censoring of subject j since it only considers subjects who lived longer than s^i.

5.4.2.3 Prediction Loss

Longitudinal measurements on time-varying covariates, such as the trajectory of biomarkers and the presence of comorbidities over time, may be highly associated with the occurrence of clinical events. Thus, we introduce an auxiliary task in the shared subnetwork, which makes predictions, $\mathbf{y}_j$, on the step-ahead covariates, $\mathbf{x}_{j+1}$, of our interest, to regularize the shared subnetwork such that the hidden representations preserve information for the step-ahead predictions. Taking account missing measurements into consideration, the prediction loss is defined as follows:

$$\mathcal{L}_3 = \beta \cdot \sum_{i=1}^{N} \sum_{j=0}^{J^i-1} \sum_{d \in \mathcal{I}} (1 - m_{j+1,d}^{i}) \cdot \zeta(x_{j+1,d}^{i}, y_{j,d}^{i}), \tag{5.9}$$

where $\beta \geq 0$ is a hyper-parameter and $\zeta(a, b) = |a - b|^2$ for continuous covariates and $\zeta(a, b) = -a \log b - (1- a) \log(1- b)$ for binary covariates. By incorporating the missing indicators, the loss is calculated for the step-ahead predictions whose actual measurements are not missing. We select $\mathcal{I}$ as a set of time-varying covariates (e.g., biomarkers or comorbidities) on which we aim to focus the network to be regularized.

5.4.3 Discussion on the Scalability

For an accurate estimation of CIFs in (5.5), it is desirable to have the time interval resolution for discretizing the time horizon (i.e., $\mathcal{T}$ in Section 5.3) to be fine rather than coarse to maintain more information on time-to-event/censoring. However, Dynamic-DeepHit might become over-fitted as it requires the number of output nodes equivalent to $|\mathcal{T}|$ (i.e., inversely proportional to the resolution of the time horizons). To prevent this, we utilize i) early stopping based on the performance metric of our interest (i.e., discriminative performance) and ii) L1 regularization over weights in the cause-specific subnetworks and the output layer. Throughout the experiments, we discretized the time with a resolution of one month that is a fine resolution for longitudinal data with regular follow-ups on a yearly basis, since the time information in the data was mostly available in month format. We show that

Dynamic-DeepHit achieves a significant gain in terms of the discriminative performance and provides the calibration performance comparable to the best performing benchmark. We provide more details in the subsequent sections.

5.5 Dataset

Experiments were conducted using retrospective longitudinal data from the UK Cystic Fibrosis Registry; this database is sponsored and hosted by the UK Cystic Fibrosis Trust[4]. The registry comprises a cohort of 10,995 patients during annual follow-ups between 2008-2015 with covariates for individual CF patients including demographics, genetic mutations, bacterial infections, comorbidities, hospitalization, lung function scores and therapeutic management. Lung transplantation (LT) is recommended for patients with end-stage respiratory failure as a means to improve life expectancy [143, 144]. Unfortunately, there are more LT candidates than available lung donors, and in addition, the LT procedure is accompanied with serious risks of subsequent post-transplant complications [145].

Meanwhile, complications due to organ transplantation and CF-associated liver disease have been reported as the most frequent causes of death among CF patients after lung-related disease, which share a number of risk factors with respiratory failure [126]. Hence, it is important that patients who are at risk of respiratory failure and other causes be provided with a joint prognosis in order to properly manage LT. More specifically, an effective LT referral policy should efficiently allocate the scarce donor lungs by identifying high-risk patients as candidates for transplant, without overwhelming the LT waiting list with low-risk patients for whom a LT might be an unnecessary exposure to the risk of post-transplant complications or be at risk of other CF-associated diseases [146].

In this work, we focused on follow-up variables that are available from 2009 – this was due to covariate mismatch between measurements recorded in 2008 and those recorded in the

[4] https://www.cysticfibrosis.org.uk/the-work-we-do/uk-cf-registry

rest of the years. Since transplantation decisions are mostly relevant for adults and deaths in children with CF are now very rare in developed countries [147], we excluded pediatric patients, and included only patients who were aged 18 years or older. Overall, out of 10,995 patients, experiments were conducted on 5,883 adult patients with total of 90 features (11 static covariates and 79 time-varying covariates). For each patient, longitudinal measurements were conducted roughly every year; the time interval between two adjacent measurements ranges from 0 to 69 months with mean of 9.20 months. Here, we discretized the time with a resolution of one month since the date information in the data was mostly available in month format. The number of yearly follow-ups was from 1 to 7 with mean of 5.34 measurements per patients. Among the total of 5,883 patients, 605 patients (10.28%) were followed until death and the remaining 5,278 patients (89.72%) were right-censored (i.e., lost to follow-up). We divided the mortality cause into: i) 491 (8.35%) deaths due to respiratory failures and ii) 114 (1.94%) deaths due to other causes including complications due to organ transplantation and CF-associated liver failure.

5.6 Experiments

The usefulness of a survival model should be assessed primarily by how well the model discriminates among predicted risks and secondarily by how well the model is calibrated. As an illustration in CF, lung transplant is the treatment of last resort for patient with end-stage respiratory failure. Successful transplant can mean many additional years of life for such patients, but there are many more patients in need of transplants than there are available donor lungs. Therefore, it is important to correctly discriminate/prioritize recipients on the basis of risk. However, if the risk predictions of a given model are not well calibrated to the truth (i.e., if there is poor agreement between predicted and observed outcomes), then the model will have little prognostic value for clinicians. As discussed above, we assess the risk predictions of Dynamic-DeepHit with respect to how well the predictions discriminate among individual risks and how accurate the predictions are.

Throughout the experiments, all patients are aligned based on their date of birth to synchronize the time for comparing risk predictions made at different points. More specifically, the time at which measurements are recorded and that at which events or censoring occur is defined as the amount of time elapsed since births. We define the set of possible survival times to be up to 100 years with a monthly time interval, i.e., $T_{\max} = 1200$). Our results are obtained using 5 random 80/20 train/test splits: we randomly separated the data into a training set (80%) and a testing set (20%) and then reserved 20% of the training set as a validation set for hyper-parameter optimization and for early-stopping to avoid over-fitting.

The hyper-parameters, such as the coefficients, the activation functions, and the number of hidden layers and nodes of each subnetwork, are chosen utilizing Random Search [148]. The permitted values of the hyper-parameters are listed in Table 5.1.

For the prediction loss in (5.9), we considered two scenarios: i) $\mathcal{I} = \{\text{FEV}_1\% \text{ predicted}\}$ for a fair comparison with the joint models, where $\text{FEV}_1\%$ predicted is a well-known biomarker of the respiratory failure and ii) $\mathcal{I}$ includes all the time-varying covariates including lung function scores, nutritional status, and comorbidities.

5.6.1 Benchmarks

We compared Dynamic-DeepHit with state-of-the-art methods that account for dynamic survival analysis under the presence of longitudinal measurements including the joint model [13], the joint model based on latent classes [132], and survival methods under landmarking approaches [12].

In particular, the joint model (**JM**)[5] was implemented using a Bayesian framework that uses MCMC algorithms [149] by modeling the time-to-event data using a cause-specific Cox proportional hazards regression and the longitudinal process using a multivariate linear mixed

[5] https://cran.r-project.org/web/packages/JMbayes/

Table 5.1: Hyper-parameters of Dynamic-DeepHit

Block	**Sets of hyper-parameters**
Initialization	Xavier initialization for weight matrix Zero initialization for bias vector
Optimization	Adam Optimizer
RNN architecture	{GRU, LSTM}
Dropout	0.6
Learning rate	10^{-4}
Mini-batch size	$\{32, 64, 128\}$
α, β, σ	$\{0.1, 1, 3, 5\}$
Nonlinearity (Attention)	{ReLU, eLU, tanh}
No. of layers (Attention)	2
No. of nodes (Attention)	$\{50, 100, 200, 300\}$
Nonlinearity (Cause-Specific)	{ReLU, eLU, tanh}
No. of layers (Cause-Specific)	$\{1, 2, 3, 5\}$
No. of nodes (Cause-Specific)	$\{50, 100, 200, 300\}$
Nonlinearity (Shared)	{ReLU, eLU, tanh}
No. of layers (Shared)	$\{1, 2, 3\}$
No. of nodes (Shared)	$\{50, 100, 200, 300\}$

model. (Due to the computational limitations of standard joint models [130], we selected only FEV$_1$% predicted for the longitudinal process.) To account for the competing risks setting, the cause-specific Cox was created by fixing an event (e.g., death from respiratory cause) and treating the other event (e.g., death from other causes) simply as a form of censoring; see [43]. The joint models integrating latent class (**JM-LC**)[6] to characterize the underlying

[6] https://cran.r-project.org/web/packages/lcmm/

heterogeneity of the cohort [132] was implemented with $G = 3$ latent classes whose parameters are associated with each class with the similar model specifications to JM.

For the landmarking approaches, we chose the landmarking times as the prediction times, which is age at 30, 40, and 50, and only patients who are at risk at these landmarking times (patients who have not experienced any event or been censored) are considered when we fit survival models at each landmarking time. Overall, the landmarking approaches are implemented utilizing the following survival models: the cause-specific version of the Cox proportional hazards model (**cs-Cox**)[7] and random survival forests under competing risks (**RSF**)[8] [30] with 1000 trees, as a non-parametric alternative of the Cox model.

5.6.2 Discriminative Performance

In this subsection, we present the performance metric that is extended to the survival setting with competing risks and longitudinal measurements, and then we evaluate Dynamic-DeepHit in terms of this metric. To assess the discriminative performance of the various methods, we use a cause-specific time-dependent concordance index ($C_k(t, \Delta t)$), which is an extension of the time-dependent concordance index[9] in [58] adapted to the competing risks setting with longitudinal measurements; similar extensions[10] are made in [150, 151]. More specifically, $C_k(t, \Delta t)$ takes both prediction and evaluation times into account to reflect

[7] https://cran.r-project.org/web/packages/survival/

[8] https://cran.r-project.org/web/packages/randomForestSRC/

[9] This metric is suitable for evaluating discriminative performance at different time horizons once risk predictions are issued with the same condition. However, since the time horizon at which risk predictions are made is not considered, this metric cannot be directly used in the longitudinal setting.

[10] This metric provides area under ROC curve (AUC) considering both the prediction and evaluation times. However, it quantifies how well a survival model can order risks at given evaluation time, while our proposed metric quantifies how well a survival model can order risks up to that evaluation time, which better represents the time-to-event setting with right-censoring

possible changes in risk over time compared to the ordinary concordance index [25], which is a widely used discriminative index in survival analysis.[11] Given the estimated CIF in (5.5), $C_k(t, \Delta t)$ for event k is defined as

$$C_k(t, \Delta t) = P\Big(\hat{F}_k(t + \Delta t|\mathcal{X}^i(t)) > \hat{F}_k(t + \Delta t|\mathcal{X}^j(t))\Big|\tau^i < \tau^j, k^i = k, \tau^i < t + \Delta t\Big), \quad (5.10)$$

where t indicates the prediction time which is the time when the prediction is made to incorporate dynamic predictions and Δt denotes the evaluation time which is the time elapsed since the prediction is made. Throughout the evaluations, $\hat{F}_k(t + \Delta t|\mathcal{X}(t))$ implies the risk of event k occurring in Δt years, which is predicted at age t given the longitudinal measurements until that age.

The discriminative performance of Dynamic-DeepHit on the CF dataset is reported in Table 5.2; means and standard deviations were obtained via 5 random splits. Throughout the evaluation, the tested prediction and evaluation times are in years. Dynamic-DeepHit outperformed the benchmarks for all evaluated prediction and evaluation times with respect to $C_k(t, \Delta t)$ for both causes. All the improvements over the benchmarks were statistically significant; we denoted $*$ for p-value < 0.01 and $\dagger$ for p-value < 0.05. More specifically, on average, Dynamic-DeepHit achieved improvements of 4.36% and 9.67% over the best benchmark (6.26% and 14.97% over JM) for death from respiratory failure and death from other causes, respectively.

To provide more fair comparison with JM, we also reported the discriminative performance of simplified versions of Dynamic-DeepHit: i) the proposed network (denoted as $\mathbf{FEV_1\%}$) whose $\mathcal{L}_3$ is computed only based on $\mathcal{I} = \{\text{FEV}_1\% \text{ predicted}\}$ and ii) the proposed network (denoted as **cause-spec.**) that is separately trained for each cause in a cause-specific manner

[11] The concordance index and its variations are based on the assumption that patients who experienced an event should be assigned a higher risk than those who lived longer (i.e., patients experienced event or was censored afterward). Thus, it naturally handles right-censoring – for example, if both patients are censored, we do not include this pair of patients as defined in (5.10).

Table 5.2: Comparison of $C_k(t, \Delta t)$ (mean ± std) for various methods. Higher the better.

Algorithms	Resp. Failure				Other Causes			
	$\Delta t = 1$	$\Delta t = 3$	$\Delta t = 5$	$\Delta t = 10$	$\Delta t = 1$	$\Delta t = 3$	$\Delta t = 5$	$\Delta t = 10$
Prediction Time $t = 30$								
cs-Cox	$0.840\pm0.09^{\dagger}$	$0.837\pm0.08^{\dagger}$	$0.837\pm0.08^{\dagger}$	$0.837\pm0.08^{\dagger}$	$0.667\pm0.10^{*}$	$0.664\pm0.10^{*}$	$0.665\pm0.10^{*}$	$0.665\pm0.10^{*}$
RSF	$0.936\pm0.01^{\dagger}$	0.932 ± 0.01	$0.931\pm0.02^{\dagger}$	$0.929\pm0.01^{\dagger}$	$0.798\pm0.04^{*}$	$0.792\pm0.04^{*}$	$0.773\pm0.05^{*}$	$0.776\pm0.05^{*}$
JM	$0.882\pm0.03^{*}$	$0.896\pm0.01^{*}$	$0.896\pm0.01^{*}$	$0.897\pm0.01^{*}$	$0.760\pm0.02^{*}$	$0.795\pm0.03^{*}$	$0.802\pm0.02^{*}$	$0.812\pm0.01^{*}$
JM-LC	$0.897\pm0.04^{\dagger}$	$0.894\pm0.05^{\dagger}$	$0.894\pm0.05^{\dagger}$	$0.894\pm0.05^{\dagger}$	$0.856\pm0.02^{*}$	$0.855\pm0.02^{*}$	$0.855\pm0.02^{*}$	$0.855\pm0.02^{*}$
[50]	$0.910\pm0.02^{*}$	$0.907\pm0.02^{*}$	$0.907\pm0.02^{*}$	$0.907\pm0.01^{*}$	$0.819\pm0.07^{\dagger}$	$0.831\pm0.07^{\dagger}$	$0.834\pm0.07^{\dagger}$	$0.839\pm0.07^{\dagger}$
Exponential	$0.895\pm0.03^{*}$	$0.890\pm0.03^{*}$	$0.890\pm0.03^{*}$	$0.890\pm0.02^{*}$	$0.824\pm0.05^{*}$	$0.825\pm0.05^{*}$	$0.824\pm0.05^{*}$	$0.824\pm0.05^{*}$
Proposed								
FEV$_1$%	0.948±0.01	0.939±0.01	0.938±0.01	0.937±0.01	0.924±0.02	0.922±0.02	0.921±0.02	0.921±0.02
cause-spec.	0.946±0.01	0.937±0.02	0.936±0.02	0.933±0.02	$0.875\pm0.04^{\dagger}$	$0.867\pm0.05^{\dagger}$	$0.862\pm0.05^{\dagger}$	$0.866\pm0.05^{\dagger}$
full-fledged	**0.949±0.01**	**0.941±0.01**	**0.942±0.01**	**0.941±0.01**	**0.929±0.02**	**0.927±0.02**	**0.925±0.02**	**0.926±0.02**
Prediction Time $t = 40$								
cs-Cox	$0.842\pm0.03^{*}$	$0.842\pm0.03^{*}$	$0.842\pm0.03^{*}$	$0.842\pm0.03^{*}$	$0.748\pm0.10^{*}$	$0.749\pm0.10^{*}$	$0.749\pm0.10^{*}$	$0.749\pm0.10^{*}$
RSF	$0.888\pm0.01^{*}$	$0.887\pm0.02^{*}$	$0.886\pm0.03^{*}$	$0.891\pm0.03^{*}$	$0.803\pm0.06^{\dagger}$	$0.771\pm0.05^{*}$	$0.749\pm0.05^{*}$	$0.746\pm0.05^{*}$
JM	$0.906\pm0.01^{*}$	$0.905\pm0.01^{*}$	$0.908\pm0.01^{*}$	$0.909\pm0.01^{*}$	$0.818\pm0.03^{*}$	$0.814\pm0.03^{*}$	$0.813\pm0.02^{*}$	$0.840\pm0.02^{*}$
JM-LC	$0.911\pm0.04^{\dagger}$	$0.910\pm0.04^{\dagger}$	$0.910\pm0.04^{\dagger}$	$0.910\pm0.04^{\dagger}$	$0.851\pm0.02^{*}$	$0.851\pm0.02^{*}$	$0.850\pm0.02^{*}$	$0.850\pm0.02^{*}$
[50]	$0.913\pm0.02^{*}$	$0.923\pm0.02^{*}$	$0.923\pm0.01^{*}$	$0.923\pm0.01^{*}$	$0.837\pm0.07^{\dagger}$	$0.845\pm0.07^{\dagger}$	$0.846\pm0.07^{\dagger}$	$0.849\pm0.07^{\dagger}$
Exponential	$0.883\pm0.03^{*}$	$0.883\pm0.03^{*}$	$0.882\pm0.03^{*}$	$0.882\pm0.03^{*}$	$0.816\pm0.04^{*}$	$0.817\pm0.04^{*}$	$0.816\pm0.04^{*}$	$0.816\pm0.04^{*}$
Proposed								
FEV$_1$%	0.956±0.01	0.958±0.01	0.957±0.01	0.957±0.01	0.934±0.02	0.931±0.02	0.931±0.02	0.931±0.02
cause-spec.	0.955±0.01	0.957±0.01	0.957±0.01	0.958±0.01	$0.907\pm0.02^{\dagger}$	$0.909\pm0.02^{\dagger}$	$0.906\pm0.03^{\dagger}$	$0.909\pm0.02^{\dagger}$
full-fledged	**0.961±0.01**	**0.963±0.01**	**0.963±0.01**	**0.963±0.01**	**0.939±0.01**	**0.938±0.01**	**0.939±0.01**	**0.939±0.01**
Prediction Time $t = 50$								
cs-Cox	$0.851\pm0.11^{\dagger}$	$0.851\pm0.11^{\dagger}$	$0.851\pm0.11^{\dagger}$	$0.851\pm0.11^{\dagger}$	$0.721\pm0.09^{*}$	$0.720\pm0.09^{*}$	$0.720\pm0.09^{*}$	$0.720\pm0.09^{*}$
RSF	$0.898\pm0.01^{*}$	$0.890\pm0.03^{*}$	$0.892\pm0.02^{*}$	$0.891\pm0.02^{*}$	$0.741\pm0.05^{*}$	$0.764\pm0.03^{*}$	$0.763\pm0.03^{*}$	$0.768\pm0.04^{*}$
JM	$0.900\pm0.01^{*}$	$0.902\pm0.01^{*}$	$0.908\pm0.01^{*}$	$0.908\pm0.01^{*}$	$0.824\pm0.03^{*}$	$0.823\pm0.02^{*}$	$0.826\pm0.01^{*}$	$0.843\pm0.02^{*}$
JM-LC	$0.916\pm0.04^{*}$	$0.916\pm0.04^{*}$	$0.916\pm0.04^{*}$	$0.916\pm0.04^{*}$	$0.852\pm0.02^{*}$	$0.852\pm0.02^{*}$	$0.852\pm0.02^{*}$	$0.853\pm0.02^{*}$
[50]	$0.929\pm0.01^{*}$	$0.929\pm0.01^{*}$	$0.929\pm0.01^{*}$	$0.929\pm0.01^{*}$	$0.851\pm0.07^{\dagger}$	$0.858\pm0.06^{\dagger}$	$0.859\pm0.06^{\dagger}$	$0.862\pm0.06^{\dagger}$
Exponential	$0.875\pm0.02^{*}$	$0.874\pm0.02^{*}$	$0.874\pm0.02^{*}$	$0.873\pm0.02^{*}$	$0.806\pm0.04^{*}$	$0.806\pm0.04^{*}$	$0.806\pm0.04^{*}$	$0.806\pm0.04^{*}$
Proposed								
FEV$_1$%	0.962±0.01	0.962±0.00	0.962±0.00	0.961±0.00	0.926±0.03	0.935±0.02	0.930±0.02	0.934±0.02
cause-spec.	0.962±0.01	0.961±0.01	0.944±0.03	0.954±0.02	$0.896\pm0.04^{\dagger}$	0.929±0.03	0.929±0.03	0.925±0.03
full-fledged	**0.968±0.00**	**0.968±0.01**	**0.967±0.01**	**0.967±0.01**	**0.941±0.01**	**0.942±0.01**	**0.943±0.01**	**0.936±0.02**

$*$ indicates p-value $<$ 0.01, † indicates p-value $<$ 0.05

(by fixing an event and treating the other event as right-censoring). As seen in Table 5.2, the simplified versions still achieved significant performance improvements over JM. It is worth to highlight that, especially for predicting the risk of death from other causes, the full-fledged network achieved performance improvement over the cause-specific version by jointly learning latent representations that are common to competing events.

To further understand the source of gains, we compare Dynamic-DeepHit with the following variations: the network in [50] which performs risk predictions based only on the last available measurements (the dynamic-RNN in the shared subnetwork is replaced with a feed-forward network and the network is trained without $\mathcal{L}_3$) and a deep network utilizing the same architecture with that of Dynamic-DeepHit whose output layer is modified to model the time-to-event data via the Exponential distribution (denoted as **Exponential**). For the comparison, the same hyper-parameter optimization is applied. Dynamic-DeepHit leverages the RNN architecture to learn the associations between the longitudinal measurements and the time-to-events, and to incorporate the history of the measurements when making risk predictions. Hence, as expected, our method outperformed our previous work in [50], which discards the historical information and relies only on the last available measurements. In contrast to the network which specifies the underlying survival process as Exponential distribution and, thus, is limited to learn the complex interactions with the covariates, our network better discriminates individual risks by directly learning the joint distribution of the first hitting time and the competing events.

5.6.3 Calibration Performance

In this subsection, we present the calibration performance metric that is extended to the survival setting with competing risks and longitudinal measurements. More specifically, to assess the calibration performance of the various methods, we use a cause-specific time-dependent Brier score ($BS_k(t, \Delta t)$), which is an extension of the Brier score [59] that implies the mean squared error adjusted for right-censoring; the same extension in $C_k(t, \Delta t)$ is applied. Given the estimated CIF, $BS_k(t, \Delta t)$ for event k is defined as

$$BS_k(t, \Delta t) = E\Big[\big(\mathbb{1}(T^i < t + \Delta t, k^i = k) - \hat{F}_k(t + \Delta t | \mathcal{X}^i(t))\big)^2\Big] \tag{5.11}$$

where t indicates the prediction time which is the time when the prediction is made to incorporate dynamic predictions and Δt denotes the evaluation time which is the time elapsed

Table 5.3: Comparison of $BS_k(t, \Delta t)$ (mean ± std) for various methods. Lower the better.

Algorithms	**Resp. Failure**				**Other Causes**			
	$\Delta t = 1$	$\Delta t = 3$	$\Delta t = 5$	$\Delta t = 10$	$\Delta t = 1$	$\Delta t = 3$	$\Delta t = 5$	$\Delta t = 10$
				Prediction Time $t = 30$				
cs-Cox	0.085±0.02	0.145±0.01	0.225±0.02	0.377±0.03	0.060±0.02	0.084±0.02	0.156±0.01	0.256±0.02
RSF	**0.044±0.01**	0.053±0.00	0.058±0.00	**0.059±0.00**	0.012±0.00	0.012±0.00	0.013±0.00	0.013±0.00
JM	0.050±0.01	**0.051±0.00**	**0.051±0.00**	0.066±0.01	0.012±0.00	0.012±0.00	0.014±0.00	0.018±0.00
JM-LC	0.053±0.01	0.062±0.00	0.065±0.00	0.066±0.00	0.012±0.00	0.012±0.00	**0.013±0.00**	**0.013±0.00**
Proposed	0.058±0.01	0.059±0.01	0.059±0.01	0.060±0.00	**0.011±0.00**	**0.012±0.00**	0.013±0.00	0.017±0.00
				Prediction Time $t = 40$				
cs-Cox	0.150±0.04	0.309±0.08	0.354±0.08	0.433±0.07	0.016±0.00	0.055±0.04	0.133±0.09	0.133±0.09
RSF	**0.057±0.00**	**0.051±0.00**	**0.054±0.00**	**0.056±0.00**	0.015±0.00	0.015±0.00	0.016±0.00	0.016±0.00
JM	0.058±0.00	0.052±0.00	0.055±0.00	0.087±0.01	0.015±0.00	0.016±0.00	0.018±0.00	0.031±0.00
JM-LC	0.063±0.00	0.064±0.00	0.065±0.00	0.067±0.00	**0.015±0.00**	0.016±0.00	0.017±0.00	**0.016±0.00**
Proposed	0.067±0.00	0.063±0.00	0.062±0.00	0.059±0.00	0.015±0.00	**0.015±0.00**	**0.016±0.00**	0.019±0.00
				Prediction Time $t = 50$				
cs-Cox	0.442±0.24	0.616±0.28	0.658±0.30	0.658±0.30	0.315±0.21	0.428±0.23	0.428±0.23	0.737±0.20
RSF	**0.055±0.00**	0.065±0.01	0.069±0.01	0.069±0.01	0.018±0.00	0.021±0.00	0.021±0.00	0.021±0.00
JM	0.056±0.00	**0.057±0.00**	**0.066±0.01**	0.111±0.01	0.017±0.00	0.022±0.00	0.028±0.00	0.054±0.01
JM-LC	0.069±0.00	0.072±0.00	0.073±0.00	0.075±0.00	0.017±0.00	0.017±0.00	0.017±0.00	**0.016±0.00**
Proposed	0.074±0.00	0.071±0.00	0.070±0.00	**0.069±0.00**	**0.016±0.00**	**0.016±0.00**	**0.017±0.00**	0.022±0.00

since the prediction is made. Throughout the evaluations, $\hat{F}_k(t + \Delta t|\mathcal{X}(t))$ implies the risk of event k occurring in Δt years, which is predicted at age t given the longitudinal measurements until that age.

In Table 5.3, we report the calibration performance in terms of Brier score (lower the better) for the CF dataset. As seen in the tables, our method achieves the performance comparable to the best performing benchmark, i.e., RSF and JM-LC, for most of the tested prediction and evaluation times.

5.6.4 Interpreting Dynamic-DeepHit Predictions

Although deep networks offer tremendous success in predictive ability including survival analysis, low interpretability of the inference process has prevented them from being widely used in medicine. In this subsection, we utilize a post-processing statistic that can be used by clinicians to interpret predictions issued by Dynamic-DeepHit and to understand the associations of covariates and survival over time. It is worth drawing a distinction between interpreting a model, versus interpreting its decision [152, 153]. While interpreting complex models (e.g deep neural networks) may sometimes be infeasible, it is often the case that clinicians only want explanations for the prediction made by the model for a given subject. To help interpret predictions issued by Dynamic-DeepHit, we leverage the partial dependence introduced in [154] by extending it to the survival setting with longitudinal measurements.

Let $\mathcal{X}_d$ be a chosen target subset of the input covariates $\mathcal{X}$ and $\mathcal{X}_{\backslash d}$ be its complement, i.e., $\mathcal{X}_d \cup \mathcal{X}_{\backslash d} = \mathcal{X}$. Then, we can rewrite the estimated CIF in (5.5) as $\hat{F}_k(\tau|\mathcal{X}) = \hat{F}_k(\tau|\mathcal{X}_d, \mathcal{X}_{\backslash d})$ to explicitly denote the dependency on variables in both subsets. The partial dependence function at time Δt, which is the time elapsed since the last measurement, for event k can be defined as a function of $\mathcal{X}_d$ as follows:

$$\gamma_k(\Delta t, \mathcal{X}_d) = \mathbb{E}_{\mathcal{X}_{\backslash d}}\left[\hat{F}_k(t_J + \Delta t|\mathcal{X}_d, \mathcal{X}_{\backslash d})\right] \approx \frac{1}{N}\sum_{i=1}^{N} \hat{F}_k(t^i_{J^i} + \Delta t|\mathcal{X}_d, \mathcal{X}^i_{\backslash d}), \tag{5.12}$$

where t_J indicates the time of the last measurement. Thus, from (5.12), we can approximately assess how the estimated CIFs are affected by different values of $\mathcal{X}_d$ on average.

To see the influence of covariates on risk predictions issued by Dynamic-DeepHit, we calculated the change in (5.12) for each covariate $\mathcal{X}_d$ for $d = 1, \cdots, d_{\mathbf{x}}$ by varying the value from its minimum, $x_{d,\min}$, to its maximum, $x_{d,\max}$:

$$\gamma_k(\Delta t, \mathcal{X}_d = x_{d,\min}) - \gamma_k(\Delta t, \mathcal{X}_d = x_{d,\max}). \tag{5.13}$$

Table 5.4 illustrates the fifteen most influential covariates for the death from respiratory failure and the death from other causes, respectively. Here, we set $\Delta t = 5$ year and the amount of increase/decrease is used to rank the influence. Here, the values imply the averaged increase/decrease of the risk predictions (by varying the covariate from its minimum to maximum) and the signs indicate whether the increase of each covariate increases (+) or decreases (-) the risk predictions.

Table 5.4: The top 15 most influential covariates with $\Delta t = 5$ year. The values indicate the amount of increase(+)/decrease(-) in the predicted risks on average and the covariates are ranked by the absolute values.

Rank	**Death Cause**	
	Resp. Failure	**Other Causes**
1	FEV_1 Predicted (-0.033)	IV ABX Days Hosp. (+0.014)
2	IV ABX Days Hosp. (+0.032)	Gram-Negative (-0.013)
3	Gram-Negative (-0.029)	FEV_1 Predicted (-0.012)
4	FEV_1 (-0.026)	FEV_1 (-0.012)
5	Weight (-0.026)	Weight (-0.011)
6	BMI (-0.025)	BMI (-0.010)
7	Colonic Stricture (-0.024)	Oral Hypo. Agents (-0.008)
8	Oral Hypo. Agents (-0.019)	Class IV Mutation (-0.008)
9	Class IV Mutation (-0.017)	IV ABX Days Home (+0.007)
10	B. Cepacia (+0.016)	Cancer (+0.007)
11	GI Bleed (non-var.) (-0.016)	GI Bleed (var.) (+0.007)
12	O_2 Continuous (+0.015)	HypertonicSaline (-0.006)
13	Drug Dornase (-0.015)	Bone Fracture (-0.006)
14	IV ABX Days Home (+0.014)	Colonic Stricture (-0.006)
15	O_2 Nocturnal (+0.013)	O_2 Nocturnal (+0.006)

IV: intravenous, ABX: antibiotics

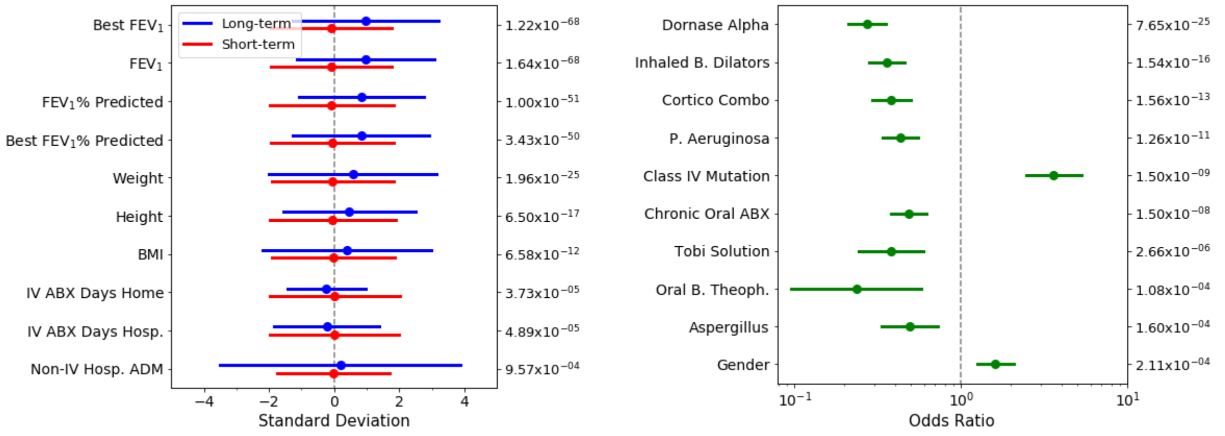
Previous studies in respiratory failures of CF patients have identified FEV_1% predicted as a strong surrogate for the survival, and have shown that a decrease in FEV_1% predicted severely increases the mortality of CF patients [6, 121]. Notably, the risk predictions on the respiratory failure made by Dynamic-DeepHit was highly influenced by FEV_1% predicted in a similar manner. In addition, days of intravenous (IV) antibiotics (ABX), which are used to treat severe bacterial infections, both in hospital and at home, and body mass index (BMI) and weight turned out to be highly influential covariates. This finding is consistent with the domain knowledge, which finds the IV ABX and hospitalization periods are often considered as key risk factors for CF patients [7] and the occurrence of malnutrition, which is often indicated by BMI, is associated with reductions in their survival [155]. More interestingly, the predicted risks for respiratory failure were significantly increased when a patient has Burkholderia cepacia (B. Cepacia), which is a rare but significant threat to CF patients colonizing in the lungs that causes infection and inflammation that deteriorates lung function [156].

For death from other causes, the partial dependence displayed the similar trend, while IV ABX days was more influential to the predicted risks than FEV$_1$% predicted was. In particular, the risk predictions for the death from other causes showed slightly different influences from other covariates, such as the indicators of cancer and GI bleeding in variceal source that is a strong sign of liver failure. Therefore, the risk factors and corresponding risk predictions issued by Dynamic-DeepHit need to be carefully interpreted with different priorities depending on the events.

5.6.5 Temporal Importance of Longitudinal Measurements

The temporal attention mechanism in the shared subnetwork renders Dynamic-DeepHit to pay special attention to time stamps at which the measurements are important for making risk predictions. To investigate the attention mechanism, we aim this subsection at finding to which patients the network focuses on the long-term (or short-term) dependency of the measurements. For ease of illustration, we define $j^* = \arg\max_{j \in \{1, \cdots, J-1\}} a_j$ as the time stamp at which the proposed network pays the most attention to.

We divide the patients into two groups based on their temporal dependency: the long-term dependency group comprises patients having the highest attention weight to earlier



(a) Mean and 95% CI for continuous covariates (b) Odds ratio and 95% CI for binary covariates

Figure 5.3: Forest plots on (a) continuous covariates and (b) binary covariates with the smallest p-values. The left column displays the covariate names and the right column denotes the corresponding p-values. (The covariates are ordered from smallest to largest.)

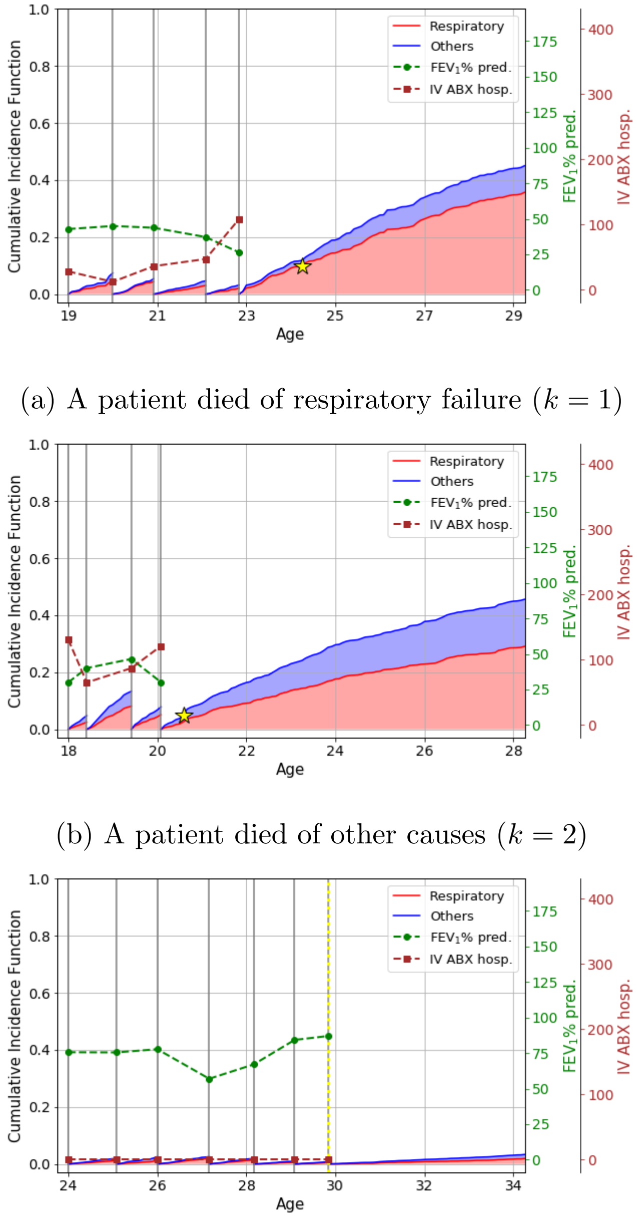
measurements, i.e., $j^* < J - 1$, and the short-term dependency group consists of patients having the highest attention weight to the most recent measurement, i.e., $j^* = J - 1$. Among 3710 patients with at least three measurements (i.e., $J \geq 3$), our network focused on the long-term dependency of longitudinal measurements for 290 patients (7.82%) and on the short-term dependency for 3420 patients (92.18%). Then, the characteristics of the two groups were compared using independent two-sample t-test for continuous covariates and Fisher's exact test for discrete covariates.

In Figure 5.3, we illustrated forest plots on twenty covariates (ten for the continuous and ten for binary covariates) with the smallest p-values, which implies strong evidence that their distributions are different in the two groups. More specifically, for each continuous covariate in Figure 5.3(a), we aligned the mean and the 95% confidence interval (CI) of each group with the overall population – this implies that how much the distribution of each group is different from the mean of the overall population in terms of its standard deviation. For an example of Best FEV$_1$, the mean of the long-term dependency group (i.e., 3.37) was approximately a standard deviation (i.e., 0.92) larger than the overall mean (i.e., 2.44) while that of the short-term dependency group (i.e., 2.36) was very close to the overall mean. For each binary covariate in Figure 5.3(b), we displayed the odds ratio (OR) and the 95% CI, which is the ratio of the odds of being in the long-term dependency group in the presence of the covariate and the odds of being in the long-term dependency group without the presence of the covariate – this statistic quantifies the strength of the association between each covariate and being in the long-term dependency group. For instance, if the OR is greater than 1, then the presence of the covariate raises the odds of being in the long-term dependency group.

Interestingly, patients in the long-term dependency group displayed, on average, factors that mitigate the predicted risks compared to those in the short-term dependency group. For continuous covariates, as seen in Figure 5.3(a), the factors include higher lung functions scores (i.e., FEV_1, FEV_1% predicted, Best FEV_1, and Best FEV_1% predicted), shorter IV ABX periods (i.e., IV ABX days at home and in hospital), richer nutritional status (i.e., weight and BMI), that decrease the predicted risks for both death from the respiratory failure and that from other causes as reported in Table 5.4. For binary covariates, as seen in Figure 5.3(b), the factors include lower bacterial infection rate (i.e., pseudomonas aeruginosa and aspergillus whose infection increases the risk of the respiratory failure [156]) and lower therapy/treatments rate (i.e., dornase alpha, cortico combo, chronic oral ABX, and tobi solution). Indeed, Dynamic-DeepHit issued lower risk predictions for patients in the long-term group; the predicted risks were 38.98% and 35.20% lower on average for respiratory failure and death from other causes, respectively.

5.6.6 Dynamic Survival Prediction

At run-time, Dynamic-DeepHit issues cause-specific risk predictions as defined in (5.5) for each subject incorporating his/her medical history. Owing to the RNN structure utilized in the shared subnetwork, whenever a new observation is made for that subject, the proposed



(a) A patient died of respiratory failure ($k = 1$)

(b) A patient died of other causes ($k = 2$)

(c) A censored patient ($k = \varnothing$)

Figure 5.4: An illustration of dynamic risk predictions issued by Dynamic-DeepHit for patients with (a) $k = 1$, (b) $k = 2$, and (c) $k = \varnothing$. Gray solid lines, yellow dotted lines, and stars indicate times at which measurement are taken, the time at which a patient is censored, and the time at which an event occurred, respectively.

method is easily able to integrate this information into the history of measurements and to issue new risk predictions in a fully dynamic fashion. It is worth highlighting that the landmarking methods can only provide risk assessment at the predefined landmarking times [12]. In Figure 5.4, we have illustrated the dynamic survival analysis for representative patients in order to show how Dynamic-DeepHit issues and updates risk predictions for different causes (including right-censoring) with new measurements being collected. Along with the predicted risks, trajectories of two highly influential covariates, $\text{FEV}_1\%$ predicted and IV ABX Days in Hospital, are illustrated to show their associations. As demonstrated in Figure 5.4, Dynamic-DeepHit was able to flexibly update the cause-specific risks by incorporating new measurements in a dynamic fashion. For example, the predicted risks for the patient in Figure 5.4(a) was relatively high compared to that of the patient in Figure 5.4(c), presumably due to the high and increasing IV ABX days in hospital and the decreasing $\text{FEV}_1\%$ predicted. The importance of this dynamic approach can be seen in Figure 5.4(a) when a sudden increase in the number of IV ABX days around at age 23 resulted in a steep increase in predicted risks.

5.7 Conclusion

In this work, we developed a novel approach, Dynamic-DeepHit, to perform dynamic survival analysis with competing risks on the basis of longitudinal data. Dynamic-DeepHit is a deep neural network which learns the estimated joint distributions of survival times and competing events, without making assumptions regarding the underlying stochastic processes. We train the network by leveraging a combination of loss functions that capture the right-censoring and the associations of longitudinal measurements with disease progression, both of which are inherent in time-to-event data. We demonstrated the utility of our proposed method through a set of experiments conducted on a cohort of 5,883 adult CF patients whose follow-ups have been recorded in the UK Cystic Fibrosis Registry. The experiments show that the proposed method significantly outperforms the cutting-edge benchmarks in terms of discriminative performance. Supported with a post-processing statistic to interpret risk predictions issued by the proposed method, the results suggest the possibility of improved dynamic analysis on disease progression that will result in more effective health care.

CHAPTER 6

Temporal Phenotyping using Deep Predictive Clustering of Disease Progression

6.1 Introduction

Chronic diseases – such as cystic fibrosis and dementia – are heterogeneous in nature, with widely differing outcomes even in narrow patient subgroups. Disease progression manifests through a broad spectrum of clinical factors, collected as a sequence of measurements in electronic health records, which gives a rise to complex progression patterns among patients [9, 157]. For example, cystic fibrosis evolves slowly, allowing for development of comorbidities and bacterial infections, and creating distinct responses to therapeutic interventions, which in turn makes the survival and quality of life substantially different [158, 159]. Identifying patient subgroups with similar progression patterns can be advantageous for understanding such heterogeneous diseases. This allows clinicians to anticipate patients' prognoses by comparing to "similar" patients and to design treatment guidelines tailored to homogeneous subgroups [11].

Temporal clustering has been recently used as a data-driven framework to partition patients with time-series observations into subgroups of patients. Recent research has typically focused on either finding fixed-length and low-dimensional representations [11, 160] or on modifying the similarity measure [161, 162] both in an attempt to apply the existing clustering algorithms to time-series observations. However, clusters identified from these approaches are purely unsupervised – they do not account for patients' observed outcomes

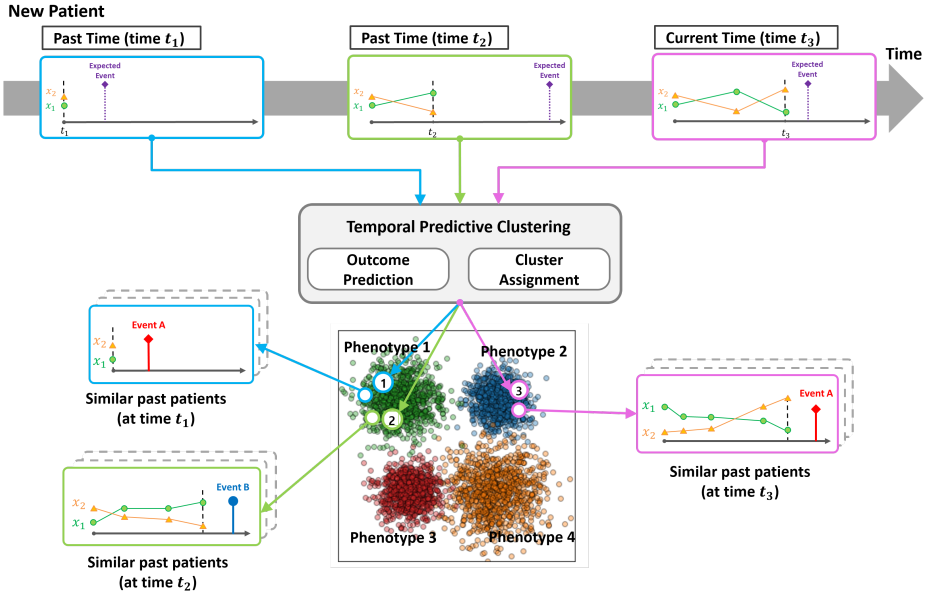


Figure 6.1: A conceptual illustration of our (real-time) clustering procedure. Here, a new patient is assigned over time to one of the four phenotypes based on the expected future event – either Event A or Event B – as new observations are collected.

(e.g., adverse events, the onset of comorbidities, etc.) – which leads to heterogeneous clusters if the clinical presentation of the disease differs even for patients with the same outcomes. Thus, a common prognosis in each cluster remains unknown which can mystify the understanding of the underlying disease progression [163, 164]. To overcome this limitation, we focus on *predictive clustering* [165] to combine predictions on the future outcomes with clustering. More specifically, we aim at finding cluster assignments and centroids by learning discrete representations of time-series that best describe the future outcome distribution. By doing so, patients in the same cluster share similar future outcomes to provide a prognostic value. Figure 6.1 illustrates a pictorial depiction of the clustering procedure.

In this work, we propose an actor-critic approach for temporal predictive clustering, which we call AC-TPC.[1] Our model consists of three networks – an *encoder*, a *selector*, and a *predictor* – and a set of centroid candidates. The key insight, here, is that we model temporal

[1] Source code available at https://github.com/chl8856/AC_TPC.

predictive clustering as learning discrete representations of the input time-series that best describe the future outcome distribution. More specifically, the encoder maps an input time-series into a continuous latent encoding; the selector assigns a cluster (i.e., maps to a discrete representation) to which the input belongs by taking the latent encoding as an input; and the predictor estimates the future outcome distributions conditioned on either the encoding or the centroid of the selected cluster (i.e., the selected discrete representation). The following three contributions render our model to achieve our goal. First, to encourage homogeneous future outcomes in each cluster, we define a clustering objective based on the Kullback-Leibler (KL) divergence between the predictor's output given the time-series, and that given the assigned centroids. Second, we transform solving a combinatorial problem of identifying clusters into iteratively solving two sub-problems: optimization of the cluster assignments and optimization of the centroids. Finally, we allow "back-propagation" through the sampling process of the selector by adopting actor-critic training [166].

Throughout the experiments, we show significant performance improvements over the state-of-the-art clustering methods on two real-world medical datasets. To demonstrate the practical significance of our model, we consider a more realistic scenario where the future outcomes of interest are high-dimensional – that is, development of multiple comorbidities in the next year – and interpreting all possible combinations is intractable. Our experiments show that our model can identify meaningful clusters that can be translated into actionable information for clinical decision-making.

6.2 Problem Formulation

Let $\mathbf{X} \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be random variables for an input feature and an output label (i.e., one or a combination of future outcome(s) of interest) with a joint distribution p_{XY} (and marginal distributions are p_X and p_Y) where $\mathcal{X}$ is the feature space and $\mathcal{Y}$ is the label space.

Here, we focus our description on C-class classification tasks, i.e., $\mathcal{Y} = \{1, \cdots, C\}$.[2] We are given a time-series dataset $\mathcal{D} = \{(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}\}_{n=1}^N$ comprising sequences of realizations (i.e., observations) of the pair $(\mathbf{X}, Y)$ for N patients. Here, $(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}$ is a sequence of T^n observation pairs that correspond to patient n and $t \in \mathcal{T}^n \triangleq \{1, \cdots, T^n\}$ denotes the time stamp at which the observations are made. From this point forward, we omit the dependency on n when it is clear in the context and denote $\mathbf{x}_{1:t} = (\mathbf{x}_1, \cdots, \mathbf{x}_t)$.

Our aim is to identify a set of K *predictive clusters*, $\mathcal{C} = \{\mathcal{C}(1), \cdots, \mathcal{C}(K)\}$, for time-series data. Each cluster consists of homogeneous data samples, that can be represented by its centroid, based on a certain similarity measure. There are two main distinctions from the conventional notion of clustering. First, we treat subsequences of each times-series as data samples and focus on partitioning $\{\{\mathbf{x}_{1:t}^n\}_{t=1}^{T^n}\}_{n=1}^N$ into $\mathcal{C}$. Hence, we define a cluster as $\mathcal{C}(k) = \{\mathbf{x}_{1:t}^n | t \in \mathcal{T}^n,\ s_t^n = k\}$ for $k \in \mathcal{K} \triangleq \{1, \cdots, K\}$ where $s_t^n \in \mathcal{K}$ is the cluster assignment for a given $\mathbf{x}_{1:t}^n$. This is to flexibly update the cluster assignment (in real-time) to which a patient belongs as new observations are being accrued over time. Second, we define the similarity measure with respect to the label distribution and associate it with clusters to provide a prognostic value. More specifically, we want the distribution of output label for subsequences in each cluster to be homogeneous and, thus, can be well-represented by the centroid of that cluster.

Let S be a random variable for the cluster assignment – that depends on a given subsequence $\mathbf{x}_{1:t}$ – and $Y|S = k$ be a random variable for the output given cluster k. Then, such property of predictive clustering can be achieved by minimizing the following Kullback-Leibler (KL) divergence: $KL(Y_t|\mathbf{X}_{1:t} = \mathbf{x}_{1:t} \| Y_t|S_t = k)$ for $\mathbf{x}_{1:t} \in \mathcal{C}(k)$ which is defined as $\int_y p(y|\mathbf{x}_{1:t})\big(\log p(y|\mathbf{x}_{1:t}) - \log p(y|s_t)\big)dy$ where $p(y|\mathbf{x}_{1:t})$ and $p(y|s_t)$ are the label distributions conditioned on a subsequence $\mathbf{x}_{1:t}$ and a cluster assignment s_t, respectively. Note that the KL divergence achieves its minimum when the two distributions are equivalent.

[2] Simple modifications can be made for regression, i.e., $\mathcal{Y} = \mathbb{R}$ and M-dimensional binary classification tasks, i.e., $\mathcal{Y} = \{0, 1\}^M$.

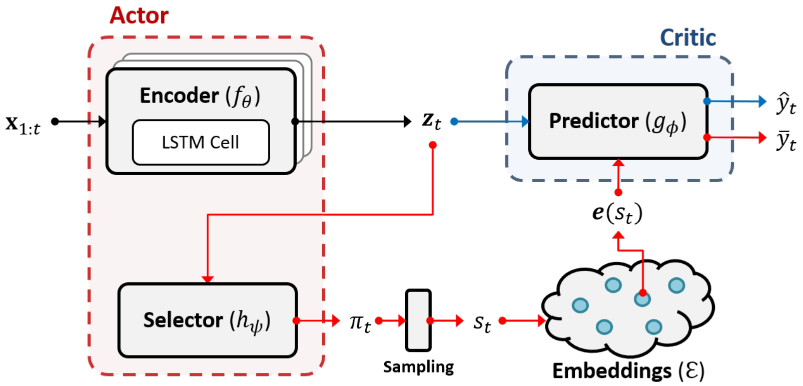


Figure 6.2: The block diagram of AC-TPC. The red line implies the procedure of estimating $p(y|S_t = s_t)$ via a sampling process and the blue line implies that of estimating $p(y|\mathbf{X}_{1:t} = \mathbf{x}_{1:t})$.

Finally, we establish our goal as identifying a set of predictive clusters $\mathcal{C}$ that optimizes the following objective:

$$\underset{\mathcal{C}}{\text{minimize}} \sum_{k \in \mathcal{K}} \sum_{\mathbf{x}_{1:t} \in \mathcal{C}(k)} KL\big(Y_t | \mathbf{X}_{1:t} = \mathbf{x}_{1:t} \big\| Y_t | S_t = k\big). \tag{6.1}$$

Unfortunately, the optimization problem in (6.1) is highly non-trivial. We need to estimate the objective function in (6.1) while solving a non-convex combinatorial problem of finding the optimal cluster assignments and cluster centroids.

6.3 Method: AC-TPC

To effectively estimate the objective function in (6.1), we introduce three networks – an *encoder*, a *selector*, and a *predictor* – and an *embedding dictionary* as illustrated in Figure 6.2. These components together provide the cluster assignment and the corresponding centroid based on a given sequence of observations and enable us to estimate the probability density $p(y|s_t)$. More specifically, we define each component as follows:

- The *encoder*, $f_\theta : \prod_{i=1}^{t} \mathcal{X} \to \mathcal{Z}$, is a RNN (parameterized by θ) that maps a (sub)sequence of a time-series $\mathbf{x}_{1:t}$ to a latent representation (i.e., encoding) $\mathbf{z}_t \in \mathcal{Z}$ where $\mathcal{Z}$ is the latent space.

- The *selector*, $h_\psi : \mathcal{Z} \rightarrow \Delta^{K-1}$, is a fully-connected network (parameterized by ψ) that provides a probabilistic mapping to a categorical distribution from which the cluster assignment $s_t \in \mathcal{K}$ is being sampled.
- The *predictor*, $g_\phi : \mathcal{Z} \rightarrow \Delta^{C-1}$, is a fully-connected network (parameterized by ϕ) that estimates the label distribution given the encoding of a time-series or the centroid of a cluster.
- The *embedding dictionary*, $\mathcal{E} = \{\mathbf{e}(1), \cdots, \mathbf{e}(K)\}$ where $\mathbf{e}(k) \in \mathcal{Z}$ for $k \in \mathcal{K}$, is a set of cluster centroids lying in the latent space which represents the corresponding cluster.

Here, $\Delta^{D-1} = \{\mathbf{q} \in [0,1]^D : q_1 + \cdots + q_D = 1\}$ is a $(D-1)$-simplex that denotes the probability distribution for a D-dimensional categorical (class) variable.

At each time stamp t, the *encoder* maps a input (sub)sequence $\mathbf{x}_{1:t}$ into a latent encoding $\mathbf{z}_t \triangleq f_\theta(\mathbf{x}_{1:t})$. Then, based on the encoding $\mathbf{z}_t$, the cluster assignment s_t is drawn from a categorical distribution that is defined by the *selector* output, i.e., $s_t \sim Cat(\pi_t)$ where $\pi_t = [\pi_t(1), \cdots, \pi_t(K)] \triangleq h_\psi(\mathbf{z}_t)$. Once the assignment s_t is chosen, we allocate the latent encoding $\mathbf{z}_t$ to an embedding $\mathbf{e}(s_t)$ in the *embedding dictionary* $\mathcal{E}$. Since the allocated embedding $\mathbf{e}(s_t)$ corresponds to the centroid of the cluster to which $\mathbf{x}_{1:t}$ belongs, we can, finally, estimate the density $p(y|s_t)$ in (6.1) as the output of the *predictor* given the embedding $\mathbf{e}(s_t)$, i.e., $\bar{y}_t \triangleq g_\phi(\mathbf{e}(s_t))$.

6.3.1 Loss Functions

In this subsection, we define loss functions to achieve our objective in (6.1); the details of how we train our model will be discussed in the following subsection.

Predictive Clustering Loss: Since finding the cluster assignment of a given sequence is a probabilistic problem due to the sampling process, the objective function in (6.1) must be defined as an expectation over the cluster assignment. Thus, we can estimate solving the objective problem in (6.1) as minimizing the following loss function:

$$\mathcal{L}_1(\theta, \psi, \phi, \mathcal{E}) = \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\Big[\sum_{t\in\mathcal{T}} \mathbb{E}_{s_t\sim Cat(\pi_t)}\big[\ell_1(y_t, \bar{y}_t)\big]\Big] \tag{6.2}$$

where $\ell_1(y_t, \bar{y}_t) = -\sum_{c=1}^{C} y_t^c \log \bar{y}_t^c$. Here, we slightly abuse the notation and denote $y = [y^1 \cdots y^C]$ as the one-hot encoding of y, and y^c and $\bar{y}^c$ indicates the c-th component of y and $\bar{y}$, respectively. It is worth to highlight that minimizing ℓ_1 is equivalent to minimizing the KL divergence in (6.1) since the former term of the KL divergence is independent of our optimization procedure.

One critical question that may arise is how to avoid trivial solutions in this unsupervised setting of identifying the cluster assignments and the centroids [167]. For example, all the embeddings in $\mathcal{E}$ may collapse into a single point or the selector simply assigns equal probability to all the clusters regardless of the input sequence. In both cases, our model will fail to correctly estimate $p(y|s_t)$ and, thus, end up finding a trivial solution. To address this issue, we introduce two auxiliary loss functions that are tailored to address this concern. It is worth to highlight that these loss functions are not subject to the sampling process and their gradients can be simply back-propagated.

Sample-Wise Entropy of Cluster Assignment: To motivate sparse cluster assignment such that the selector ultimately selects one dominant cluster for each sequence, we introduce sample-wise entropy of cluster assignment which is given as

$$\mathcal{L}_2(\theta, \psi) = \mathbb{E}_{\mathbf{x}\sim p_X}\Big[-\sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}} \pi_t(k)\log\pi_t(k)\Big] \tag{6.3}$$

where $\pi_t = [\pi_t(1) \cdots \pi_t(K)] = h_\psi(f_\theta(\mathbf{x}_{1:t}))$. The sample-wise entropy achieves its minimum when π_t becomes an one-hot vector.

Embedding Separation Loss: To prevent the embeddings in $\mathcal{E}$ from collapsing into a single point, we define a loss function that encourages the embeddings to represent different label distributions, i.e., $g_\phi(\mathbf{e}(k))$ for $k \in \mathcal{K}$, from each other:

$$\mathcal{L}_3(\mathcal{E}) = -\sum_{k\neq k'} \ell_1(g_\phi(\mathbf{e}(k)), g_\phi(\mathbf{e}(k'))) \tag{6.4}$$

where ℓ_1 is reused to quantify the distance between label distributions conditioned on each cluster. We minimize (6.4) when updating the embedding vectors $\mathbf{e}(1), \cdots, \mathbf{e}(K)$.

6.3.2 Optimization

The optimization problem in (6.1) is a non-convex combinatorial problem because it comprises not only minimizing the KL divergence but also finding the optimal cluster assignments and centroids. Hence, we propose an optimization procedure that iteratively solves two subproblems: i) optimizing the three networks – the encoder, selector, and predictor – while fixing the embedding dictionary and ii) optimizing the embedding dictionary while fixing the three networks. We provide the pseudo-code for optimizing our AC-TPC in Algorithm 3 and that for initializing the parameters in Algorithm 4.

6.3.2.1 Optimizing the Three Network

Finding predictive clusters incorporates the sampling process which is non-differentiable. Thus, to render "back-propagation", we utilize the training of actor-critic models [166]. More specifically, we view the combination of the encoder (f_θ) and the selector (h_ψ) as the "actor" parameterized by $\omega_A = [\theta, \psi]$, and the predictor ($g_\phi$) as the "critic". The critic takes as input the the output of the actor (i.e., the cluster assignment) and estimates its value based on the sample-wise predictive clustering loss (i.e., $\ell_1(y_t, \bar{y}_t)$) given the chosen cluster. This, in turn, renders the actor to change the distribution of selecting a cluster to minimize such loss. Thus, it is important for the critic to perform well on the updated output of the actor while it is important for the actor to perform well on the updated loss estimation. As such, the parameters for the actor and the critic need to be updated iteratively.

Given the embedding dictionary $\mathcal{E}$ fixed (thus, we will omit the dependency on $\mathcal{E}$), we train the actor, i.e., the encoder and the selector, by minimizing a combination of the predictive clustering loss $\mathcal{L}_1$ and the entropy of cluster assignments $\mathcal{L}_2$, which is given by

Algorithm 3 Pseudo-code for Optimizing AC-TPC

```
**Input:** Dataset $\mathcal{D} = \{(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}\}_{n=1}^{N}$, number of clusters $K$, coefficients $(\alpha, \beta)$,
        learning rate $(\eta_A, \eta_C, \eta_E)$, mini-batch size $n_{mb}$, and update step $M$
**Output:** AC-TPC parameters $(\theta, \psi, \phi)$ and the embedding dictionary $\mathcal{E}$
Initialize parameters $(\theta, \psi, \phi)$ and the embedding dictionary $\mathcal{E}$ via Algorithm 4
**repeat**
    *Optimize the Encoder, Selector, and Predictor*
    **for** $m = 1, \cdots, M$ **do**
        Sample a mini-batch of $n_{mb}$ data samples: $\{(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}\}_{n=1}^{n_{mb}} \sim \mathcal{D}$
        **for** $n = 1, \cdots, n_{mb}$ **do**
            Calculate the assignment probability:   $\pi_t^n = [\pi_t^n(1) \cdots \pi_t^n(K)] \leftarrow h_\psi(f_\theta(\mathbf{x}_{1:t}^n))$
            Draw the cluster assignment:   $s_t^n \sim Cat(\pi_t^n)$
            Calculate the label distributions:   $\bar{y}_t^n \leftarrow g_\phi(\mathbf{e}(s_t^n))$ and $\hat{y}_t^n \leftarrow g_\phi(f_\theta(\mathbf{x}_{1:t}^n))$
        **end for**
        Update the encoder $f_\theta$ and selector $h_\psi$:
            $\theta \leftarrow \theta - \eta_A \left( \frac{1}{n_{mb}} \sum_{n=1}^{n_{mb}} \sum_{t=1}^{T^n} \ell_1(y_t^n, \bar{y}_t^n) \nabla_\theta \log \pi_t^n(s_t^n) - \alpha \nabla_\theta \sum_{k=1}^{K} \pi_t^n(k) \log \pi_t^n(k) \right)$
            $\psi \leftarrow \psi - \eta_A \left( \frac{1}{n_{mb}} \sum_{n=1}^{n_{mb}} \sum_{t=1}^{T^n} \ell_1(y_t^n, \bar{y}_t^n) \nabla_\psi \log \pi_t^n(s_t^n) - \alpha \nabla_\psi \sum_{k=1}^{K} \pi_t^n(k) \log \pi_t^n(k) \right)$
        Update the predictor $g_\phi$:
            $\phi \leftarrow \phi - \eta_C \frac{1}{n_{mb}} \sum_{n=1}^{n_{mb}} \sum_{t=1}^{T^n} \nabla_\phi \ell_1(y_t^n, \bar{y}_t^n)$
    **end for**
    *Optimize the Cluster Centroids*
    **for** $m = 1, \cdots, M$ **do**
        Sample a mini-batch of $n_{mb}$ data samples: $\{(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}\}_{n=1}^{n_{mb}} \sim \mathcal{D}$
        **for** $n = 1, \cdots, n_{mb}$ **do**
            Calculate the assignment probability:   $\pi_t^n = [\pi_t^n(1) \cdots \pi_t^n(K)] \leftarrow h_\psi(f_\theta(\mathbf{x}_{1:t}^n))$
            Draw the cluster assignment:   $s_t^n \sim Cat(\pi_t^n)$
            Calculate the label distributions:   $\bar{y}_t^n \leftarrow g_\phi(\mathbf{e}(s_t^n))$
        **end for**
        **for** $k = 1, \cdots, K$ **do**
            Update the embeddings $\mathbf{e}(k)$:
                $\mathbf{e}(k) \leftarrow \mathbf{e}(k) - \eta_E \left( \frac{1}{n_{mb}} \sum_{n=1}^{n_{mb}} \sum_{t=1}^{T^n} \nabla_{\mathbf{e}(k)} \ell_1(y_t^n, \bar{y}_t^n) - \gamma \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \nabla_{\mathbf{e}(k)} \ell_1\big(g_\phi(\mathbf{e}(k)), g_\phi(\mathbf{e}(k'))\big) \right)$
        **end for**
        Update the embedding dictionary:   $\mathcal{E} \leftarrow \{\mathbf{e}(1), \ldots \mathbf{e}(K)\}$
    **end for**
**until** convergence
```

Algorithm 4 Pseudo-code for pre-training AC-TPC

```
**Input:** Dataset $\mathcal{D} = \{(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}\}_{n=1}^{N}$, number of clusters $K$, learning rate $\eta$, mini-batch size $n_{mb}$
**Output:** AC-TPC parameters $(\theta, \psi, \phi)$ and the embedding dictionary $\mathcal{E}$
Initialize parameters $(\theta, \psi, \phi)$ via Xavier Initializer
Pre-train the Encoder and Predictor
repeat
    Sample a mini-batch of $n_{mb}$ data samples: $\{(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}\}_{n=1}^{n_{mb}} \sim \mathcal{D}$
    for $n = 1, \cdots, n_{mb}$ do
        Calculate the label distributions:   $\hat{y}_t^n \leftarrow g_\phi(f_\theta(\mathbf{x}_{1:t}^n))$
    end for
        $\theta \leftarrow \theta - \eta \frac{1}{n_{mb}} \sum_{n=1}^{n_{mb}} \sum_{t=1}^{T^n} \nabla_\theta \ell_1(y_t^n, \hat{y}_t^n)$        $\phi \leftarrow \phi - \eta \frac{1}{n_{mb}} \sum_{n=1}^{n_{mb}} \sum_{t=1}^{T^n} \nabla_\phi \ell_1(y_t^n, \hat{y}_t^n)$
until convergence
Initialize the Cluster Centroids
Calculate the embedding dictionary $\mathcal{E}$ and initial cluster assignments $c_t^n$
        $\mathcal{E}, \{\{c_t^n\}_{t=1}^{T^n}\}_{n=1}^{N} \leftarrow \texttt{K-means}(\{\{\mathbf{z}_t^n\}_{t=1}^{T^n}\}_{n=1}^{N}, K)$
Pre-train the Selector
repeat
    Sample a mini-batch of $n_{mb}$ data samples: $\{(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}\}_{n=1}^{n_{mb}} \sim \mathcal{D}$
    for $n = 1, \cdots, n_{mb}$ do
        Calculate the assignment probability:   $\pi_t^n = [\pi_t^n(1) \cdots \pi_t^n(K)] \leftarrow h_\psi(f_\theta(\mathbf{x}_{1:t}^n))$
    end for
    Update the selector $h_\psi$:
        $\psi \leftarrow \psi + \eta \frac{1}{n_{mb}} \sum_{n=1}^{n_{mb}} \sum_{t=1}^{T^n} \sum_{k=1}^{K} c_t^n(k) \log \pi_t^n(k)$
until convergence
```

$\mathcal{L}_A(\theta, \psi, \phi) = \mathcal{L}_1(\theta, \psi, \phi) + \alpha \mathcal{L}_2(\theta, \psi)$ where $\alpha \geq 0$ is a coefficient chosen to balance between the two losses. To derive the gradient of this loss with respect $\omega_A = [\theta, \psi]$, we utilize the ideas from actor-critic models [166] in (6.5).

$$\begin{aligned}\nabla_{\omega_A} \mathcal{L}_A(\theta, \psi, \phi) &= \mathbb{E}_{\mathbf{x}, y \sim p_{XY}} \left[\nabla_{\omega_A} \left(\sum_{t=1}^{T} \mathbb{E}_{s_t \sim Cat(\pi_t)} \big[\ell_1(y_t, \bar{y}_t) \big] \right) \right] + \alpha \nabla_{\omega_A} \mathcal{L}_2(\theta, \psi) \\ &= \mathbb{E}_{\mathbf{x}, y \sim p_{XY}} \left[\sum_{t=1}^{T} \mathbb{E}_{s_t \sim Cat(\pi_t)} \big[\ell_1(y_t, \bar{y}_t) \nabla_{\omega_A} \log \pi_t(s_t) \big] \right] + \alpha \nabla_{\omega_A} \mathcal{L}_2(\theta, \psi),\end{aligned} \tag{6.5}$$

where the second equality comes from the following derivation of the former term:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x},y\sim p_{XY}}\left[\nabla_{\omega_A}\left(\sum_{t=1}^{T}\mathbb{E}_{s_t\sim Cat(\pi_t)}\left[\ell_1(y_t,\bar{y}_t)\right]\right)\right] &= \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\left[\nabla_{\omega_A}\left(\sum_{t=1}^{T}\sum_{s_t\in\mathcal{K}}\pi_t(s_t)\ell_1(y_t,\bar{y}_t)\right)\right] \\
&= \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\left[\sum_{t=1}^{T}\sum_{s_t\in\mathcal{K}}\nabla_{\omega_A}\pi_t(s_t)\ell_1(y_t,\bar{y}_t)\right] \\
&= \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\left[\sum_{t=1}^{T}\sum_{s_t\in\mathcal{K}}\frac{\nabla_{\omega_A}\pi_t(s_t)}{\pi_t(s_t)}\pi_t(s_t)\ell_1(y_t,\bar{y}_t)\right] \\
&= \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\left[\sum_{t=1}^{T}\sum_{s_t\in\mathcal{K}}\pi_t(s_t)\ell_1(y_t,\bar{y}_t)\nabla_{\omega_A}\log\pi_t(s_t)\right] \\
&= \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\left[\sum_{t=1}^{T}\mathbb{E}_{s_t\sim Cat(\pi_t)}\left[\ell_1(y_t,\bar{y}_t)\nabla_{\omega_A}\log\pi_t(s_t)\right]\right].
\end{aligned}$$

Note that since no sampling process is considered in $\mathcal{L}_2(\theta,\psi)$, we can simply derive $\nabla_{\omega_A}\mathcal{L}_2(\theta,\psi)$.

Iteratively with training the actor, we train the critic, i.e., the predictor, by minimizing the predictive clustering loss $\mathcal{L}_1$ as the following: $\mathcal{L}_C(\phi)=\mathcal{L}_1(\theta,\psi,\phi)$ whose gradient with respect to ϕ can be givens as $\nabla_\phi\mathcal{L}_C(\phi)=\nabla_\phi\mathcal{L}_1(\theta,\psi,\phi)$. Note that since the critic is independent of the sampling process, the gradient can be simply back-propagated.

6.3.2.2 Optimizing the Cluster Centroids

Now, once the parameters for the three networks (θ,ψ,ϕ) are fixed (thus, we omit the dependency on θ, ψ, and ϕ), we updated the embeddings in $\mathcal{E}$ by minimizing a combination of the predictive clustering loss $\mathcal{L}_1$ and the embedding separation loss $\mathcal{L}_3$, which is given by $\mathcal{L}_E(\mathcal{E})=\mathcal{L}_1(\mathcal{E})+\beta\mathcal{L}_3(\mathcal{E})$ where $\beta\geq 0$ is a coefficient chosen to balance between the two losses.

6.3.2.3 Initializing AC-TPC via Pre-Training

Since we transform the combinatorial optimization problem in (6.1) into iteratively solving two sub-problems, initialization is crucial to achieve better optimization as a similar concern has been addressed in [167].

Therefore, we initialize our model based on the following procedure. First, we pre-train the encoder and the predictor by minimizing the following loss function based on the predicted label distribution given the latent encodings of input sequences, i.e., $\hat{y}_t \triangleq g_\phi(\mathbf{z}_t) = g_\phi(f_\theta(\mathbf{x}_{1:t}))$, as the following:

$$\mathcal{L}_I(\theta, \phi) = \mathbb{E}_{\mathbf{x},y\sim p_{XY}}\Big[-\sum_{t\in\mathcal{T}} \ell_1(y_t, \hat{y}_t)\Big]. \tag{6.6}$$

Minimizing (6.6) encourages the latent encoding to be enriched with information for accurately predicting the label distribution. Then, we perform K-means (other clustering method can be also applied) based on the learned representations to initialize the embeddings $\mathcal{E}$ and the cluster assignments $\{\{s_t^n\}_{t=1}^{T^n}\}_{n=1}^N$. Finally, we pre-train the selector h_ψ by minimizing the cross entropy treating the initialized cluster assignments as the true clusters.

6.4 Related Work

Temporal clustering, also known as time-series clustering, is a process of unsupervised partitioning of the time-series data into clusters in such a way that homogeneous time-series are grouped together based on a certain similarity measure. Temporal clustering is challenging because i) the data is often high-dimensional – it consists of sequences not only with high-dimensional features but also with many time points – and ii) defining a proper similarity measure for time-series is not straightforward since it is often highly sensitive to distortions [168]. To address these challenges, there have been various attempts to find a good representation with reduced dimensionality or to define a proper similarity measure for times-series [169].

Recently, [170] and [171] proposed temporal clustering methods that utilize low-dimensional representations learned by RNNs. These works are motivated by the success of applying deep neural networks to find "clustering friendly" latent representations for clustering static data [167, 172]. In particular, authors in [170] utilized a modified LSTM auto-encoder to find the latent representations that are effective to summarize the input time-series and conducted K-means on top of the learned representations as an ad-hoc process. Similarly, authors in [171] proposed a bidirectional-LSTM auto-encoder that jointly optimizes the reconstruction loss for dimensionality reduction and the clustering objective. However, these methods do not associate a target property with clusters and, thus, provide little prognostic value about the underlying disease progression.

Our work is most closely related to SOM-VAE [173]. This method jointly optimizes a static variational auto-encoder (VAE), that finds latent representations of input features, and a self-organizing map (SOM), that allows to map the latent representations into a more interpretable discrete representations, i.e., the embeddings. However, there are three key differences between our work and SOM-VAE. First, SOM-VAE aims at minimizing the reconstruction loss that is specified as the mean squared error between the original input and the reconstructed input based on the corresponding embedding. Thus, similar to the aforementioned methods, SOM-VAE neither associates future outcomes of interest with clusters. In contrast, we focus on minimizing the KL divergence between the outcome distribution given the original input sequence and that given the corresponding embedding to build association between future outcomes of interest and clusters. Second, to overcome non-differentiability caused by the sampling process (that is, mapping the latent representation to the embeddings), [173] applies the gradient copying technique proposed by [174], while we utilize the training of actor-critic model [166]. Finally, while we flexibly model time-series using LSTM, SOM-VAE handles time-series by integrating a Markov model in the latent representations. This can be a strict assumption especially in clinical settings where a patient's medical history is informative for predicting the future clinical outcomes [31].

6.5 Experiments

In this section, we provide a set of experiments using two real-world time-series datasets. We iteratively update the three networks – the encoder, selector, and predictor – and the embedding dictionary as described in Section 6.3.2. For the network architecture, we constructed the encoder utilizing a single-layer LSTM [141] with 50 nodes and constructed the selector and predictor utilizing two-layer fully-connected network with 50 nodes in each layer, respectively. The parameters (θ, ψ, ϕ) are initialized by Xavier initialization [175] and optimized via Adam optimizer [176] with learning rate of 0.001 and keep probability 0.7. We chose the balancing coefficients $\alpha, \beta \in \{0.001, 0.01, 0.1, 1.0\}$ utilizing grid search that achieves the minimum validation loss in (6.2); the effect of different loss functions are further investigated in the experiments. Here, all the results are reported using 5 random 64/16/20 train/validation/test splits.

6.5.1 Real-World Datasets

We conducted experiments to investigate the performance of AC-TPC on two real-world medical datasets.

UK Cystic Fibrosis registry (UKCF)[3]: This dataset records annual follow-ups for 5,171 adult patients (aged 18 years or older) enrolled in the UK CF registry over the period from 2008 and 2015, with a total of 25,012 hospital visits. Each patient is associated with 89 variables (i.e., 11 static and 78 time-varying features), including information on demographics and genetic mutations, bacterial infections, lung function scores, therapeutic managements, and diagnosis on comorbidities. We set the development of different comorbidities in the next year as the label of interest at each time stamp.

[3] https://www.cysticfibrosis.org.uk

Alzheimer's Disease Neuroimaging Initiative (ADNI)[4]**:** This dataset consists of 1,346 patients in the Alzheimer's disease study with a total of 11,651 hospital visits, which tracks the disease progression via follow-up observations at 6 months interval. Each patient is associated with 21 variables (i.e., 5 static and 16 time-varying features), including information on demographics, biomarkers on brain functions, and cognitive test results. We set predictions on the three diagnostic groups – normal brain functioning, mild cognitive impairment, and Alzheimer's disease – as the label of interest at each time stamp.

6.5.2 Benchmarks

We compare AC-TPC with clustering methods ranging from conventional approaches based on K-means to the state-of-the-art approaches based on deep neural networks. All the benchmarks compared in the experiments are tailored to incorporate time-series data as described below:

Dynamic time warping followed by K-means: Dynamic time warping (DTW) is utilized to quantify pairwise distance between two variable-length sequences and, then, K-means is applied (**KM-DTW**).

K-means with deep neural networks: To handle variable-length time-series data, we utilize our encoder and predictor that are trained based on (6.6) for fixed-length dimensionality reduction. Then, we apply K-means on the latent encodings $\mathbf{z}$ (**KM-E2P ($\mathcal{Z}$)**) and on the predicted label distributions $\hat{y}$ (**KM-E2P ($\mathcal{Y}$)**), respectively.

Extensions of DCN [167]: Since the DCN is designed for static data, we replace their static auto-encoder with a sequence-to-sequence network to incorporate time-series data (**DCN-S2S**).[5] To associated with the label distribution, we compare a DCN whose static

[4] https://adni.loni.usc.edu

[5] This extension is a representative of recent deep learning approaches for clustering of both static data [167, 172] and time-series data [170, 171] since these methods are built upon the same concept – that

Table 6.1: Comparison table of benchmarks.

Methods	**Handling Time-Series**	**Clustering Method**	**Similarity Measure**	**Label Provided**	**Label Associated**
KM-DTW	DTW	K-means	DTW	N	N
KM-E2P ($\mathcal{Z}$)	RNN	K-means	Euclidean in $\mathcal{Z}$	Y	Y (indirect)
KM-E2P ($\mathcal{Y}$)	RNN	K-means	Euclidean in $\mathcal{Y}$	Y	Y (direct)
DCN-S2S	RNN	K-means	Euclidean in $\mathcal{Z}$	N	N
DCN-E2P	RNN	K-means	Euclidean in $\mathcal{Z}$	Y	Y (indirect)
SOM-VAE	Markov model	embedding mapping	reconstruction loss	N	N
SOM-VAE-P	Markov model	embedding mapping	prediction loss	Y	Y (direct)
Proposed	RNN	embedding mapping	KL divergence	Y	Y (direct)

auto-encoder is replaced with our encoder and predictor (**DCN-E2P**) to focus dimensionality reduction while preserving information for label prediction.

SOM-VAE [173]: We compare with SOM-VAE – though, this method aims at visualizing input – since it naturally clusters time-series data (**SOM-VAE**). In addition, we compare with a variation of SOM-VAE by replacing the decoder with our predictor to find embeddings that capture information for predicting the label (**SOM-VAE-P**). For both cases, we set the dimension of SOM to K.

It is worth highlighting that the label information is provided for training DCN-E2P, KM-E2P, and SOM-VAE-P while the label information is not provided for training KM-DTW, DCN-S2S, and SOM-VAE. We compared and summarized major components of the benchmarks in Table 6.1.

is, applying deep networks for dimensionality reduction to conduct conventional clustering methods, e.g., K-means.

6.5.3 Performance Metrics

Clustering Performance: We applied the following three standard metrics for evaluating clustering performances when the ground-truth cluster label is available: *purity score*, *normalized mutual information* (NMI) [177], and *adjusted Rand index* (ARI) [178]. More specifically, the purity score assesses how homogeneous each cluster is (ranges from 0 to 1 where 1 being a cluster consists of a single class), the NMI is an information theoretic measure of how much information is shared between the clusters and the labels that is adjusted for the number of clusters (ranges from 0 to 1 where 1 being a perfect clustering), and ARI is a corrected-for-chance version of the Rand index which is a measure of the percentage of correct cluster assignments (ranges from -1 to 1 where 1 being a perfect clustering and 0 being a random clustering).

When the ground-truth label is not available, we utilize the average *Silhouette index* (SI) [179] which measures how similar a member is to its own cluster (homogeneity within a cluster) compared to other clusters (heterogeneity across clusters). Formally, the SI for a subsequence $\mathbf{x}^n_{1:t} \in \mathcal{C}^k$ can be given as follows: $SI(n) = \frac{b(n)-a(n)}{\max(a(n),b(n))}$ where $a(n) = \frac{1}{|\mathcal{C}^k|-1}\sum_{m\neq n} \|y^n_t - y^m_t\|_1$ and $b(n) = \min_{k'\neq k} \frac{1}{|\mathcal{C}^{k'}|}\sum_{m\in\mathcal{C}^{k'}} \|y^n_t - y^m_t\|_1$. Here, we used the L1-distance between the ground-truth labels of the future outcomes of interest since our goal is to group input subsequences with similar future outcomes.

Prediction Performance: To assess the prediction performance of the identified predictive clusters, we utilized both a*rea under receiver operator characteristic curve* (AUROC) and *area under precision-recall curve* (AUPRC) based on the label predictions of each cluster and the ground-truth binary labels on the future outcomes of interest. Note that the prediction performance is available only for the benchmarks that incorporate the label information during training.

Table 6.2: Performance comparison on the UKCF and ADNI datasets.

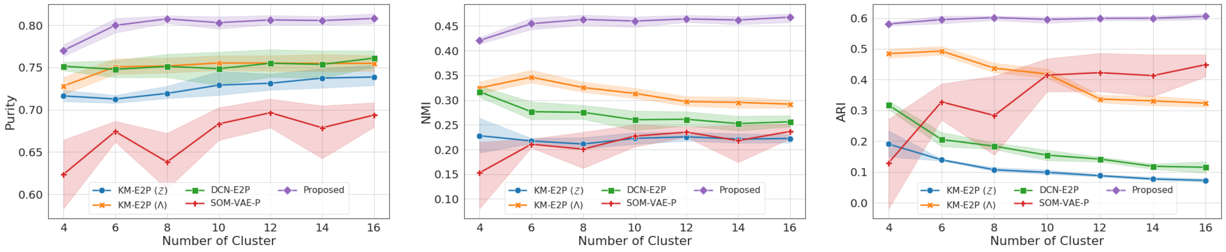
Dataset	**Method**	**Purity**	**NMI**	**ARI**	**AUROC**	**AUPRC**
UKCF	KM-DTW	0.573±0.01*	0.010±0.01*	0.014±0.01*	N/A	N/A
	KM-E2P ($\mathcal{Z}$)	0.719±0.01*	0.211±0.01*	0.107±0.01*	0.726±0.01*	0.425±0.02*
	KM-E2P ($\mathcal{Y}$)	0.751±0.01*	0.325±0.01*	0.440±0.02*	0.807±0.00*	0.514±0.01*
	DCN-S2S	0.607±0.06*	0.059±0.08*	0.063±0.09*	N/A	N/A
	DCN-E2P	0.751±0.02*	0.275±0.02*	0.184±0.01*	0.772±0.03*	0.487±0.03*
	SOM-VAE	0.573±0.01*	0.006±0.00*	0.006±0.01*	N/A	N/A
	SOM-VAE-P	0.638±0.04*	0.201±0.05*	0.283±0.17†	0.754±0.05*	0.331±0.07*
	Proposed	**0.807±0.01**	**0.463±0.01**	**0.602±0.01**	**0.843±0.01**	**0.605±0.01**
ADNI	KM-DTW	0.566±0.02*	0.019±0.02*	0.006±0.02*	N/A	N/A
	KM-E2P ($\mathcal{Z}$)	0.736±0.03†	0.249±0.02	0.230±0.03†	0.707±0.01*	0.509±0.01
	KM-E2P ($\mathcal{Y}$)	0.776±0.05	0.264±0.07	0.317±0.11	0.756±0.04	0.503±0.04
	DCN-S2S	0.567±0.02*	0.005±0.00*	0.000±0.01*	N/A	N/A
	DCN-E2P	0.749±0.06	0.261±0.05	0.215±0.06†	0.721±0.03†	0.509±0.03
	SOM-VAE	0.566±0.02*	0.040±0.06*	0.011±0.02*	N/A	N/A
	SOM-VAE-P	0.586±0.06*	0.085±0.08*	0.038±0.06*	0.597±0.10†	0.376±0.05*
	Proposed	**0.786±0.03**	**0.285±0.04**	**0.330±0.06**	**0.768±0.02**	**0.515±0.02**

$*$ indicates p-value < 0.01, $\dagger$ indicates p-value < 0.05

6.5.4 Clustering Performance

We start with a simple scenario where the true class (i.e., the ground-truth cluster label) is available and the number of classes is tractable. In particular, we set $C = 2^3 = 8$ based on the binary labels for the development of three common comorbidities of cystic fibrosis – diabetes, ABPA, and intestinal obstruction – in the next year for the UKCF dataet and $C = 3$ based on the mutually exclusive three diagnostic groups for the ADNI dataset. We compare AC-TPC against the aforementioned benchmarks with respect to the clustering and prediction performance in Table 6.2.

As shown in Table 6.2, AC-TPC achieved performance gain over all the tested benchmarks in terms of both clustering and prediction performance – where most of the improvements were



(a) The averaged purity score.

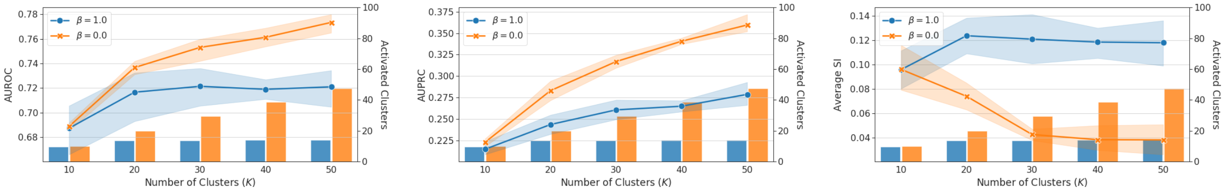
(b) The averaged NMI.

(c) The averaged ARI.

Figure 6.3: The purity score, NMI, and ARI (mean and 95% confidence interval) for the UKCF dataset ($C = 8$) with various K.

statistically significant with p-value < 0.01 or p-value < 0.05 – for both datasets. Importantly, clustering methods – i.e., KM-DTW, DCN-S2S, and SOM-VAE – that do not associate with the future outcomes of interest identified clusters that provide little prognostic value on the future outcomes (note that the true class is derived from the future outcome of interest). This is clearly shown by the ARI value near 0 which indicates that the identified clusters have no difference with random assignments. Therefore, similar sequences with respect to the latent representations tailored for reconstruction or with respect to the shape-based measurement using DTW can have very different outcomes.

In Figure 6.3, we further investigate the purity score, NMI, and ARI by varying the number of clusters K from 4 to 16 on the UKCF dataset in the same setting with that stated above (i.e., $C = 8$). Here, the three methods – i.e., KM-DTW, DCN-S2S, and SOM-VAE – are excluded for better visualization. As we can see in Figure 6.3, our model rarely incur performance loss in both NMI and ARI while the benchmarks (except for SOM-VAE-P) showed significant decrease in the performance as K increased (higher than C). This is because the number of clusters identified by AC-TPC (i.e., the number of activated clusters where we define cluster k is activated if $|\mathcal{C}(k)| > 0$) was the same with C most of the times, while the DCN-based methods identified exactly K clusters (due to the K-means). Since the NMI and ARI are adjusted for the number of clusters, a smaller number of identified



(a) AUROC

(b) AUPRC

(c) Average SI

Figure 6.4: AUROC, AUPRC, and average SI (mean and 95% confidence interval) and the number of activated clusters for the UKCF dataset ($C = 2^{22}$) with various K.

clusters yields, if everything being equal, a higher performance. In contrast, while our model achieved the same purity score for $K \geq 8$, the benchmark showed improved performance as K increased since the purity score does not penalize having many clusters. This is an important property of AC-TPC that we do not need to know a priori what the number of cluster is which is a common practical challenge of applying the conventional clustering methods (e.g., K-means).

The performance gain of our model over SOM-VAE-P (and, our analysis is the same for SOM-VAE) comes from two possible sources: i) SOM-VAE-P mainly focuses on visualizing the input with SOM which makes both the encoder and embeddings less flexible – this is why it performed better with higher K – and ii) the Markov property can be too strict for time-series data especially in clinical settings where a patient's medical history is informative for predicting the future clinical outcomes [31].

6.5.5 Multiple Future Outcomes – a Practical Scenario

In this experiment, we focus on a more practical scenario where the future outcome of interest is high-dimensional and, thus, the number of classes based on all the possible combinations of future outcomes becomes intractable. Suppose that we are interested in the development of M comorbidities in the next year whose possible combinations grow exponentially $C = 2^M$. Interpreting such a large number of patient subgroups will be a daunting task which hinders

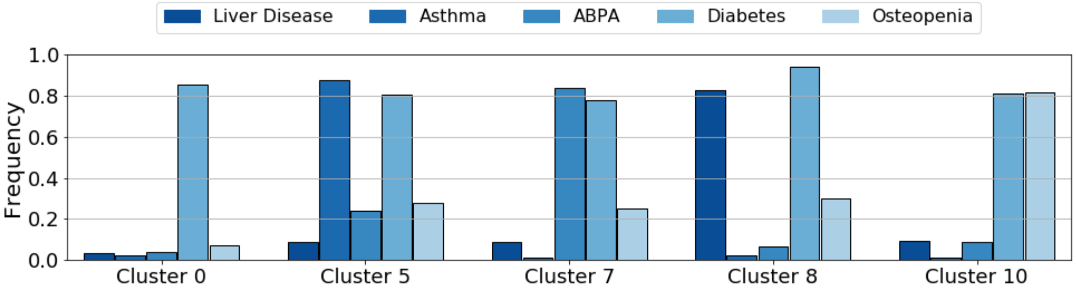


Figure 6.5: Clusters with high-risk of developing diabetes.

the understanding of underlying disease progression. Since different comorbidities may share common driving factors [180], we hope our model to identify much smaller underlying (latent) clusters that govern the development of comorbidities. Here, to incorporate with M comorbidities (i.e., M binary labels), we redefine the output space as $\mathcal{Y} = \{0, 1\}^M$ and modify the predictor and loss functions, accordingly.

We identified 12 clusters of patients based on the next-year development of 22 different comorbidities in the UKCF dataset and reported 5 clusters in Figure 6.5 – Cluster 0, 5, 7, 8, and 10 – with the frequency of developing important comorbidities in the next year. Here, we selected the 5 clusters that have the highest risk of developing diabetes in the next year, and the frequency is calculated in a cluster-specific fashion using the true label.

Although all these clusters displayed high risk of diabetes, the frequency of other co-occurred comorbidities was significantly different across the clusters. In particular, around 89% of the patients in Cluster 5 experienced asthma in the next year while it was less than 3% of the patients in the other cluster. Interestingly, "leukotriene" – a medicine commonly used to manage asthma – and "FEV$_1$% predicted" – a measure of lung function – were the two most different input features between patients in Cluster 5 and those in the other clusters. We observed similar findings in Cluster 7 with ABPA, Cluster 8 with liver disease, and Cluster 10 with osteopenia. Therefore, by grouping patients who are likely to develop a similar set of comorbidities, our method identified clusters that can be translated into actionable information for clinical decision-making.

6.5.6 Trade-Off between Clustering and Prediction

In predictive clustering, the trade-off between the clustering performance (for better interpretability) – which quantifies how the data samples are homogeneous within each cluster and heterogeneous across clusters with respect to the future outcomes of interest – and the prediction performance is a common issue. The most important parameter that governs this trade-off is the number of clusters. More specifically, increasing the number of clusters will make the predictive clusters have higher diversity to represent the output distribution and, thus, will increase the prediction performance while decreasing the clustering performance. One extreme example is that there are as many clusters as data samples which will make the identified clusters fully individualized; as a consequence, each cluster will lose interpretability as it no longer groups similar data samples.

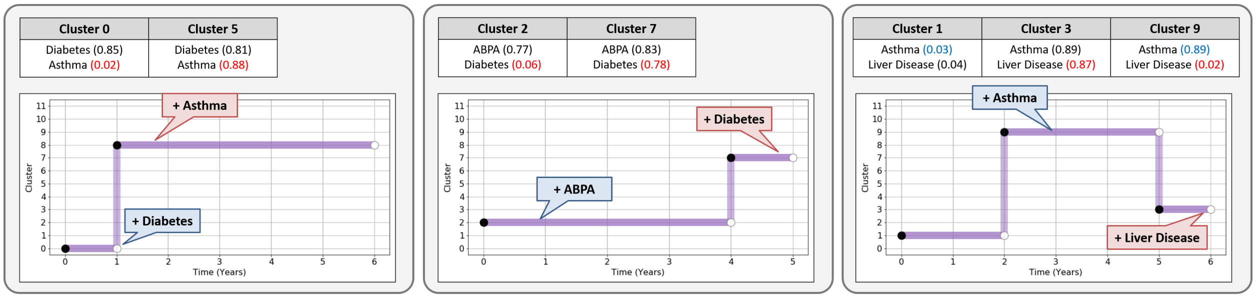
To highlight this trade-off, we conduct experiments under the same experimental setup with that of Section 6.5.5. For the performance measures, we utilized the AUROC and AUPRC to assess the prediction performance, and utilized the average SI to assess the clustering performance. To control the number of activated clusters, we set $\beta = 0$ and $\beta = 1$ (since the embedding separation loss in (6.4) controls the activation of clusters) and reported the performance by increasing the number of possible clusters K, i.e., the dimension of the embedding dictionary.

As can be seen in Figure 6.4, the prediction performance increased with a increasing number of identified clusters due to the higher diversity to represent the label distribution while making the identified clusters less interpretable. That is, the cohesion and separation among clusters become ambiguous as shown in the low average SI. On the other hand, when we set $\beta = 1.0$ (which is selected based on the validation loss in 6.2), our method consistently identified a similar number of clusters for $K > 20$, i.e., 13.8 on average, in a data-driven fashion and provided slightly reduced prediction performance with significantly better interpretability, i.e., the average SI 0.120 on average.

6.5.7 How Does the Temporal Phenotypes Change over Time?

In this subsection, we demonstrate run-time examples of how AC-TPC flexibly updates the cluster assignments over time with respect to the future development of comorbidities in the next year. Figure 6.6 illustrates three representative patients:

- **Patient A** had diabetes from the beginning of the study and developed asthma as an additional comorbidity at $t = 2$. Accordingly, AC-TPC changed the temporal phenotype assigned to this patient from Cluster 0, which consists of patients who are very likely to develop diabetes but very unlikely to develop asthma in the next year, to Cluster 5, which consists of patients who are likely to develop both diabetes and asthma in the next year, at $t = 1$.
- **Patient B** had ABPA from the beginning of the study and developed diabetes at $t = 5$. Similarly, AC-TPC changed the temporal phenotype assigned to this patient from Cluster 2, which consists of patients who are likely to develop ABPA but not diabetes in the next year, to Cluster 7, which consists of patients who are likely to develop both ABPA and diabetes in the next year, at $t = 4$.
- **Patient C** had no comorbidity at the beginning of the study, and developed asthma and liver disease as additional comorbidities, respectively at $t = 3$ and $t = 6$. AC-TPC changed the temporal phenotypes assigned to this patient from Cluster 1 to Cluster 9 at $t = 2$ and then to Cluster 3 at $t = 5$. The changes in the temporal phenotypes were consistent with the actual development of asthma and liver disease considering the distribution of comorbidity development in the next year – that is, Cluster 1 consists of patients who are not likely to develop any comorbidities in the next year, Cluster 9 consists of patients who are likely to develop asthma but not liver disease, and Cluster 3 consists of patients who are likely to develop asthma and liver disease in the next year.



Patient A **Patient B** **Patient C**

Figure 6.6: An illustration of run-time examples of AC-TPC on three representative patients.

6.6 Conclusion

In this work, we introduced AC-TPC, a deep learning approach for predictive clustering of time-series data. We defined novel loss functions to encourage each cluster to have homogeneous future outcomes (e.g., adverse events, the onset of comorbidities, etc.) and designed optimization procedures to avoid trivial solutions in identifying cluster assignments and the centroids. Throughout the experiments on two real-world datasets, we showed that our model achieves superior clustering performance over state-of-the-art methods and identifies meaningful clusters that can be translated into actionable information for clinical decision-making.

We believe ASAC has wide-ranging applications, both in cost reduction but also for things such as planning, in which patients can be told when they might expect to need their next check-up and for what (i.e. personalized screening).

Part III

Application to Clinical Data

CHAPTER 7

Clinical Impact: Predicting Cancer-Specific Mortality in Prostate Cancer

7.1 Contributions

Accurate prognostication is crucial in treatment decisions made for men diagnosed with non-metastatic prostate cancer. Current models rely on pre-specified variables, which limits their performance. We aimed to investigate a novel machine learning approach to develop an improved prognostic model for predicting 10-year prostate cancer-specific mortality and compare its performance with existing validated models.

Evidence before this study. Prognostic models for non-metastatic prostate cancer have hitherto been built using traditional statistical modeling with pre-specified variables and interactions. These typically place patients into risk 'groups' or 'categories' using clinico-pathological variables. A major aim for future healthcare however is to make treatment decisions more personalized. This is particularly important for men diagnosed with non-metastatic prostate cancer, where treatment choices and decisions are complex. Machine learning systems offer the possibility of individualizing predictions for these men but there are no such tools in use. We searched PubMed up to April 10, 2020, using the search phrase "prostate cancer artificial intelligence". This identified very few previous machine learning studies in prostate cancer prognostics, with the majority being relatively small, single ethnic cohort and proof-of-concept studies. In particular, there are no studies in large population cohorts and importantly none that have compared model performance or added value against currently used risk prediction models.

Added value of this study. This study used a very large (n =171,942) multiethnic, population-based, prospectively-maintained dataset to produce a machine learning-trained model to predict 10-year prostate cancer-specific mortality (PCSM). To do this we used a novel algorithm called Survival Quilts which exploits an ensemble of traditional and machine learning-based modeling techniques. The survival function learned by Survival Quilts is a combination of survival profiles from these techniques, and is therefore optimized to account for both discriminative performance and calibration. The Survival Quilts model produced in this study predicted 10-year PCSM with good discrimination and was well calibrated. In comparison to 9 other models in current clinical use, the model derived by the Survival Quilts algorithm showed comparable discrimination in predicting outcome and this was maintained when stratified by age and ethnicity. We further observed that it may add benefit when applied in a clinical decision model analysis. This study adds value by demonstrating for the first time the advantages inherent with a data-driven, variable-agnostic, machine-learning approach in predicting PCSM. This approach will only improve with further training on new datasets and the addition of further variables (for example, new imaging or molecular markers). With further development and refinement, this could be used clinically to provide superior, more individualized survival predictions.

Implications of all the available evidence. Clinicians and patients need to balance the risks of treatment benefit versus harms and consider multiple variables that may affect prognosis. Machine learning algorithms inherently lend themselves to quickly integrating data from multiple variables, such as those in prostate cancer for individual prognostic modeling. The data-driven and variable-agnostic approach inherent to machine learning also allows for an ‘information-gain’ from hitherto unsuspected contributing factors. Machine learning therefore could form the basis for a new era of prognostic models that more accurately predict individualized survival outcomes and enhance decision making information in prostate cancer and indeed other cancers.

Findings. 647,151 men with prostate cancer were enrolled into the SEER database, of whom 171,942 were included in this study. Discrimination improved with greater granularity, and multivariable models outperformed tier-based models. The Survival Quilts model showed good discrimination (C-index 0.829, 95% CI 0.820-0.838) for 10-year prostate cancer-specific mortality, which was similar to the top-ranked multivariable models: PREDICT Prostate (0.820, 0.811-0.829) and Memorial Sloan Kettering Cancer Center (MSKCC) nomogram (0.787, 0.776-0.798). All three multivariable models showed good calibration with low Brier scores (Survival Quilts 0.036, 95% CI 0.035-0.037; PREDICT Prostate 0.036, 0.035-0.037; MSKCC 0.037, 0.035-0.039). Of the tier-based systems, the Cancer of the Prostate Risk Assessment model (C-index 0.782, 95% CI 0.771-0.793) and Cambridge Prognostic Groups model (0.779, 0.767-0.791) showed higher discrimination for predicting 10-year prostate cancer-specific mortality. C-indices for models from the National Comprehensive Cancer Care Network, Genitourinary Radiation Oncologists of Canada, American Urological Association, European Association of Urology, and National Institute for Health and Care Excellence ranged from 0.711 (0.701-0.721) to 0.761 (0.750-0.772). Discrimination for the Survival Quilts model was maintained when stratified by age and ethnicity. Decision curve analysis showed an incremental net benefit from the Survival Quilts model compared with the MSKCC and PREDICT Prostate models currently used in practice.

7.2 Introduction

Prostate cancer is the commonest male cancer worldwide and its global incidence is rising [181]. Of men diagnosed, over 80% present with non-metastatic disease. Treatment decisions are particularly complex, needing to balance the risks of progression with therapy-related morbidity [182]. Accurate prognostication is therefore crucial for identifying who benefits most from treatment [183, 184].

Many nationally and internationally-endorsed tools for risk modeling are available. Most stratify men into risk groups and are derived from the 3-tiered D'Amico system, originally developed to predict biochemical recurrence (BCR) [185–190]. However, BCR is a poor surrogate for survival and prognostic models should therefore be based upon survival outcomes [191, 192]. As demonstrated recently, the simple combination of prostate-specific antigen (PSA), grade, and stage can enable effective prognostic models, and refining group-stratification systems can improve model discrimination [185, 189, 190, 193]. Work from our group and others have further demonstrated that using continuous data rather than categorization can make prognostication more accurate and personalized for clinical decision-making [194–196]. For example, the PREDICT Prostate tool and Memorial Sloan Kettering Cancer Center (MSKCC) nomogram have demonstrated high discriminative ability for predicting survival in robust external validation, and both are available as accessible web-based decision aids for patients and clinicians [194, 195].

However, even these more individualized models rely on traditional statistical modeling, with pre-specified variables and interactions. Machine learning (ML) is a data-driven application of artificial intelligence whereby systems automatically learn and improve without explicit programming. Accordingly, ML is able to autonomously exploit datasets to identify new variables and more complex relationships between them. Its application is growing rapidly in healthcare and is increasingly being used to develop novel prognostic models in several diseases [197]. We hypothesize that ML may produce a superior predictive model in prostate cancer too. For prostate cancer prognostication, ML has so far been restricted to small, proof-of-concept studies without comparison to reference standards [198–201]. Here, using a recently described, novel ML survival model, Survival Quilts, we exploited a large, national observational dataset to test a ML-trained model for predicting 10-year prostate cancer-specific mortality (PCSM) in men with non-metastatic disease [202]. We further compared its performance against available models in current clinical practice.

7.3 Methods

7.3.1 Data source and study population

Data collected through the prospectively-maintained Surveillance, Epidemiology, and End Results (SEER) program were used for this study. SEER collects data regarding cancer diagnoses and survival for approximately 30% of the United States population, and benefits from extensive quality review [203]. Men aged between 35 and 95 years diagnosed with histologically confirmed non-metastatic prostate cancer (site code C61.9) between January 1, 2000 and December 31, 2016 were included. Intact data were required for PSA, Gleason score, stage and prostate cancer specific mortality (PCSM). The primary outcome of interest was PCSM at 10 years. Time-to-event/censoring was derived from the date of diagnosis or the date of last contact (either death or the last follow-up). Biopsy core involvement was available in 66,885/171,942 (38.9%) of the men in the final cohort and derived by mean imputation where missing. Biopsy core involvement was defined as the number of cares positive for cancer as a percentage of the total number of cores taken. Access to the SEER database does not need formal ethics approval and is covered by its open access policy: https://seer.cancer.gov/data/access.html.

7.3.2 Model development

The following variables, measured at diagnosis, were included in model development: age, PSA, primary and secondary Gleason grades/Grade groups, T-stage, total number of cores examined, and core positivity (number of cancerous cores divided by number of cores taken). Magnetic resonance imaging (MRI), comorbidity, and treatment received data were not available. We derived our ML-based survival model using Survival Quilts; an open-source software developed to automate deployment of ML in survival analysis [202]. Survival Quilts is an ensemble of different survival models. Because the different models exhibit varying discriminative performance and calibration accuracy from one dataset to another, Survival Quilts learns to automatically weigh these models and tune the parameters of each individual model in a single ensemble for the dataset at hand. The survival function learned by Survival Quilts is a combination of the survival profiles produced by many models, optimized to account for both discriminative performance and calibration. This renders Survival Quilts a superset of many existing statistical models and ML-based models for survival prediction. As it is automated, Survival Quilts also provides a way to free researchers from choosing one particular survival model without need for in-depth knowledge of ML. The 4 models included in this study ranged from traditional statistical models to state-of-the-art deep learning models: Cox proportional hazards, random survival forest, conditional inference survival forest, and DeepHit models [204–206]. The turning parameters were chosen via grid search based on the validation performance on the C-index for predicting PCSM at 10 years, as described [202]. The SEER cohort was randomly split into a 64:16:20 ratio for training, validation, and testing sets and was generated using the Python package scikit-learn. For model evaluation, we used bootstrapping of 10,000 patients in the testing set, averaging over 100 iterations. Time-dependent concordance indices (C-index) and Brier scores for model discrimination and calibration, respectively, were calculated. Model calibration, reflecting predicted versus observed outcomes, was also assessed by visual inspection of calibration plots. Discrimination was assessed in the full cohort and then stratified by age groups based on the cohort median. This resulted in the following age groups: age <65 (n =79,003) and age ≥65 (n =92,939). We also stratified by different ethnicities (Black, White, other). The code for this analysis is freely available at: https://bitbucket.org/mvdschaar/mlforhealthlabpub/src/master/alg/survivalquilts.

7.3.3 Head-to-head comparison

The Survival Quilts model was compared with 9 other prognostic models in current clinical use. These included the tier-based Cambridge Prognostic Groups (CPG), European Association of Urology (EAU), National Institute for Health and Care Excellence (NICE), Genitourinary Radiation Oncologists of Canada (GUROC), American Urological Association (AUA), and

National Comprehensive Cancer Care Network (NCCN) models [185–190]. Comparison was also made against the point-based Cancer of the Prostate Risk Assessment (CAPRA) score, and the multivariable MSKCC nomogram and PREDICT Prostate model [194–196]. Due to the lack of treatment and co-morbidity data for the PREDICT Prostate model, we removed this variable from hazard calculation. Using the testing set, model performance at 10 years was compared by calculating C-index to demonstrate how well models discriminate PCSM risk. A sensitivity analysis was also performed without the biopsy core involvement variable. Decision curve analysis (DCA) was used to calculate a clinical "net benefit" for one or more prediction model in comparison to default strategies of treating all or no patients regardless of prognosis. Risk predictions on the probability of 10-year PCSM for each of the three models was calculated across a range of threshold probabilities and plotted versus intervention for no patients (none) and intervention for all patients (all). We then compare the net benefit of following these intervention strategies against use of the top 3 models for an intervention based on prognosis i.e. that an intervention is prescribed for patients with a predicted risk that exceeds a given risk threshold. We defined the net benefit as the value achieved by making decisions based on model predictions. The statistical tools used for these analyses included R and Python.

7.4 Results

7.4.1 Cohort description

Figure 7.1 shows our data assembly process. 647,151 men were enrolled into the SEER database with prostate cancer in the study period. Of these, 7,340 did not have survival data, 21,528 presented with evidence of lymph nodes or metastasis and 446,294 had missing data for at least 1 of the essential domains of PSA, Gleason Grade, or clinical stage. 47 men outside of the age range 35-95 years were also excluded. The final study population therefore included 171,942 men. Mean age was 65.6 years with mean PSA 10.1 ng/ml. The

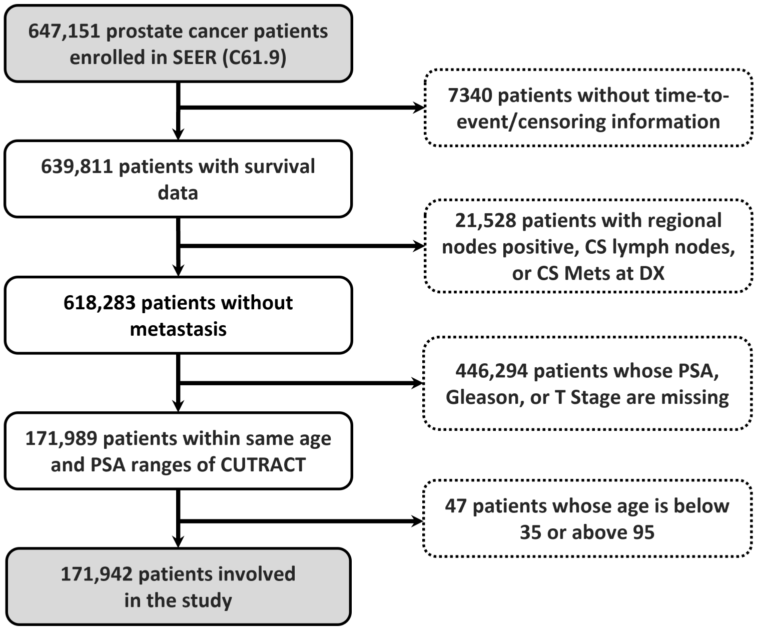

647,151 prostate cancer patients enrolled in SEER (C61.9)
7340 patients without time-to-event/censoring information
639,811 patients with survival data
21,528 patients with regional nodes positive, CS lymph nodes, or CS Mets at DX
618,283 patients without metastasis
446,294 patients whose PSA, Gleason, or T Stage are missing
171,989 patients within same age and PSA ranges of CUTRACT
47 patients whose age is below 35 or above 95
171,942 patients involved in the study

Figure 7.1: Patient data selection process.

majority of men were White (134,139/171,942 (78.0%)), with 24,488/171,942 (14.2%) and 8,925/171,942 (5.2%) of Black and Asian ethnicity, respectively. The majority of cancers were stage T1-T2 (168,573/171,942 (98.0%)) and grade group 1-3 (146,666/171,942 (85.3%)). Only a very low proportion of patients (0-0.01%) had primary Gleason <3, and so were not excluded. Median time to event for men who died with prostate cancer was 4.4 years and for the remaining cohort (including other causes of death) was 6.2 years, giving an overall median follow up of 6.1 years. By 10 years, 2,469/171,942 of the cohort died of prostate cancer, and 26,488/171,942 of other causes.

7.4.2 Survival Quilts model performance

The Survival Quilts model in this study incorporated age, PSA, biopsy involvement, clinical stage, and histological grade. The C-index for predicting PCSM was consistently high in training, validation and testing sets (0.829, 95% confidence interval (CI): 0.820-0.838) and with excellent calibration (Brier score 0.036, 95% CI: 0.035-0.037) (Table 7.1, Figure 7.2).

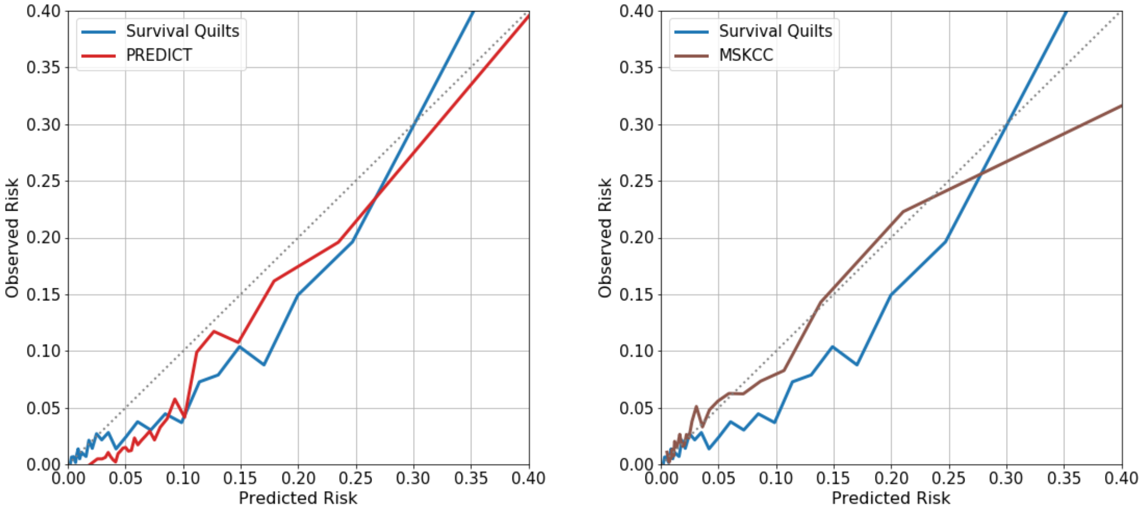
Table 7.1: Discrimination and calibration of each model at predicting 10-year prostate cancer-specific mortality.

Model	**C-index** (95% CI)	**Brier score** (95% CI)
Survival Quilts	0.829 (0.820, 0.838)	0.036 (0.035, 0.037)
PREDICT Prostate	0.820 (0.811, 0.829)	0.036 (0.035, 0.037)
MSKCC	0.787 (0.776, 0.798)	0.037 (0.035, 0.039)
CAPRA	0.782 (0.771, 0.793)	0.037 (0.035, 0.039)
CPG	0.779 (0.767, 0.791)	0.037 (0.035, 0.039)
NCCN	0.761 (0.750, 0.772)	0.038 (0.036, 0.040)
GUROC	0.750 (0.739, 0.761)	0.039 (0.037, 0.041)
AUA	0.749 (0.738, 0.760)	0.039 (0.037, 0.041)
EAU	0.711 (0.701. 0.721)	0.039 (0.037, 0.041)
NICE	0.711 (0.701. 0.721)	0.039 (0.037, 0.041)

C-index was also high when the cohort was subdivided by age (Table 7.2). Here model performance was marginally better in men aged under 65 (C-index 0.834, 95% CI: 0.817-0.851) compared to older men (C-index 0.797, 95% CI: 0.786-0.808). We next tested performance in different ethnic groups. Here the Survival Quilts model performed consistently well with very little difference in the C-index between White, Black and men of other ethnicities (C-indices 0.815-0.836) (Table 7.2).

7.4.3 Head-to-head comparison

Survival Quilts model performance compared favorably to current tier-based and multivariable models. Table 7.1 shows C-index and Brier score for each model. Amongst the tier-based systems, the CAPRA and CPG models showed higher discrimination for predicting PCSM



(a) vs PREDICT Prostate

(b) vs MSKCC model

Figure 7.2: Calibration plots of observed versus predicted risk. Prostate cancer-specific mortality at 10 years, assessed in men aged 35–95 years with non-metastatic prostate cancer. Survival Quilts model compared with the top two performing prognostic models: (left) PREDICT Prostate and (right) MSKCC model.

Table 7.2: Comparative C-index for 10-year prostate cancer-specific mortality (age-stratified)

Model	**Age** $<$ **65** $(n = 79,003)$	**Age** $\geq$ **65** $(n = 92,939)$
	C-index (95% CI)	**C-index** (95% CI)
Survival Quilts	0.834 (0.817, 0.851)	0.797 (0.786, 0.808)
PREDICT Prostate	0.819 (0.802, 0.836)	0.789 (0.778, 0.800)
MSKCC	0.830 (0.813, 0.847)	0.749 (0.737, 0.761)
CAPRA	0.818 (0.801, 0.835)	0.742 (0.730, 0.754)
CPG	0.824 (0.807, 0.841)	0.742 (0.729, 0.755)
NCCN	0.807 (0.790, 0.824)	0.725 (0.713, 0.737)

with C-indices of 0.782 (95% CI: 0.771-0.793) and 0.779 (95% CI: 0.767-0.791), respectively. C-indices for NCCN, GUROC, AUA, EAU and NICE models ranged from 0.711 (95% CI: 0.701-0.721) to 0.761 (95% CI: 0.750-0.772). The multivariable models generally discriminated patients better with C-indices of 0.820 (95% CI: 0.811-0.829) and 0.787 (95% CI: 0.776-0.798) for the PREDICT Prostate and MSKCC models, respectively. The Survival Quilts model had similarly high C-index in this cohort (0.829). Model discrimination was also maintained when the cohort was stratified by age (Table 7.2). All 3 models also showed good calibration with low Brier scores (Survival Quilts 0.036, 95% CI: 0.035-0.037; PREDICT Prostate 0.036, 95% CI: 0.035-0.037; MSKCC 0.037, 95% CI: 0.035-0.039) (Table 7.1, Figure 7.2). We further tested if these comparisons were valid given that the PREDICT Prostate and MSKCC models were originally derived from different cohorts. To do this we re-fitted the PREDICT Prostate and MSKCC models to the training set, before reapplying them to the validation set. Here we found similar performance characteristics for the PREDICT Prostate model and a better MSKCC model performance. Both models continued to perform comparably with the Survival Quilt model. Finally, given that biopsy core data was only available in less than half the cohort, we reassessed the PREDICT Prostate and Survival Quilts models without this variable and found similar comparative performance characteristics.

7.4.4 Decision curve analysis

We next assessed model performance using DCA in the context of considering the impact on decision making for treatment (e.g. surveillance versus radial therapy). The heterogeneous profile of the patient population renders a uniform treatment strategy (treat all or treat none) inferior to strategies informed by any one of the 3 models (Figure 7.3). Across the 3 models MSKCC provided the least net benefit while the Survival Quilts provided the greatest gain. The gain from the Survival Quilts model was particularly seen with threshold probabilities of risk between 0.1 and 0.3 with added net incremental benefits across each threshold compared to the PREDICT Prostate model. The difference was even greater when compared to the

MSKCC model.

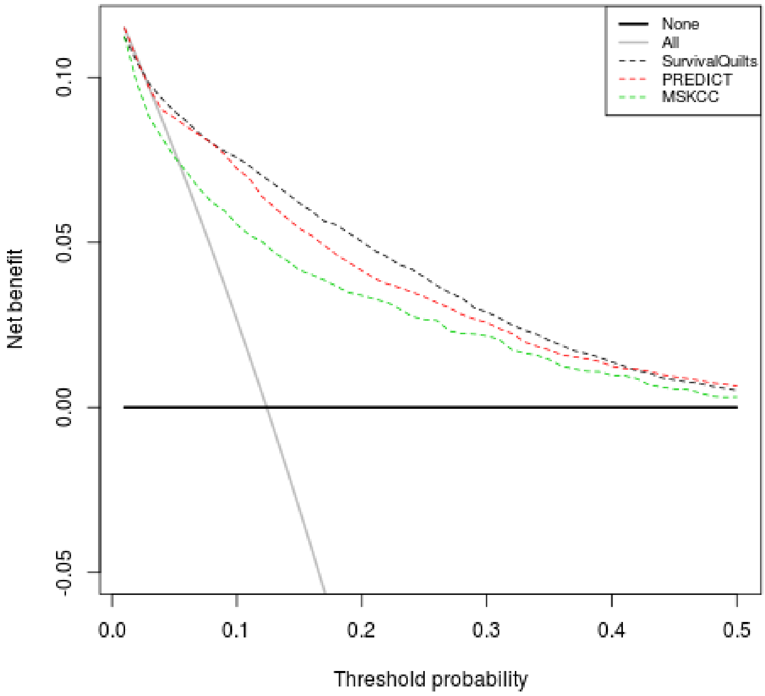


Figure 7.3: Decision curve analysis. The clinical net benefit for each prediction model is calculated across a range of risk threshold probabilities. Clinical net benefit is defined as the minimum probability of disease at which further intervention would be warranted. MSKCC=Memorial Sloan Kettering Cancer Center.

7.4.5 Discussion

In this paper we have used a very large dataset to develop and test a ML-trained prognostic model for predicting 10-year PCSM and assessed its performance against a range of tiered and multivariable prediction models. To our knowledge, our study is also the first to use the SEER cohort to compare numerous models for predicting PCSM. In comparative analysis we observed that multivariable models consistently outperform tiered systems consistent with previous head to head comparisons [193, 207]. Our study further introduces a potentially better approach by utilizing a novel ML algorithm that automatically combines optimal attributes from different modeling methods.

There are very few ML studies in prostate cancer prognostics, and these have included relatively small cohorts in terms of model development. The only study to assess PCSM trained several artificial neural network (ANN) models with 19 pretreatment variables in 7,267 Korean men [198]. Koo et al.'s [198] 'long short-term memory' ANN model produced a C-index of 0.815 for both 10-year PCSM and all-cause mortality discrimination. However, the model has never been externally validated. The largest study to date using ML in prostate cancer prognostics was performed by Lin et al., utilizing data from 8581 Taiwanese men. Using a support vector machine-trained model incorporating comorbidity data with standard clinico-pathological variables (but not PSA), an accuracy of 0.852 was achieved in predicting cancer-related post-treatment recurrence and mortality [199].

At present, datasets are often limited by which variables prostate cancer specialists have traditionally considered important and therefore collected (for example, clinico-pathological and comorbidity variables). Our study supports the intuitive notion that model performance improves with greater granularity, and ML-trained models should have a particular advantage when considering incorporating new variables [193, 198–201, 208]. As an example, Donovan et al. combined standard variables with 5 molecular biomarkers and automated histopathological image analysis to derive a prediction tool for BCR after treatment [200]. Their Precise Post-Op model had a C-index of 0.77 for recurrence-free survival. Zhang et al. combined somatic mutation signatures in a 43 gene panel with the NICE risk criteria and improved the area under the curve for prediction of post-surgical BCR from 0.62 to 0.75 [201].

The data-driven and variable-agnostic approach inherent to ML also allows for an 'information-gain' from hitherto unsuspected contributing factors. For example, the ML AutoPrognosis model for predicting cardiovascular risk was trained on 473 variables and identified walking pace as the third most important variable for death after systolic blood pressure and body mass index [197]. The Survival Quilts method used in our paper similarly permits a ‘modeling-gain’ whereby the most robust model amongst several can be objectively chosen without prior presumptions regarding model characteristics and variable interactions [202]. Notably, by just using a few standard clinico-pathological factors, the method was able to achieve high C-indices and excellent calibration. In DCA, we also found an incremental gain in net benefit when the Survival Quilts was applied in comparison to using the other 2 top performing models. There is no accepted consensus on what is a clinically useful range for net benefit in treatment prognostic models [209]. In clinical practice if the uncertainty is 10% or less, then a decision model is not really needed. We therefore reasoned that threshold probabilities higher than this would gain from using a decision model. In this analysis we particularly found gain when the threshold probabilities of risk were between 10% and 30%. We accept that there may be other interpretations of a clinically important range but believe that this is a pragmatic approach to define a range where prognostic model improvements have a net clinical benefit. It is likely that training the Survival Quilts framework on multiple large datasets incorporating more factors will produce an even more superior model than we have so far achieved. Owing to the relatively autonomous nature of ML, such models could be quickly and automatically updated whenever new data become available. A further opportunity is the ability to continuously add new data as the patient’s treatment journey progresses and visualize how this impacts prognosis; something not currently possible with static prognostic models.

This study does have some important limitations. Our ML-based model was trained in a large, contemporary, ethnically-heterogeneous population using real world data from a high-quality database [204]. Indeed, it is by far the largest study applying ML to prostate cancer prognostics so far described in the literature. However, the cohort distribution is heavily slanted to earlier stage disease as it represents a predominantly PSA-screened population. Consequently, it also had relatively few death events and follow-up was limited. It was encouraging, however, to observe that the performance of the other prognostic models we tested was very consistent with the results seen in other large population studies where there is a more balanced case-mix [193, 207]. Although our final cohort included over 170,000 men, we did not have data for a larger starting population so cannot account for any bias this missingness might have introduced when the final cohort was derived. We also did not explore the geographical distribution of this United States cohort nor any social differences, so cannot comment on any impact this may have had on our results. Similarly, biopsy core data had to be imputed for a significant portion of the included cohort. We acknowledge, however, that using imputation for such a large amount of information may have introduced a bias in our study. As the SEER database does not collect comorbidity data, we also could not model the impact of comorbidity on outcomes, nor could we consider the effects of treatment, both of which are key parameters of other tested models (for example, PREDICT Prostate). Comparison between different models also may introduce inherent bias because of input variable heterogeneity. Changes in performance may therefore reflect the input variable heterogeneity. Furthermore, SEER also does not collect data on prostate MRI though it is unclear if MRI findings will improve current prognostic capabilities [210]. We additionally did not have any molecular markers to assess, although their addition to standard models does show some promise [211]. It remains to be seen how useful these tests are and how their addition to models like Survival Quilts, or indeed the PREDICT Prostate or MKSCC nomogram, improve performance given their significant additional cost burden. This study was specially focused on non-metastatic cancer but in future work we would be keen to take the methods here and apply it to the metastatic setting where there is a paucity of robust and validated models.

7.5 Conclusion

A novel ML-trained model is capable of predicting PCSM at 10 years with comparable performance to the best existing models. ML may confer numerous potential future advantages, especially its potential to readily incorporate new data, self-training and evolving variables. Consequently, ML represents a unique future framework for producing more granular, individualized and iterative prognostic models. This study also demonstrates, in a PSA-screened population, the critical need to move away from tier-based risk grouping and increasingly utilize multivariable and more personalized prognostic models to guide patient management.

CHAPTER 8

Clinical Impact: Outcome-Oriented Deep Temporal Phenotyping of Breast Cancer Progression

8.1 Contribution

Chronic diseases evolve slowly throughout a patient's lifetime creating heterogeneous progression patterns that make clinical outcomes remarkably varied across individual patients. A tool capable of identifying temporal phenotypes based on the patients' different progression patterns and clinical outcomes would allow clinicians to better forecast disease progression by recognizing a group of *similar* past patients, and to better design treatment guidelines that are tailored to specific phenotypes.

Added value of this study. To build such a tool, we adopt and improve a deep learning-based temporal phenotyping method [212] to discover outcome-oriented temporal phenotypes of disease progression considering *what type* of clinical outcomes will occur and *when* based on the longitudinal observations. More specifically, we model clinical outcomes throughout a patient's longitudinal observations via time-to-event (TTE) processes whose conditional intensity functions are estimated as non-linear functions using a recurrent neural network.

Findings. We perform a set of experiments on real-world data which was collected by the UK National Cancer Registration and Analysis Service (NCRAS). The data contains a cohort of 11,779 female patients (between age 15 to 90) diagnosed with stage III breast cancer, whose observations are collected over their follow-up periods. We focused on stage III breast cancer patients due to their heterogeneity in disease progression and the development of adverse clinical events (e.g., recurrence) and earlier death [213, 214]. For these patients, identifying temporal phenotypes and understanding the underlying progression can provide actionable intelligence. Throughout the evaluation, we show that the proposed method identifies temporal phenotypes that are strongly associated with future clinical outcomes and achieves significant gain in the homogeneity and heterogeneity measures over existing methods. In addition, we analyze driving factors that lead to transitions between the identified temporal phenotypes and thereby enable us to better understand the underlying disease progression within the longitudinal context.

8.2 Introduction

The progression of chronic diseases (such as cancer and diabetes) manifests through a broad spectrum of longitudinally-linked clinical features and outcomes, which we refer to as *clinical pathways*. This leads to heterogeneous progression patterns that may vary greatly across individual patients. Therefore, temporal phenotyping has become a crucial tool in identifying patient subgroups to address such heterogeneity. By transforming the raw information in clinical pathways into clinically relevant and interpretable information [10], temporal phenotypes allow clinicians to better forecast disease progression with reduced uncertainty and design treatment guidelines that are tailored to patient subgroups [11].

The fundamental idea of temporal phenotyping to understand the underlying disease progression is to group patients based on the *similarity* in their clinical pathways. However, there are many different notions of similarity which make the identified phenotypes substantially different [11, 161, 170, 215]. Under the traditional notion of clustering (e.g., K-means [216]), recent approaches focused on either adjusting similarity measures for longitudinal observations [161, 215] or finding low-dimensional representations [11, 170] of longitudinal observations to group clinical pathways. However, these approaches identify temporal phenotypes in a purely unsupervised fashion, thereby discarding already available information about patients' clinical outcomes (such as recurrence of cancer and death) and the timing when these outcomes occurred which both play a significant role in understanding the underlying disease progression and in reasoning about the future clinical outcomes. Therefore, discarding this valuable information may lead to problems in effectiveness and timeliness of clinical interventions as past researches (see e.g., [163, 164, 212]) have shown that clinical outcomes and the timing of such outcomes can significantly vary for patients even in the same temporal phenotype.

In this work, we introduce a different notion: outcome-oriented temporal phenotyping. This characterizes temporal phenotypes of the underlying disease progression in relevance to the *type* and the *timing* of clinical outcomes which will occur based on the clinical pathways. More specifically, patient pathways are grouped together into the same phenotype such that these pathways share similar future outcomes and timing. Additionally, the phenotype assigned to a patient needs to be flexibly updated as the disease evolves which can be manifested through new observations of clinical features and/or clinical outcomes accrued over time. By doing so, we ensure that clinicians can leverage temporal phenotyping as an *actionable tool* to recognize similar past patients (for whom a pathway with an endpoint was already collected) for reasoning about future outcomes as well as life-changing disease manifestations which may cause a patient to transit between phenotypes. A pictorial depiction of our notion of temporal phenotyping in comparison to the traditional notion is illustrated in Figure 8.1.

To this goal, we utilize and extend a deep learning-based temporal clustering method [212] to discover outcome-oriented deep temporal phenotypes, which we call ODTP, of clinical pathways based on the framework of neural discrete representation learning [173, 174]. ODTP models clinical outcomes in the variable-length and irregularly-spaced clinical pathways as time-to-event (TTE) processes whose conditional intensity functions are estimated as non-linear functions of the latent representations of a recurrent neural network (RNN).

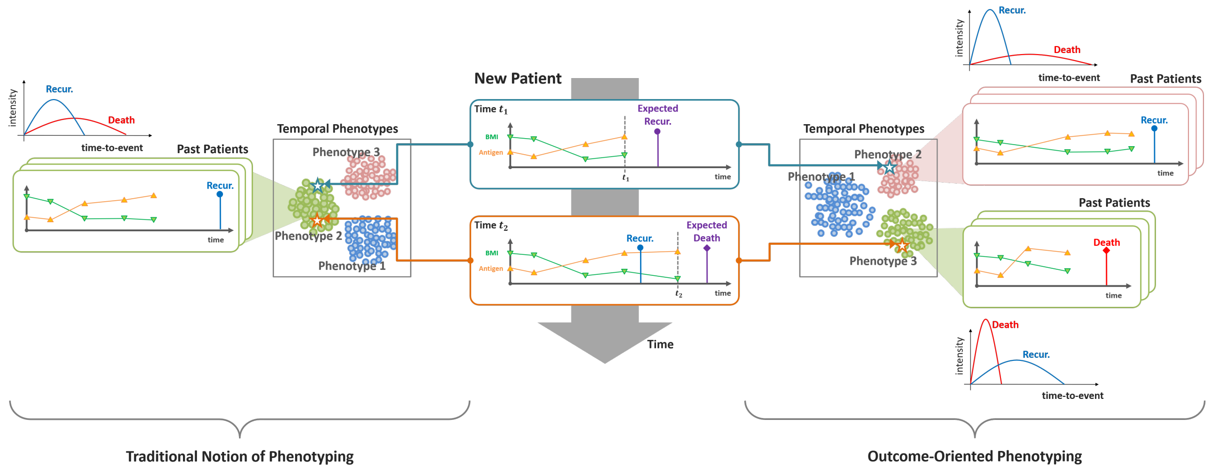


Figure 8.1: A conceptual illustration of our (real-time) temporal phenotyping procedure. In this example, we focus on patients who are diagnosed with breast cancer where the clinical outcomes of our interest are the recurrence of cancer and cancer-related death. Note that, during run-time, the new patient is assigned to one of three phenotypes as new observations are collected over time. In our notion (i.e. outcome-oriented) of phenotyping, the new patient is assigned to Phenotype 2 at time t_1, which consists of past patients with a high risk of recurrence. Then, this new patient is assigned to Phenotype 3 at time t_2, which consists of past patients who died from cancer-related death in the near future, due to the increased risk of cancer-related death. However, in the traditional notion of phenotyping, the new patient remains at the same phenotype at both t_1 and t_2 since the longitudinal observations remained very similar (in entire sequence perspective) to the past patients of this phenotype.

Temporal phenotypes are identified by our novel loss function that is designed to learn discrete latent representations that best characterize the TTE processes for the clinical pathways. The key insight here is that learning the centroids of each phenotype and the assignments of the pathways to these phenotypes can be achieved by learning a finite number of discrete representations, called embeddings, and the mapping from input pathways to these embeddings, respectively.

We demonstrate the power of ODTP by applying it to a real-world heterogeneous cohort of 11,779 stage III breast cancer patients from the UK National Cancer Registration and Analysis Service. The experiments show that ODTP identifies temporal phenotypes that are strongly associated with the future clinical outcomes and achieves significant gain on the homogeneity and heterogeneity measures over existing methods. Furthermore, we are able to identify the key driving factors that lead to transitions between phenotypes which can be translated into actionable information to support better clinical decision-making.

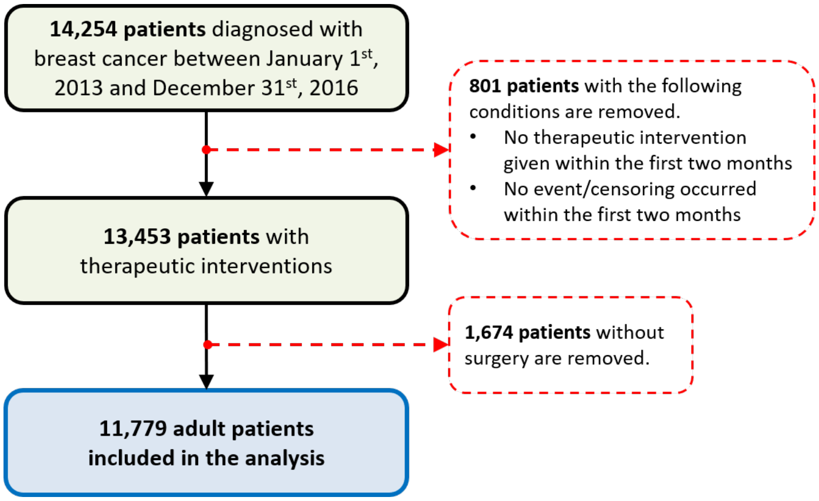
8.3 Methods

8.3.1 Data source and study population

The UK National Cancer Registration and Analysis Service (NCRAS)[1] comprises clinical pathways of patient care in the form of clinical observations, therapeutic interventions, and clinical incidents that occur to cancer patients. Overall, out of 14,254 female patients between 15 and 90 years who were diagnosed with stage III breast cancer between 2013 and 2016, we focused our experiments on 11,779 patients who had properly scheduled cancer treatments. More specifically, all breast cancer patients should start treatment – either drug treatment first then surgery (e.g., neoadjuvant therapy) or breast cancer surgery first (e.g., mastectomy or lumpectomy) – within the first 1 to 2 months after the diagnosis. In the case of neoadjuvant therapy which is often given for a total of 3 to 6 months, breast cancer surgery should take place within the next 1 to 2 months. Hence, we first excluded patients who were not given any therapeutic interventions within the first two months after diagnosis and without having any terminal events (i.e., death or right-censoring). Then, patients without any breast cancer surgery record within the first eight months after the diagnosis were excluded. Figure 8.2 depicts a flow chart of the data assembly process involved in our analysis.

Among 11,779 patients, 1,922 patients died from cancer, other cancer, and other causes

[1] http://www.ncin.org.uk/home



14,254 patients diagnosed with breast cancer between January 1st, 2013 and December 31st, 2016
801 patients with the following conditions are removed.
• No therapeutic intervention given within the first two months
• No event/censoring occurred within the first two months
13,453 patients with therapeutic interventions
1,674 patients without surgery are removed.
11,779 adult patients included in the analysis

Figure 8.2: Patient data selection process.

(including cardiovascular disease) and 9,857 patients were right-censored. The follow-up period (i.e., time-to-event/censoring) ranges from 1 to 2,049 days with an average of 795.4 days. Overall, we consider 4 clinical outcomes of our interest: 2 recurrent events and 2 terminal events. The terminal events are i) 1,409 (11.96%) *deaths due to cancer*, and ii) 513 (4.36%) *deaths due to other causes*. It is important for patients who are at risk of death from cancer to be provided with a joint prognosis of risk of other causes in order to properly manage therapeutic interventions. For instance, chemotherapy, which maintains a prominent role in treating many forms of cancer, increases the risk of cardiovascular side effects [16, 217]. The recurrent events are other tumor diagnosis and recurrence (including local, regional, and distant recurrence). These recurrent events are crucial since the diagnosis of other tumors or recurrence of cancer leads to a significant rise in mortality due to cancer. Among 11,779, patients, 464 patients (3.94%) and 1,936 patients (16.44%) experienced *other tumor diagnosis* and *recurrence* during their pathways, respectively. Throughout the experiments, all patients are aligned based on the diagnosis of breast cancer.

The data comprises of 14 static features that are observed at the time of diagnosis and 26 time-varying features that are collected over follow-up periods. The static features include demographics, tumor assessments – such as grades, morphology, and TNM (tumor, nodes, metastasis) stages –, hormone receptor status, laterality, and comorbidity information. The time-varying features include indicators for in-patient hospital visits, tumor examinations (i.e., radiology and pathology), therapeutic interventions (i.e, chemotherapy, radiotherapy, and hormone therapy), surgery, diagnosis of other tumors, and relapse/recurrence of the breast cancer. For each patient, the time interval between two adjacent longitudinal measurements ranges from 1 to 1,413 days with a mean of 19.95 days. Here, we discretized the time with a resolution of days since the date information in the data was mostly available in that format. The number of follow-up observations was from 1 to 373 with a mean of 40.87 observations per patient.

8.4 Model Development

8.4.1 Clinical Pathway Data

A *clinical pathway* is a longitudinally-linked series of clinical features and outcomes that are systematically collected in disease registries data as well as in data from EHRs to describe the disease progression of a patient. Each pathway consists of an *observation sequence* containing longitudinal observations of static and time-varying covariates, and an *outcome sequence* containing a series of one or more clinical outcome event(s) that occurred to a patient during his/her follow-up. Outcome events include recurrent events (such as relapse of cancer) that can repeatedly occur throughout a patient's pathway and terminal events (such as death from cancer or right-censoring) that terminate further observations. We assume that every patient experiences exactly one terminal event at the end of his/her pathway. Figure 8.3 depicts clinical pathways of representative patients dying due to cancer or having the endpoint censored.

Formally, for each patient n, define $\mathcal{X}^n_{J^n} = (\mathbf{x}^n_j, t^n_j)^{J^n}_{j=1}$ to be an observation sequence which comprises J^n longitudinal observations where $\mathbf{x}^n_j \in \mathbb{R}^d$ denotes the observed covariates and $t^n_j \in \mathbb{R}_{\geq 0}$ is the timing at the j-th observation, respectively. Also, define $\mathcal{Y}^n_{L^n} = (m^n_\ell, \tau^n_\ell)^{L^n}_{\ell=1}$ to

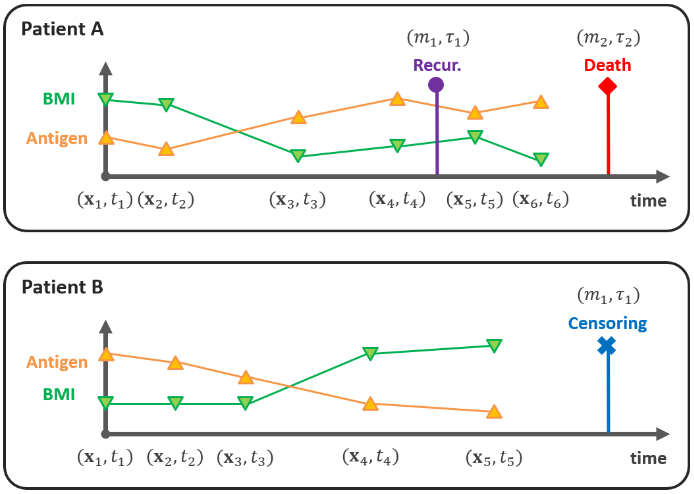


Figure 8.3: An illustration of clinical pathways (observation sequence: antigen and BMI, outcome sequence: recurrence and death) of representative patients diagnosed with breast cancer with different terminal events.

be an outcome sequence which consists of $L^n \geq 1$ outcome events where $m_\ell^n \in \{1, \cdots, M, \varnothing\}$ is the event type among M possible events and $\tau_\ell^n \in \mathbb{R}_{\geq 0}$ denotes the timing of the ℓ-th outcome event, respectively, with $\varnothing$ indicating right-censoring. Note that irregular time intervals between observations and/or outcome events can be generally described by the actual timestamps t_j^n and τ_ℓ^n. Throughout, we assume that the observation times, event times, and censoring times are aligned based on a synchronization event such as the entry to a clinical study. Hereafter, we omit the dependency on n for ease of notation when it is clear in the context.

For each clinical pathway, let $\tilde{\mathbf{x}}_i$ be input features at time step i that aggregate both observed covariates and outcome events available at the corresponding timestamp $\tilde{t}_i \in \mathcal{T}$ as the following:

$$\tilde{\mathbf{x}}_i = \begin{cases} (\mathbf{x}_{j^*}, m_{\ell^*}, \tilde{t}_i) & \text{if } \tilde{t}_i \in \{t_1, \cdots, t_J\}, \tilde{t}_i \in \{\tau_1, \cdots, \tau_L\} \\ (\mathbf{x}_{j^*}, *, \tilde{t}_i) & \text{if } \tilde{t}_i \in \{t_1, \cdots, t_J\}, \tilde{t}_i \notin \{\tau_1, \cdots, \tau_L\} \\ (*, m_{\ell^*}, \tilde{t}_i) & \text{if } \tilde{t}_i \notin \{t_1, \cdots, t_J\}, \tilde{t}_i \in \{\tau_1, \cdots, \tau_L\} \end{cases}$$

where $\mathcal{T} \triangleq \{\tilde{t}_i, \cdots, \tilde{t}_I\} = \{t_1, \cdots, t_J\} \cup \{\tau_1, \cdots, \tau_L\}$ indicates an ordered set of I available observation and/or event times. Here, $*$ indicates that the observation or the outcome event is not available at the corresponding timestamp. $j^* = \arg_{j\in\{1,\cdots,J\}}(t_j = \tilde{t}_i)$ and $\ell^* = \arg_{\ell\in\{1,\cdots,L\}}(\tau_\ell = \tilde{t}_i)$ indicate the covariates and the event that are observed at timestamp $\tilde{t}_i$, respectively. For example, $\mathcal{X}_2 = (\mathbf{x}_1, 1), (\mathbf{x}_2, 3)$ and $\mathcal{Y}_2 = (m_1, 1), (m_2, 4)$ can be expressed as $\tilde{\mathbf{x}}_1 = (\mathbf{x}_1, m_1, 1), \tilde{\mathbf{x}}_2 = (\mathbf{x}_2, *, 3), \tilde{\mathbf{x}}_3 = (*, m_2, 4)$. Then, we can finally denote the *history* of a patient's clinical pathway up to time $\tilde{t}_i$ as $\tilde{\mathbf{x}}_{1:i} \triangleq (\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_i)$.

8.4.2 Modeling Clinical Pathways via Time-to-Event Processes

To understand the underlying progression of the target disease, we model the outcome sequence throughout a clinical pathway via time-to-event (TTE) processes. That is, given the history of observations and outcome events at each time step i, we focus on the type and the timing of the next outcome event in chronological order.[2]

Define random variables (RVs) $T_1, \cdots, T_M$ and $T_\varnothing$ where $T_m \in \mathbb{R}_{\geq 0}$ denotes the time to the next outcome event of cause $m \in \{1, \cdots, M\}$ and $T_\varnothing \in \mathbb{R}_{\geq 0}$ denotes the time to censoring event. We assume that T_m for $m \in \{1, \cdots, M, \varnothing\}$ is drawn from a conditional density function that depends on the history of a patient's pathway. Given the pathway of each patient up to a certain time point, we only observe the occurrence time for the earliest next outcome event (including right-censoring), i.e., $T = \min(T_1, \cdots, T_M, T_\varnothing)$ and $E = \arg\min_{m\in\{1,\cdots,M,\varnothing\}} T_m$.

The *cause-specific conditional hazard function* $h_m(s|\tilde{\mathbf{x}}_{1:i})$ [218] represents the instantaneous risk of the next outcome event of type m occurring given the history $\tilde{\mathbf{x}}_{1:i}$, and is formally defined as:

$$h_m(s|\tilde{\mathbf{x}}_{1:i}) = \lim_{ds\to 0} \frac{P(s \leq T \leq s + ds, E = m|\tilde{\mathbf{x}}_{1:i}, T \geq s)}{ds} \tag{8.1}$$

[2] The main distinction from modeling the outcome sequence via multi-variate temporal point processes is that we do not consider longitudinal observations as a part of the event sequence. This is because observations, in general, are not event-driven but are collected based on clinical guidelines or regular physical examinations.

where s denotes the time elapsed since the latest observation time $\tilde{t}_i$. Then, we can express the probability of the next outcome event (m, τ) given the history $\tilde{\mathbf{x}}_{1:i}$ as the following: $P(T = \tau - \tilde{t}_i, E = m|\tilde{\mathbf{x}}_{1:i}) = h_m(\tau - \tilde{t}_i|\tilde{\mathbf{x}}_{1:i})S(\tau - \tilde{t}_i|\tilde{\mathbf{x}}_{1:i})$ if event m occurred (i.e., $m \neq \varnothing$) and $P(T > \tau - \tilde{t}_i|\tilde{\mathbf{x}}_{1:i}) = S(\tau - \tilde{t}_i|\tilde{\mathbf{x}}_{1:i})$ if right-censored (i.e., $m = \varnothing$). Here, $S(s|\cdot) = \exp\big(-\int_0^s h(u|\cdot)du\big)$ is the survival function which captures the probability of a patient's event-free survival up to s and $h(s|\cdot) = \sum_{m=1}^{M} h_m(s|\cdot)$ denotes the overall hazard function.

Parametric Assumption. We assume that the cause-specific conditional hazard functions follow the *Weibull* distribution [219], which is one of the most common parametric forms to analyze TTE processes due to its convenient closed-form expressions. That is, given the history $\tilde{\mathbf{x}}_{1:i}$, (8.1) can be simplified as:

$$h_m(s|\tilde{\mathbf{x}}_{1:i}) = p\lambda_m(\tilde{\mathbf{x}}_{1:i})\big(\lambda_m(\tilde{\mathbf{x}}_{1:i})s\big)^{p-1} \tag{8.2}$$

where $\lambda_m(\tilde{\mathbf{x}}_{1:i}) > 0$ is the *conditional intensity function* given $\tilde{\mathbf{x}}_{1:i}$ and $p > 0$ is the shape parameter.[3]

Overall, the log-likelihood of the outcome sequence with L outcome events throughout a patient's entire pathway can be derived as $\sum_{\ell=1}^{L}\sum_{i\in\mathcal{I}_\ell}\log P(E = m_\ell, T = \tau_\ell - \tilde{t}_i|\tilde{\mathbf{x}}_{1:i})$ where $\mathcal{I}_\ell = \{i : \tau_{\ell-1} \leq \tilde{t}_i < \tau_\ell\}$ (with $\tau_0 = 0$) denotes a set of time steps between the timestamp at which the previous outcome event occurred (i.e., $\tau_{\ell-1}$) and that at which the next outcome event occurs (i.e., τ_ℓ). Here, the conditional probability of an outcome event and the timing (m, τ) given the history $\tilde{\mathbf{x}}_{1:i}$ in the log-likelihood of the outcome sequence can be derived as

[3] The Weibull distribution is a generalization of the exponential distributions. For instance, when $p = 1$, it reduces to the exponential distribution and has constant hazard function over time, while the hazard function is increasing and decreasing over time when $p > 1$ and $p < 1$, respectively.

follows:

$$\begin{aligned}\log P(E=m, T=\tau-\tilde{t}_i|\tilde{\mathbf{x}}_{1:i}) &= \mathbb{1}_{\{m\neq\varnothing\}}\cdot\log\big(h_m(\tau-\tilde{t}_i|\tilde{\mathbf{x}}_{1:i})S(\tau-\tilde{t}_i|\tilde{\mathbf{x}}_{1:i})\big) \\ &\qquad + \mathbb{1}_{\{m=\varnothing\}}\cdot\log S(\tau-\tilde{t}_i|\tilde{\mathbf{x}}_{1:i}) \\ &= \mathbb{1}_{\{m\neq\varnothing\}}\cdot\log h_m(\tau-\tilde{t}_i|\tilde{\mathbf{x}}_{1:i}) + \log S(\tau-\tilde{t}_i|\tilde{\mathbf{x}}_{1:i}) \\ &= \mathbb{1}_{\{m\neq\varnothing\}}\cdot\log\big(p\lambda_m(\tilde{\mathbf{x}}_{1:i})^p(\tau-\tilde{t}_i)^{p-1}\big) - \lambda(\tilde{\mathbf{x}}_{1:i})^p(\tau-\tilde{t}_i)^p\end{aligned} \tag{8.3}$$

where $\lambda(\cdot)=\sum_{m=1}^{M}\lambda_m(\cdot)$. Hence, the problem of accurately estimating the log-likelihood of an outcome sequence throughout a pathway boils down to accurately estimating the conditional intensity functions $\lambda_m(\cdot)$ for $m\in\{1,\cdots,M\}$ as a function of the pathway.

8.4.3 Modeling TTE Processes via NNs

We use an RNN to model the underlying dynamics of the outcome sequences throughout clinical pathways. The key idea here is to determine the conditional intensity functions in (8.2) from the latent representations (i.e., the hidden states) of the RNN. This allows learning of complex dependencies of the cause-specific conditional hazard functions on the history of observations and outcome events (i.e., previous event types and the timings). The network, which we refer to as the *TTE network*, comprises an *encoder* that captures the underlying dynamics given a pathway and an *estimator* that estimates the conditional intensity functions based on the encoder output. The two biggest distinctions from the previous work in [159] come from modeling a sequence of both recurrent and terminal events in a single framework, and, more importantly, further utilizing the latent representations for temporal phenotyping.

The encoder, $f^\theta:\prod_{j=1}^{i}(\mathbb{R}^d\times\{1,\cdots,M\}\times\mathbb{R}_{>0})\rightarrow\mathcal{Z}$, is an RNN (parameterized by θ) that takes a sequence of tuples $\tilde{\mathbf{x}}_{1:i}$ – i.e., the pathway that contains available observations, outcome events, and the timing up to the i-th time step – as inputs and maps the input sequence to latent representations $\mathbf{z}_i\triangleq f^\theta(\tilde{\mathbf{x}}_{1:i})\in\mathcal{Z}$ at each time step i.

The estimator, $g^\phi:\mathcal{Z}\rightarrow\mathbb{R}^M_{>0}$, is a fully-connected network (parameterized by ϕ) that estimates the cause-specific conditional intensity functions in (8.2) given the latent represen-

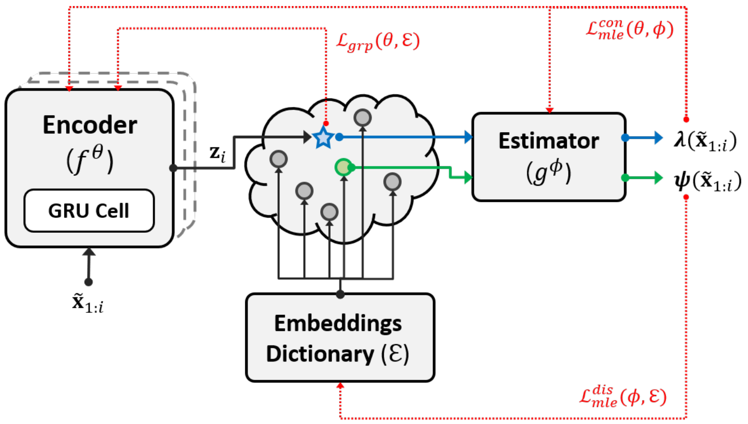

$\mathcal{L}_{grp}(\theta,\mathcal{E})$
$\mathcal{L}^{con}_{mle}(\theta,\phi)$
Encoder
(f^{θ})
GRU Cell
$\mathbf{z}_i$
Estimator
(g^{ϕ})
$\boldsymbol{\lambda}(\tilde{\mathbf{x}}_{1:i})$
$\boldsymbol{\psi}(\tilde{\mathbf{x}}_{1:i})$
$\tilde{\mathbf{x}}_{1:i}$
Embeddings
Dictionary $(\mathcal{E})$
$\mathcal{L}^{dis}_{mle}(\phi,\mathcal{E})$

Figure 8.4: An illustration of the network architecture. The network parameters that are updated by each loss function are highlighted in the red dotted lines.

tation of the input sequence at each time step i, that is, $\boldsymbol{\lambda}(\tilde{\mathbf{x}}_{1:i}) = [\lambda_1(\tilde{\mathbf{x}}_{1:i}), \cdots, \lambda_M(\tilde{\mathbf{x}}_{1:i})] \triangleq g^{\phi}(\mathbf{z}_i) = [g^{\phi}_1(\mathbf{z}_i), \cdots, g^{\phi}_M(\mathbf{z}_i)]$ where $g^{\phi}_m(\cdot)$ denotes the m-th element of $g^{\phi}(\cdot)$.

8.4.4 Outcome-Oriented Deep Temporal Phenotyping

Our goal is to identify temporal phenotypes that characterize the underlying disease progression in terms of what type of outcome events will occur next and when, on the basis of patients' pathways. To ensure such a prognostic value, we want the identified phenotypes to have the following properties: (i) patients' pathways in the same phenotype need to share a similar expected future in terms of the type and the timing of the next outcome event. (ii) the phenotype assigned to a patient needs to be flexibly updated as new observations and/or outcome events are accrued over time. To this end, we formalize temporal phenotyping as learning discrete representations that best characterize the TTE processes of next outcome events throughout the pathways. The key insight here is that learning embeddings (i.e., a finite number of latent representations available for discrete representation learning) and the mappings from pathways to these embeddings can be viewed as learning the centroids of each phenotype (i.e., the representative representations of each phenotype) and the assignments of the pathways to these phenotypes, respectively.

In this section, we propose a deep learning method, which we call ODTP, that identifies outcome-driven temporal phenotypes of clinical pathways under the framework of discrete representation learning. The proposed method comprises the following components:

- an *encoder*, f^θ, and an *estimator*, g^ϕ, to flexibly model outcome sequences in the pathways as introduced in the previous section; and
- an *embedding dictionary* $\mathcal{E} = \{\mathbf{e}(1), \cdots, \mathbf{e}(K)\}$ that is a set of K embedding vectors and *mappings* to those embedding vectors $\{s_i^n\}$ that together describe the phenotype centroids and assignment to those phenotypes.

A schematic illustration of ODTP is depicted in Figure 8.4.

Let $s_i \in \{1, \cdots, K\}$ be the *phenotype assignment* at time step i and $\mathcal{E} = \{\mathbf{e}(1), \cdots, \mathbf{e}(K)\}$ where $\mathbf{e}(k) \in \mathcal{Z}$ be the *embedding dictionary.* Then, we define $\bar{\mathbf{z}}_i \triangleq \mathbf{e}(s_i) \in \mathcal{Z}$ to be the *embedding*, a discrete representation of clinical pathways in the latent space. At each time step i, the discrete representation can be obtained via the following: First, we find an *encoding* $\mathbf{z}_i = f^\theta(\tilde{\mathbf{x}}_{1:i})$ (i.e., a continuous representation in the latent space) of an input pathway $\tilde{\mathbf{x}}_{1:i}$ as an output of the encoder. Then, the encoding is mapped to the closest *embedding* based on the phenotype assignments s_i and the embedding dictionary $\mathcal{E}$; formally, $\bar{\mathbf{z}}_i = \mathbf{e}(s_i)$ where $s_i^n = \arg\min_k \|\mathbf{e}(k) - \mathbf{z}_i^n\|_2^2$.

Since the embedding $\bar{\mathbf{z}}_i$ corresponds to the centroid of the phenotype to which $\tilde{\mathbf{x}}_{1:i}$ belongs, we can, finally, estimate the discretized conditional intensity function in (8.2) for the assigned phenotype as an output of the estimator given the embedding, i.e., $\boldsymbol{\psi}(\tilde{\mathbf{x}}_{1:i}) = [\psi_1(\tilde{\mathbf{x}}_{1:i}), \cdots, \psi_M(\tilde{\mathbf{x}}_{1:i})] \triangleq g^\phi(\bar{\mathbf{z}}_i)$.

We optimize the network components in an iterative fashion: i) updating network parameters of the encoder and the estimator (θ, ϕ) and ii) updating phenotype assignments s_i and the embedding dictionary $\mathcal{E}$.

Table 8.1: Comparison of different *similarity* notions

Methods	**Similarity Notions**
TTE-KM	similarity in latent representations tailored for predicting the type and the timing of the next clinical outcomes
S2S-DCN	similarity in latent representations of observation sequences
SurvTree	similarity in log-rank statistics of the timing of the next clinical outcomes
JLCM	similarity in model specifications for the longitudinal and the TTE processes
ODTP (ours)	similarity in latent representations and log-likelihoods of the outcome sequences

8.4.5 Head-head comparison

We compare ODTP with four well-known clustering methods for temporal phenotyping with different notions of *similarity*; see Table 8.1 for the summary. Since the benchmarks are not directly applicable to clinical pathway data, we adapted each method as described below:

- **TTE-KM**: the K-means clustering ch7:alg:orithm [216] requires quantification of the pairwise similarity between clinical pathways that can contain a different number of observations at irregular time intervals [220]. To address this issue, we first trained the TTE network with the log-likelihood loss to provide fixed-length and low-dimensional representations of clinical pathways. Then, we applied the K-means on the trained latent representations $\{\{\mathbf{z}^n_{1:i}\}_{i=1}^{I^n}\}_{n=1}^N$ based on the Euclidean distance.

- **S2S-DCN**: Following the recent success of Deep Clustering Network (DCN) [167], a state-of-the-art method that utilizes a deep neural network to cluster complex static data, we replaced the fully-connected networks of the encoder-decoder structure with a sequence-to-sequence network as an extension of DCN to incorporate clinical pathways.

In particular, we implemented the encoder and the decoder using GRU with the same number of layers and nodes with those of the TTE network, respectively.[4]

- **SurvTree**: Survival tree [221] is a decision tree designed for TTE analysis that utilizes the log-rank statistic as the splitting rule to measure the similarity between two children nodes (i.e., leaves). Thus, each leaf in the tree naturally becomes a predictive cluster [165] that contains instances with similar TTE outcomes. To train SurvTree, we model the next clinical outcomes and the timing as TTE outcomes as defined in Section 8.4.2. The difference is we only use the latest available observation (i.e., $\tilde{\mathbf{x}}_i$) as the input covariates and treat each event type separately (by treating other event types as censored) to handle the clinical pathways. The temporal phenotypes are identified as the leaves of the trained survival tree for death due to cancer.

- **JLCM**: Joint latent class model (JLCM) is a mixture model (thus, a model-based clustering method) that characterizes the heterogeneity in the population by integrating K homogeneous latent classes each of which shares the same model specifications for longitudinal and time-to-event processes. We implemented the JLCM in [169] utilizing R package `lcmm`[5]. To avoid issues with collinearity, we first performed Cox regression (as a validation step), and then, the JLCM is trained based on the retained features having coefficient p-values < 0.1.

8.5 Results

In this section, we provide a set of experiments using the real-world longitudinal time-to-event data described in the previous section. All the results are reported using 5 random 64/16/20

[4] This extension is a representative of the recently proposed deep learning approaches for clustering both in the static setting [167, 172] and in the longitudinal setting [170, 171]. All these methods are built upon the same concept of dimensionality reduction using an autoencoder followed by K-means clustering.

[5] https://cran.r-project.org/web/packages/lcmm

train/validation/test splits. For the network architecture, we construct the encoder (f^{θ}) utilizing two-layer GRU [140] with 50 nodes in each layer and the estimator (g^{ϕ}) utilizing single-layer fully-connected network with 100 nodes in each layer. The parameters θ, ϕ are initialized by Xavier initialization [175] and optimized via Adam optimizer [176] with learning rate $\eta_1 = \eta_2 = 0.001$ applying dropout with keep probability 0.6. We fixed the balancing coefficient $\alpha = 0.1$ throughout the experiments where the coefficient is selected among $\alpha = \{0.001, 0.01, 0.1, 1.0\}$ utilizing grid search that achieves the minimum validation loss.

8.5.1 Cohesion and Separation

In this section, we evaluate the temporal phenotyping methods with respect to the Silhouette index (SI) [179] which measures how similar a member is to its own phenotype (cohesion) compared to members of other phenotypes (separation). To do so, we adopt the Jensen-Shannon (JS) divergence between TTE distributions on the next outcomes given the clinical pathways as the similarity measure.[6] Due to our parametric assumption, the JS divergence between two Weibull distributions can be easily computed in a closed-form: $JS(\lambda||\lambda') = \frac{1}{2}\big(\frac{\lambda}{\lambda'}\big)^p + \frac{1}{2}\big(\frac{\lambda'}{\lambda}\big)^p - 1$ where λ and λ' are intensity functions of the two Weibull distributions, respectively; please refer to [223] for the full derivation. Let $\mathcal{C}_k = \big\{n \mid s^n_{I^n} = k\big\}$ be a set of clinical pathways that are assigned to phenotype k. Then, the SI for patient $n \in \mathcal{C}_k$ can be formally given as follows:

$$SI(n) = \frac{b(n) - a(n)}{\max\big(a(n), b(n)\big)} \tag{8.4}$$

$$a(n) = \frac{1}{|\mathcal{C}_k| - 1} \sum_{n,\ell \in \mathcal{C}_k, \ell \neq n} JS(\boldsymbol{\lambda}(\tilde{\mathbf{x}}^n_{1:I^n})||\boldsymbol{\lambda}(\tilde{\mathbf{x}}^{\ell}_{1:I^{\ell}})) \text{ and } b(n) = \min_{k' \neq k} \frac{1}{|\mathcal{C}_{k'}|} \sum_{\ell \in \mathcal{C}_{k'}} JS(\boldsymbol{\lambda}(\tilde{\mathbf{x}}^n_{1:I^n})||\boldsymbol{\lambda}(\tilde{\mathbf{x}}^{\ell}_{1:I^{\ell}}))$$

[6] The JS divergence is a proper *distance* measure as it satisfies the non-negativity, identity, and symmetry properties [222].

where $a(n)$ and $b(n)$ are the average intra-phenotype divergence and the average nearest-phenotype divergence, respectively. A high SI indicates better cohesion within a phenotype – such that the next clinical outcomes of a member in a phenotype are well matched to its assigned phenotype – and better separation across phenotypes – such that the next clinical outcomes of a member are poorly matched to its neighboring phenotypes.

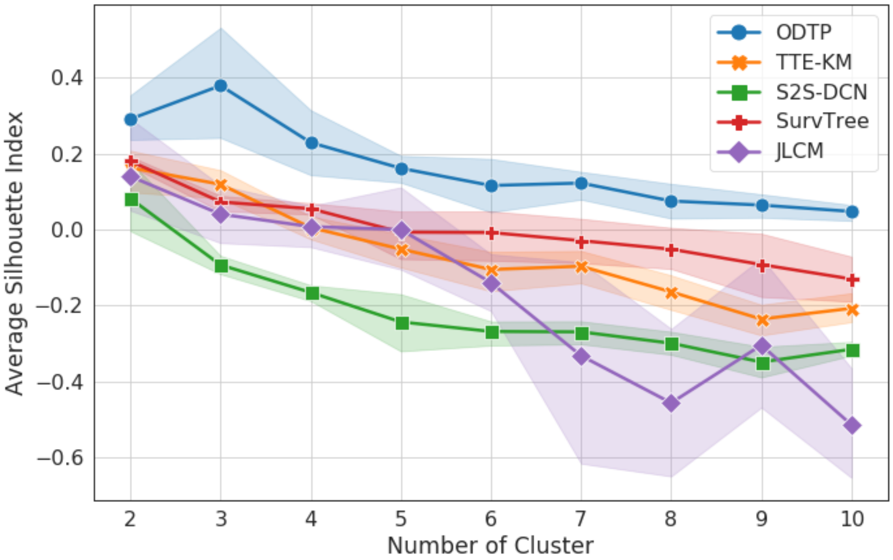


Figure 8.5: The Silhouette index performance (mean and 95% confidence interval) with various K. (Higher the better.)

In Figure 8.5, we reported the averaged SI by varying the number of phenotypes from 2 to 10 based on the clinical guidance of our medical collaborator.[7] ODTP significantly outperformed all the tested benchmarks in terms of the performance metric that assess both intra-phenotype homogeneity (i.e, cohesion) and inter-phenotype heterogeneity (i.e., separation). In particular, our method achieved significant gain over S2S-DCN and the JLCM whose phenotypes are not associated with predictions on the next clinical outcomes,

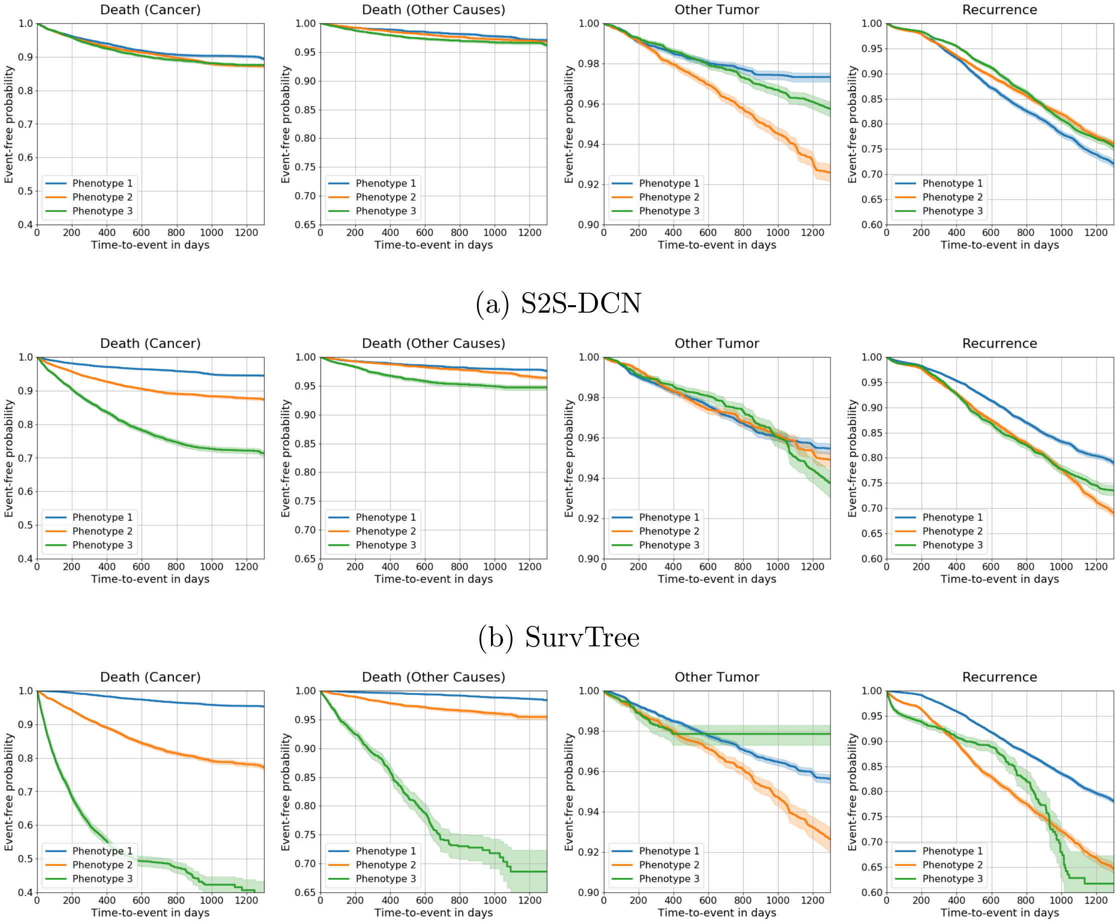
[7] For the JLCM, we instead reported the SI using the training set since the R package `lcmm` does not provide latent class assignments for hold-out samples (i.e., the testing set).

and, thereby, provide little prognostic value. This is because the latent representations in which S2S-DCN performs clustering are focused on reconstructing the observation sequence rather than predicting the type and the timing of the next clinical outcomes. Similarly, the phenotypes identified by JLCM share the same model specifications for the underlying longitudinal and TTE processes. Moreover, ODTP provided a significant improvement over TTM-KM and SurvTree that identify phenotypes by associating the latent representations with predictions on the next clinical outcomes or by splitting phenotypes based on the log-rank statistic of the next clinical outcomes. The poor performance of TTE-KM comes from two sources: i) the phenotype assignments and centroids of TTE-KM are obtained in an ad-hoc fashion and ii) the phenotype centroids do not properly describe the distribution of the next clinical outcomes. The gain of ODTP over SurvTree comes from the fact that the predictions of SurvTree on the next clinical outcomes are not very accurate since only a small number of leaves (equivalent to K) are used for building the tree and the history of clinical pathways are not fully utilized.

Selecting the Number of Phenotypes. The main challenge of phenotyping is the fact that, in general, we do not know a priori what the number of phenotypes should be. To address this, the averaged SI has been commonly utilized as a useful criterion for selecting the number of phenotypes in a data-driven fashion [224]. That is, the maximum value of the averaged SI over different K implies the best number of phenotypes. Therefore, we choose $K = 3$ in our analysis as it provides the maximum average SI (see Figure 8.5).

8.5.2 Analysis on the Identified Temporal Phenotypes

In this section, we further analyze the three identified temporal phenotypes. For ease of illustration, we numbered them in order of risk of death due to cancer (the values in parentheses imply the percentage of clinical pathways – at any time step i.e., $\{\{\tilde{\mathbf{x}}^n_{1:i}\}_{i=1}^{I^n}\}_{n=1}^N$ – assigned to each phenotype):



(a) S2S-DCN

(b) SurvTree

(c) ODTP (ours)

Figure 8.6: Comparison of distributions of the next clinical outcomes in terms of Kaplan-Meier curves for each type.

- **Phenotype 1:** A low-risk group (66.02%). Patients assigned to this phenotype have a small probability of developing any adverse clinical outcomes in the near future.

- **Phenotype 2:** An intermediate-risk group (26.13%). Patients assigned to this phenotype have an intermediate probability of death due to cancer or other causes but have a higher chance of developing other tumor or recurrence in the near future compared to patients in Phenotype 3.

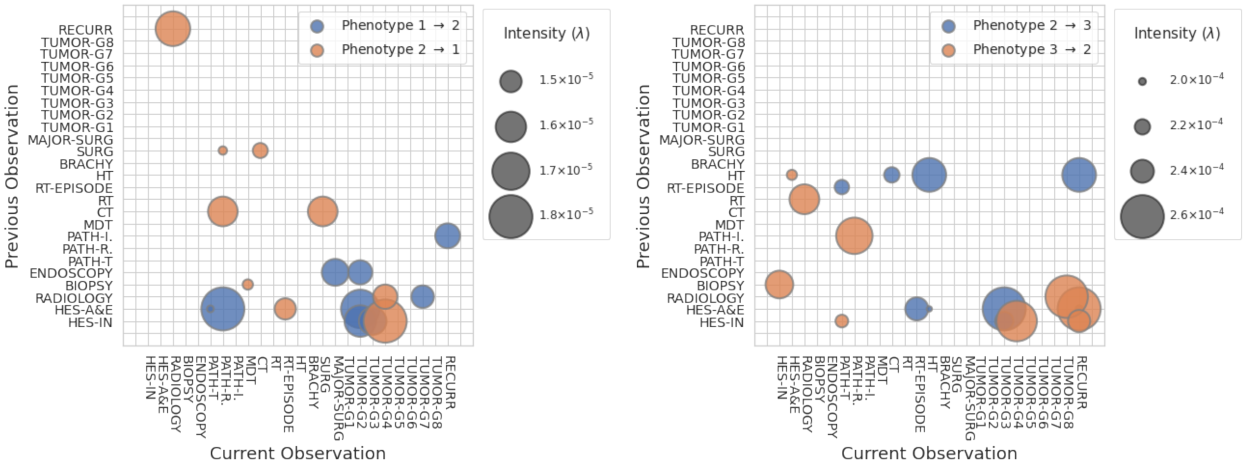
- **Phenotype 3:** A high-risk group (7.85%). Patients assigned to this phenotype have the highest chance of dying due to cancer or dying due to other causes.

Each patient can stay in one of the three identified phenotypes throughout her pathway and the phenotype assignment may change as new longitudinal observations or clinical outcomes are accrued over time.

8.5.2.1 Clinical Outcome Perspective

We now visualize how the identified phenotypes have heterogeneous clinical outcomes utilizing Kaplan-Meier (KM) curves on the *ground truth* time to the next outcome event for each event type (treating other event types as censoring) in Figure 8.6c. It is worth highlighting that the type and the timing of the next clinical outcomes are not available when phenotypes are assigned. As can be seen in Figure 8.6c, the phenotypes identified by ODTP have very heterogeneous clinical outcomes especially with respect to the time to death due to cancer or death due to other causes. Moreover, Phenotype 2 and Phenotype 3 have distinguishable distributions on the time to other tumor diagnosis or time to recurrence such that patients who are assigned to Phenotype 2 have a higher chance of developing other tumors or recurrence.

Contrarily, Figure 8.6a and 8.6b show that phenotypes identified based on the traditional notion of similarity do not properly group pathways based on the next clinical outcomes. More specifically, S2S-DCN finds phenotypes that have distributions on the time to outcome events that are indistinguishable across different phenotypes, except for the distribution on the time to other tumor diagnosis. Similarly, phenotypes that are identified by SurvTree are less heterogeneous than those identified by our method especially for time to death due to other causes or time to other tumor diagnosis. To summarize, traditional notions of similarity result in phenotyping often assigns heterogeneous phenotypes even for patients with similar clinical outcomes. This results in a lack of common prognosis in each phenotype which may mystify the understanding of the underlying disease progression [163, 164].



(a) Transitions between Phenotype 1 and 2

(b) Transitions between Phenotype 2 and 3

Figure 8.7: Comparisons of the transitions between identified phenotypes based on the current observation (x-axis) given the previous observation (y-axis) with respect to change in the conditional intensities for death from cancer. The value (node size) is averaged over all the observed transitions and the color indicates whether the transition is towards higher risk phenotypes (blue) or towards lower risk phenotypes (orange).

8.5.2.2 Longitudinal Observation Perspective

To further investigate what makes ODTP change phenotype assignments under the context of clinical pathways, we provide a scatter plot in Figure 8.7 that illustrates what the new (current) observations (i.e., $\tilde{\mathbf{x}}_i$) and the previous observations (i.e., $\tilde{\mathbf{x}}_{i-1}$) were when phenotype transitions were triggered. Here, larger points indicate a higher averaged value of the difference between conditional intensity functions at the previous and those at the current observations when the corresponding phenotype transitions are made. Formally, this can be quantified as the following:

$$\frac{1}{|\mathcal{S}_{p_1,p_2}|}\sum_{(n,i)\in\mathcal{S}_{p_1,p_2}}\lambda_m(\tilde{\mathbf{x}}^n_{1:i})-\lambda_m(\tilde{\mathbf{x}}^n_{1:i-1})$$

where $\mathcal{S}_{p_1,p_2}=\{(n,i)|\ s^n_{i-1}=p_1, s^n_i=p_2\}$ denotes a set of clinical pathway n and time step i pairs at which phenotype transitions from Phenotype p_1 to Phenotype p_2 are made. In Fig

8.7, we select 10 (previous and current) observation pairs which give the biggest difference for the corresponding phenotype transitions.

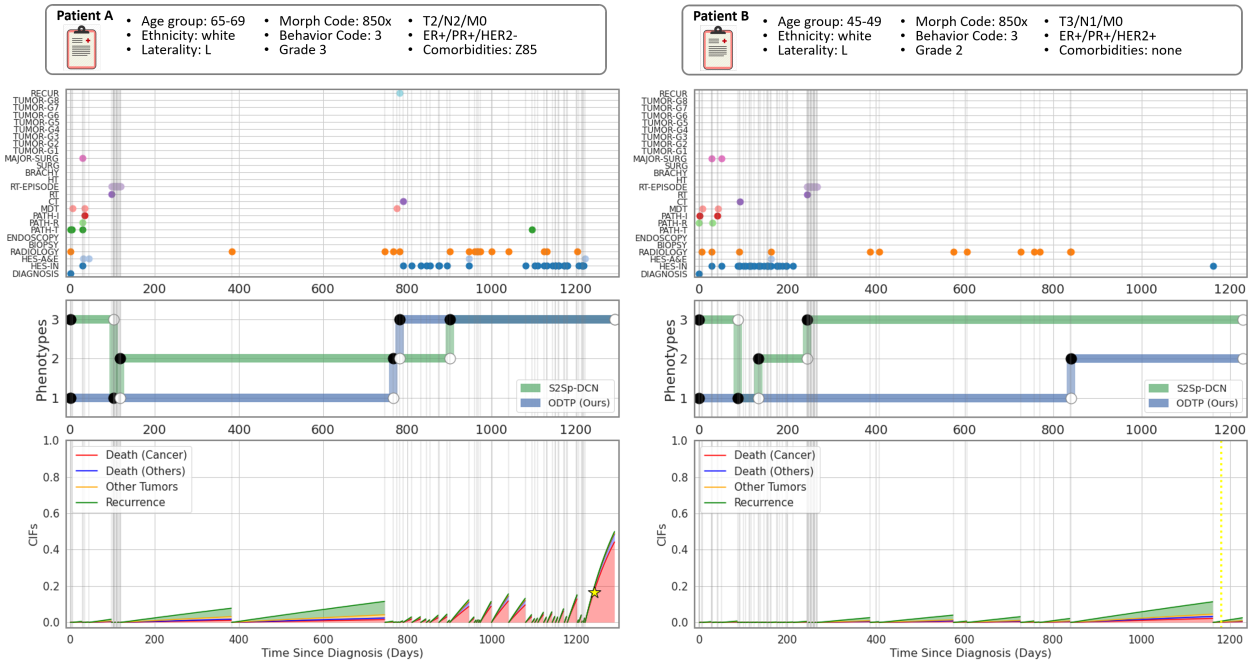
As seen in Figure 8.7, transitions from one phenotype to another phenotype behave differently based on previous and current observations, and which phenotype a patient currently belongs to. We make the following observations from Figure 8.7:

- **Transitions from Phenotype 1 to Phenotype 2:** Recurrence of breast cancer, additional tumor diagnoses – cancer in digestive organs (TUMOR-G2), other cancers (TUMOR-G7) –, and pathology reports (PATH-R.) are strong indicators for recurrence of cancer or additional tumor. Therefore, these observations significantly increased the risk of death due to cancer and triggered the transition to the higher risk group i.e., Phenotype 2.

- **Transitions from Phenotype 2 to Phenotype 1:** Pathology report (PATH-R.) or breast surgery following chemotherapy (CT) decreased the risk of death due to cancer for patients in Phenotype 2 as it reflects the patient is under neoadjuvant treatments. Moreover, additional tumor diagnosis on skin cancer (TUMOR-G4) also decreased the risk as it is well-known as one of the least fatal cancers [225].

- **Transitions from Phenotype 2 to Phenotype 3:** Recurrence of breast cancer, additional tumor diagnoses in the lung (TUMOR-G3), and consecutive hormone therapies (HT) were the top three driving factors that significantly increased the risk of death due to cancer. This is consistent with domain knowledge that the recurrence of breast cancer may increase the mortality rate due to cancer [226] and that lung cancer is by far the leading cause of cancer death [225]. Moreover, we expect that consecutive hormone therapies after initial treatment increased the risk as this implies new tumor diagnosis or recurrence.

- **Transition from Phenotype 3 to Phenotype 2:** Contrarily, recurrence of breast cancer, diagnosis of skin cancer (TUMOR-G4), and diagnosis of benign tumor or carcinoma in situ (TUMOR-G8) decreased the risk of death due to cancer. We presume that for



(a) A patient died of cancer.

(b) A patient right-censored.

Figure 8.8: An illustration of run-time examples of ODTP on two representative patients: (a) a patient who died of cancer and (b) a patient who is right-censored. We displayed (top) time-varying observations, (middle) phenotype assignments, and (bottom) risk predictions – in terms of CIFs in (8.5) for the clinical outcomes of our interest – that are dynamically updated as ODTP collects more observations along the pathways. Here, gray solid lines, yellow dotted lines, and stars indicate times at which new observations are recorded, the patient is censored, and an event occurred, respectively.

the patients in Phenotype 3 – who are at very high risk of death due to cancer – such observations are less fatal compared to what would have been observed otherwise.

8.5.3 How Does the Phenotype Assignment Change over Time?

In this experiment, we demonstrate a run-time example of how ODTP flexibly updates the phenotype assignment of a patient as new observations are collected over time. Figure 8.8

illustrates two representative patients – (a) a patient who has died from cancer and (b) a patient who was right-censored – with respect to the following components: the pathway, the corresponding phenotypes assigned by ODTP over time with comparison to that of S2S-DCN, and the predicted risks of the next clinical outcome. The predicted risk is given in terms of cumulative incidence function, defined based on (8.2) as follows:

$$F_m(s|\tilde{\mathbf{x}}_{1:i}) = \left(\frac{\lambda_m(\tilde{\mathbf{x}}_{1:i})}{\lambda(\tilde{\mathbf{x}}_{1:i})}\right)^p (1 - \exp(-(\lambda_m(\tilde{\mathbf{x}}_{1:i})s)^p)) \tag{8.5}$$

where $F_m(s|\tilde{\mathbf{x}}_{1:i})$ denotes the probability of next clinical outcome with type m occurs before or at time s given the input pathway up to time step i. It is worth highlighting that ODTP reissues risk predictions that start from 0 due to the fact that this patient is alive at the time when a new observation is collected.

There are two main points to be highlighted in Figure 8.8. First, ODTP provides phenotype assignments that are well-associated with the risk predictions on the next clinical outcomes. For Patient A, ODTP changed the phenotype assignment from Phenotype 1 to Phenotype 2 on around Day 750 (since diagnosis) because she had an increased risk of recurrence which indeed occurred on Day 781. Furthermore, as a response to the recurrence of breast cancer, this patient is reassigned to Phenotype 3 where she remains until she died due to cancer on Day 1243. For Patient B, the risk predictions on the next clinical outcomes were low and, thus, this patient remained in Phenotype 1 and Phenotype 2 until she was censored without having any adverse clinical outcomes. In contrast, S2S-DCN issued inconsistent phenotype assignments and thus frequent transitions across different phenotypes throughout the representative patients' pathways. Second, ODTP accurately predicts the next clinical outcomes by capturing the influence of static covariates and by incorporating new observations in a dynamic fashion. Particularly, the risk predictions on the next clinical outcomes (especially death due to cancer and recurrence) for Patient A, who died from cancer on Day 1243, were consistently higher (with a steeper slope) than those for Patient B, who were right-censored on Day 1179. This is because Patient A had more risk factors such as higher age, higher grade, HER2 negative – which is well-known for its difficulty in treating cancer with hormone therapy drugs [227, 228] – and comorbidity history of the previous malignant neoplasm. Moreover, our method significantly increased the risk predictions for death due to cancer as a response to the recurrence that occurred to Patient A.

8.6 Conclusion

In this work, we develop a deep learning approach to outcome-oriented phenotyping of clinical pathways with variable-length and irregularly-spaced observations, and recurrent and terminal clinical outcomes of interests. Our method models what type of clinical outcomes will occur and when throughout the clinical pathways utilizing an RNN. Identification of outcome-oriented temporal phenotypes is carried out by learning the discrete representations that best characterize the clinical outcomes based on the proposed novel loss functions. Throughout experiments on real-world data, we show that the proposed method identifies phenotypes that are strongly associated with future clinical outcomes and achieves superior performance with respect to homogeneity and heterogeneity measures compared to the benchmarks. Moreover, our posthoc analyses find driving factors of transitions between the identified phenotypes that can be translated into actionable information for better clinical decision-making.

While we provided a single clinical example, using the UK's National Cancer Registry data, our work is applicable and generalizable to other clinical datasets, which we leave as future work, where temporal phenotyping is important for understanding chronic diseases such as cancer.

Bibliography

[1] R. S. Evans, “Electronic health records: Then, now, and in the future,” *Yearb. Med. Inform.*, pp. S48–S61, 2016.

[2] S. Shilo, H. Rossman, and E. Segal, “Axes of a revolution: challenges and promises of big data in healthcare,” *Nat Med*, vol. 26, pp. 29–38, 2020.

[3] D. Blumenthal and M. Tavenner, “The”meaningful use” regulation for electronic health records,” *New England Journal of Medicine*, vol. 363(6), pp. 501–504, August 2010.

[4] D. G. Altman and J. M. Bland, “Time to event (survival) data,” *BMJ*, vol. 317, no. 7156, pp. 468–469, 1998.

[5] P. Wang, Y. Li, and C. K. Reddy, “Machine learning for survival analysis: A survey,” *arXiv preprint arXiv:1708.04649*, 2017.

[6] T. G. Liou, F. R. Adler, and D. Huang, “Use of lung transplantation survival models to refine patient selection in cystic fibrosis,” *American Journal of Respiratory and Critical Care Medicine*, vol. 171(9), pp. 1053–1059, 2005.

[7] L. Nkam, J. Lambert, A. Latouche, G. Bellis, P. Burgel, and M. Hocine, “A 3-year prognostic score for adults with cystic fibrosis,” *Journal of Cystic Fibrosis*, vol. 16(6), pp. 702–708, November 2017.

[8] D. Li, R. Keogh, J. Clancy, and R. Szczesniak, “Flexible semiparametric joint modeling: an application to estimate individual lung function decline and risk of pulmonary exacerbations in cystic fibrosis,” *Emerging Theme in Epidemiology*, vol. 14, December 2017.

[9] L. Samal, A. Wright, B. Wong, J. Linder, and D. Bates, “Leveraging electronic health records to support chronic disease management: the need for temporal data views,” *Informatics in Primary Care*, vol. 19(2), pp. 65–74, 2011.

[10] G. Hripcsak and D. J. Albers, “Next-generation phenotyping of electronic health records.,” *J Am Med Inform Assoc*, vol. 20(1), pp. 117–121, January 2013.

[11] X. Zhang, J. Chou, J. Liang, C. Xiao, Y. Zhao, H. Sarva, C. Henchcliffe, and F. Wang, “Data-driven subtyping of parkinson’s disease using longitudinal clinical records: A cohort study,” *Scientific Reports*, vol. 9(797), pp. 1–12, January 2019.

[12] H. C. van Houwlingen, “Dynamic prediction by landmarking in event history analysis,” *Scandinavian Journal of Statistics*, vol. 34(1), pp. 70–85, March 2007.

[13] R. Henderson, P. Diggle, and A. Dobson, “Joint modelling of longitudinal measurements and event time data,” *Biostatistics*, vol. 1(4), pp. 465–480, December 2000.

[14] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, “A survey of methods for explaining black box models,” *ACM computing surveys (CSUR)*, vol. 51, no. 5, p. 93, 2019.

[15] M.-L. T. Lee and G. A. Whitmore, “Threshold regression for survival analysis: Modeling event times by a stochastic process reaching a boundary,” *Statistical Science*, vol. 21(4), pp. 501–513, November 2006.

[16] R. J. Koene, A. E. Prizment, A. Blaes, and S. H. Konety, “Shared risk factors in cardiovascular disease and cancer,” *Circulation*, vol. 133, pp. 1104–1114, March 2016.

[17] J. Yoon, A. M. Alaa, M. Cadeiras, and M. van der Schaar, “Personalized donor-recipient matching for organ transplantation,” *In Proceedings of the 31th AAAI Conference on Artificial Intelligence (AAAI 2017)*, 2017.

[18] K. Ahuja, W. R. Zame, and M. van der Schaar, “Dpscreen: Dynamic personalized screening,” *In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[19] D. R. Cox, "Regression models and life tables (with discussion)," *Journal of the Royal Statistical Society. Series B*, vol. 34, pp. 187–220, 1972.

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

[21] D. Faraggi and R. Simon, "A neural network model for survival data," *Statistics in Medicine*, vol. 14, pp. 73–82, January 1995.

[22] J. Katzman, U. Shaham, J. Bates, A. Cloninger, T. Jiang, and Y. Kluger, "Deep survival: A deep cox proportional hazards network," *arXiv preprint arXiv:1606.00931*, 2016.

[23] M. Luck, T. Sylvain, H. Cardinal, A. Lodi, and Y. Bengio, "Deep learning for patient-specific kidney graft survival analysis," *arXiv preprint arXiv:1705.10245*, 2017.

[24] L. Antolini, P. Boracchi, and E. Biganzoli, "A time-dependent discrimination index for survival data," *Statistics in Medicine*, vol. 24, pp. 3927–3944, December 2005.

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

[26] E. L. Kaplan and P. Meier, "Nonpararic estimation from incomplete observations," *American Statistical Association*, vol. 53(282), pp. 457–481, June 1958.

[27] M.-L. T. Lee and G. A. Whitmore, "Proportional hazards and threshold regression: Their theoretical and practical connections," *Lifetime Data Analysis*, vol. 16, pp. 196–214, December 2010.

[28] K. A. Doksum and A. Hóyland, "Models for variable-stress accelerated life testing experiments based on wiener processes and the inverse gaussian distribution," *American*

Statistical Association and American Society for Quality, vol. 34(1), pp. 74–82, February 1992.

[29] I. M. Longini, W. S. Clark, R. H. Byers, J. W. Ward, W. W. Darrow, G. F. Lemp, and H. W. Hethcote, “Statistical analysis of the stages of hiv infection using a markov model,” *Statistics in Medicine*, vol. 8(7), pp. 831–843, July 1989.

[30] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer, “Random survival forests,” *The Annals of Applied Statistics*, vol. 2(3), pp. 841–860, September 2008.

[31] R. Ranganath, A. Perotte, N. Elhadad, and D. Blei, “Deep survival analysis,” *In Proceedings of the 1st Machine Learning for Healthcare Conference (MLHC 2016)*, 2016.

[32] C. N. Yu, R. Greiner, H. C. Lin, and V. Baracos, “Learning patient-specific cancer survival distributions as a sequence of dependent regressors,” *In Proceedings of the 24th Conference on Neural Information Processing Systems (NIPS 2011)*, 2011.

[33] T. Fernández, N. Rivera, and Y. W. Teh, “Gaussian processes for survival analysis,” *In Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016)*, 2016.

[34] A. M. Alaa and M. van der Schaar, “Deep multi-task gaussian processes for survival analysis with competing risks,” *In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[35] T. A. Gooley, W. Leisenring, J. Crowley, and B. E. Storer, “Estimation of failure probabilities in the presence of competing risks: New representations of old estimators,” *Statistics in Medicine*, vol. 18(6), pp. 695–706, March 1999.

[36] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” *In Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pp. 160–167, 2008.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 770–778, June 2016.

[38] J. F. Lawless, *Statistical Models and Methods for Lifetime Data, 2nd Edition*. Wiley, 2002.

[39] W. R. Z. J. Yoon, M. C. A. Banerjee, A. M. Alaa, and M. van der Schaar, "Personalized survival predictions for cardiac transplantation via trees of predictors," *arXiv preprint arXiv:1704.03458*, 2017.

[40] E. Bilal, J. Dutkowski, J. Guinney, I. S. Jang, B. A. Logsdon, G. Pandey, B. A. Sauerwine, Y. Shimoni, H. K. M. Vollan, B. H. Mecham, O. M. Rueda, J. Tost, C. Curtis, M. J. Alvarez, V. N. Kristensen, S. Aparicio, A.-L. Børresen-Dale, C. Caldas, A. Califano, S. H. Friend, T. Ideker, E. E. Schadt, G. A. Stolovitzky, and A. A. Margolin, "Improving breast cancer survival analysis through competition-based multidimensional modeling," *PLoS Computational Biology*, May 2013.

[41] T. M. Therneau, *A Package for Survival Analysis in S*, 2015. version 2.38.

[42] H. Ishwaran and U. B. Kogalur, *Random Forests for Survival, Regression and Classification (RF-SRC)*, 2017. R package version 2.4.2.

[43] B. Haller, G. Schmidt, and K. Ulm, "Applying competing risks regression models: an overview," *Lifetime Data Analysis*, vol. 19, pp. 33–58, January 2013.

[44] J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman, and Z. Ghahramani, "A general framework for constrained bayesian optimization using information-based search," *Journal of Machine Learning Research*, vol. 17, pp. 1–53, 2016.

[45] G. Ridgeway, "The state of boosting," *Computing Science and Statistics*, pp. 172–181, 1999.

[46] H. Binder and M. Schumacher, "Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models," *BMC Bioinformatics*, vol. 9(1), 2008.

[47] T. Fernández, N. Rivera, and Y. W. Teh, "Gaussian processes for survival analysis," *In Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016)*, 2016.

[48] T. Hothorn, K. Hornik, and A. Zeileis, "Unbiased recursive partitioning: A conditional inference framework.," *Journal of Computational and Graphical Statistics*, vol. 15(3), pp. 651–674, 2006.

[49] A. Bellot and M. van der Schaar, "Boosted trees for risk prognosis," *In Proceedings of the 3st Machine Learning for Healthcare Conference (MLHC 2018)*, 2018.

[50] C. Lee, W. R. Zame, J. Yoon, and M. van der Schaar, "Deephit: A deep learning approach to survival analysis with competing risks," *In Proceedings of the 32th AAAI Conference on Artificial Intelligence (AAAI 2018)*, 2018.

[51] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA," *Journal of Machine Learning Research*, vol. 18(25), pp. 1–5, 2017.

[52] A. M. Alaa and M. van der Schaar, "Autoprognosis: Automated clinical prognostic modeling via bayesian optimization with structured kernel learning," *In Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, 2018.

[53] A. Wey, J. Connett, and K. Rudser, "Combining parametric, semi-parametric, and non-parametric survival models with stacked survival models," *Biostatistics*, vol. 16(3), pp. 537–549, February 2015.

[54] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[55] K. Kandasamy, J. Schneider, and B. Póczos, "High dimensional bayesian optimisation and bandits via additive models," *In Proceedings of the 32th International Conference on Machine Learning (ICML 2015)*, 2015.

[56] J. Nocedal and S. J. Wright, *Numerical Optimization, 2nd Edition.* Springer, 2006.

[57] R. B. Gramacy, G. A. Gray, S. L. Digabel, H. K. Lee, P. Ranjan, G. Wells, and S. M. Wild, "Modeling an augmented lagrangian for blackbox constrained optimization," *Technometrics*, vol. 58(1), pp. 1–11, 2016.

[58] T. A. Gerds, M. W. Kattan, M. Schumacher, and C. Yu, "Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring," *Statistics in Medicine*, vol. 32(13), pp. 2173–2184, 2013.

[59] U. B. Mogensen, H. Ishwaran, and T. A. Gerds, "Evaluating random forests for survival analysis using prediction error curves," *Journal of Statistical Software*, vol. 50(11), 2012.

[60] C. M. Wong, N. M. Hawkins, M. C. Petrie, P. S. Jhund, R. S. Gardner, C. A. Ariti, K. K. Poppe, N. Earle, G. A. Whalley, I. B. Squire, R. N. Doughty, and J. J. McMurray, "Heart failure in younger patients: the meta-analysis global group in chronic heart failure (MAGGIC)," *European Heart Journal*, vol. 35(39), pp. 2714–2721, June 2014.

[61] W. A. Knaus, 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, "The SUPPORT prognostic model. objective estimates of survival for seriously ill hospitalized adults. study to understand prognoses and preferences for outcomes and risks of treatments," *Annals of Internal Medicine*, vol. 122(3), pp. 191–203, February 1995.

[62] C. Curtis, S. P. Shah, S.-F. Chin, G. Turashvili, O. M. Rueda, M. J. Dunning, D. Speed, A. G. Lynch, S. Samarajiwa, Y. Yuan, S. Gräf, G. Ha, G. Haffari, A. Bashashati, R. Russell, S. McKinney, M. Group, A. Langerød, A. Green, E. Provenzano, G. Wishart, S. Pinder, P. Watson, F. Markowetz, L. Murphy, I. Ellis, A. Purushotham, A.-L. Børresen-Dale, J. D. Brenton, S. Tavaré, C. Caldas, and S. Aparicio, "The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups," *Nature*, vol. 486, pp. 346–352, June 2012.

[63] J. F. Hayes, L. Marston, K. Walters, J. R. Geddes, M. King, and D. P. J. Osborn, "Adverse renal, endocrine, hepatic, and metabolic events during maintenance mood stabilizer treatment for bipolar disorder: A population-based cohort study," *PLOS Medicine*, vol. 13(8), p. e1002058, August 2016.

[64] J. L. Hill, "Bayesian nonparametric modeling for causal inference," *Journal of Computational and Graphical Statistics*, vol. 20, no. 1, pp. 217–240, 2011.

[65] S. Athey and G. Imbens, "Recursive partitioning for heterogeneous causal effects," *Proceedings of the National Academy of Sciences*, vol. 113, no. 27, pp. 7353–7360, 2016.

[66] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1228–1242, 2018.

[67] S. Athey, J. Tibshirani, S. Wager, *et al.*, "Generalized random forests," *The Annals of Statistics*, vol. 47, no. 2, pp. 1148–1178, 2019.

[68] P. R. Hahn, J. S. Murray, C. M. Carvalho, *et al.*, "Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion)," *Bayesian Analysis*, vol. 15, no. 3, pp. 965–1056, 2020.

[69] A. M. Alaa and M. van der Schaar, "Bayesian inference of individualized treatment

effects using multi-task gaussian processes," *Advances in Neural Information Processing Systems*, vol. 30, pp. 3424–3432, 2017.

[70] A. Alaa and M. van der Schaar, "Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design," in *International Conference on Machine Learning*, pp. 129–138, 2018.

[71] F. Johansson, U. Shalit, and D. Sontag, "Learning representations for counterfactual inference," in *International conference on machine learning*, pp. 3020–3029, PMLR, 2016.

[72] U. Shalit, F. D. Johansson, and D. Sontag, "Estimating individual treatment effect: Generalization bounds and algorithms," *In Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML 2016)*, 2016.

[73] F. D. Johansson, N. Kallus, U. Shalit, and D. Sontag, "Learning weighted representations for generalization across designs," *arXiv preprint arXiv:1802.08598*, 2018.

[74] C. Shi, D. Blei, and V. Veitch, "Adapting neural networks for the estimation of treatment effects," in *Advances in Neural Information Processing Systems*, pp. 2507–2517, 2019.

[75] N. Hassanpour and R. Greiner, "Counterfactual regression with importance sampling weights.," in *IJCAI*, pp. 5880–5887, 2019.

[76] N. Hassanpour and R. Greiner, "Learning disentangled representations for counterfactual regression," in *International Conference on Learning Representations*, 2020.

[77] S. Assaad, S. Zeng, C. Tao, S. Datta, N. Mehta, R. Henao, F. Li, and L. C. Duke, "Counterfactual representation learning with balancing weights," in *International Conference on Artificial Intelligence and Statistics*, pp. 1972–1980, PMLR, 2021.

[78] A. Curth and M. van der Schaar, "Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms," *Proceedings of the 24th International*

Conference on Artificial Intelligence and Statistics (AISTATS) 2021 (To Appear); arXiv preprint arXiv:2101.10943, 2021.

[79] G. Tutz, M. Schmid, *et al.*, *Modeling discrete time-to-event data.* Springer, 2016.

[80] J. P. Klein and M. L. Moeschberger, *Survival analysis: techniques for censored and truncated data*, vol. 1230. Springer, 2003.

[81] M. J. Van der Laan and S. Rose, *Targeted learning: causal inference for observational and experimental data.* Springer Science & Business Media, 2011.

[82] O. M. Stitelman and M. J. van der Laan, "Collaborative targeted maximum likelihood for time to event data," *The International Journal of Biostatistics*, vol. 6, no. 1, 2010.

[83] O. M. Stitelman, C. W. Wester, V. De Gruttola, and M. J. van der Laan, "Targeted maximum likelihood estimation of effect modification parameters in survival analysis," *The international journal of biostatistics*, vol. 7, no. 1, 2011.

[84] W. Cai and M. J. van der Laan, "One-step targeted maximum likelihood estimation for time-to-event outcomes," *Biometric Methodology*, 2019.

[85] S. Tabib and D. Larocque, "Non-parametric individual treatment effect estimation for survival data with random forests," *Bioinformatics*, vol. 36, no. 2, pp. 629–636, 2020.

[86] N. C. Henderson, T. A. Louis, G. L. Rosner, and R. Varadhan, "Individualized treatment effects with censored data via fully nonparametric bayesian accelerated failure time models," *Biostatistics*, vol. 21, no. 1, pp. 50–68, 2020.

[87] W. Zhang, T. D. Le, L. Liu, Z.-H. Zhou, and J. Li, "Mining heterogeneous causal effects for personalized cancer treatment," *Bioinformatics*, vol. 33, no. 15, pp. 2372–2378, 2017.

[88] Y. Cui, M. R. Kosorok, S. Wager, and R. Zhu, "Estimating heterogeneous treatment effects with right-censored data via causal survival forests," *arXiv preprint arXiv:2001.09887*, 2020.

[89] P. Chapfuwa, S. Assaad, S. Zeng, M. J. Pencina, L. Carin, and R. Henao, "Enabling counterfactual survival analysis with balanced representations," in *Proceedings of the Conference on Health, Inference, and Learning*, pp. 133–145, 2021.

[90] J. Pearl, *Causality.* Cambridge University Press, 2009.

[91] D. B. Rubin, "Causal inference using potential outcomes: Design, modeling, decisions," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 322–331, 2005.

[92] I. Díaz, O. Savenkov, and K. Ballman, "Targeted learning ensembles for optimal individualized treatment rules with time-to-event outcomes," *Biometrika*, vol. 105, no. 3, pp. 723–738, 2018.

[93] W. M. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," *arXiv preprint arXiv:1812.11806*, 2018.

[94] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

[95] C. Cortes, Y. Mansour, and M. Mohri, "Learning bounds for importance weighting.," in *Nips*, vol. 10, pp. 442–450, Citeseer, 2010.

[96] F. D. Johansson, U. Shalit, N. Kallus, and D. Sontag, "Generalization bounds and representation learning for estimation of potential outcomes and causal effects," *arXiv preprint arXiv:2001.07426*, 2020.

[97] F. D. Johansson, D. Sontag, and R. Ranganath, “Support and invertibility in domain-invariant representations,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 527–536, PMLR, 2019.

[98] M. Cuturi and A. Doucet, “Fast computation of wasserstein barycenters,” in *International Conference on Machine Learning*, pp. 685–693, PMLR, 2014.

[99] J. Byrd and Z. Lipton, “What is the effect of importance weighting in deep learning?,” in *International Conference on Machine Learning*, pp. 872–881, PMLR, 2019.

[100] D. Xu, Y. Ye, and C. Ruan, “Understanding the role of importance weighting for deep learning,” *In Proceedings of the 10th International Conference on Learning Representations (ICLR 2021)*, 2021.

[101] J. Yoon, J. Jordon, and M. van der Schaar, “Ganite: Estimation of individualized treatment effects using generative adversarial nets,” in *International Conference on Learning Representations*, 2018.

[102] S. R. Künzel, J. S. Sekhon, P. J. Bickel, and B. Yu, “Metalearners for estimating heterogeneous treatment effects using machine learning,” *Proceedings of the national academy of sciences*, vol. 116, no. 10, pp. 4156–4165, 2019.

[103] E. H. Kennedy, “Optimal doubly robust estimation of heterogeneous causal effects,” *arXiv preprint arXiv:2004.14497*, 2020.

[104] X. Nie and S. Wager, “Quasi-oracle estimation of heterogeneous treatment effects,” *Biometrika*, vol. 108, no. 2, pp. 299–319, 2021.

[105] D. Faraggi and R. Simon, “A neural network model for survival data,” *Statistics in medicine*, vol. 14, no. 1, pp. 73–82, 1995.

[106] P. Chapfuwa, C. Tao, C. Li, C. Page, B. Goldstein, L. C. Duke, and R. Henao, "Adversarial time-to-event modeling," in *International Conference on Machine Learning*, pp. 735–744, PMLR, 2018.

[107] J. A. Steingrimsson, L. Diao, and R. L. Strawderman, "Censoring unbiased regression trees and ensembles," *Journal of the American Statistical Association*, vol. 114, no. 525, pp. 370–383, 2019.

[108] J. A. Steingrimsson and S. Morrison, "Deep learning for survival outcomes," *Statistics in medicine*, vol. 39, no. 17, pp. 2339–2349, 2020.

[109] C. C. Brown, "On the use of indicator variables for studying the time-dependence of parameters in a response-time model," *Biometrics*, pp. 863–872, 1975.

[110] E. Biganzoli, P. Boracchi, L. Mariani, and E. Marubini, "Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach," *Statistics in medicine*, vol. 17, no. 10, pp. 1169–1186, 1998.

[111] M. F. Gensheimer and B. Narasimhan, "A scalable discrete-time survival model for neural networks," *PeerJ*, vol. 7, p. e6257, 2019.

[112] K. Ren, J. Qin, L. Zheng, Z. Yang, W. Zhang, L. Qiu, and Y. Yu, "Deep recurrent survival analysis," *In Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, 2019.

[113] H. Kvamme and Ø. Borgan, "Continuous and discrete-time survival prediction with neural networks," *arXiv preprint arXiv:1910.06724*, 2019.

[114] C. Lee, W. Zame, J. Yoon, and M. van der Schaar, "Deephit: A deep learning approach to survival analysis with competing risks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[115] C.-N. Yu, R. Greiner, H.-C. Lin, and V. Baracos, “Learning patient-specific cancer survival distributions as a sequence of dependent regressors,” *Advances in Neural Information Processing Systems*, vol. 24, pp. 1845–1853, 2011.

[116] L. Hu, J. Ji, and F. Li, “Estimating heterogeneous survival treatment effect in observational data using machine learning,” *arXiv preprint arXiv:2008.07044*, 2020.

[117] C. Louizos, U. Shalit, J. Mooij, D. Sontag, R. Zemel, and M. Welling, “Causal effect inference with deep latent-variable models,” *In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017)*, 2017.

[118] S. Pölsterl, “scikit-survival: A library for time-to-event analysis built on top of scikit-learn,” *Journal of Machine Learning Research*, vol. 21, no. 212, pp. 1–6, 2020.

[119] H. Uno, T. Cai, M. J. Pencina, R. 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*, vol. 30, no. 10, pp. 1105–1117, 2011.

[120] A. Bellot and M. van der Schaar, “Tree-based bayesian mixture model for competing risks,” *In Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS 2018)*, 2018.

[121] S. D. Aarona, A. L. Stephenson, D. W. Cameron, and G. A. Whitmore, “A statistical model to predict one-year risk of death in patients with cystic fibrosis,” *Journal of Clinical Epidemiology*, vol. 68, pp. 1336–1345, 2015.

[122] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *In Proceedings of the 27th Conference on Neural Information Processing Systems (NIPS 2014)*, 2014.

[123] A. Karpathy and F.-F. Li, “Deep visual-semantic alignments for generating image de-

scriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39(4), pp. 664–676, April 2017.

[124] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *In Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, 2013.

[125] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[126] N. Kobelska-Dubiel, B. Klincewicz, and W. Cichy, "Liver disease in cystic fibrosis," *Prz Gastroenterol*, vol. 9(3), pp. 136–141, June 2014.

[127] J. G. Ibrahim, H. Chu, and L. M. Chen, "Basic concepts and methods for joint models of longitudinal and survival data," *Journal of Clinical Oncology*, vol. 28(16), pp. 2796–2801, June 2010.

[128] J. Barrett and L. Su, "Dynamic predictions using flexible joint models of longitudinal and time-to-event data," *Statistics in Medicine*, vol. 36(9), pp. 1447–1460, April 2017.

[129] E. R. Brown, J. G. Ibrahim, and V. DeGruttola, "A flexible b-spline model for multiple longitudinal biomarkers and survival," *Biometrics*, vol. 61(1), pp. 64–73, March 2005.

[130] G. L. Hickey, P. Philipson, A. Jorgensen, and R. Kolamunnage-Dona, "Joint modelling of time-to-event and multivariate longitudinal outcomes: Recent developments and issues," *BMC Medical Research Methodology*, vol. 16, p. 117, September 2016.

[131] Y. Liu, L. Liu, and J. Zhou, "Joint latent class model of survival and longitudinal data: An application to cpcra study," *Computational Statistics & Data Analysis*, vol. 91, pp. 40–50, November 2015.

[132] E.-R. Andrinopoulou, K. Nasserinejad, R. Szczesniak, and D. Rizopoulos, “Integrating latent classes in the bayesian shared parameter joint model of longitudinal and survival outcomes,” *arXiv preprint arXiv:1502.02072*, 2018.

[133] Y. Zheng and P. J. Heagerty, “Partly conditional survival models for longitudinal data,” *Biometrics*, vol. 61, pp. 379–391, March 2005.

[134] P. J. Heagerty and Y. Zheng, “Survival model predictive accuracy and roc curves,” *Biometrics*, vol. 61, pp. 92–105, March 2005.

[135] H. C. van Houwelingen and H. Putter, “Dynamic predicting by landmarking as an alternative for multi-state modeling: an application to acute lymphoid leukemia data,” *Lifetime Data Analysis*, vol. 14(4), pp. 447–463, December 2008.

[136] A. A. Tsiatis and M. Davidian, “Joint modeling of longitudinal and time-to-event data: an overview,” *Statistica Sinica*, vol. 1(4), pp. 809–834, July 2004.

[137] L. Deng, G. E. Hinton, and B. Kingsbury, “New types of deep neural network learning for speech recognition and related applications: An overview,” *In Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pp. 8599–8603, 2013.

[138] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, “Massively multitask networks for drug discovery,” *arXiv preprint arXiv:1502.02072*, 2015.

[139] H. Harutyunyan, H. Khachatrian, D. C. Kale, and A. Galstyan, “Multitask learning and benchmarking with clinical time series data,” *arXiv preprint arXiv:1703.07771*, 2017.

[140] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[141] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9(8), pp. 1735–1780, 1997.

[142] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45(11), pp. 2673–2681, November 1997.

[143] T. Liou, F. Adler, B. Cahill, S. FitzSimmons, D. Huang, J. Hibbs, and B. Marshall, “Survival effect of lung transplantation among patients with cystic fibrosis,” *JAMA*, vol. 286(21), pp. 2683–2689, December 2001.

[144] M. Hofer, C. Benden, I. Inci, C. Schmid, S. Irani, R. Speich, W. Weder, and A. Boehler, “True survival benefit of lung transplantation for cystic fibrosis patients: the zurich experience,” *The Journal of Heart and Lung Transplantation*, vol. 28(4), pp. 334–339, April 2009.

[145] N. Mayer-Hamblett, M. Rosenfeld, J. Emerson, C. Goss, and M. Aitken, “Developing cystic fibrosis lung transplant referral criteria using predictors of 2-year mortality,” *American Journal Respiratory Critical Care Medicines*, vol. 166, pp. 1550–1555, December 2002.

[146] T. G. Liou, F. R. Adler, and D. Huang, “Use of lung transplantation survival models to refine patient selection in cystic fibrosis,” *American Journal Respiratory Critical Care Medicines*, vol. 171, pp. 1053–1059, May 2005.

[147] D. S. Urquhart, L. P. Thia, J. Francis, S. A. Prasad, C. Dawson, C. Wallis, and I. M. Balfour-Lynn, “Deaths in childhood from cystic fibrosis: 10-year analysis from two london specialist centres,” *Archives of Disease in Childhood*, vol. 98(2), p. 123–127, February 2013.

[148] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, pp. 281–305, February 2012.

[149] D. Rizopoulos, "The r package jmbayes for fitting joint models for longitudinal and time-to-event data using mcmc," *Journal of Statistical Software*, vol. 72(7), 2016.

[150] D. Rizopoulos, G. Molenberghs, and E. M. Lesaffre, "Dynamic predictions with time-dependent covariates in survival analysis using joint modeling and landmarking," *Biometrical Journal*, vol. 59(6), pp. 1261–1276, November 2017.

[151] K. Suresh, J. M. Taylor, D. E. Spratt, S. Daignault, and A. Tsodikov, "Comparison of joint modeling and landmarking for dynamic prediction under an illness-death model," *Biometrical Journal*, vol. 59(6), pp. 1277–1300, November 2017.

[152] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," *In Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016)*, 2016.

[153] A. Avati, K. Jung, S. Harman, L. Downing, A. Ng, and N. H. Shah, "Improving palliative care with deep learning," *arXiv preprint arXiv:1711.06402*, 2017.

[154] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29(5), pp. 1189–1232, 2001.

[155] A. L. Stephenson, L. A. Mannik, S. Walsh, M. Brotherwood, R. Robert, P. B. Darling, R. Nisenbaum, J. Moerman, and S. Stanojevic, "Longitudinal trends in nutritional status and the relation between lung function and BMI in cystic fibrosis: a population-based cohort study," *The American Journal of Clinical Nutrition*, vol. 97(4), p. 822–827, April 2013.

[156] B. Fauroux, N. Hart, S. Belfar, M. Boulé, I. Tillous-Borde, D. Bonnet, E. Bingen, and A. Clément, "Burkholderia cepacia is associated with pulmonary hypertension and increased mortality among cystic fibrosis patients," *Journal of Clinical Microbiology.*

[157] J. Yoon, C. Davtyan, and M. van der Schaar, "Discovery and clinical decision support for personalized healthcare," *IEEE J Biomed Health Inform.*, vol. 21(4), pp. 1133–1145, 2017.

[158] K. J. Ramos, B. S. Quon, S. L. Heltshe, N. Mayer-Hamblett, E. D. Lease, M. L. Aitken, N. S. Weiss, and C. H. Goss, "Heterogeneity in survival in adult patients with cystic fibrosis with $\text{FEV}_1 < 30\%$ of predicted in the united states," *Chest*, vol. 151(6), pp. 1320–1328, June 2017.

[159] C. Lee, J. Yoon, and M. van der Schaar, "Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data," *IEEE Transactions on Biomedical Engineering*, April 2019.

[160] A. Rusanov, P. V. Prado, and C. Weng, "Unsupervised time-series clustering over lab data for automatic identification of uncontrolled diabetes," *In Proceedings of the 4th IEEE International Conference on Healthcare Informatics (ICHI)*, 2016.

[161] A. Giannoula, A. Gutierrez-Sacristán, A. Bravo, F. Sanz, and L. I. Furlong, "Identifying temporal patterns in patient disease trajectories using dynamic ping: A population-based study," *Scientific Reports*, vol. 8(4216), pp. 1–14, March 2018.

[162] D. T. A. Luong and V. Chandola, "A k-means approach to clustering disease progressions," *In Proceedings of the 5th IEEE International Conference on Healthcare Informatics (ICHI)*, 2017.

[163] A. Boudier, S. Chanoine, S. Accordini, J. M. Anto, X. B. na, J. Bousquet, P. Demoly, J. Garcia-Aymerich, F. Gormand, J. Heinrich, C. Janson, N. Künzli, R. Matran, C. Pison, C. Raherison, J. Sunyer, R. Varraso, D. Jarvis, B. Leynaert, I. Pin, and V. Siroux, "Data-driven adult asthma phenotypes based on clinical characteristics are associated with asthma outcomes twenty years later," *Allegy*, vol. 74(5), pp. 953–963, May 2019.

[164] W. M. Wami, F. Buntinx, S. Bartholomeeusen, G. Goderis, C. Mathieu, and M. Aerts, "Influence of chronic comorbidity and medication on the efficacy of treatment in patients with diabetes in general practice," *The British Journal of General Practice*, vol. 63(609), pp. 267–273, March 2013.

[165] H. Blockeel, S. Dzeroski, J. Struyf, and B. Zenko, *Predictive Clustering.* Springer New York, 2017.

[166] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," *In Proceedings of the 13th Conference on Neural Information Processing Systems (NIPS 2000)*, 2000.

[167] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," *In Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, 2017.

[168] C. A. Ratanamahatana, E. Keogh, A. J. Bagnall, and S. Lonardi, "A novel bit level time series representation with implications for similarity search and clustering," *In Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005)*, 2005.

[169] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering – a decade review," *Information Systems*, vol. 53, pp. 16–38, May 2015.

[170] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware lstm networks," *In Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2017)*, 2017.

[171] N. S. Madiraju, S. M. Sadat, D. Fisher, and H. Karimabadi, "Deep temporal clustering: Fully unsupervised learning of time-domain features," *arXiv preprint arXiv:1802.01059*, 2018.

[172] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” *In Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, 2017.

[173] V. Fortuin, M. Hüser, F. Locatello, H. Strathmann, and G. Rätsch, “SOM-VAE: Interpretable discrete representation learning on time series,” *In Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, 2019.

[174] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” *In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[175] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, 2010.

[176] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[177] N. X. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance,” *Journal of Machine Learning Research*, vol. 11(1), pp. 2837–2854, October 2010.

[178] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of Classification*, vol. 2(1), pp. 193–218, December 1985.

[179] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[180] N. J. Ronan, J. Elborn, and B. J. Plant, “Current and emerging comorbidities in cystic fibrosis,” *Presse Med.*, vol. 46(6), pp. 125–138, June 2017.

[181] H. E. Taitt, "Global trends and prostate cancer: A review of incidence, detection, and mortality as influenced by race, ethnicity, and geographic location," *Am J Mens Health*, vol. 12, no. 6, pp. 1807–1823, 2018.

[182] J. L. Donovan, F. C. Hamdy, and J. A. e. a. Lane, "Patient-reported outcomes after monitoring, surgery, or radiotherapy for prostate cancer," *N Engl J Med*, vol. 375, no. 15, pp. 1425–1437, 2016.

[183] F. C. Hamdy, J. L. Donovan, and J. A. e. a. Lane, "10-year outcomes after monitoring, surgery, or radiotherapy for localized prostate cancer," *N Engl J Med*, vol. 375, no. 15, pp. 1415–1424, 2016.

[184] T. J. Wilt, K. M. Jones, and M. J. e. a. Barry, "Follow-up of prostatectomy versus observation for early prostate cancer," *N Engl J Med*, vol. 377, no. 2, pp. 132–142, 2017.

[185] V. J. Gnanapragasa, A. Lophatananon, K. A. Wright, K. R. Muir, A. Gavin, and D. C. Greenberg, "Improving clinical risk stratification at diagnosis in primary prostate cancer: A prognostic modelling study," *PLoS Med*, vol. 13, no. 8, p. e1002063, 2016.

[186] N. Mottet, J. Bellmunt, and M. e. a. Bolla, "Eau-estro-siog guidelines on prostate cancer. part 1: Screening, diagnosis, and local treatment with curative intent," *Eur Urol*, vol. 71, no. 4, pp. 618–629, 2017.

[187] J. Graham, P. Kirkbride, K. Cann, E. Hasler, and M. Prettyjohns, "Prostate cancer: summary of updated nice guidance," *BMJ*, vol. 348, no. 1, 2014.

[188] H. Lukka, P. Warde, and T. e. a. Pickles, "Controversies in prostate cancer radiotherapy: consensus development," *Can J Urol*, vol. 8, no. 4, pp. 1314–1322, 2001.

[189] M. G. Sanda, J. A. . Cadeddu, and E. e. a. Kirkby, "Clinically localized prostate cancer: Aua/astro/suo guideline. part i: Risk stratification, shared decision making, and care options," *J Urol*, vol. 199, no. 3, pp. 683–690, 2018.

[190] J. L. Mohler, A. J. Armstrong, and R. R. e. a. Bahnson, “Prostate cancer, version 1,” *J Natl Compr Canc Netw*, vol. 14, no. 1, pp. 19–30, 2016.

[191] F. M. Jhaveri, C. D. Zippe, E. A. Klein, and P. A. Kupelian, “Biochemical failure does not predict overall survival after radical prostatectomy for localized prostate cancer: 10-year results,” *Urology*, vol. 54, no. 5, pp. 884–890, 1999.

[192] M. W. Kattan, K. R. Hess, and M. B. e. a. Amin, “American joint committee on cancer acceptance criteria for inclusion of risk models for individualized prognosis in the practice of precision medicine,” *CA Cancer J Clin*, vol. 66, no. 5, pp. 370–374, 2016.

[193] R. Zelic, H. Garmo, and D. e. a. Zugna, “Predicting prostate cancer death with different pretreatment risk stratification tools: A head-to-head comparison in a nationwide cohort study,” *Eur Urol*, vol. 77, no. 2, pp. 180–188, 2020.

[194] D. R. Thurtle, D. C. Greenberg, L. S. Lee, H. H. Huang, P. D. Pharoah, and V. J. Gnanapragasam, “Individual prognosis at diagnosis in nonmetastatic prostate cancer: Development and external validation of the predict prostate multivariable model,” *PLoS Med*, vol. 16, no. 3, p. e1002758, 2019.

[195] A. J. Stephenson, M. W. Kattan, and J. A. e. a. Eastham, “Prostate cancer-specific mortality after radical prostatectomy for patients treated in the prostate-specific antigen era,” *J Clin Oncol*, vol. 27, no. 26, pp. 4300–4305, 2009.

[196] R. assessment for prostate cancer metastasis and mortality at the time of diagnosis, “Cooperberg, m. r. and broering, j. m. and carroll, p. r.,” *J Natl Cancer Inst*, vol. 101, no. 12, pp. 878–887, 2009.

[197] A. M. Alaa, T. Bolton, E. Di Angelantonio, J. H. F. Rudd, and M. van der Schaar, “Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 uk biobank participants,” *PLoS One*, vol. 14, no. 5, p. e0213653, 2019.

[198] K. C. Koo, K. S. Lee, and S. e. a. Kim, "Long short-term memory artificial neural network model for prediction of prostate cancer survival outcomes according to initial treatment strategy: development of an online decision-making support system.," *World J Urol*, 2020.

[199] Y. T. Lin, M. T. Lee, Y. C. Huang, C. K. Liu, Y. T. Li, and M. Chen, "Prediction of recurrence-associated death from localized prostate cancer with a charlson comorbidity index-reinforced machine learning model," *Open Med (Wars)*, vol. 14, no. 1, pp. 593–606, 2019.

[200] M. J. Donovan, G. Fernandez, and R. e. a. Scott, "Development and validation of a novel automated gleason grade and molecular profile that define a highly predictive prostate cancer progression algorithm-based test," *Prostate Cancer Prostatic Dis*, vol. 21, no. 4, pp. 594–603, 2018.

[201] S. Zhang, Y. Xu, and X. e. a. Hui, "Improvement in prediction of prostate cancer prognosis with somatic mutational signatures," *J Cancer*, vol. 8, no. 16, pp. 3261–3267, 2017.

[202] C. Lee, W. R. Zame, J. Yoon, and M. van der Schaar, "Temporal quilting for survival analysis," *In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019)*, 2019.

[203] E. National Cancer Institute Surveillance, E. R. P.-S. A. W. Group, Adamo, M. P., and B. J. A. et al., "Validation of prostate-specific antigen laboratory values recorded in surveillance, epidemiology, and end results registries," *Cancer*, vol. 123, no. 4, pp. 697–703, 2017.

[204] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer, "Random survival forests," *Ann Appl Stat*, vol. 2, no. 3, pp. 841–860, 2008.

[205] T. Hothorn, K. Hornik, and A. Zeileis, “Unbiased recursive partitioning: A conditional inference framework,” *J Comput Graph Stat*, vol. 15, no. 3, pp. 651–674, 2006.

[206] C. Lee, W. R. Zame, J. Yoon, and M. van der Schaar, “Deephit: A deep learning approach to survival analysis with competing risks,” *In Proceedings of the 32th AAAI Conference on Artificial Intelligence (AAAI 2018)*, 2018.

[207] D. Thurtle, O. Bratt, P. Stattin, P. Pharoah, and V. Gnanapragasam, “Comparative performance and external validation of the multivariable predict prostate tool for non-metastatic prostate cancer: a study in 69,206 men from prostate cancer data base sweden (pcbase),” *BMC Med*, vol. 18, no. 1, p. 139, 2020.

[208] D. Thurtle, S. H. Rossi, B. Berry, P. Pharoah, and V. Gnanapragasam, “Models predicting survival to guide treatment decision-making in newly diagnosed primary non-metastatic prostate cancer: a systematic review.,” *BMJ Open*, vol. 9, no. 6, p. e029149, 2019.

[209] B. Van Calster, L. Wynants, J. F. M. Verbeek, J. Y. Verbakel, E. Christodoulou, A. J. Vickers, and E. W. Roobol, M. J. amd Steyerberg, “Reporting and interpreting decision curve analysis: A guide for investigators,” *Eur Urol*, vol. 74, no. 6, pp. 796–804, 2018.

[210] A. Vickers, S. V. Carlsson, and M. Cooperberg, “Routineuse of magnetic resonance imaging for early detection of prostate cancer is not justified by the clinical trial evidence,” *Eur Urol*, vol. 78, no. 3, pp. 310–313, 2020.

[211] D. E. Spratt, J. Zhang, and S.-J. M. et al., “Development and validation of a novel integrated clinical-genomic risk group classification for localized prostate cancer,” *J Clin Oncol*, vol. 36, no. 6, pp. 581–590, 2018.

[212] C. Lee and M. van der Schaar, “Temporal phenotyping using deep predictive clustering of disease progression,” *In Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 2020.

[213] A. Nandakumar, G. K. Rath, and A. C. K. et al., “Decreased survival with mastectomy vis-à-vis breast-conserving surgery in stage ii and iii breast cancers: A comparative treatment effectiveness study,” *Journal of Global Oncology*, vol. 3(4), pp. 304–313, August 2017.

[214] R. Takahashi, U. Toh, and N. I. et al., “Treatment outcome in patients with stage III breast cancer treated with neoadjuvant chemotherapy,” *Experimental and Therapeutic Medicine*, vol. 6(5), pp. 1089–1095, September 2013.

[215] J. C. Ho, J. Ghosh, and J. Sun, “Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization,” *In Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2014)*, 2014.

[216] S. Lloyd, “Least squares quantization in pcm,” *IEEE Transaction on Information Theory*, vol. 28(2), pp. 129–137, March 1982.

[217] J. V. McGowan, R. Chung, A. Maulik, I. Piotrowska, J. M. Walker, and D. M. Yellon, “Anthracycline chemotherapy and cardiotoxicity,” *Cardiovasc. Drugs Ther.*, vol. 31(63), pp. 63–75, 2017.

[218] R. L. Prentice, J. D. Kalbfleisch, J. A. V. Peterson, N. Flournoy, V. T. Farewell, and N. E. Breslow, “The analysis of failure times in the presence of competing risks,” *Biometrics*, vol. 34(4), pp. 541–554, 1978.

[219] D. W. Hosmer, S. Lemeshow, and S. May, *Applied Survival Analysis: Regression Modeling of Time-to-Event Data.* John Wiley & Sons, Inc, 2008.

[220] T. W. Liao, “Clustering of time series data–a survey,” *Pattern Recognition*, vol. 38(11), pp. 1857–1874, November 2005.

[221] M. LeBlanc and J. Crowley, "Survival trees by goodness of split," *Journal of the American Statistical Association*, vol. 88(422), pp. 457–467, June 1993.

[222] T. M. Osán, D. G. Bussandri, and P. W. Lamberti, "Monoparametric family of metrics derived from classical jensen-shannon divergence," *Physica A*, vol. 495, pp. 336–344, 2018.

[223] C. Bauckhage, "Computing the kullback-leibler divergence between two weibull distributions," *arXiv preprint arXiv:1310.3713*, 2013.

[224] R. Lletí, M. C. Ortiz, L. A. Sarabia, and M. S. Sánchez, "Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes," *Analytica Chimica Acta*, vol. 515(1), pp. 87–100, July 2004.

[225] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2020," *CA Cancer J. Clin*, vol. 70(1), pp. 7–30, February 2020.

[226] H. Kim, D. H. Choi, W. Park, S. J. Huh, S. J. Nam, J. E. Lee, J. S. Ahn, and Y.-H. Im, "Prognostic factors for survivals from first relapse in breast cancer patients: analysis of deceased patients," *Radiat Oncol J.*, vol. 31(4), p. 222–227, 2013.

[227] A. M. Chiu, M. Mitra, L. Boymoushakian, and H. A. Coller, "Integrative analysis of the inter-tumoral heterogeneity of triple-negative breast cancer," *Scientific Reports*, vol. 8, 2018.

[228] W. D. Foulkes, I. E. Smith, and J. Reis-Filho, "Triple-negative breast cancer," *N. Engl. J. Med.*, vol. 363, p. 1938–1948, 2010.