Learning Individual Readmission-Free Survival Distributions using Longitudinal Medical Events

by

Sarah (Sacha) Maren Davis

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

The rate of 30-day hospital readmission is a common measurement of hospital quality, which can affect the funding a hospital receives. Over a quarter of readmissions are estimated to be preventable with adequate interventions, but these interventions are themselves costly. For this reason, many projects have attempted to determine which individuals are at a high risk of readmission, and thus whose prognosis may improve with further testing and treatment. There are two common approaches to this prediction problem. (1) Formulate risk indices, such as the LACE score. These are common in a hospital setting; however, the simplicity often leads to poor predictive performance. (2) Use machine learning to transform a set of hand-selected features into the probability of readmission at a single future time-point. Unfortunately, feature engineering is time-consuming, and a physician may care about predictions at time points other than 30 days. Both approaches rely heavily on domain knowledge.

In this thesis, I use Neural Multi-Task Logistic Regression (N-MTLR) to model all-cause readmission-free survival as a function of time. N-MTLR, despite producing probability predictions for all future time-points, out-performs XGBoost and Deep Learning approaches trained specifically to predict readmissions at 30 days (AUROC 0.821 ± 0.004 (Std.Dev) versus 0.814 ± 0.003 and 0.810 ± 0.005). Further, I show that N-MTLR, augmented with a sequence model, can learn a patient's representation directly from their history of medical codes, predicting 30-day all-cause readmission with an AUROC of 0.846 ± 0.003 using only sequences of administrative medical codes as input. This approach significantly outperforms the LACE baseline of 0.659 ± 0.001. These results demonstrate the merit of medical code sequences to represent a patient's past, and N-MLTR to model a patient's future.

Preface

The work detailed in Chapter 3 and parts of Chapter 4 of this thesis was published in BMC Health Services Research in November of 2022[2], barring minor changes in target definition, study population, and feature calculation. My collaborators are responsible for the initial idea of detecting Albertan 30-day hospital readmissions using features informed by a linked administrative data Medical Concept Embedding dictionary. J. Zhang initially hand-engineered the **Detailed** feature set in `R`, which I took inspiration from for my own repository written in Python. Study design involving deep learning, sequence and before-index features, and individual survival distributions is my own. All writing (except for the first paragraph of 2.4, part of 3.1.1, and a few sentences in the discussion) and figures are my own. All code used to generate results reported in this dissertation is my own, or is cited from the internet as appropriate.

This study and all associated protocols were approved by the Health Research Ethics Board of the University of Alberta (Study ID Pro00082041). Research was conducted in a manner adhering to all relevant guidelines and regulations. Informed consent was waived by the Health Research Ethics Board Health Panel (University of Alberta), as the data were de-identified prior to access.

Don't fret over pennies.

Acknowledgements

Words cannot express how grateful I am to my supervisor, Dr. Russell Greiner, for the mentorship during (and leading up to) my master's degree. My life trajectory since 2018 would have looked very different without his priceless insights, infectious enthusiasm, and belief in me as a student.

Beyond being a brilliant researcher, Russ has an unquestionable ability to cultivate a team of people who are somehow just as kind as they are smart. A thank you is in order for everyone from the Greiner lab who I have had the pleasure of talking shop with over the last three years. In particular, Weijie Sun deserves an acknowledgement for answering so many of my questions, and for doing everything I've been doing, but slightly earlier and slightly better. Additionally, Yousef Nademi provided many valuable comments and edits on the first (much longer) draft of this dissertation.

A further thank-you goes out to all my co-authors and collaborators from my time here: Luke Kumar, Amira Aissiou, Neeraj Kumar, Ruchika Verma, Michael Parkes, Daniel Skubleny, Ilbin Lee, Mostafa Rezaei, Jin Zhang, Shazan Jabbar, and David Konkin. Ilbin Lee is the reason this project (and the BMC HSR paper) happened; I am so thankful for all the hours he spent grappling with difficult problems and holding everything together. I would also like to thank Sandra Dewan from IST for getting my data to where it needed to be: no number of edible arrangements would adequately express my gratitude for her ability to pull a solution out of a tangled mess.

I would be remiss not to mention everyone I came to know through the Artificial Intelligence in Medicine Student Society (now the Artificial Intelligence in Medical

Systems Society), even though my involvement with the club probably made my degree take 50% longer than it would have otherwise. Thank you especially to Ehsan Misaghi for our time together as co-leads—those conferences weren't going to plan themselves, and all the challenges we took on together made me a more experienced and confident person.

Thank you to my friends who worked alongside me at the Amii lab: to Sheila Schoepp, in particular, for making the CSC building such a fun and welcoming place to be. The folks at the Alberta Machine Intelligence Institute also deserve a thank-you for their unwavering efforts to create and maintain a sense of community among AI researchers at the University of Alberta. I'd like to thank Alex Kearney for being so effortlessly cool, and dragging me out to my first department social events when I was a newly-in-person master's student. In fact, to everyone who I've gotten close to over an Amii beer or two—or more—on a Friday: thank you for the laughs, the spirited discussions, and for consistently keeping life interesting. Thank you especially to Rohini Das, Justin Stevens, Erfan Miahi, Maliha Sultana, Aidan Bush, Alex Lewandowski, and Andrew Jacobson for really making my time (both inside and outside of school) memorable. And of course, to my dear Revan: nothing, and nobody, could have made this experience more special than you have. Thank you sincerely for all the love and support—and for helping me realize that I needed to pivot my thesis project last year.

To my wonderful high-school and undergrad friends who kept my sanity in check (more-or-less) while I was off in my computing science world, thank you.

Finally, my family has truly had my back through all the ups and down over the last three years—of which there were **many**. My parents, Vince and Cindy Davis, always pushed me to be the best I could, and I hope this makes them proud.

Table of Contents

List of Tables

List of Figures

Abbreviations

AHS Alberta Health Services.

AI Artificial Intelligence.

DNN Deep Neural Network.

ED Emergency Department.

EHR Electronic Health Record.

EMR Electronic Medical Record.

HSP Health Service Provider.

ISD Individual Survival Distribution.

KM Kaplan Meier.

ML Machine Learning.

N-MTLR Neural Multi-Task Logistic Regression.

NLP Natural Language Processing.

PH Proportional Hazards.

Chapter 1

Introduction

1.1 Motivation

Far outreaching its roots in religion and magic, medicine represents an intricately interconnected system of knowledge, data, and practice that aims to keep a population healthy. This discipline has long been the subject of multiple competing and conspiring demands. Resource availability, pressures to scale, and capacity for strategic future planning have all affected its trajectory.

The history of medicine can helpfully be viewed as analogous to the establishment of a major city. Around the turn of the second century, the Romans founded Barcino, a settlement in what is now the Catalina region of northeast Spain. Sprawling roads were built in all directions as the population soared over the next two millennia, areas were developed then repurposed many times over, control of the territory was disputed and changed, and a large encircling defensible wall was erected. Because of this continued growth, the town was suffocating under its own weight by the time of the industrial revolution. Still confined by its medieval walls, Frankenstein-esque archways and scaffolds atop buildings and roadways were constructed to support more lodgings. With each passing year, the consequences of uncompromising and future-blind over-development was felt increasingly by the citizens.

In the 19th century, during a time of fascination with urban renewal, a major intervention was proposed and implemented: demolishing much of the wall and organizing new neighbourhoods into a brilliantly replicable pattern of major and minor city blocks. This imposition of centralized city planning changed the fingerprint of the city in a major way; intentional organization led to increased capacity, utility, and scalability. What we see as the result of this history is modern-day Barcelona (Figure 1.1), with its striking *L'Eixample* (and surrounding) districts.

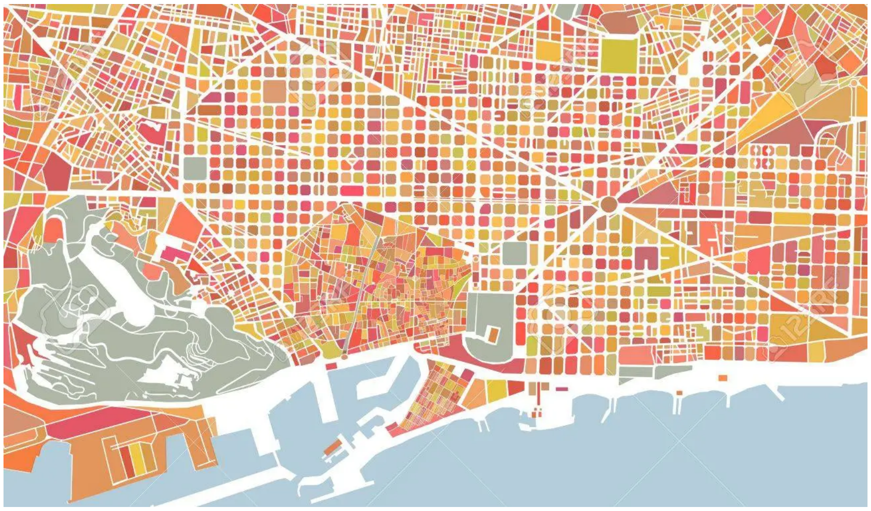


Figure 1.1: **Twenty-first century overhead view of Barcelona, Spain**. Credit: Shutterstock.

Up until a recent stage of modernity, medicine predominantly evolved organically, similar to Old Barcelona, alongside the needs of humanity. Just as the development of Old Barcelona was the result of pressures to support a growing population **presently**, so too has medicine developed for purposes **other than providing the optimal care to some hypothetical future population**. The mass amounts of data captured—prescription histories, insurance claims, laboratory test results, medical imaging, free-form text cataloguing patient-physician interactions—are to ensure medical care offerings meet a certain minimum standard, and to ensure that medical professionals are compensated for their hours. Much like with our developing city example, medical care provision is a patchwork of quick-fixes and approaches that maximize, perhaps short-sightedly, the effectiveness-to-cost ratio. Many artifacts of this reactive dynamic exist in our medical datasets, such as the often-sub-optimal data quality and heavy siloing. It can be said with certainty that paving the way to improve medicine through Machine Learning (ML) has not been at the forefront of medical administrators' minds throughout most of history.

We are currently watching medicine undergo a rapid transformation with the introduction of reliable, digitized, standardized, and readily-available health record keeping. An example is the adoption of Alberta's ConnectCare[1]. One day, we will look back at this era as medicine's own *L'Eixample* (literally, *"The Expansion"*); this is a crucial step towards the modernization of health. However, this reconstitution takes time, and current medical systems are under significant stress *now*. Furthermore, even if issues of current electronic health record robustness and integration are ignored, the longitudinal nature of electronic health data makes its pairing with traditional supervised learning techniques awkward at best. Thus, if we are going to practically improve predicting clinical outcomes (such as hospital readmissions) through the use of ML in a timely fashion, exploring ways around constraints of current and future health data is an imperative.

Unfortunately, our efforts are moot if we do not set out to solve the right problems. The application of Artificial Intelligence (AI) has rapidly gained traction in medical literature since the mid-2000s—yet, the adoption of AI in practical medical contexts is dazzlingly scarce. Using education to reduce wariness of new technologies and canning solutions into user-friendly software may help, but does not paint the complete picture. The architect behind Barcelona's *Eixample*, Ildefons Cerdà, who has been praised for his meticulous attention to detail (and for his founding contributions to the study of urbanization) would have understood this. During the design process, he investigated how the working classes lived, mapped optimal distances to amenities, and studied important associations between factors like street width and disease. The result was a plan that not only supported the growth of the city, but enhanced community well-being. To make medical AI tools whose benefits outweigh

[1] https://www.albertahealthservices.ca/cis/cis.aspx

the risks, computing scientists and developers must, throughout the entire lifecycle of an AI system, cater intentionally and deliberately to specific needs of the medical end-user.

Thus, for those who wish to contribute to the future of medical AI, there exists the need to shift our philosophy in (at minimum) two ways. First, we must focus on cleverly circumventing the current limitations of our current and future data; and second, we must practice intentionality whilst choosing which problems to solve and how. From here, we can truly begin our journey.

1.2 Thesis Objectives

This dissertation contains two contributions to the larger body of medical AI knowledge.

1. To effectively use the wealth of historic longitudinal medical data available to us, we adopt techniques that allow us to embed knowledge about healthcare usage as a dictionary of dense vectors. We show that embedding-informed medical history representations are useful for predicting 30-day all-cause hospital readmissions, and in the more general task of predicting time-to-all-cause-hospital-readmissions. These outcomes reinforce the idea that adapting medical data to suit the fortuitously analogous context of Natural Language Processing can be a viable strategy for extracting clinical insights agnostic to disease cohort.

2. Instead of predicting the probability of readmission at a single future timepoint, we use models that generate individual readmission-free survival distributions (i.e., the probability of survival at all future time points) directly from the sequence of a patient's past medical codes. The combination of learning from the progression of a patient's medical history to model a patient's prognosis in the future achieves impressive performance on the 30-day all-cause readmission prediction task, captures much of the information represented in

hand-engineered features automatically, and allows us to ask questions about the time-to-readmissions for each patient.

1.3 Thesis Outline

We begin with a review of pertinent literature in Chapter 2. Next follows the exploration of ways to represent medical histories (Chapter 3) and validating these representations in predictive tasks (Chapter 4). We close with a discussion of the project's implications in Chapter 5, and conclusions and future work in Chapter 6.

Chapter 2

Context and Literature Review

2.1 The State of Affairs

An Electronic Health Record (EHR) contains widely-accessible information about the holistic health of an individual from many different sources [3]. In Alberta, Canada, this takes the form of Alberta Netcare[1], maintained by Alberta Health Services (AHS). As an individual moves amid Health Service Providers (HSPs), information is uploaded to a centralized and secure storage system creating a trail of rich medical information, covering medication history, lab test results, diagnostic images and reports, hospital visits, and more. "Administrative data" is a subcategory of health data captured within and beyond the Netcare EHR, and includes (but is not limited to) medical expense claims, hospital in-patient episodes, ambulatory care provision, medication dispensation events, and patient demographic information.

While Netcare and other data sharing endeavors have increased the speed and efficiency with which care can be provided, "How can these integrated data be used to accurately predict health outcomes?" is a question of interest for both researchers and doctors. Understandably, many different health data schemes and structures create problems that impede using these data to the fullest extent. The prevalence of "missing values" is one barrier. Phung et al. [4] categorize types of missing data into two categories: **systematic** (due to changing data capture protocols) and **non-**

[1] https://www.albertanetcare.ca/

deterministic (due to times where individuals fail to provide the requested data). Indeed, examples of these types of missing data are abound in Alberta Netcare. For example, the *numeric result* and *abnormal status* of many lab tests are not captured in their designated structured field—instead, the field may read "see doctor's note", which contains test findings in a wall of unstructured text. Further, values for important variables (e.g., demographic features like sex) occasionally go uncaptured if a patient arrives at the hospital in poor shape and/or lacking in proper identification documents.

However, null and misplaced values do not paint the complete picture of where we truly struggle with the lack of data in EHRs. Imagine you would like to examine lab test results, or a patient's historical medical codes, to predict clinical outcomes. Thousands of lab tests are available for order in Alberta hospitals, and more than 35,000 different medical codes were used in Alberta Netcare in the 2010s. Vectors containing test results or medical events for a hospital visit would be high-dimensional and rich in zeroes. Further, oversight in high-level data governance can exacerbate this sparsity issue, as exemplified in Alberta by the reliance on multiple versions of the International Classification of Diseases (ICD) coding system for different administrative data sources. A comprehensive mapping between these versions (ICD-9 and ICD-10-CM) is not straightforward [5], which makes standardization of linked datasets burdensome. It is true that many models of interest to clinicians (e.g., survival models) are often intolerant to sparse representations of multicollinear features, which are prevalent in medical data [6]. So, what are we to do? Since defining a tabulation of medical data representing historical information is not straightforward, the new question becomes "How can these integrated data be **represented** to accurately predict health outcomes?"

One way to represent historical medical information in tabular form is through **hand-engineering**, which refers to the process an analyst performs, informed by domain knowledge, to calculate the value of some feature from a larger collection of data. Using hand-engineered features in simple and practical heuristics in medicine far predates hand-engineered features as inputs to ML models. The LACE index [7] is one example of this practice. LACE attempts to rank a patient's risk of hospital readmission by capturing information about the **L**ength of the current hospital stay, if the current episode started in the emergency room (**A**cuity of admission), the presence of **C**omorbidities (Charlson Comorbidity Index, or CCI [8]), and the patient's **E**mergency department usage over the last six months. However, the simplicity of this approach often leads to lackluster performance compared to more recent methods. Further, some information needed at time-of-calculation is only available at discharge, whereas planning for targeted interventions starts at hospital admission [9]. Alberta Health Services does make use of the LACE index (as per the ConnectCare manual[2]), which makes it a reasonable baseline against which to compare more sophisticated readmission-prediction models (Section 3.3.1, 4.5.1). Extensions of the LACE score (such as LACE+) improve upon the original in AUROC performance [10]; however, some score components such as case mix group (CMG) codes—which the authors themselves admit are "computationally expensive"—are not practical or straightforward to calculate with the data we have. Other notable readmission-related heuristics like the HOSPITAL [11] and B-PREPARED [12] scores (which require lab test results or the administration of a self-report respectively) also show moderate performance and are not applicable universally. Au et al. [13] note that for predicting 30-day readmission due to heart failure, LACE, LACE+, and CCI scores only led to an AUROC of between 0.57 and 0.61.

Simple heuristics for predicting outcomes (where the contributions of different features towards the final score are prescribed rather than learned) most often involve only a small number of features. This is likely because manually defining the weightings of each of these factors becomes increasingly difficult to validate with more complex inputs. Machine learning has enabled the automatic learning of these weights to

[2] https://manual.connect-care.ca/workflows/patient-movement/transition-planning

model a future adverse outcome. Covariates designated as "highly predictive" for a particular task are often published as *risk factors* in much of recent literature. For example, Philbin et al. (1999) use logistic regression to predict congestive heart failure, and derive a risk scoring system based on factors related to an admission, such as black race, Medicare and Medicaid insurance, ischemic heart disease, idiopathic cardiomyopathy, prior cardiac surgery, peripheral vascular disease, and others [14]. Many risk factors for 30-day readmissions (all-cause and otherwise) have been identified this way from different research groups, including hospital-acquired Clostridium difficile infection [15], cancer, pulmonary, liver, and kidney disease [16], residence out-of-area, major or minor lack of procedure applied during index hospitalization, hypertension [17], and maintenance chemotherapy, Gabapentin ordered at index admission, and $\geq$16 abnormal laboratory test results [18]—alongside many others. Engineering attributes (like those mentioned above) from medical history data often require the use of validated algorithms to calculate (e.g., when identifying the presence of chronic conditions [19]). This is a time-consuming process and does not scale well with an increasing number of features.

Further, most of the features mentioned previously only capture atemporal (demographic or chronic condition-related) features, or features that only concern the current hospitalization. Adequately capturing more of the past may prove advantageous for tasks that concern the future of a patient. Medical codes (indicating diagnoses, procedures, prescriptions, and beyond) are used universally in health centres, and contain important information about an individual's health history. Glossaries of medical codes in hospitals can number in the tens of thousands; if one were to represent a patient as a multi-hot or count-based vector of medical codes, the resulting vector would be highly sparse. In 2015, Yerex et al. used various feature extraction techniques from this sparse representation to decrease the dimension of the feature space for readmission prediction [20], and Sideris et al. [21] use a similar technique for predicting heart failure severity. Grzyb et al. tried using hash maps to decrease the feature space [22] . However, the choice to represent medical medical events in this multi-hot way discards any sense of order or temporality—*when* the codes are attributed could be as important as the codes themselves. Some authors address this by splitting the health history of an individual into multiple time-bins, where the medical events in each bin are represented using a multi-hot feature vector [23]. In this example, Boltzmann machines were used to embed the medical events from within these time-bins to predict suicide risk prediction, but this technique is not appropriate when the observation window for each individual is not the same length. Others apply medical event standardization procedures to represent histories before applying machine learning, e.g., using the Fast Healthcare Interoperability Resources (FHIR) format [24]. While effective, the flexibility of FHIR is restricted due to its proprietary nature. This motivates the exploration of open-source algorithms that can compute patient representations (that can incorporate the sequence of medical events in a patient's history) to predict adverse medical outcomes (Figure 2.1). The adverse medical outcome we focus on in this dissertation is the all-cause hospital readmission (Section 4.1.1).

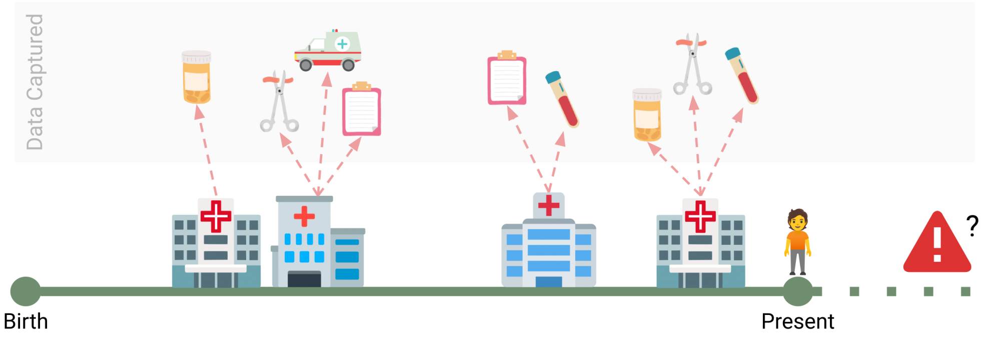


Figure 2.1: **How can we use the health of historical health data we have to gain insight into the future of a patient?**

2.2 Natural Language Processing and Predicting Health Outcomes

Much of humanity's knowledge is stored in natural language. Computationally parsing, distilling, and understanding this inherently hierarchical information promises many benefits. The base unit of meaning is a **token**, which is often viewed as a word or grouping of characters. Many tokens make up a **sentence**, and many sentences make up a **document**. Many documents comprise a **corpus**. Idiosyncrasies at any of these levels can cause difficulties with representation. Vocabularies of tokens are complex, high-dimensional, and lead to sparse representations—words, sentences, and documents come in varying lengths, and context and higher-order structures can drastically change meaning. Natural Language Processing (NLP) refers to a suite of algorithms designed specifically for handing these characteristics. Supervised and semi/self-supervised NLP models often circumvent the aforementioned issues by learning low-dimensional embeddings of words and combining them, informedly, into sentence or document representations.

NLP is widely applicable to medical problems due to its power to represent information. While it can characterize factors that influence care from cancer support discussion forums [25] and write published medical literature [26], most relevant to this project is the use of NLP to predict downstream clinical outcomes. For example, Huang et al. used BERT to embed free-form clinical notes, then fine tuned the architecture to predict 30-day hospital readmissions. This model attained an AUROC score of 0.714 [27]. Golas et al. show that adding information from clinical notes can significantly improve 30-day heart failure readmission prediction [28]. While promising, free-form clinical text is noisy, sometimes unavailable due to its potentially identifying nature, and was not included alongside our data extraction for this project.

To employ NLP in the absence of free-form clinical text, the structural similarities between natural language and sequences of medical codes can be exploited. A previously-mentioned example is the work of Rajkomar et al. [24], who apply the FHIR format to represent historical medical codes, and use these representations as input for LSTM (Long Short Term Memory) models and TANNs (Time-Aware Neural Networks). This procedure performs notably well on many tasks, including predicting in-hospital mortality (AUROC 0.93–0.94), 30-day unplanned readmission (AUROC 0.75–0.76), and prolonged length of hospital stay (AUROC 0.85–0.86). Pham et al. introduced DeepCare [29] for modeling disease progression and predicting diabetes-related readmissions, also relying on longitudinal medical event data fed into LSTMs. Doctor AI (from E. Choi et al., 2016 [30]) uses Gated Recurrent Unit neural networks [31] (GRU networks) to mimic how a physician diagnoses patients—by predicting, at the end of each hospital episode, *all* diagnosis and prescription codes in the subsequent visit. Suo and colleagues augmented GRU neural networks with an attention mechanism to monitor disease progression via the diagnosis results of previous records [32]. GRUs have been further utilized by Chakraborty in 2021 [33] (who note that GRU-based models far out-perform simpler models on the task of generating readmission risk scores from sequential Claims data), and by E. Choi et al. [34] [35]); the authors show in both publications that incorporating temporally-sensitive features improves upon heart failure-specific readmission prediction. Given the previous successes of the GRU network and medical code sequence combination (and the computational efficiency of GRU networks compared to other models like LSTMs), GRU networks are our sequence model of choice to represent patient histories for predicting readmission-related outcomes in this project (Sections 4.3.3 and 4.4.2).

This “patient medical code history = natural language document” analogy can also be used to represent medical concepts, codes, or events in vector form. Most projects do this by making use of the “distributional hypothesis” to capture medical meaning—that is, words (or medical codes) that appear in similar settings (are used to describe a patient close in time to one another) have similar meanings. Herein lie all the projects relying on various implementations of the Word2Vec embedding architecture, originally introduced in 2013 by Mikolov et al. [36]. Y. Choi et al. first proposed the use of Word2Vec to generate vector embeddings of medical codes from Claims data, and demonstrate how related medical concepts appear meaningfully similar in vector space [37] using this technique. Even though this idea was proposed in 2016, these "**Medical Concept Embeddings**" (MCEs) have remained of interest to researchers in recent years. For example, in 2022, Wang et al. demonstrate that manifold learning shows promise in refining Word2Vec-generated medical concept embeddings [38].

MCEs, and the medical meaning captured therein, can provide rich information for machine learning models, increasing downstream task performance. In the same year as Y. Choi proposed the medical concept embedding [37], E. Choi and coauthors proposed the same idea[3], and demonstrated its usefulness on the downstream binary classification task of (heart-failure-related) 30-day hospital readmission prediction. E. Choi's idea is simple—to represent a patient, extract embeddings corresponding to some number of recent medical codes, perform a summation on these vectors, and use the resulting vector to represent a patient's medical history. This simple representation, combined with a Deep Neural Network, achieved an AUROC higher than 0.8, which is high compared to numbers reported for the same task in other studies. An extension of this approach, Med2Vec (also from E. Choi [39]) simultaneously learns representations of both medical codes and entire hospital visits. While Med2Vec visit-level embeddings out-perform the summation of Word2Vec medical concept embeddings (on the task of predicting future medical codes), this multi-level architecture relies on clear boundaries between medical "episodes" in the data, which is not always realistic, and indeed is not straightforward with Alberta Netcare data. Creating Medical Concept Embeddings is not restricted to Word2Vec—for example,

[3] You can imagine how confusing this was.

RNN and topic modeling approaches were hybridized by Xiao et al. to jointly learn event and patient representations (similar to Med2Vec) for predicting readmissions for congestive heart failure patients. However, the AUROC score did not exceed 0.62 [40]. CEHR-BERT [41] incorporates temporal information from EHR data into event and patient-level representations, once again to only moderate success in the task of 30-day heart failure readmission. Some have attempted to use generative language models (GLMs) to represent patients based on sequences of historical medical codes. However, GLM-based representations did not decidedly out-perform the use of summed Word2Vec medical code embeddings [42] on the task of predicting 30-day all-cause hospital readmission. For these reasons, we use the original procedure from E. Choi: create a MCE dictionary using Word2Vec and sum recent medical code embeddings to create a representation (Sections 3.2.2, 3.3.5).

While not strictly borrowing from natural language processing, multiple projects make use of Convolutional Neural Network (CNN) filters to "sweep" across events in medical histories to learn higher-level, sequence-preserving, increasingly-abstract patient representations. Zhang et al. [43] use this approach to predict comorbidity risk, and Cheng et al. [44] use it to predict the risk of onset of chronic conditions. Nguyen et al. propose Deepr [45], a CNN that detects clinical motifs and predicts all-cause hospital readmissions at 3 and 6 months. Deepr is initialized with Word2Vec embeddings of medical concepts, which led to improved performance; initializing GRU-RNNs with MCE dictionaries can similarly lead to higher performance at evaluation [35].

In all these publications, the problem is formulated as a classification task—often predicting a 1 if the event of interest happens within a certain time-frame, and 0 otherwise. This is limiting in two key ways. One, an event's presence or absence at a particular time-point is only a coarse level of information, and may not paint a complete picture of a patient's prognosis. Two, restricting a prediction to a particular time-point (e.g., 30-days) requires the (perhaps arbitrary) definition of a time-point, when the event predictions at other time points may also be of medical interest. These sacrifices are not necessary; classes of algorithms exist that can make use of timing information and can make predictions at many/all time-points.

2.3 Event-Free Survival Prediction

The information stored in the binary target label of “30-day readmission” is quite coarse; there is no measure of gradation to distinguish those who were readmitted in one week vs. one month, nor those who were readmitted in 31 days vs. never readmitted within the study period. If we instead consider some measure of time-until-event as the target, the model can learn a more fine-grained relationship between the input features and the adverse outcome of interest. However, if we would like to consider this temporal information, a problem arises. A model would ideally capture information about events at differing future timepoints with approximately equal robustness; however, some patients may move away, or they may never experience an adverse outcome during the study period. Any event that disconnects us from measuring our adverse outcome of interest is called a “censorship” (Figure 2.2). Garmedia et al. (and others) choose a naive approach to predict time until emergency-room readmission, opting to remove all censored patients and treat it as a regression problem [46]—however, this approach causes the model to chronically underestimate survival time [47] which compromises overall utility.

Survival analysis and survival prediction models are used to understand how the future of a “system” (e.g., patient, machine, business) may unfold with respect to some event of interest (e.g., death, mechanical failure, bankruptcy) [47], and are formulated specifically to deal with these censoring events. The fact that the numeric target of interest for survival analysis problems is often partially obscured in the training dataset differentiates it from a classic regression problem. Survival models learn from a **survival dataset**, where every system is associated with both a “time-to-event” and a bit denoting if the event indicates true “survival time” or censorship. Haider et al. [48] suggest three axes on which a survival prediction model may be

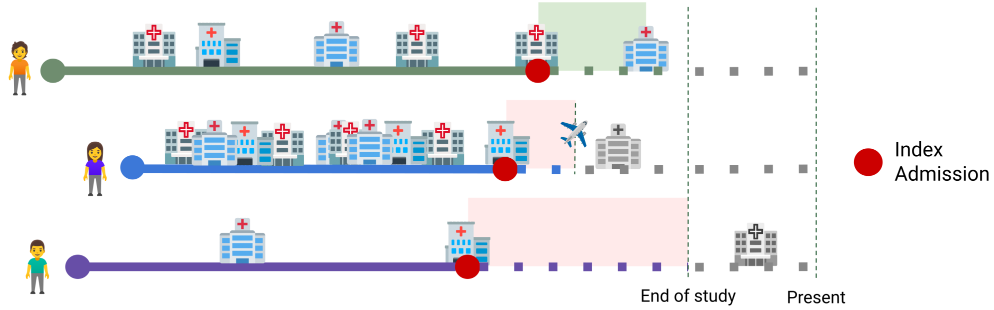


Figure 2.2: **Censorship illustration with three example patients**. Patient 1 is uncensored, as we understand their time-until-event. Patients 2 and 3 are both censored, as Patient 2 moved away before experiencing an event, and Patient 3 remained event-free until the end of the study. Some patients may never experience an adverse outcome.

classified; whether the model's predicted output is 1) a risk score (R) vs. probability of event (P), 2) a single value (1_{t^*}, $1_{\forall}$) vs. a range of values over future timepoints (∞), and 3) applied on the individual level (i) vs. to a cohort or population (g).

The Cox Proportional Hazards (CoxPH) model [49] (discussed further in Section 4.4.1) falls into the category $[R, 1_{\forall}, i]$, generating a single, individualized risk score. "Proportional hazards" refers to the assumption that a covariate's effect on survival does not change with the passage of time. Given the model's simplicity, it is used often in medical survival analysis. In 2013, McAlister et al. [50] used CoxPH trained on time-dependent covariates to understand the risk of readmission or death after a heart failure event using Alberta Health Services data. Mixon et al.(2016) attempt to understand how discharge "preparedness" influences 30 and 90-day readmission risk using CoxPH [51], and similarly, Glasgow et al. use CoxPH to understand how leaving against medical advice impacts readmission risk [52]. Grzyb and coauthors propose a multi-task CoxPH model for predicting risk of 30-day unplanned readmission and binary 30-day unplanned readmission [22], but performance was lackluster.

While risk scores generated by CoxPH (and others algorithms, such as Random Survival Forests [53]) are commonplace for understanding the effect of different factors on survival, modeling survival probabilities (especially over multiple timepoints) is more useful for understanding the progressions of individual patients in a standalone way. The Gail model [54] and PredictDepression [55] both use a time-to-event dataset to predict the probability of event-free survival ("event" being breast cancer onset and depressive episode onset, respectively) at a particular time-point. These algorithms do not provide information for times other than the chosen t_*, and would be classified as $[P, 1_{t_*}, i]$. The Kaplan-Meier estimator [56] ($[P, \infty, g]$) models event-free survival as a function of time for a population or sub-population of individuals, but is more useful for comparing treatment or cohort effects, and neglects individual-level survival information. The aforementioned limitations motivate the need for survival-prediction systems that can model the individualized probability of event-free survival at arbitrary future time-points—i.e., models that generate **Individualized Survival Distributions** (ISDs). The CoxPH risk score can be combined with a baseline survival function to generate an individual survival distribution, such as with the Kalbfleisch-Prentice [57] extension or using Breslow's Estimator [58]. Multi-Task Logistic Regression (MTLR) [59] is an algorithm specifically designed to use a survival dataset to generate an Individual Survival Distributions, either alone or combined with Deep Neural Networks (DNNs) [60]. MTLR trains a sequence of connected logistic regression modules which each predict the probability of event-free survival within a particular future time range. This approach has been used to model length-of-hospital-stay for COVID-19 patients [61], time until readmission for COVID-19 patients using ECG inputs [62], cardiovascular-related hospitalizations for hypertensive individuals [63], breast cancer onset [64], Alzheimer's disease progression [65], and many other classes of medical events. We employ both CoxPH with Breslow's estimator and MTLR to model all-cause-readmission-free survival.

Very few publications have harnessed both NLP-powered medical concept embeddings and algorithms that model individual time-to-event probability distributions. In 2022, Kalmady and colleagues used Med2Vec representations of medical events and MTLR to model ISDs for heart-failure-specific hospital readmissions [66]. However, the Med2Vec hospital visit embeddings used to predict readmissions only contained information from the index admission, ignoring the events of an individual's previous medical history. To the best of my knowledge, nobody has explored the combination of ISD-generating algorithms and sequence models to represent medical histories (with and without the addition of contextual medical concept embeddings).

2.4 Practical Considerations for Readmission Prediction

As of the early 2010s, hospital readmissions cost approximately 2 billion Canadian dollars per year in Canada [67] and 26 billion US dollars per year in the United States [68]. In the US, the Centers for Medicare & Medicaid Services financially penalize hospitals with high 30-day readmission rates [69], making a reliable 30-day readmission prediction system highly sought after in medical AI literature. Studies estimate that anywhere from 10-60% of these readmissions are avoidable [70] [11] [71]. Predicting the readmission risk of individual patients alongside a hospital stay could help better target expensive transitional care interventions, which may save money and shed light on the complex factors associated with re-hospitalization events. Even with the most conservative estimates of readmission preventability, the potential savings from better anticipating and targeting hospital readmissions (in both dollars and suffering) are massive. Despite these promised benefits, and the continually ballooning body of medical AI literature [72], very few of these systems are implemented in practice [73]. In this section, I explore and coalesce some of the factors that may be deepening the divide between what we know about predicting readmissions versus what we can use.

Perhaps the problem lies in the way we define our target populations. In a review of the literature for hospital readmission risk prediction up until 2019, only 17 out of 41 eligible studies were all-cause—the remaining were specific to disease cohort, with most being about heart failure [9]. Although rates of hospitalization for heart failure are increasing in the US, these admissions still only account for approximately 0.5% of all hospital visits [74]. If a heart-failure-specific readmission model were to be integrated with an EHR, its output would not be validated on (or even apply to) roughly 995 out of every 1000 hospital discharges[4]. In a general hospital setting, an increasing number of cohort-specific models would be necessary to cover greater fractions of admissions; this approach would be costly and eventually plateau in practicality. For this reason, we choose to predict all-cause readmissions (with as few cohort-related caveats[5] as possible) to ensure maximal model coverage.

The data used as model input (and therefore **when** the readmission prediction can be made), may also compromise model utility. The majority of studies use information from the duration of the index hospitalization [9], despite the fact that strategizing about treatment courses and discharge often begins at the first day of a hospital admission [75]. Predicting readmissions at the time of admission is widely considered to be a more difficult problem [18], which many studies [18] [76] [77] attempt to tackle for various cohorts; however, few compare them to their full length-of-stay counterpart models. One exception is Nguyen et al. [15], who show that adding features from the hospital stay only moderately increased readmission prediction performance (AUROC of 0.69 vs 0.64-0.67). In our study, alongside full-stay models, we examine versions of our proposed systems that use only information **at** or **prior to** admission.

Choosing models wisely (and involving a diversity of opinions surrounding these choices) may increase downstream model usability. Simple models—even if inaccurate—are the ones most often deployed [24], as the demand for model interpretability is prevalent in the medical field [78]. We propose simple versions of our models (both in model complexity and feature set used), an added benefit being the lower computational cost associated with training and evaluation. Further, many papers reviewed were published in association with a quantitative department (e.g., computing

[4] These systems may still be useful at a heart-condition-specific ward or clinic.

[5] See Section 4.1.1 for our all-cause readmission definition.

science), with no co-authors in (or acknowledgements to) medical sciences experts. Cronin et al. [75] developed a real-time 30-day readmission prediction system that was implemented at the Massachusetts General Hospital, despite their model achieving an AUROC score of only 0.705 on retrospective testing data. The authors note the importance of inviting hospitalists to be part of the model development process, as this was crucial for the project's success. Two physicians (Drs. Raj Padwal and Finlay A. McAlister) contributed to the design of our lightweight 30-day readmission models and the patient representation step to ensure compatibility with the healthcare system. Another factor that may stave off model adoption is administrative decision ambiguity—hospital administrators may not wish to arbitrarily decide "this hospital needs a 30-day readmission prediction system", rather than "this hospital needs a 1-year readmission prediction system". The merits of Individual Survival Distributions—explored in this study—to model readmissions (rather than a risk score or single probability) were noted in Section 2.3.

Chapter 3

Representing Medical Histories for Readmission Prediction

Imagine your life represented as a timeline. No doubt there are markers that indicate many types of events—from meaningful, like a wedding, to something as mundane as renewing a driver's license. For most of us, a recognizable number of these markers are related to our health. First, you were born, likely in a hospital. Various vaccines and strep throat swabs were administered throughout your childhood. You fell off your bicycle in eighth grade and broke your arm, and in the same year, you were prescribed anti-dandruff shampoo and accutane. In your early twenties you had your appendix removed, which resulted in complications that were monitored closely by your general practitioner. Like it or not, we spend a non-trivial percentage of our lives in and around health centres, and an non-trivial amount of effort to avoid returning once we leave.

Healthcare data is longitudinal in nature, taking the form of episodes of events separated by variable lengths of time. The number of events can vary wildly between individuals. These irregularities (Figure 3.1), alongside imperfect data capture, cause considerable difficulty when trying to leverage this rich information for predictive machine learning. Previously, the favoured approach has been hand-engineering features to capture important details from a patient's medical history. This takes the form of whatever a physician believes to be an important factor for the task at hand; for example, the number of ER visits in the previous six months is likely related to the patient's risk of hospital readmission. However, this engineering process is labour-intensive, subjective, and highly specific to both the task and the idiosyncrasies of a hospital's data capture and storage processes. This necessitates the exploration of ways to represent a patient automatically (potentially making use of **all** historical medical events), rather than manually. In this chapter, we examine a technique that allows us to capture important medical meaning numerically, which can be used downstream to represent patient medical histories in a fashion that is a) amenable to machine learning, b) model-agnostic, and c) can take into account the ordering of (and timing between) events[1].

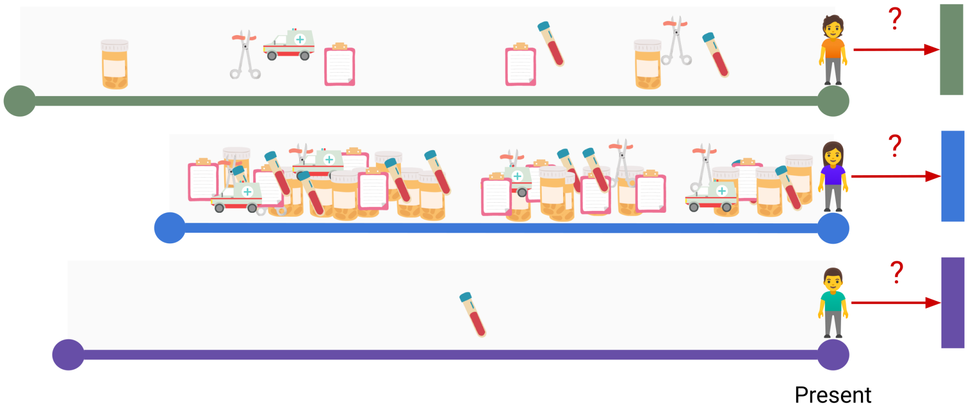


Figure 3.1: **How can we represent highly diverse medical histories in a fixed-width vector?**

3.1 Data

3.1.1 Sources and Characterization

Linked administrative health data collected within the province of Alberta were used for this study. We extracted the following information for all individuals who interacted with Alberta Health Services from years 2011 to 2017:

[1] For the purposes of this project, we define a "medical event" as the ascription of a medical code or an emergent/non-emergent admission or discharge.

1. Hospitalizations from Discharge Abstract Database (DAD), including admit and discharge dates, discharge disposition, diagnosis codes, and procedure codes

2. Ambulatory visits from National Ambulatory Care Reporting System (NACRS), including visit date, emergent status, disposition, diagnosis codes, and procedure codes

3. Physician office visits from insurance data (Claims), including visit date and procedure/diagnosis codes from primary care physicians (family medicine), internal medicine specialists, and general surgery specialists

4. Drug prescriptions from Pharmaceutical Information Network (PIN), including prescription date and Anatomical Therapeutic Chemical (ATC) code

The extracted data also included sex, age, and the first three alpha-numerics of postal code for each individual. Diagnosis codes were ICD-10-CA except those in Claims, which are ICD-9. All procedure codes are following the Canadian Classification of Health Interventions (CCI), except those in Claims, which were the Health Service Canadian Classification of Procedures Extended Codes (CCPX). All data used in this study were collected pre-ConnectCare, and were extracted and anonymized by the Alberta Strategy for Patient Oriented Research (SPOR) SUPPORT Unit.

3.1.2 Index Admissions and Study Population

We build representations for patients who were discharged from any Albertan hospital during the index period of January 1, 2015 to December 31, 2016 (Figure 3.2). We select valid index episodes (each constituting an entire hospital "event", potentially including transfers between locations), thereby determining our target population, using the following procedure (Figure 3.3A).

The set of all records from 2011 to 2017 contained 520,960 patients. Admissions from routine hospital admissions for baby births were not included in the initial data

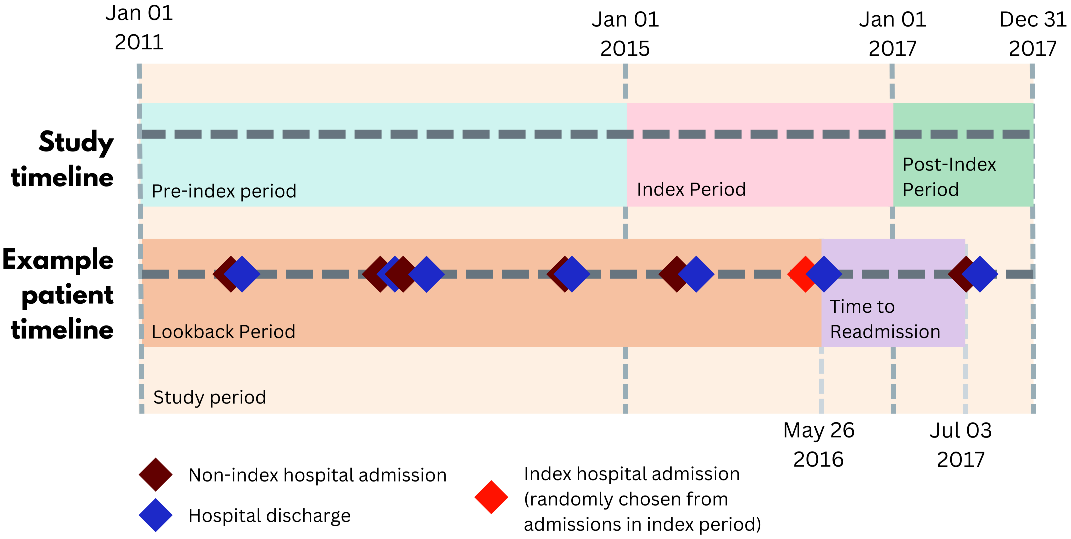


Figure 3.2: **Study and example patient timelines.** The study timeline is split into three periods: the *pre-index period* for training Medical Concept Embeddings, the *index period* from which index admissions are selected, and the *post-index* period to determine at least one year of a patient's future post-index-admission-discharge. The patient timeline is split into two periods: the *lookback period* of patient history (which includes all information until the index admission discharge) and the *time-to-readmission* (the period of readmission-free survival). Time-to-readmission is calculated from the discharge of the index admission to the day of the next admission. If time-to-readmission extends beyond the end of the study period, it becomes time-to-censorship.

extraction. DAD records whose discharge date occurred within 2015 and 2016 were selected (number of unique patients $n = 472339$). Patients who were associated with at least one record from outside of Alberta ($n = 21024$) or had an invalid patient identifier ($n = 408$) were removed. Patients with only psychiatric admissions (ICD-10-CM diagnosis of F00-F99 except F10-F19, $n = 16140$) were excluded, alongside all other psychiatric admissions, due to characteristic patterns of readmission separating them from other hospital users. Patients whose only hospitalization ended in death (discharge disposition code 07, $n = 3931$) were also excluded. Records for each of the remaining 430836 patients that were separated by no more than one day (regardless of discharge disposition) were collapsed into admission "episodes"[2]. Episodes ending

[2] See Section 4.1 for how this was later used to define our target variables.

with further transfers (discharge disposition 01), in-hospital deaths (discharge disposition 07), or a failure to return from pass (discharge disposition 12) were removed from the selection pool, leaving 421089 patients. Patients who reportedly experienced at least one post-mortality hospital admission ($n = 1$) were also removed, leaving 421088 unique patients. Throughout 2015 and 2016, the same individual may have been admitted and discharged multiple times. Our study randomly retrains one index episode per-individual. Table 3.1 contains summary statistics for each patient and associated index admission.

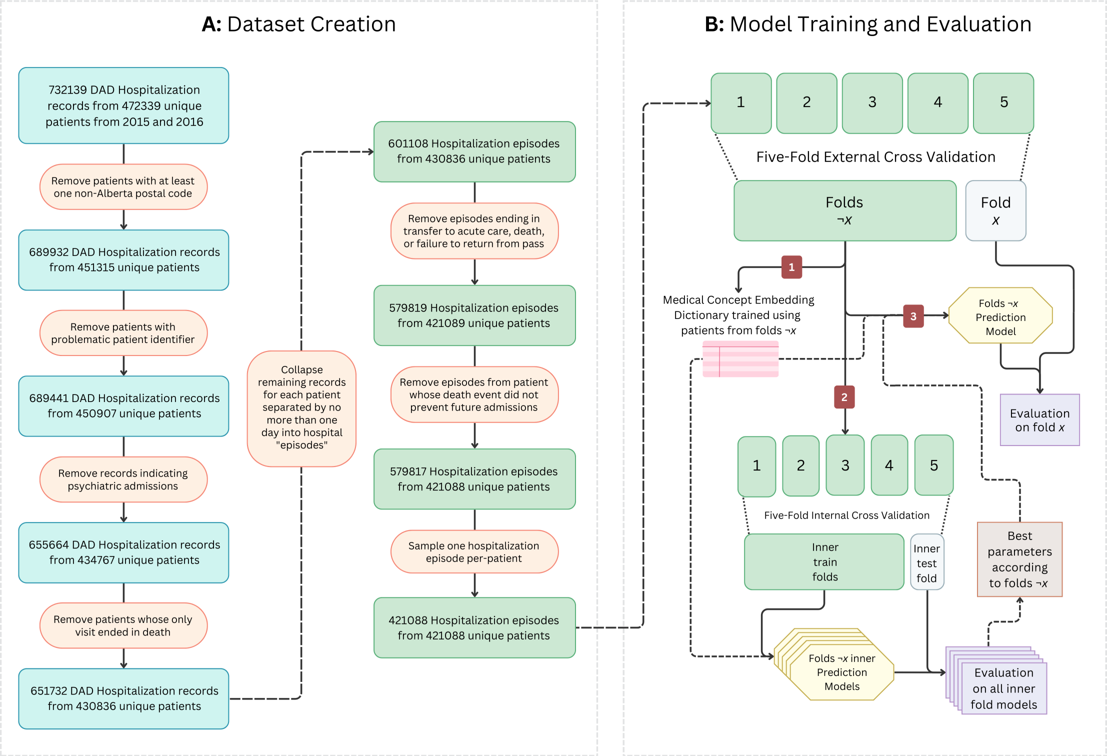

A: Dataset Creation
732139 DAD Hospitalization records from 472339 unique patients from 2015 and 2016
Remove patients with at least one non-Alberta postal code
689932 DAD Hospitalization records from 451315 unique patients
Remove patients with problematic patient identifier
689441 DAD Hospitalization records from 450907 unique patients
Remove records indicating psychiatric admissions
655664 DAD Hospitalization records from 434767 unique patients
Remove patients whose only visit ended in death
651732 DAD Hospitalization records from 430836 unique patients
Collapse remaining records for each patient separated by no more than one day into hospital "episodes"
601108 Hospitalization episodes from 430836 unique patients
Remove episodes ending in transfer to acute care, death, or failure to return from pass
579819 Hospitalization episodes from 421089 unique patients
Remove episodes from patient whose death event did not prevent future admissions
579817 Hospitalization episodes from 421088 unique patients
Sample one hospitalization episode per-patient
421088 Hospitalization episodes from 421088 unique patients
B: Model Training and Evaluation
1
2
3
4
5
Five-Fold External Cross Validation
Folds ¬x
Fold x
1
Medical Concept Embedding Dictionary trained using patients from folds ¬x
3
Folds ¬x Prediction Model
Evaluation on fold x
2
1
2
3
4
5
Five-Fold Internal Cross Validation
Inner train folds
Inner test fold
Best parameters according to folds ¬x
Folds ¬x inner Prediction Models
Evaluation on all inner fold models

Figure 3.3: **Overview of procedures used for dataset procurement, model training, and model evaluation.** (A) Procedure used to arrive at the final dataset of index admissions and study population. (B) Details of external cross-validation for evaluation and internal cross-validation for tuning.

Table 3.1: **Study population summary statistics, divided by 30-day readmission status**. Variables reported correspond to Bare features used as machine learning input (Section 3.3.2). Each row of the dataset corresponds to one hospital admission, randomly selected for each patient from candidate admissions in the index period.

	All	Not Readmitted within 30 Days	Readmitted within 30 Days
	n = 421088	n = 399716	n = 21372
	Number (%)	Number (%)	Number (%)
	or	or	or
Variable	Mean (Std.Dev)	Mean (Std.Dev)	Mean (Std.Dev)
Sex (Female)	263200 (62.5%)	251826 (63.0%)	11374 (53.2%)
Age			
<1	12541 (3.0%)	10432 (2.6%)	2109 (9.9%)
1 - 14	19824 (4.7%)	19162 (4.8%)	662 (3.1%)
15 - 24	33486 (8.0%)	32545 (8.1%)	941 (4.4%)
25 - 64	238950 (56.7%)	230346 (57.6%)	8604 (40.3%)
≥65	116287 (27.6%)	107231 (26.8%)	9056 (42.4%)
Discharge Disposition			
02: Transferred to Continuing Care	10197 (2.4%)	9534 (2.4%)	663 (3.1%)
03: Transferred to Other Facility	2155 (0.5%)	2074 (0.5%)	81 (0.4%)
04: Discharged to home or home setting with support services	33385 (7.9%)	29730 (7.4%)	3655 (17.1%)
05: Discharged home (no support services required)	371257 (88.2%)	354684 (88.7%)	16573 (77.5%)
06: Discharge against medical advice or absent without leave	4094 (1.0%)	3694 (0.9%)	400 (1.9%)
Length of Stay	7.17 (24.86)	7.02 (24.87)	9.92 (24.48)
Admission Acuity	206845 (49.1%)	193650 (48.4%)	13195 (61.7%)
Charlson Comorbidity Index	3.24 (6.56)	3.03 (6.31)	7.22 (9.41)
Emergency Department Usage (6m)	1.0 (2.41)	0.95 (2.29)	1.84 (3.94)
LACE Score	6.53 (4.62)	6.39 (4.54)	9.29 (5.25)

3.2 Representing Medical Concepts

The success of transfer learning has shown that providing a starting-off point can improve the performance of a new model [79], while also decreasing the time a model takes to converge. Further, black-box machine learning is the subject of ire in the medical community [78]. In the pursuit of effectively representing a patient's medical history (Section 3.3), an intermediate step—one that can be sanity-checked—could increase accuracy and ease unrest in the minds of medical administrators and practitioners. One way to do this is by first capturing information about medical codes through examining the provision of medical care.

3.2.1 The Medical Biography

Medical histories viewed sequentially are similar to natural language passages in many ways. This is first evident at the word level, with a vocabulary of medical codes (or "tokens") that convey specific meaning when assigned. Similarly, medical "episodes"—defined as single contiguous experiences with the healthcare system involving the incurrence of medical codes—may be viewed as analogous to sentences. On the patient level, medical sentences are separated by "punctuation", which indicate eras of relative wellbeing. All these sentences, taken in order, represent the story of a patient's life history with the healthcare system—a document containing their "medical biography" (Figure 3.4). The medical biographies containing information from 2011 to 2014, from patients discharged from Alberta hospitals during 2015 and 2016, constitute our "medical corpus".

To construct each person's medical biography, records with the same patient identifier were extracted from the DAD, NACRS, Claims, and PIN datasets, and were sorted by timestamp. These records contained 36834 unique codes indicating emergent and non-emergent admissions, diagnoses, procedures, prescriptions, and acute care and ambulatory discharge dispositions. Highly specific codes were "rounded" to

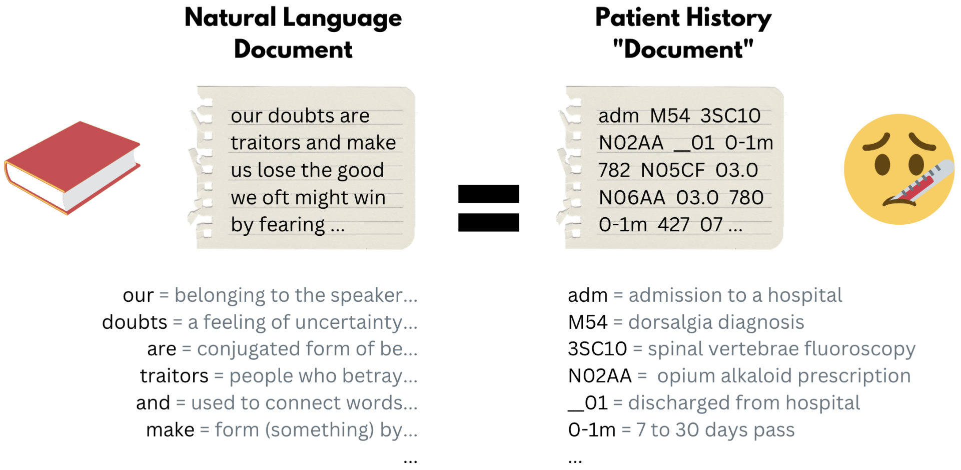

Natural Language Document
Patient History "Document"
our doubts are traitors and make us lose the good we oft might win by fearing ...
adm M54 3SC10 N02AA _01 0-1m 782 N05CF 03.0 N06AA 03.0 780 0-1m 427 07 ...
our = belonging to the speaker...
doubts = a feeling of uncertainty...
are = conjugated form of be...
traitors = people who betray...
and = used to connect words...
make = form (something) by...
...
adm = admission to a hospital
M54 = dorsalgia diagnosis
3SC10 = spinal vertebrae fluoroscopy
N02AA = opium alkaloid prescription
_01 = discharged from hospital
0-1m = 7 to 30 days pass
...

Figure 3.4: **The Patient Document analogy.** Just as words can be read sequentially to understand the meaning of a passage, medical codes "read" sequentially encompasses important health history information about a patient.

decrease granularity; the first three alphanumerics of ICD-9 and ICD-10-CM codes and the first five alphanumerics of ATC and CCI codes were kept, and CCPX codes were cleft once place after the decimal. Codes with fewer than 100 occurrences were replaced with the generic code "RAREWORD". These steps decreased the vocabulary size to under 4000[3]. Following suggestions from E. Choi and Nguyen [30] [35] [45], codes documented more than seven days apart for a single patient were separated by a "timecode": "0-1m" for fewer than 31 days, "2-3m" for an interval of 31 to 90 days, and "3-6m", "6-12m", and "12+m" for periods of three to six, six to twelve, and more than twelve months respectively. Further, long, repeating stretches of codes were often observed in the data; for example, patients with chronic conditions may refill a single prescription on a near-weekly basis for many years. For this reason, continuous single-code repeats between timecodes were replaced with a single instance of the code.

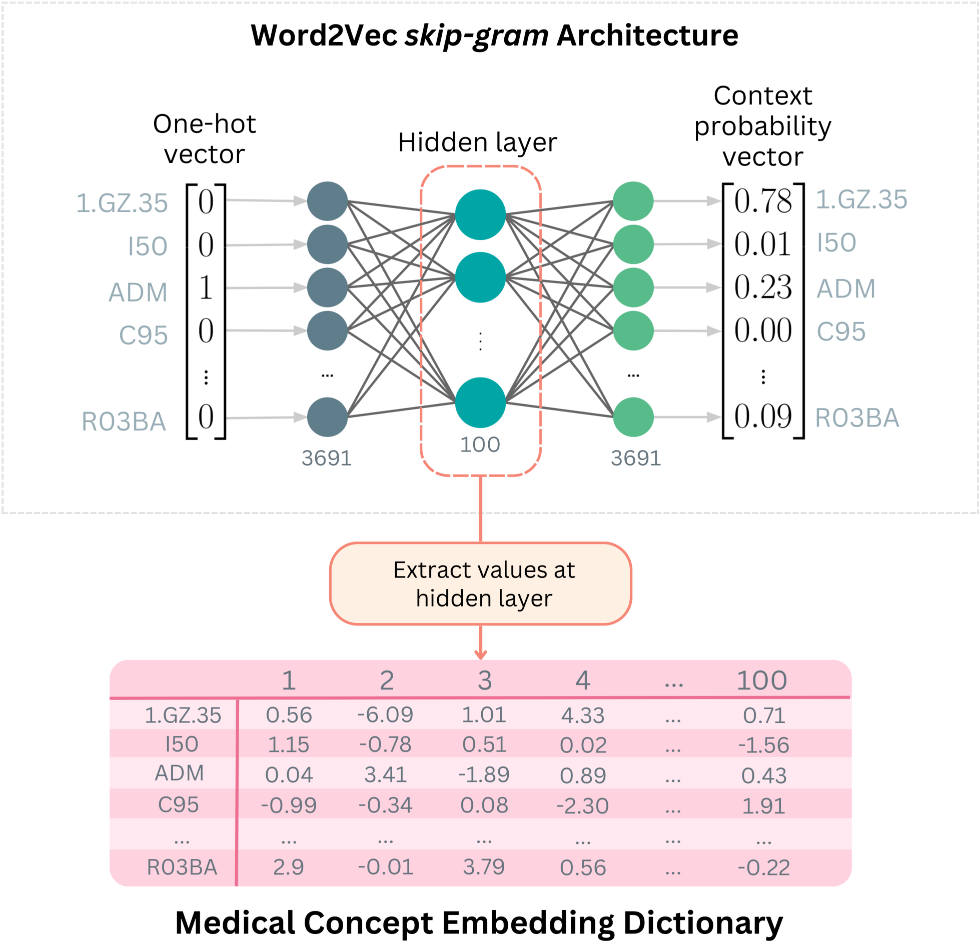
[3] The exact number varied between folds.

3.2.2 Medical Code Embeddings using Word2Vec

Word2Vec [36] uses a single-hidden-layer neural network architecture to learn low-dimensional vector embeddings of words (Figure 3.5). Two implementations of this architecture exist: *CBOW* (Continuous Bag-Of-Words)—concerned with predicting a masked word given its immediate context—and (*skip-gram*), where the word is used to predict its immediate context. For both formulations, training requires pairs of vectors, both the size of the vocabulary ($|V|$): a one-hot vector where each index is associated with a word $w \in V$, and a Context Probability Vector (CPV), where each element holds the probability that every other word in V occurs in some predefined context of w. The CPV for each w is calculated from the sentences in the training corpus. For our purposes, we use the *skip-gram* algorithm with a context window size of ± 5 and a hidden layer with 100 nodes. Word2Vec is a **self-supervised** algorithm in the sense that no outcome label is used in the generation of the embeddings.

When the network weights are trained to completion, the lower-dimensional representation for each word is calculated by extracting the values at the network's hidden nodes. The values at the output layer—those that hold the prediction for the CPV—are discarded. Intuitively, resulting embeddings representing similar concepts should themselves be similar, relying on the assumption that related concepts appear in analogous contexts.

We use Word2Vec to develop a dictionary of **Medical Concept Embeddings** (MCEs) for every medical code (e.g., prescription) and event (e.g., time-skip) that appears with sufficient frequency in the medical biographies found in our training data corpus. Some further considerations exist in this pursuit. Timestamps associated with certain medical codes (used to determine the order of codes in a medical biography) are occasionally not reflective of the exact time of medical event. For example, Claims dataset entries are captured when the insurance claim is processed, rather than when the actual care was provided. In addition, many codes are assigned on the same day


Word2Vec *skip-gram* Architecture
One-hot vector
Hidden layer
Context probability vector
1.GZ.35
I50
ADM
C95
R03BA
0
0
1
0
⋮
0
3691
100
3691
0.78
0.01
0.23
0.00
⋮
0.09
1.GZ.35
I50
ADM
C95
R03BA
Extract values at hidden layer
Medical Concept Embedding Dictionary

	1	2	3	4	...	100
1.GZ.35	0.56	-6.09	1.01	4.33	...	0.71
I50	1.15	-0.78	0.51	0.02	...	-1.56
ADM	0.04	3.41	-1.89	0.89	...	0.43
C95	-0.99	-0.34	0.08	-2.30	...	1.91
...	...	...	...	...	...	...
R03BA	2.9	-0.01	3.79	0.56	...	-0.22

Figure 3.5: **An example Word2Vec *skip-gram* architecture with a hidden layer size of 100 and a vocabulary size of 3691**. This would generate 100-dimensional embeddings for each of the 3691 medical codes in V.

in datasets where only the event **date** (not event **time**) is captured. To account for these issues, codes occurring within medical sentences (i.e., between timecode "punctuation") were randomly re-ordered (following the suggestion of E. Choi *et al.* in [80] and [34]). This shuffling has previously been shown to decrease bias found in Word2Vec models [81]. If a new medical code (that did not exist in the V of our training set) occurred in the test set, the code would be represented using the embedding associated with our generic "RAREWORD". Our MCEs were trained only on data that was (1) from the pre-index period (Figure 3.2), and (2) was **not** used to determine our final reported performance metrics. For more information about the train/test splits, see Section 4.2 and Figure 3.3B.

3.2.3 Medical Code Embedding Illustrative Examples

One can determine how similar a vector $\vec{a}$ is from a vector $\vec{b}$ (where $\vec{a}, \vec{b} \in \mathbb{R}^n$) using cosine similarity: the cosine of the angle between the two vectors:

$$S_C(\vec{a}, \vec{b}) \quad = \quad cos(\theta) \quad = \quad \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \tag{3.1}$$

If Medical Code Embeddings contain the desired contextual medical information, vectors representing similar concepts should themselves be similar. As a sanity-check, for four prevalent chronic conditions: asthma, diabetes, hypertension, and heart failure (each with their own ICD-10-CM diagnosis code), we examine the top five most similar codes to each using cosine similarity.

Results are shown in Table 3.2. Five out of five of the extracted similar codes for each of the chronic conditions were deemed medically related to the original condition by our physician collaborators, and the dictionary of Medical Code Embeddings was given verbal approval.

3.3 Patient Representations for Clinical Predictive Tasks

This section describes how we transform a patient's health history and general information into input for our predictive models. Some features are either available directly alongside longitudinal health record information, or are hand-engineered based on domain knowledge. Others contain strings detailing medical history biographies, or are informed by these biographies and the Medical Code Embedding dictionaries obtained in Section 3.2.2.

In the pursuit of training useful models, we can categorize features into those that use information from the duration of the index admission, and those that do not. A model that predicts readmissions would be best used at the **beginning**

Table 3.2: **Medical Concept Embeddings sanity check**. We report the five most cosine similar codes associated with four highly prevalent chronic conditions.

Condition Code	Condition Name	Similar Code	Cosine Similarity	Similar Code Type	Description
E11	Type 2 diabetes mellitus	E14	0.777	ICD-10	Unspecified diabetes mellitus
		N08	0.752	ICD-10	Glomerular disorders in diseases classified elsewhere
		I10	0.743	ICD-10	Essential (primary) hypertension
		E78	0.661	ICD-10	Disorders of lipoprotein metabolism and other lipidemias
		A10BX	0.612	ATC	Drugs used in diabetes: Other blood glucose lowering drugs
I10	Essential hypertension	E11	0.743	ICD-10	Type 2 diabetes mellitus
		E78	0.729	ICD-10	Disorders of lipoprotein metabolism and other lipidemias
		I25	0.63	ICD-10	Chronic ischemic heart disease
		N08	0.572	ICD-10	Glomerular disorders in diseases classified elsewhere
		Z95	0.514	ICD-10	Presence of cardiac and vascular implants and grafts
J46	Asthma	493	0.722	ICD-9	Asthma
		1.GZ.35	0.671	CCI	Inhalation pharmacotherapy
		J98	0.608	ICD-10	Other respiratory disorders
		R03AC	0.6	ATC	Selective beta-2-adrenoreceptor agonists
		R03BA	0.594	ATC	Glucocorticoids
I50	Heart Failure	428	0.863	ICD-9	Heart failure
		I42	0.805	ICD-10	Cardiomyopathy
		I25	0.706	ICD-10	Chronic ischemic heart disease
		I48	0.688	ICD-10	Atrial fibrillation and flutter
		I34	0.674	ICD-10	Nonrheumatic mitral valve disorders

of an index admission to facilitate the planning of the care trajectory [75], when information such as "length of stay" is as-of-yet unknown. For all the features detailed in Sections 3.3.1 to 3.3.3, we note if they require information from the duration of the index admission (`incl_idx`), or if they only use information up until and including the time of admission (`b4_idx`). See Table 3.3 for details.

3.3.1 LACE Score

The index admission "length of stay" is calculated using the admit and discharge dates of the index episode. We extract the the emergent status of the index admission by searching for emergent NACRS records (associated with MIS codes 713100000, 715130000, 715140000, 713102000) from the day before or the day of the beginning of the index episode. The `comorbidipy`[4] library is used to calculate Charlson Comorbidity Index [8] from DAD data. NACRS is queried to find the number of times an individual was admitted to the emergency department in the six months preceding their index admission. The singular **LACE** score (ranging from 0-19), is calculated for each patient's index episode i using the above four features and mappings originally detailed by van Walraven et al. [7].

3.3.2 Bare Features

Bare features are defined as those that:

1. appear in raw form alongside every index admission, or

2. must be manually calculated, but given what is already available in Alberta Netcare, would not require extra work on the part of an analyst or expert to engineer.

"Raw" features include an individual's sex (M, F, or O), age, and discharge disposition (the code characterizing the patient's departure from the hospital). Alongside

[4] https://comorbidipy.readthedocs.io/

the raw features, we include the LACE score and all the features that were determined to calcuate LACE—namely the length of stay, emergent status of admission, the Charlson Comorbidity Index calculated using information from the index admission, and the number of times emergency services were utilized in the previous six months.

3.3.3 Detailed Features

Beside the **Bare** features that would surely appear alongside every index admission, 38 extra features that capture important information from an individual's medical history and hospital usage are also included. This is meant to approximate what information is possible to represent using domain knowledge and manual extraction from administrative health data. These **Supplementary** features were chosen from surrounding medical AI literature with help from our physician collaborators, and are combined with the **Bare** features to form our **Detailed** feature set.

From the span of the index episode, we capture the number of: procedures (total), procedures (unique), ICD-10-CM diagnoses codes (unique), prescription (total), prescriptions (unique), and days prescribed (total). Prior to the index admission, from DAD we capture the number of hospital episodes (in 6 months, 1 year), and the total length of stay in hospital in the previous 6 months and 2 years. From Claims, we calculate the number of visits to General Practitioners (GPs), General Surgeons (GNSG), and Internal Medicine Specialists (INMD) in the previous 6 months and 1 year. From the previous 2, 3, and 4 years, the binary variable of 'visited_GP" and the total cost of the user to the health system were also calculated from Claims. We extracted (from the two years preceding the index episode) the number of prescriptions (total), prescriptions (unqiue) and prescribed days (total) from PIN. NACRS can tell us the number of emergent and non-emergent admissions in the previous 6 months and 1 year. The discharge disposition of the patient's most recent admission (if any) is captured from DAD, and the Charlson Comorbidity Index (not including records the index episode) is calculated from DAD diagnoses as described in Section 3.3.2. Finally, the binary presences of asthma, chronic heart failure, diabetes, and hypertension from the previous two years were calculated from ICD-9 and ICD-10-CM codes in Claims, DAD, and NACRS using an algorithm validated by Tonelli et al. [19]—examining records both including and excluding those associated with the index episode—to form two different sets of four comorbidity features.

3.3.4 Raw Sequence Features

A number of models that take sequential input (**Seq** input) are tested as part of this study. Strings containing the chronological medical biographies of each patient from the entire study period (as they exist after the post-processing of Section 3.2.1) are prepared in the following ways and associated with each index episode.

`incl_idx` **Seq features:** The 200 most recent codes accumulated **before and during** the index episode (ending at the day of discharge) are retained.

`b4_idx` **Seq features:** We retain the 199 most recent codes accumulated **before** the index episode (ending the day before the index admission). A new admission token "`ADM`" is added to the end of the list of codes, representing the index admission event. For many people, their index episode is the first (and potentially only) interaction with the healthcare system captured these data. In this case, we assign a trivial medical biography ("`ADM`"), only denoting that an index admission event happened.

3.3.5 Aggregated Sequence Features

The MCE dictionary (Section 3.2.2) is used to map a sequence of events from a patient's medical biography to a sequence of medical code embeddings. Recall that $\vec{w} \in \mathbb{R}^{100}$ for each medical code embedding $\vec{w}$. A simple summation of the k most recent code embeddings in a patient's i care history (prior to and potentially including the index admission) is used to create a feature vector $\vec{b}_k^{(i)} \in \mathbb{R}^{100}$. $k = 20$ or

Table 3.3: **All clinical features used in this study**. Features are categorized by the information they require—using information from the duration of the index admission (`incl_idx`) or not (`b4_idx`), and whether the feature would be guaranteed directly available from AHS (**Bare**) or may need to be manually engineered (**Supplementary**). The **Bare** and **Supplementary** features together form the **Detailed** feature sets.

	Bare	Supplementary
incl_idx	Index episode dicharge disposition	Total number of procedures in index
	Index episode length-of-stay	Number of unique procedures in index
	LACE Score	Number of unique ICD codes in index
	If index episode admission was emergent	Total number of prescription records in index
	Index-informed Charlson Comorbidity Score	Number of unique drugs prescribed in index
		Index-informed presence of hypertension
		Index-informed presence of diabetes
		Index-informed presence of asthma
		Index-informed presence of chronic heart failure
b4_idx	Age at admission	Discharge disposition of most recent admission
	Sex	Number of admissions (6m, 2y)
	Number of emergency admissions (6m)	Total length-of-stay (6m, 2y)
		Number of general practitioner visits (6m, 1y)
		Number of general surgery visits (6m, 1y)
		Number of internal medicine visits (6m, 1y)
		Presence of a general practitioner visit (2y, 3y, 4y)
		Total cost to the healthcare system (2y, 3y, 4y)
		Total number of prescription records (2y)
		Number of unique drugs prescribed (2y)
		Number of days prescribed (2y)
		Number of emergency admissions (1y)
		Number of non-emergency admissions (6m, 1y)
		Index-uninformed presence of hypertension
		Index-uninformed presence of diabetes
		Index-uninformed presence of asthma
		Index-uninformed presence of chronic heart failure
		Index-uninformed Charlson Comorbidity Score

$k = 25$ is used to compute this representation, selected using internal cross-validation separately for each fold. Please see Appendix B for more details about the search for the most favourable k for this summation. We create different versions of these **AggSeq** features using both `incl_idx` and `b4_idx` medical biographies.

The intuition behind the summation step lies in an important property of Word2Vec, that certain vector operations performed on the numeric vectors (e.g., addition, subtraction) are meaningful. A well-known example demonstrating this is:

$$\overrightarrow{\text{king}} - \overrightarrow{\text{man}} + \overrightarrow{\text{woman}} \quad \approx \quad \overrightarrow{\text{queen}} \tag{3.2}$$

Assuming that data for training the numeric representations are representative enough, the above equation should hold for a set of Word2Vec vector representations. Therefore, the summation of the vectors representing a patient's medical history is a compelling way approximating the patient, as they exist at the time of admission or discharge, in vector space.

Chapter 4

Predicting Clinical Outcomes

4.1 Tasks

We now explore how these previously-defined patient representations may be used for predicting adverse downstream clinical events. The two tasks of interest are 1) **binary readmission prediction** (detecting the presence of a readmission to the hospital within 30 days), and 2) **readmission-free survival prediction** (modeling the more general progression of a patient, examining the probability of "surviving" without a readmission event as a function of time.) Of interest in this project is how:

1. different models,

2. using different combinations of features (**LACE**, **Bare**, **Detailed**, **Agg**regated-**Seq**uence, and raw-**Seq**uence),

3. calculated with versus without information from the index admission,

perform on our selected tasks. Table 4.1 summarizes the five models used. Figures 4.1 and 4.2 showcase the architectures and inputs used to create these models, separated by their reliance (or lack thereof) on deep learning. We call deep-learning models "heavyweight" models, and non-deep-learning models "lightweight".

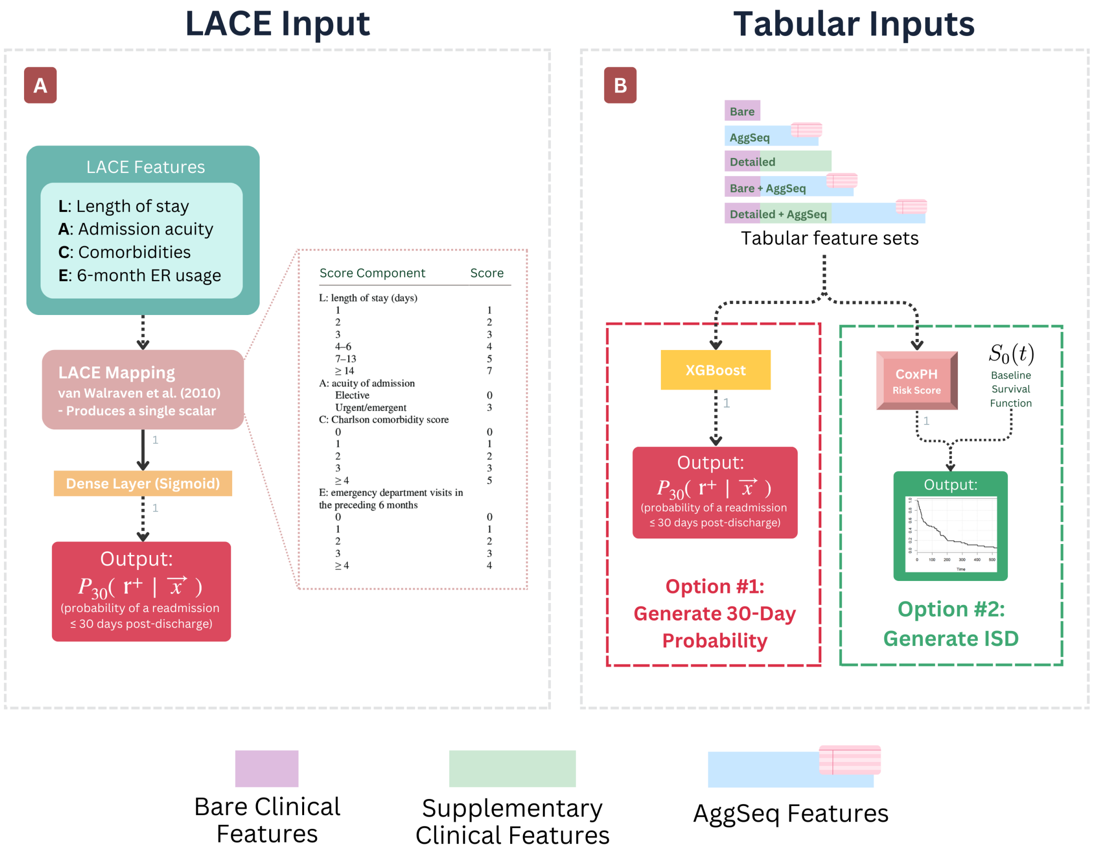
Table 4.1: **A summary of models used in this study**. Each of the five models is classified based on the complexity of their architecture and the type of survival information modeled.

	Lightweight - No deep learning - Cannot take sequence inputs	**Heavyweight** - Deep learning - Can take sequence inputs
Models **risk of readmission**	LACE Model (LACE)	-
Models **probability of readmission at 30 days**	XGBoost (XGB)	Deep Neural Network (DNN)
Models **individual readmission-free survival distribution**	CoxPH with Baseline Survival Function (Cox)	Neural Multi-Task Logistic Regression (N-MTLR)

4.1.1 Label Calculation

We define an all-cause **readmission event** for a singular individual, i, as the first **hospital episode** (Section 3.1.2) whose start date is at least two days after the discharge date of the **index episode**, and begins in either the index period or the post-index period (Figure 3.2). More concretely, this definition allows the models to learn associations between hospitalizations that are **not** due to psychiatric events or baby births and the **first successor hospitalizations** that is also not due to psychiatric events or baby births.

The label necessary to train our 30-day binary readmission models is `readmitted` ($r^{(i)}$). For the readmission-free survival models, the labels required are `time` ($t^{(i)}$, time until event) & `event` ($\delta^{(i)}$, censorship status). We begin with the set of all selected index hospital episodes (see Footnote 2 in Section 3.1.2) that end in the index period (see Figure 3.2). For every i's index admission, we calculate the number of days from discharge until the next valid hospital episode from the same patient (`TTR`: time to readmission), and the number of days from discharge until the end of the study period (`TTLTF`: time to lost to followup). A standard assumption is that an individual's `TTR`



Score Component	Score
L: length of stay (days)	
1	1
2	2
3	3
4–6	4
7–13	5
≥ 14	7
A: acuity of admission	
Elective	0
Urgent/emergent	3
C: Charlson comorbidity score	
0	0
1	1
2	2
3	3
≥ 4	5
E: emergency department visits in the preceding 6 months	
0	0
1	1
2	2
3	3
≥ 4	4

Figure 4.1: **Architectures and inputs of lightweight (non-deep-learning) models used in this study.** (a) LACE model taking the LACE score as input, (b) XGBoost and CoxPH models taking tabular features as input. The CoxPH risk score is combined with a baseline survival function to generate an individual survival distribution. LACE Score table credit: Saluk et al. [82].

and `TTLTF` would be independent. If no future admissions exist, `TTR` is assigned to be ∞. The numeric `time` label $t^{(i)}$ and the binary `event` label $\delta^{(i)}$ are defined as follows:

$$t^{(i)} = \min(\texttt{TTR}, \texttt{TTLTF}) \tag{4.1}$$

$$\delta^{(i)} = \begin{cases} 1, & \text{if } \texttt{TTR} \neq \infty \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

and the binary "30-day readmission" target is defined as follows:

$$r^{(i)} = \begin{cases} 1, & \text{if} \quad t_i \geq 30 \quad \wedge \quad \delta_i = 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

The distributions of $t^{(i)}$ and $\delta^{(i)}$ are visualized in Figure 4.3, and a Kaplan-Meier

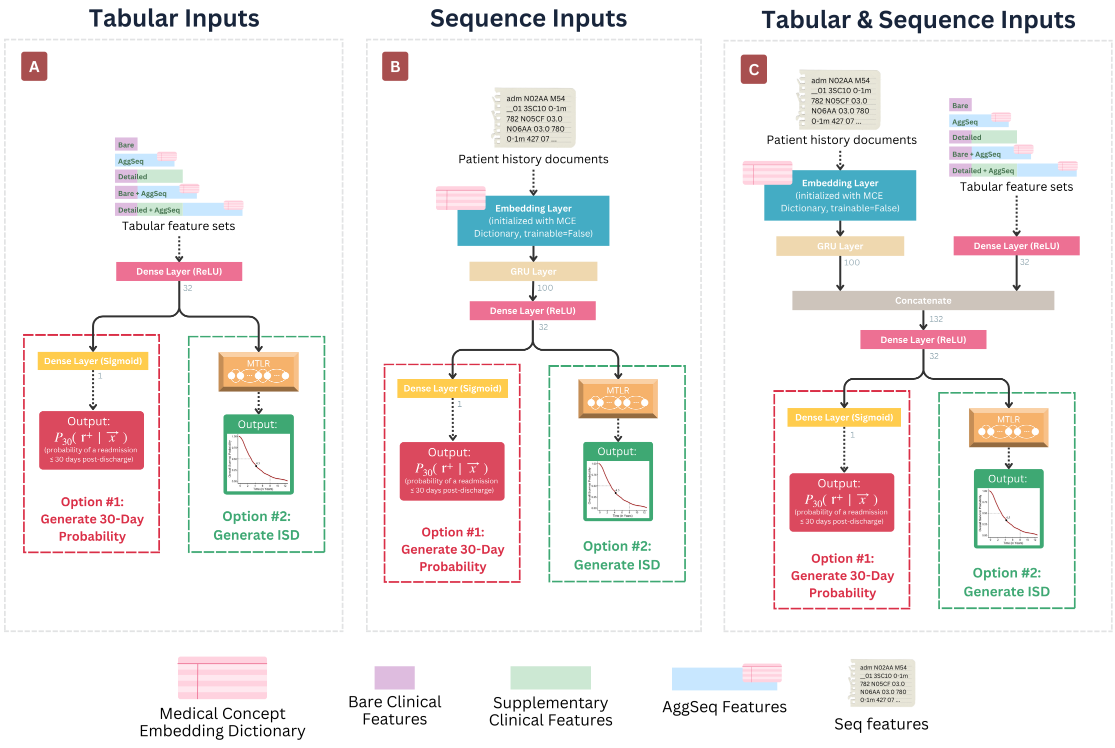

Tabular Inputs
Sequence Inputs
Tabular & Sequence Inputs
A
B
C
Bare
AggSeq
Detailed
Bare + AggSeq
Detailed + AggSeq
Tabular feature sets
adm N02AA M54 __01 35C10 0-1m 782 N05CF 03.0 N06AA 03.0 780 0-1m 427 07 ...
Patient history documents
Embedding Layer (initialized with MCE Dictionary, trainable=False)
GRU Layer
100
Dense Layer (ReLU)
32
Concatenate
132
Dense Layer (Sigmoid)
1
MTLR
Output:
$P_{30}(r^+ \mid \vec{x})$
(probability of a readmission ≤ 30 days post-discharge)
Option #1: Generate 30-Day Probability
Option #2: Generate ISD
Medical Concept Embedding Dictionary
Bare Clinical Features
Supplementary Clinical Features
AggSeq Features
Seq features

Figure 4.2: **Architectures and inputs of heavyweight (deep learning) models used in this study.** Option #1 corresponds to the DNN model, and Option #2 corresponds to N-MTLR. (a) Heavyweight model architectures only taking tabular features as input, (b) heavyweight models taking only sequence features, and (c) heavyweight models that take in both tabular and sequence features.

plot of dataset-wide survival, split by sex, can be seen in Figure 4.4. 5.08% of individuals experienced a readmission event ($r^{(i)} = 1$) within 30 days. The mean $t^{(i)}$ is 603.8 (Std.Dev 296.5 days). The maximum event time is 1095 days, and the minimum event time is 2 days. 71.8% of cases are censored ($\delta^{(i)} = 0$).

Censorship Events

Censorship obfuscates the true time until the outcome of interest. While multiple types of censorship exist (including left and interval censoring, where the beginning and some middle section of the time-until-event are unknown respectively), the only relevant type of censorship in this study is **right censorship**, where we lose track of an individual some time before they experience the outcome of interest (Figure 2.2).

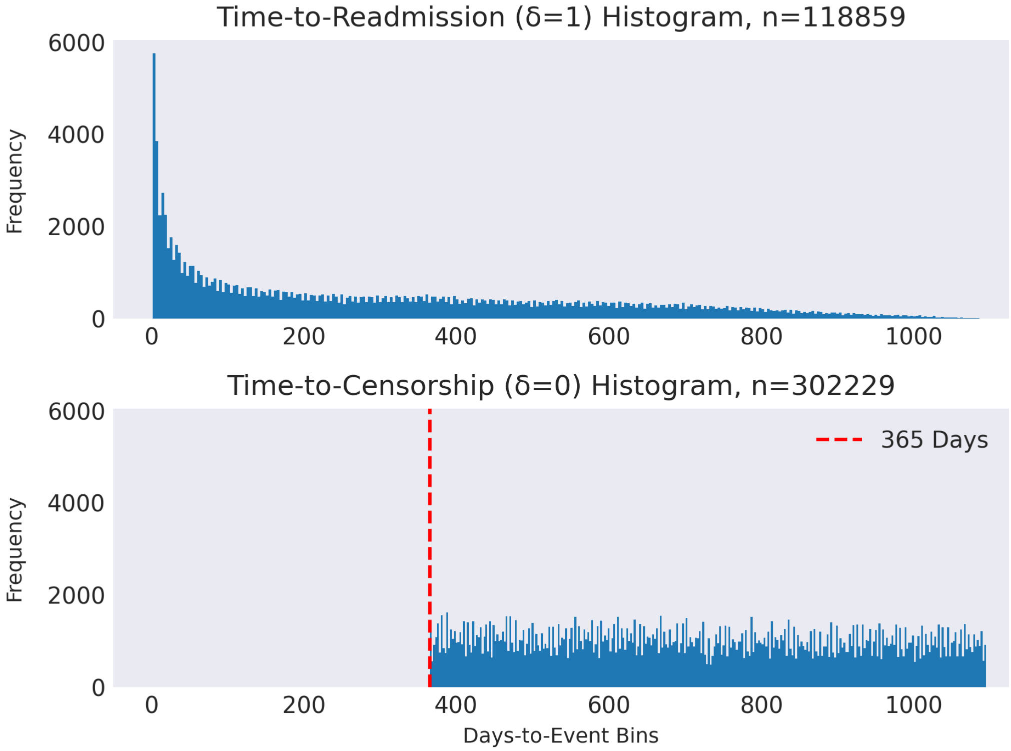


Figure 4.3: **The differing distributions of $t^{(i)}$ based on $\delta^{(i)}$.** Note that in "Time-to-Censorship", the x-axis starts at 365 days, which is the censoring time of individuals whose index episode discharge coincided with the last day of the index period.

One potential cause of right censorship is an individual moving out of the geographical region which is considered for defining the study population. In our case, there exist some individuals whose postal codes are not consistently in Alberta, but we did not have enough information to directly determine *when* they moved out of the province and began using other health services. These 21,024 individuals were excluded from the study (Section 3.1.2). The remaining (measurable) censoring in our dataset's survival label is due to the end of the study/post-index period on December 31st, 2017. There exists no patient i, under our definition of $r^{(i)}$, with a measurably missing/incomplete (or "censored") $r^{(i)}$ target, as our index selection period is succeeded by a one-year period where even readmissions occurring in the last 30 days of our index period are successfully captured. A complicating factor is that patient deaths are only collected within these available data if the death occurred

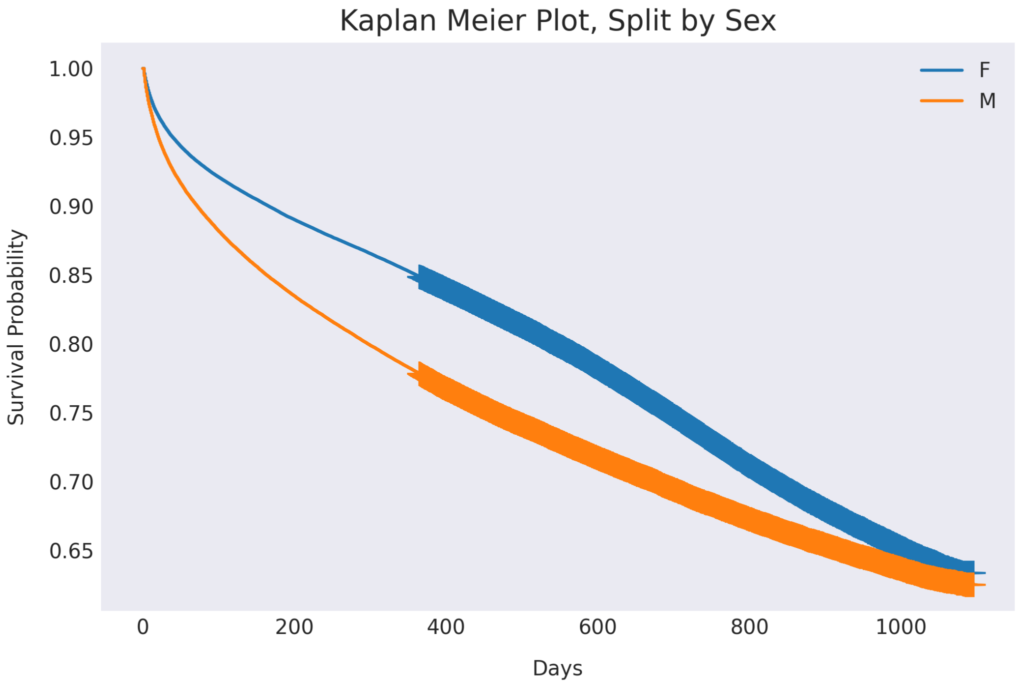


Figure 4.4: **Kaplan Meier group survival curves, separated by sex.** Crosses starting at $x = 365$ indicate censoring events. Individuals with no sex indicator are excluded from this visualization. Note that the y-axis does not go down to zero.

during a hospital visit, leading to a DAD discharge disposition of `07`. Therefore, those who pass away **outside** the hospital setting are indistinguishable from those who live readmission-free for all our targets: $t^{(i)}$, $\delta^{(i)}$, and $r^{(i)}$. Further discussion of the implications of this can be found in Chapter 5.

4.2 Data Splits

Five-fold external cross validation is used to report all final numbers, and a five-fold internal cross-validation within each of the non-test folds was used for parameter tuning. The experimental set-up is detailed in Figure 3.3B.

The set of index episodes of interest were randomly split into five folds. Each fold acted as the test set in turn. The validation set was chosen as the fold positioned directly after the test fold (or was taken to be 1 if the test fold was 5). The training set consisted of every fold that was not used in testing or validating. The MCE dictionary was calculated from the training and validation sets at the beginning of each outer fold, and was used for feature generation and model training for all within-fold experiments. Internal-cross-validation was used to tune any hyperparameters: the outer-fold training set was again split into five folds, and the best model for each outer-fold was selected based on the average loss of each model setting on the five inner folds. The best experimental settings were then used to train a model on the entire outer-fold training set (using evaluations on validation set to prevent overfitting), and then tested on the outer-fold test set. Outer test folds all contained 84217 or 84218 patients. Performances across all outer folds are averaged, and the standard deviation is examined to understand performance consistency.

4.3 Binary Readmission Models

The models described in this section use input features representing a patient's demographics and/or medical history to predict the true value of $r^{(i)}$: whether the patient i exists a hospital readmission within 30 days of the index episode discharge.

4.3.1 Lightweight Model: LACE Baseline

The baseline to which we compare our more sophisticated readmission models is a one that uses the **LACE** risk score as a single input feature. Recall that the calculation of the **LACE** index is described in Section 3.3.1. **LACE** is used in a practical setting by binning patients into categories (e.g., "high risk" and "low risk"), which requires the definition of a risk threshold. A simple model can be trained to find the threshold that most accurately bins patients based on readmission status. We implement logistic regression to find this threshold (following the procedure described by Damery et al. [83]) using a `Pytorch` model with a single sigmoid layer, and binary cross-entropy loss for training (Figure 4.1A).

4.3.2 Lightweight Model: XGBoost

XGBoost (E**X**treme **G**radient **Boost**ing) [84] is a popular ensemble-based model with a structure consisting of multiple decision trees. XGBoost is trained using a combination of bootstrapping (selecting subsets of the training data with replacement and training multiple "weak learners"), and boosting, in which weak learners are trained on the residuals of previous weak learners. The efficiency of XGBoost makes it an attractive lightweight option that would be minimally computationally burdensome if implemented in a practical hospital setting. We fit `xgboost`'s XGBClassifier using default parameters on combinations of tabular **Bare**, **Detailed** and **AggSeq** inputs (Figure 4.1B). We use the `predict_proba` function to extract the predicted 30-day readmission probabilities for each patient i.

4.3.3 Heavyweight Model: Deep Neural Network (DNN)

We test deep-learning architectures to take advantage of extra (and more flexible) representational power for the task of 30-day readmission prediction. Combinations of tabular features (**Bare**, **Detailed** and **AggSeq**) and medical history **Seq**uences form the inputs. See Figure 4.2 for an overview of the architecture, inputs, and outputs.

Tabular Inputs

Tabular features enter a 32-unit hidden layer with a Rectified Linear Unit (ReLU, $R(x) = max(0, x)$) activation function.

Sequence Inputs

Patient biographies are padded to the maximum length of 200. This sequence information is fed into an embedding layer, which can be initialized with the self-supervised Word2Vec MCE dictionary. If an embedding dictionary is provided, the embedding layer is frozen such that no further training of embeddings is allowed. The embedding size is set to 100 to match the dimension of our pre-trained embeddings. From the embedding layer, the representations pass into a fully Gated Recurrent Unit (GRU) [85] layer. The hidden state of a normal recurrent unit at time/position t is defined as follows [86]:

$$h_t(x_t) \;=\; \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{4.4}$$

where x_t is the input vector at time (or position) t, h_t is the hidden layer representation of all inputs up until and including x_t, σ_h is the activation function, and W_h, U_h, and b_h are all vectors or matrices of learnable parameters. GRU layers improve upon normal recurrent layers by introducing an "update gate" vector z_t and a "reset gate" vector r_t:

$$z_t(x_t) \;=\; \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{4.5}$$

$$r_t(x_t) \;=\; \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{4.6}$$

At each t, a candidate hidden state is calculated, $\hat{h}_t$, which incorporates information in h_{t-1} using the reset gate:

$$\hat{h}_t(x_t) \;=\; \phi_h(W_h x_t + U_r(r_t \odot h_{t-1}) + b_r) \tag{4.7}$$

which is weighted against h_{t-1} in the final calculation of h_t by the update gate z_t:

$$h_t(x_t) \;=\; (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{4.8}$$

This allows tokens (in our case, medical codes) that do not contribute ample predictive information to be diminished in importance upon the calculation of the final h_T. The forget gate can also help overcome the vanishing gradient problem, where inputs seen relatively early in the sequence contribute relatively little to the final representation.

Outputs

To use both tabular and sequence features, the output representations of both types of features are concatenated into a 132-dimensional output vector, which is fed into another 32 ReLU densely connected hidden layer. This is connected to the predictor module, a single-unit densely connected sigmoid layer. To use sequential features only, the GRU-RNN patient representation h_T is fed into a dense 32-unit layer with a ReLU activation, which is connected directly to the predictor module. To use tabular features only, the output of the tabular section is fed directly into the predictor module.

DNNs were implemented as a custom model class in `PyTorch`. Models were trained for up to 1000 epochs on the training set, and the validation set was used to monitor for overfitting. Early stopping was used with a patience of 10 (i.e., the model was allowed to train up to 10 extra epochs without improvement on the validation loss) and a minimum improvement per-epoch of 0.001. The best model settings according to the validation loss were used for evaluation on the test set.

4.3.4 Binary Readmission Model Evaluation

Preamble

For our binary task, $r^{(i)} = 1$ if an individual is readmitted within 30 days of the index episode i, and $r^{(i)} = 0$ otherwise. The predicted class, according to a model M that outputs a probability of readmission $P_M(r^{(i)} = 1|\vec{x}^{(i)})$, and a selected threshold τ, is calculated as

$$\hat{r}^{(i)} = \begin{cases} 1, & \text{if } P_M(r^{(i)} = 1|\vec{x}^{(i)}) \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{4.9}$$

This leads to four possible combinations of $r^{(i)}$ and $\hat{r}^{(i)}$:

1. True positive ($\texttt{TP}_\tau$): $r^{(i)} = 1 \;\wedge\; \hat{r}^{(i)} = 1$

2. False positive ($\texttt{FP}_\tau$): $r^{(i)} = 0 \;\wedge\; \hat{r}^{(i)} = 1$

3. False negative ($\texttt{FN}_\tau$): $r^{(i)} = 1 \;\wedge\; \hat{r}^{(i)} = 0$

4. True negative ($\texttt{TN}_\tau$): $r^{(i)} = 0 \;\wedge\; \hat{r}^{(i)} = 0$

Accuracy, or simply the proportion of correct predictions, is an intuitive way of evaluating a classifier:

$$\textbf{Accuracy}_\tau = \frac{TP_\tau + TN_\tau}{TP_\tau + FP_\tau + TN_\tau + FN_\tau} \tag{4.10}$$

However, the number of non-readmitted patients in our data outnumbers the readmitted patients by 20 times. This means a no-skill classifier that only predicts the majority class could achieve an accuracy of 95%. Further, we do not have an exact definition of the *cost* associated with each of the true and false positives and negatives: therefore, determining an optimal τ is beyond the scope of this project. Appendix A uses Youden's J-statistic as τ to report results for these (and other) threshold-dependent metrics.

Receiver Operating Characteristic AUC @ 30 Days

The most common method of evaluating binary classification models in medical literature is by reporting the Area Under the Receiver Operating Characteristic (AUROC) curve. This measures how well a model makes the trade-off between the True Positive Rate (`TPR`, also called sensitivity/recall) and the False Positive Rate (`FPR`, or 1−specificity):

$$\texttt{TPR}_\tau = \frac{TP_\tau}{TP_\tau + FN_\tau} \tag{4.11}$$

$$\texttt{FPR}_\tau = \frac{FP_\tau}{TN_\tau + FP_\tau} \tag{4.12}$$

when the threshold τ used to assign $\hat{r}^{(i)}$ varies between 0 and 1. Intuitively, the AUROC is the probability that a randomly selected patient who experienced a hospital readmission will be ranked higher by the model than a randomly selected patient who did not experience a readmission. A perfect AUROC score is 1, which means that a model has perfectly ranked individuals—i.e., every readmitted person was ranked more severely than every non-readmitted person. An AUROC of 0.5 is what would be expected for a random model with no discriminative ability, where every person who experienced the event has a 50% chance of being ranked more severely than one who did not experience the event.

AUROC is a discriminative measure in the sense that it measures a model's ability to discriminate between "high risk" and "low risk" patients. We calculate the AUROC using the predicted probability of readmission at 30 days as a stand-in for risk. The calculation of AUROC is generally equivalent to how the c-score/c-statistic/concordance is calculated, but these may use different predicted values as the risk scores.

Precision-Recall AUC @ 30 Days

The Area Under the Precision-Recall Curve (AUPRC) is a single-valued way to summarize how well a model balances precision and recall. Precision, P_τ (the fraction of positive predictions that were correct), and recall, R_τ (the fraction of positive cases that were correctly identified), at a particular threshold τ are defined as follows:

$$P_\tau = \frac{TP_\tau}{TP_\tau + FP_\tau} \tag{4.13}$$

$$R_\tau = \frac{TP_\tau}{TP_\tau + FN_\tau} = \texttt{TPR}_\tau \tag{4.14}$$

The AUPRC score is then defined thusly:

$$\textbf{AUPRC} = \sum_\tau (R_\tau - R_{\tau-1})P_\tau \tag{4.15}$$

A no-skill classifier would lead to an AUPRC score equal to the percentage of the data in the positive class (i.e., those who experienced a readmission). AUPRC is also commonly referred to as the AP (average precision) score.

Brier Score @ 30 Days

The Brier Score [87] is a strictly proper scoring rule sensitive to a model's discrimination and calibration [88]. Intuitively, the Brier score is the mean squared error between the "forecast" for an instance i from model M (in our case, $P_M(r^{(i)} = 1|\vec{x}^{(i)})$) and the true observation ($r^{(i)}$: 1 or 0—whether a readmission event was experienced or not). Over every patient i, one can calculate the Brier score as:

$$\mathbf{BS} \;=\; \frac{1}{N}\sum_{i=1}^{N} \left(P_M(r^{(i)} = 1|\vec{x}^{(i)}) \;-\; r^{(i)}\right)^2 \tag{4.16}$$

A random classifier that predicts a 50% probability of event for every instance would result in an overall Brier score of $0.5^2 = 0.25$. Censoring is often handled in Brier scoring by re-weighting using the inverse probability of censorship. However, as shown in Figure 4.3, there are no censored observations with a time $t < 365$. The Brier score can be extended to handle multiple timepoints, which is discussed further in Section 4.4.3.

4.4 Readmission-Free Survival Models

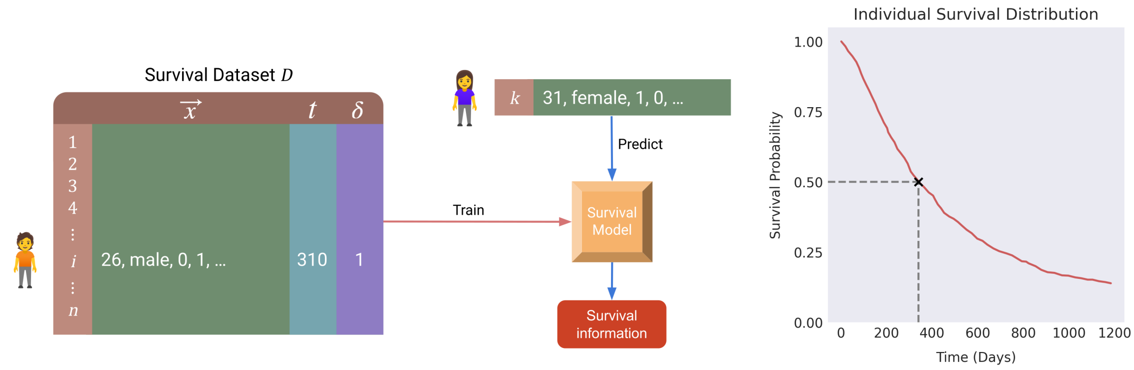
Recall that survival prediction concerns the use of a "survival dataset" or "time-to-event" dataset:

$$D \;=\; \{[\vec{x}^{(i)}, t^{(i)}, \delta^{(i)}]\}_{i\in N} \tag{4.17}$$

where $\vec{x}^{(i)}$ contains information about a patient's demographic information and medical history, $t^{(i)}$ is the time until event (readmission or censorship), and $\delta^{(i)}$ is a bit indicating if the event was a readmission (1) or censorship (0). We consider two models that learn **Individual Survival Distributions** (ISDs, denoted $S_M(t, \vec{x}^{(i)})$)—each of which provide, for all future time ts, an estimate of having survived until at least t without experiencing the event of interest. See Figure 4.5 for details.

4.4.1 Lightweight Model: Cox Proportional Hazards

Cox Proportional Hazards [49] (CoxPH) is most often used to generate individualized single-scalar risk scores. If the risk score for patient A is higher than the risk score for patient B, the model is predicting that A will be readmitted sooner than B. CoxPH first models an individual hazard function (otherwise known as the failure rate—roughly the chance that an event will occur at time t given event-free survival up



(a) Survival prediction experimental set-up

(b) ISD

Figure 4.5: **Survival prediction and individual survival distributions**. (a) A survival dataset D is used to train a survival model, which can provide insights into the future of a new patient. (b) The structure of an Individual Survival Distribution (ISD). The x axis is measured in time, and the y axis denotes the probability of reaching that time event-free. A prediction for time-to-event can be reached be examining the median survival time (cross and dotted lines).

until t) as:

$$h_{cox}(t, \vec{x}^{(i)}) \;=\; \lambda_0(t)\exp(\vec{\theta}^{\intercal}\vec{x}^{(i)}) \tag{4.18}$$

where $\vec{\theta}$ represents a vector of learned weights, and $\lambda_0(t)$ is a baseline hazard function. If we assume the effect of covariates remains constant over time, $\lambda_0(t)$ can be ignored, and $\exp(\vec{\theta}^{\intercal}\vec{x}^{(i)})$ can be taken as an individual risk score $\in \mathbb{R}^+$.

Given a baseline survival function $S_0(t)$, $\exp(\vec{\theta}^{\intercal}\vec{x}^{(i)})$ can be extended to generate an Individualized Survival Distribution:

$$S_{cox}(t, \vec{x}^{(i)}) \;=\; S_0(t)^{\exp(\vec{\theta}^{\intercal}\vec{x}^{(i)})} \tag{4.19}$$

To generate ISDs representing readmission-free survival, CoxPH is implemented using `scikit-survival` with Breslow's estimator used to approximate $S_0(t)$ [58]. See Figure 4.1 for model inputs and schema.

4.4.2 Heavyweight Model: (Neural) Multi-Task Logistic Regression

The Multi-Task Logistic Regression (MTLR) algorithm for generating ISDs was introduced by Yu et al. in 2011 [59]. MTLR models event-free survival progression using a system of dependent logistic regressors trained to predict a sequence of bits indicating an individual's event status over time. First, the time-to-event range from the training set is split into J bins, where each bin covers a time interval $a_j = [t_{j-1}, t_j)$ such that $j \in [\![1, J]\!]$, $t_0 = 0$, and $t_J = \infty$. For each i, an outcome vector $\vec{y}^{(i)}$, is created as follows:

$$\vec{y} \; = \; \begin{bmatrix} y_1 = 0 \\ y_2 = 0 \\ \vdots \\ y_{s-1} = 0 \\ y_s = 1 \\ \vdots \\ y_J = 1 \end{bmatrix} \tag{4.20}$$

where each y_j is an indication of whether an individual's event has already happened or not, and s indicates the beginning of the first post-event interval. If an individual is censored (i.e., $\delta^{(i)} = 0$), every y_j such that $j \geq s$ is designated as "unknown". The loss function used to train the matrix of logistic model parameters Θ is as follows:

$$l(\Theta) \; = \; \sum_{i=1}^{N} \delta^{(i)} \log\left(f(a_s, \vec{x}^{(i)})\right) + (1 - \delta^{(i)}) \log(S(t_{s-1}, \vec{x}^{(i)})) + \alpha_{l2}\left(\|\Theta\|^2\right) \tag{4.21}$$

where $f(a_s, \vec{x}^{(i)})$ and $S(t_{s-1}, \vec{x}^{(i)})$ are density and survival functions respectively, and the final term acts as a model weight regularizer. Upon the completion of training, the $\vec{y}$ for a new patient is fed into the system, which generates a survival prediction probability of each of the J intervals. Interpolation can be used to examine the probabilities of readmission at more granular timepoints.

The simple logistic models used to model survival probabilities for each time bin may be replaced by a more complex architectures: using neural networks for this purpose is called N-MTLR [60]. N-MTLR can capture more sophisticated information from the training set, and lend flexibility in the types of input the model can learn from. We use N-MTLR to learn ISDs that model readmission-free survival time. The output vector $\overrightarrow{y^{(i)}}$ is calculated for every i with $J = 30$ time bins. $J = 30$ was chosen because it made a favourable trade-off between detail captured and training time. The time-bin boundaries are calculated such that an approximately equal number of uncensored events happen in each. A custom model class was designed in PyTorch that uses the `torchMTLR`[1] implementation for the predictor module. As detailed in Section 4.3.3, **Seq** inputs are transformed using an embedding layer (potentially initialized with the MCE dictionary). **Tabular** inputs are fed through a 32-hidden-node ReLU layer. To use only sequence inputs, the GRU layer connects to a 32-hidden-node ReLU layer, which is used as input to the MTLR predictor module. To use only tabular inputs, the MTLR predictor module is connected directly to the 32ReLU layer. To use both, the outputs of the GRU layer and the 32ReLU tabular layer are concatenated, fed through another 32ReLU layer, and connected to the MTLR predictor module. These architectures are visualized in Figure 4.2.

4.4.3 Readmission-Free Survival Model Evaluation

Evaluations @ 30 Days

To directly compare our Readmission-Free Survival models (that generate ISDs) to our 30-day readmission models, we use the ISD's $1 - S_M(t = 30, \vec{x}^{(i)})$ (equivalent to the probability of a readmission by 30-days) to calculate AUROC @ 30-days, AUPRC @ 30 days, and Brier Score @ 30 Days (Section 4.3.4). Recall that censoring is not a concern at this time-slice due to the censoring distribution, as nobody is censored earlier than 30 days (Figure 4.3).

Concordance

Concordance (also known as the c-statistic or c-index) is a discriminative measure commonly used to evaluate a model that assigns individuals a risk score $r(\vec{x}^{(i)})$ based on a feature vector $\vec{x}^{(i)}$. Concordance is a generalization of the AUROC score that applies to *all* risk scores $\in \mathbb{R}^+$, rather than only probabilities $\in [0, 1]$. If $r(\vec{x}^{(i)}) >$

[1] https://github.com/mkazmier/torchmtlr/tree/master

$r(\vec{x}^{(j)})$ according to M, the model is claiming that i will experience the adverse outcome of interest before j. An ISD provides a wealth of information, meaning there are multiple numbers that could act as a predicted risk score; we use the negative of the median predicted time-to-readmission provided by the ISD:

$$r(\vec{x}^{(i)}) \; = \; -\hat{t}_{0.5}^{(i)} \tag{4.22}$$

to calculate the concordance. This is in contrast to our use of AUROC @ 30 days, where the risk is calculated as $r(\vec{x}^{(i)}) = 1 - S_M(t = 30, \vec{x}^{(i)})$. If a model with a concordance of 0.8 predicts that $r(\vec{x}^{(i)}) > r(\vec{x}^{(j)})$, i's event will happen before j's 80% of the time over all pairs of i and j.

There are $\binom{n}{2}$ comparable pairs of patients to compare in a fully uncensored case. If V_U denotes the set of patients who do not experience a censorship event, the uncensored concordance statistic is calculated as:

$$\textbf{C-index}(V_U, r(\cdot)) \; = \; \frac{1}{\binom{|V_U|}{2}} \sum_{[\vec{x}^{(i)}, t^{(i)}] \in V_U} \; \sum_{[\vec{x}^{(j)}, t^{(j)}] \in V_U : t^{(i)} < t^{(j)}} \mathcal{I}[r(\vec{x}^{(i)}) > r(\vec{x}^{(j)})] \tag{4.23}$$

Note that the time t is strictly "time until readmission" in the set of uncensored instances. To account for censoring, the calculation of the model's concordance only considers pairs of all individuals where the smaller event time of the two is **not** censored to be "comparable". As with AUROC, the c-statistic ranges from 0 (perfectly incorrect) to 1 (perfectly correct), with a no-skill classifier resulting in a score of 0.5.

Integrated Brier Score

The Brier Score (4.3.4), normally calculated for a particular timepoint t, can be extended to cover a series of time points. For an uncensored dataset V_U, a maximum timepoint τ, and an estimated survival distribution S, the integrated Brier score is

$$\textbf{IBS}(V_U, S(\cdot|\cdot)) \; = \; \frac{1}{\tau} \int_0^{\tau} \textbf{BS}_t(V_U, S(t|\cdot)) dt \tag{4.24}$$

which provides an average Brier score over all timepoints considered. To account for the information we lose due to censoring, Graf et al. propose that the Inverse Probability of Censoring Weights (IPCW) provides a reasonable re-weighting scheme [89] for calculating the Brier score at a particular timepoint. A “censoring survival function” of sorts (estimated using a Kaplan Meier curve, but with all $\delta^{(i)}$ bits flipped) is used to weight the patients who die later more highly in the calculation than those whose event is earlier. Intuitively, this helps to overcome the sparsity of readmission observations as time increases.

L1-Loss

Hospital-readmission ISDs can be used to consider the question “How long will it be until a patient experiences a readmission event?”—thereby viewing the survival prediction problem as akin to regression. “What is the average distance between the predicted number and the true number?” is an intuitive way to evaluate a regression model. We use the median survival time $\hat{t}_{0.5}^{(i)}$ predicted by the ISD as our time-to-event prediction, which can be directly compared to the true survival time for uncensored instances. The L1-Loss, evaluated on the uncensored set of instances V_U, is as follows:

$$\textbf{L1-Loss} \;=\; \frac{1}{|V_U|}\sum_{i \in V_U} |t^{(i)} - \hat{t}_{0.5}^{(i)}| \tag{4.25}$$

Censored observations makes calculating the difference between the actual and predicted survival times difficult. We use a variant [90] of L1-loss that considers pseudo-observations ([91] and [92]) as surrogate event values in the case of censoring. Given a survival dataset D and a censored patient i's censoring time $c^{(i)}$, the surrogate event time $e_{PO}^{(i)}$ is calculated as follows:

$$e_{PO}^{(i)}(c_i, D) \;=\; N \times \mathbb{E}_t[S_{\mathrm{KM}(D)}(t)] \;-\; (N-1) \times \mathbb{E}_t[S_{\mathrm{KM}(D^{-c^{(i)}})}(t)] \tag{4.26}$$

where N is the number of individuals in the dataset, $\mathbb{E}_t[S_{\mathrm{KM}(D)}(t)]$ is the expected survival time of individuals in the dataset (including patient i) according to the Kaplan-Meier estimator, and $\mathbb{E}_t[S_{\mathrm{KM}(D^{-c^{(i)}})}(t)]$ is the Kaplan-Meier estimated survival time for individuals in dataset D without patient i included. The pseudo-observation of a censored individual can be viewed as the “contribution” a patient makes to the unbiased time estimation. This value will always be greater than the censoring time, c_i. We then estimate the **L1-Loss** over the entire dataset using the following weighting scheme:

$$\mathbb{E}[\textbf{L1-Loss}] \;=\; \frac{1}{\sum_{i\in D} w_i}\sum_{i\in D} w_i|(1-\delta^{(i)})\cdot e^{(i)}_{PO}(c^{(i)}, D) \;+\; \delta^{(i)}\cdot t^{(i)} \;-\; \hat{t}^{(i)}_{0.5}| \tag{4.27}$$

where uncensored subjects are assigned $w_i = 1$, and censored subjects are assigned $w_i = 1 - S_{\mathrm{KM}(D)}(c^{(i)})$. This weighting allows us to consider patients we know more about more highly.

1-calibration

1-calibration helps us understand if the sets of predicted readmission probabilities (at a particular time-point t_*) approximately correspond to the number of events one would expect to see. For a dataset of uncensored patients, probabilities of readmission at t_* are sorted. A number of bins B is chosen (here, $B = 10$), and the $1/B$ individuals with the highest predicted probabilities are assigned to the first bin (B_1), the second $1/B$ individuals based on predicted probability are assigned to the second bin (B_2), and so on for every further bin $j \in \{1, 2, ..., B\}$. Next, for each bin, we calculate the *number of events expected to occur*:

$$p_j \;=\; \sum_{\vec{x}^{(i)}\in B_j} (1 - S(t_*|\vec{x}^{(i)})) \tag{4.28}$$

Let O_j be the number of patients in the j^{th} bin who experienced a readmission in the 30 days following episode i. We can then compute Hosmer-Lemeshow (HL) test statistic:

$$\mathbf{HL}(V_U, S(t_*|\cdot)) \;=\; \sum_{j=1}^{B} \frac{(O_j - p_j)^2}{p_j(1-\frac{p_j}{|B_j|})} \tag{4.29}$$

Censoring can be accounted for using a within-bin Kaplan Meier curve in place of O_j [93]. For a model 1-calibrated at t_*, the **HL** is expected to follow a χ^2_{B-2} distribution. This allows us to compute a p-value, which generally, if less than 0.05, suggests the model is not meaningfully calibrated at this time. However, because the

HL test statistic is calculated with respect to the number of individuals in the dataset, minor deviations in calibration may cause an otherwise well-performing model to fail the statistical test on population-level data. A model's predictions at one time (e.g., $t = 30$) may be calibrated whilst being un-calibrated at other timepoints (e.g., $t = 365$).

D-Calibration

D-calibration is a novel measure introduced by Haider et al. [48], which aims to overcome 1-calibration's restriction to a particular time-point t_*. Recall that the median survival time $\hat{t}^{(i)}_{0.5}$ generated for a patient i by an ISD model can be used as a prediction for time-to-event. It follows that for a trustworthy time-to-readmission prediction model (with no censoring), one would expect approximately 50% of the patients to be readmitted before their $t^{(i)}_{0.5}$, and the other 50% would be admitted after. This intuition can be extended to deciles. For each uncensored patient i, we examine the probability $S_{ISD}(t^{(i)}, \vec{x}^{(i)})$ that their readmission event happened by their real readmission time. Patients are then sorted based on $S_M(t^{(i)}, \vec{x}^{(i)})$, and assigned to $B = 10$ bins. For a D-calibrated model, we would expect roughly 10% of patients to be readmitted in each of the 10 bins. From here, a straightforward Pearson's X^2 test can be applied to determine if the bins are uniform. A p-value > 0.05 is a good indication that the predicted survival curve is provides trustworthy predictions. The details of how to cope with censored individuals can be seen in Haider et al.'s manuscript.

Like 1-calibration, D-calibration relies on a parametric statistical test. Similarly again, calculating the test statistic relies on the number of samples, which can cause calibrated-appearing models to generate a calibration-test-failing $p < 0.05$.

Concordance, IBS, L1-loss, 1-calibration, and D-calibration were calculated using the `SurvivalEVAL` repository[2].

[2] https://github.com/shi-ang/SurvivalEVAL

4.5 Results

In this section, we report multiple evaluation metrics to understand the effect that 1) feature set and 2) model choice have on the effectiveness of a readmission predictor. Each result is the average of model performances using external five-fold cross-validation, and the spread of these performances is measured with the standard deviation in the text and tables, and standard error in the figures. Fold-wise paired t-tests on AUROC (for 30-day readmission models) and concordance (for ISD models) are used to statistically compare performances, unless otherwise stated. We do not adjust the p-value for multiple comparisons.

4.5.1 AUROC, AUPRC, and Brier Score @ 30 Days

30-day AUROC, AUPRC, and Brier score are calculated for all five model architectures (Table 4.1) and combinations of feature sets (**LACE**, **Bare**, **Detailed**, **AggSeq**, and **Seq**). AUROC and AUPRC scores for all models and features are plotted in Figure 4.6. AUROC, AUPRC, and Brier scores are shown in Table 4.2.

Using the XGBoost model, **Bare** and **Detailed** feature sets provide enough information for AUROCs of 0.7546±0.005 and 0.8002±0.003 respectively. **AggSeq+Bare** features with XGBoost (AUROC 0.8025 ± 0.005) leads to statistically indistinguishable performance from XGBoost using **Detailed** feature set ($p = 0.121$). Recall that the **Detailed** features include **Bare** features. XGBoost using **AggSeq+Detailed** (AUROC 0.8138 ± 0.003) features statistically outperforms the same model when trained on **Detailed** features alone ($p = 0.0002$). When paired with the **AggSeq** features alone, XGBoost achieves an average AUROC of 0.7884 ± 0.002.

When comparing XGBoost and Deep Neural Networks using tabular features alone, XGBoost achieves statistically better performance on **Bare** (0.7546±0.005 vs. 0.7147± 0.003, $p < 0.00001$) and **Detailed** (0.8002±0.003 vs. 0.7821±0.004, $p = 0.001$) feature sets. The DNN performs marginally better than XGBoost on the **AggSeq** feature vector (0.7948 ± 0.005 vs. 0.7884 ± 0.002, $p < 0.04$), and the two achieve indistinguishable performance using **AggSeq+Bare** ($p = 0.439$) and **AggSeq+Detailed** ($p = 0.10$) features. Adding **Seq** features to the **Detailed** feature DNN leads to an AUROC of 0.8212 ± 0.003. This **Detailed+Seq** DNN achieves better results than both the **Detailed+AggSeq** DNN (0.8103 ± 0.005, $p = 0.0003$) and **Detailed+AggSeq** XGBoost model (0.8138 ± 0.003, $p = 0.001$).

Seq inputs paired with N-MTLR models show the highest performance numbers out of every feature and model combination, with AUROCs of 0.8460 ± 0.003 (**Seq**), 0.8467 ± 0.004 (**Bare+Seq**), and 0.8491 ± 0.004 (**Detailed+Seq**). According to a one-way ANOVA, there exists no detectable difference between these three means ($p = 0.497$). Given the same heavyweight deep-learning architecture and feature combinations, predicting 30-day readmissions using the N-MTLR ISD output is significantly better than using the DNN 30-day binary output (**Seq**: $\Delta = 0.0346$, $p = 0.0003$; **Seq+Bare**: $\Delta = 0.0293$, $p = 0.0001$; **Seq+Detailed**: $\Delta = 0.0279$, $p < 0.00001$). Between the lightweight models, XGBoost achieves higher scores than CoxPH on every feature set, with the most notable difference appearing when only **Bare** features are used (0.7546 ± 0.005 vs. 0.6585 ± 0.002). The simple **LACE Index** baseline model achieves an AUROC score of 0.6587 ± 0.003. This performance is statistically comparable to CoxPH performance on the **Bare** feature set alone ($\Delta = 0.0002$, $p = 0.236$), but is significantly out-performed by the next lowest-performing model on the Bare feature set, the DNN ($p < 0.00001$). Visually, performance increases when using features that capture medical histories with increasing detail (Figure 4.6).

For all evaluations already reported, the sequence embedding layer of the heavyweight models was initialized with our Medical Concept Embedding dictionary (Section 3.2.2), and the training of this layer was frozen. Recall that DNN and N-MTLR models accept sequence inputs. When initialized with the MCE dictionary, performance increased for both the DNN model (AUROC: 0.7887 ± 0.005 to 0.8144 ± 0.007, $\Delta = 0.0257$, $p = 0.005$) and N-MTLR (AUROC: 0.8392 ± 0.004 to 0.8460 ± 0.003,

Table 4.2: **AUROC, AUPRC, and Brier score evaluations of different combinations of models and feature sets at 30 days**. **Bold** indicates the highest score within model (column). Results are reported as Metric±Std.Dev.

Features			**30-Day AUROC±Std.Dev**				
Clinical Features	AggSeq Features	Seq Features	Logistic Regression	XGBoost	CoxPH	DL	N-MTLR
LACE Score	-	-	**0.6587±0.003**	-	-	-	-
Bare	-	-	-	0.7546±0.005	0.6585±0.002	0.7147±0.003	0.7348±0.004
Detailed	-	-	-	0.8002±0.003	0.7279±0.003	0.7821±0.004	0.7938±0.005
-	✓	-	-	0.7884±0.002	0.7558±0.004	0.7948±0.005	0.8015±0.003
Bare	✓	-	-	0.8025±0.005	0.7634±0.005	0.8035±0.003	0.8087±0.006
Detailed	✓	-	-	**0.8138±0.003**	**0.7718±0.004**	0.8103±0.005	0.8183±0.004
-	-	✓	-	-	-	0.8114±0.007	0.8460±0.003
Bare	-	✓	-	-	-	0.8174±0.006	0.8467±0.004
Detailed	-	✓	-	-	-	**0.8212±0.003**	**0.8491±0.004**

Features			**30-Day AUPRC±Std.Dev**				
Clinical Features	AggSeq Features	Seq Features	Logistic Regression	XGBoost	CoxPH	DL	N-MTLR
LACE Score	-	-	**0.0996±0.001**	-	-	-	-
Bare	-	-	-	0.1682±0.004	0.1165±0.002	0.133±0.002	0.1458±0.004
Detailed	-	-	-	0.2274±0.006	0.1378±0.002	0.1934±0.005	0.2046±0.006
-	✓	-	-	0.2361±0.004	0.1420±0.004	0.2222±0.007	0.2246±0.006
Bare	✓	-	-	0.2595±0.003	0.1530±0.003	0.2351±0.005	0.2388±0.009
Detailed	✓	-	-	**0.2877±0.005**	**0.1683±0.003**	0.2509±0.006	0.2604±0.004
-	-	✓	-	-	-	0.2406±0.013	0.3325±0.007
Bare	-	✓	-	-	-	0.2576±0.011	0.3343±0.006
Detailed	-	✓	-	-	-	**0.2666±0.002**	**0.3383±0.009**

Features			**30-Day Brier Score±Std.Dev**				
Clinical Features	AggSeq Features	Seq Features	Logistic Regression	XGBoost	CoxPH	DL	N-MTLR
LACE Score	-	-	**0.0471±0.001**	-	-	-	-
Bare	-	-	-	0.0453±0.001	0.0470±0.001	0.0463±0.001	0.0459±0.001
Detailed	-	-	-	0.0435±0.001	0.0462±0.001	**0.0423±0.001**	0.0441±0.001
-	✓	-	-	0.0436±0.001	0.0459±0.001	0.0437±0.001	0.0436±0.001
Bare	✓	-	-	0.0428±0.001	0.0456±0.001	0.0432±0.001	0.0431±0.001
Detailed	✓	-	-	**0.0419±0.001**	**0.0452±0.001**	0.0427±0.001	0.0423±0.001
-	-	✓	-	-	-	0.0431±0.001	0.0404±0.001
Bare	-	✓	-	-	-	0.0426±0.001	0.0402±0.001
Detailed	-	✓	-	-	-	0.0444±0.001	**0.0401±0.001**

Table 4.3: **Comparison of heavyweight model performances when trained from scratch vs. initialized with the MCE dictionary**. Only **Seq** features are used as input. Results are reported as Metric±Std.Dev.

	DNN			**N-MTLR**		
	AUROC	AUPRC	Brier	AUROC	AUPRC	Brier
Trained from scratch	0.7887±0.005	0.2188±0.008	0.0440±0.001	0.8392±0.004	0.3086±0.008	0.0411±0.001
Initialized with MCE	**0.8114±0.007**	**0.2406±0.013**	**0.0431±0.001**	**0.8460±0.003**	**0.3325±0.007**	**0.0404±0.001**

$\Delta = 0.0068$, $p = 0.001$). These scores, alongside AUPRC and Brier, are reported in Table 4.3.

4.5.2 Readmission-Free Survival Models: Other Evaluations

Additionally, we report the c-statistic, integrated brier score, and L1-loss with pseudo-observations for models that generate survival curves (Table 4.4). The highest concordance statistic (0.7522 ± 0.004) is achieved by N-MTLR paired with **Detailed+Seq** features. This model also achieves the lowest IBS overall (0.1310 ± 0.001), and the lowest L1-loss (1104.9 ± 15.4 days). Among N-MTLR models (according to three separate one-way ANOVA tests), the performance using **Seq**, **Bare+Seq**, and **Detailed+Seq** features is not significantly different in IBS ($p = 0.15$) and l1-PO ($p = 0.94$), but is marginally different in concordance ($p = 0.04$). Across N-MTLR, **Detailed+AggSeq** features lead to a higher c-statistic (0.7322 ± 0.005) than when using the **Detailed** feature set alone (0.7223 ± 0.005), $p = 0.004$. N-MTLR using **Bare+AggSeq** features matches the performance of N-MTLR using **Detailed** features ($p = 0.315$). When using **AggSeq+Detailed** features, the concordance does not significantly differ between CoxPH and N-MTLR (0.7322±0.005 vs. 0.7305±0.001, $p = 0.133$). However, N-MTLR using **Seq** features alone does significantly outperform the best CoxPH model (0.7533 ± 0.004 vs. 0.7305 ± 0.001, $p = 0.001$). For the CoxPH model, the performance and spread of L1-loss with pseudo-observations is highly variable, with a standard deviation of up to 1561 days. The standard deviations for L1-loss calculated for N-MTLR models do not exceed 21 days. Survival

Table 4.4: **Comparison of the ISD models, CoxPH and N-MTLR, according to the Concordance, Integrated Brier Score (IBS) and L1-Loss calculated with pseudo-observations**. **Bold** indicates the highest score among feature sets (within-column). Results are reported as Metric±Std.Dev.

Features			**CoxPH**			**N-MTLR**		
Clinical Features	AggSeq Features	Seq Features	Concordance	IBS	l1-PO	Concordance	IBS	l1-PO
Bare	-	-	0.6652±0.001	0.1526±0.001	1425.9±142.4	0.6937±0.003	0.1514±0.001	1183.3±16.9
Detailed	-	-	0.7052±0.001	0.1472±0.001	2304.9±1561.1	0.7223±0.003	0.1427±0.001	1156.0±16.5
-	✓	-	0.7109±0.001	0.1474±0.001	**1224.4±2.9**	0.7167±0.003	0.1416±0.001	1154.8±20.7
Bare	✓	-	0.7205±0.001	0.1443±0.001	1311.8±139.1	0.7245±0.005	0.1383±0.001	1133.0±19.3
Detailed	✓	-	**0.7305±0.001**	**0.1417±0.001**	1641.5±741.8	0.7322±0.005	0.1355±0.001	1114.5±19.5
-	-	✓	-	-	-	0.7458±0.004	0.1319±0.001	1108.5±17.9
Bare	-	✓	-	-	-	0.7504±0.001	0.1313±0.001	1105.0±17.7
Detailed	-	✓	-	-	-	**0.7533±0.004**	**0.1310±0.001**	**1104.9±15.4**

curves generated by the two ISD models (CoxPH and N-MTLR) for three example patients can be seen in Figure 4.7. None of the reported models are 1-calibrated at 30 days (or D-calibrated) with a test set size of 84217 or 84218. Plots showing the bins used to calculate the 1-calibration and D-calibration test statistics can be see in Figure 4.8 for our best-performing model—N-MTLR with **Detailed+Seq** features.

4.5.3 Binary Readmission Models: Feature Importances

Feature importances according to model "gain" can be extracted from a trained XGBoost classifier. To rank a feature by "gain" is to calculate the increase in accuracy before and after a branch is split on that attribute, averaged over each tree in the ensemble model. We examine the ten most important features according to gain for our XGBoost models trained on **AggSeq+Bare** (dimension of 100+14 after one-hot-encoding) and **Aggseq+Detailed** (dimension of 100+59 after one-hot-encoding) features.

When only **Bare** clinical features are included alongside **AggSeq** features, five of the ten highest ranking features come from the automatically-generated **AggSeq** vector. This decreases to 3/10 when the entire suite of **Detailed** hand-engineered features are included as predictors alongside the **AggSeq** feature vector. The LACE score appears as the most important feature from the **AggSeq+Bare** set, and the second-most important input feature out of all **AggSeq+Detailed** features. The gain was not distinguishable from zero for features indicating male sex (for **AggSeq+Bare**) and some previous-admission discharge dispositions (`02`, `03`, `06`, `12`), the presence of a GP visit in the previous 2, 3, and 4 years, and the presence of hypertension and asthma (for **AggSeq+Detailed**).

4.5.4 Pre-Index Readmission Prediction Models

We test XGBoost, CoxPH, Deep Learning, and N-MTLR on the task of predicting readmissions from only information within our study period captured *at or before the moment of admission*. The baseline LACE model was not included, as one cannot calculate the index admission length-of-stay (**L**) until discharge. Models that include features from the duration of the index admission consistently perform more favourably than models only using information from the time of admission and before (Figure 4.9). The AUROC of the best-performing model overall (**Detailed+Seq** N-MTLR) decreased from 0.8491 ± 0.004 to 0.7643 ± 0.003. As before, when **Seq** features are included, N-MTLR achieves a better AUROC than the otherwise equivalent DNN models (**Seq**: $p = 0.0003$, **Seq+Bare**: $p = 0.003$, **Seq+Detailed**: $p = 0.00009$, respectively). Differently than with `incl_idx` features, the DNN model trained on **Detailed+Seq** input is out-performed by the DNN model trained on **Detailed+AggSeq** inputs: 0.7429 ± 0.005 vs. 0.7485 ± 0.004, $p = 0.01$. This is also true for DNN models with **Bare+Seq** vs. **Bare+AggSeq**: 0.7344 ± 0.008 vs. 0.7421 ± 0.005, $p = 0.01$. All scores (AUROC, AUPRC, and Brier at 30-days) are reported in Table 4.5.

Table 4.6 provides ISD-specific evaluations. The highest concordance statistic using only `b4idx` features (0.7162 ± 0.002) is achieved by N-MTLR paired with **Detailed+Seq** features. N-MTLR with **Detailed+Seq** features also achieves the lowest IBS and L1-loss (0.1427 ± 0.001; 1140.9 ± 11.2 days). None of the models are

1-calibrated at 30 days or D-calibrated.

Table 4.5: `b4idx` **AUROC, AUPRC, and Brier score evaluations of different combinations of models and feature sets at 30 days**. **Bold** indicates the highest score within model (column). Results are reported as Metric±Std.Dev.

b4idx Features			**30-Day AUROC±Std.Dev**			
Clinical Features	AggSeq Features	Seq Features	XGBoost	CoxPH	DL	N-MTLR
Bare	-	-	0.7008±0.004	0.5937±0.004	0.6631±0.017	0.6934±0.001
Detailed	-	-	0.7306±0.006	0.6994±0.003	0.7300±0.005	0.7343±0.001
-	✓	-	0.7224±0.005	0.7146±0.004	0.7333±0.003	0.7421±0.004
Bare	✓	-	0.7340±0.005	0.7096±0.003	0.7421±0.005	0.7515±0.004
Detailed	✓	-	**0.7400±0.004**	**0.7247±0.003**	**0.7485±0.004**	0.7585±0.004
-	-	✓	-	-	0.7399±0.002	0.7600±0.003
Bare	-	✓	-	-	0.7344±0.008	**0.7643±0.004**
Detailed	-	✓	-	-	0.7428±0.005	**0.7643±0.003**

b4idx Features			**30-Day AUPRC±Std.Dev**			
Clinical Features	AggSeq Features	Seq Features	XGBoost	CoxPH	DL	N-MTLR
Bare	-	-	0.1176±0.002	0.0784±0.002	0.1007±0.004	0.1074±0.001
Detailed	-	-	0.1520±0.004	0.1248±0.002	0.1511±0.005	0.1607±0.005
-	✓	-	0.1330±0.004	0.1224±0.002	0.1459±0.002	0.1494±0.003
Bare	✓	-	0.1484±0.003	0.1226±0.003	0.1582±0.005	0.1659±0.002
Detailed	✓	-	**0.1577±0.003**	**0.1418±0.001**	**0.1694±0.003**	0.1789±0.001
-	-	✓	-	-	0.1455±0.002	0.1714±0.004
Bare	-	✓	-	-	0.1426±0.006	0.1816±0.003
Detailed	-	✓	-	-	0.1608±0.005	**0.1829±0.003**

b4idx Features			**30-Day Brier Score±Std.Dev**			
Clinical Features	AggSeq Features	Seq Features	XGBoost	CoxPH	DL	N-MTLR
Bare	-	-	0.0467±0.001	0.0481±0.001	0.0472±0.001	0.0468±0.001
Detailed	-	-	**0.0459±0.001**	0.0467±0.001	0.0458±0.001	0.0454±0.001
-	✓	-	0.0465±0.001	0.0466±0.001	0.0459±0.001	0.0457±0.001
Bare	✓	-	0.0461±0.001	0.0468±0.001	0.0456±0.001	0.0455±0.001
Detailed	✓	-	**0.0459±0.001**	**0.0463±0.001**	**0.0452±0.001**	0.0451±0.001
-	-	✓	-	-	0.0458±0.001	0.0451±0.001
Bare	-	✓	-	-	0.0459±0.001	0.0451±0.001
Detailed	-	✓	-	-	0.0455±0.001	**0.0448±0.001**

Table 4.6: `b4_idx` **comparison of the ISD models, CoxPH and N-MTLR, according to the Concordance, Integrated Brier Score (IBS) and L1-Loss calculated with pseudo-observations for censored individuals**. **Bold** indicates the highest score among feature sets (within-column). Results are reported as Metric±Std.Dev.

b4idx Features			**CoxPH**			**N-MTLR**		
Clinical Features	AggSeq Features	Seq Features	Concordance	IBS	l1-PO	Concordance	IBS	l1-PO
Bare	-	-	0.6241±0.001	0.1589±0.001	1360.0±111.6	0.6611±0.003	0.1575±0.001	1208.9±19.4
Detailed	-	-	0.6878±0.001	0.1502±0.001	1388.0±357.6	0.7015±0.002	0.1488±0.001	1175.1±21.9
-	✓	-	0.6907±0.001	0.1504±0.001	**1227.2±12.4**	0.6821±0.007	0.1495±0.001	1188.9±20.6
Bare	✓	-	0.6926±0.000	0.1502±0.001	1493.2±400.8	0.7011±0.002	0.1470±0.001	1168.2±12.4
Detailed	✓	-	**0.7058±0.000**	**0.1471±0.001**	1401.6±253.5	0.7083±0.002	0.1446±0.001	1155.4±14.1
-	-	✓	-	-	-	0.7103±0.003	0.1443±0.001	1151.8±18.1
Bare	-	✓	-	-	-	0.7157±0.004	**0.1427±0.001**	**1140.9±11.2**
Detailed	-	✓	-	-	-	**0.7162±0.002**	0.1429±0.002	1147.9±17.7

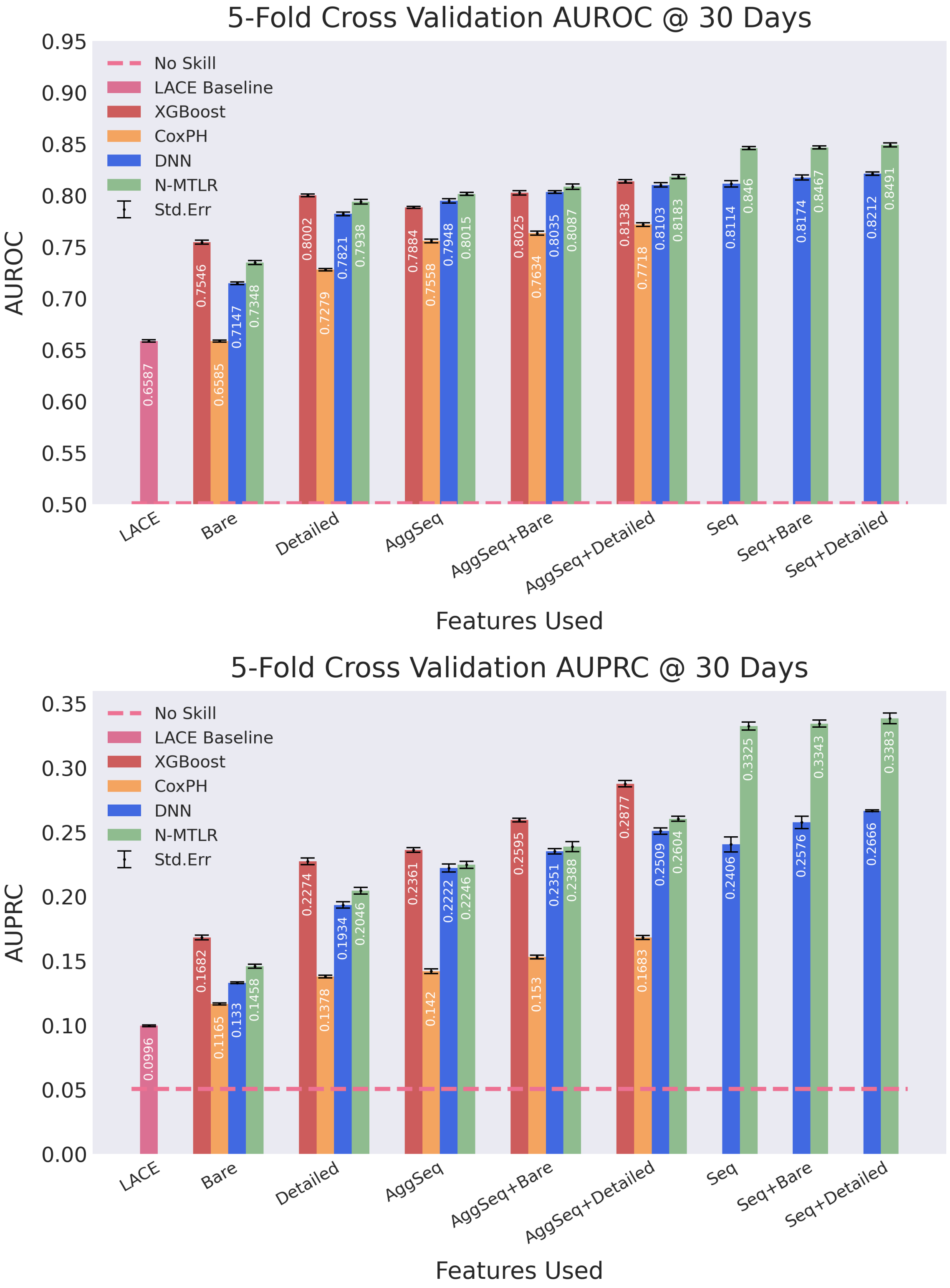


Figure 4.6: **Plots showing AUROC and AUPRC scores using different feature and model combinations.**

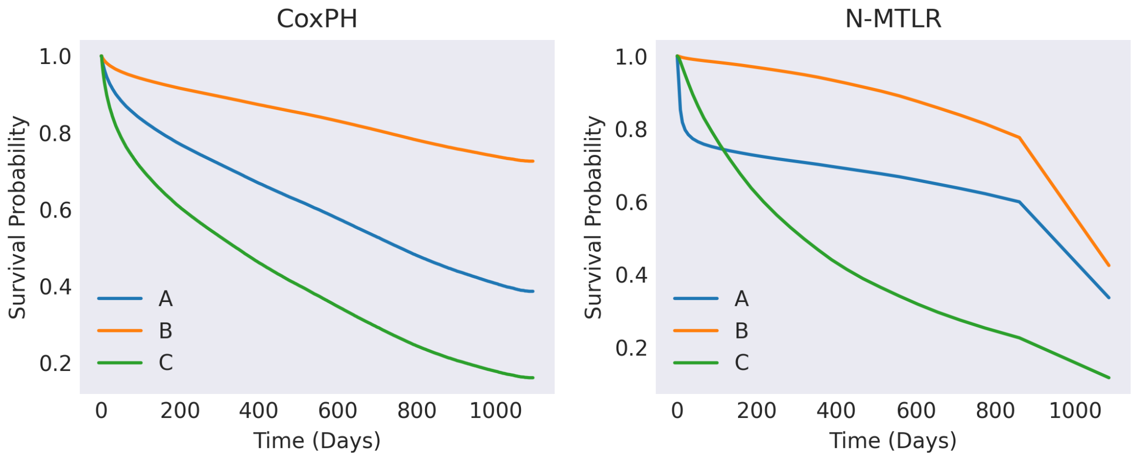


Figure 4.7: **Survival curves generated for three example patients by ISD models: CoxPH and N-MTLR.** According to the proportional hazards assumption relied on by CoxPH, the curves for different patients will never cross.

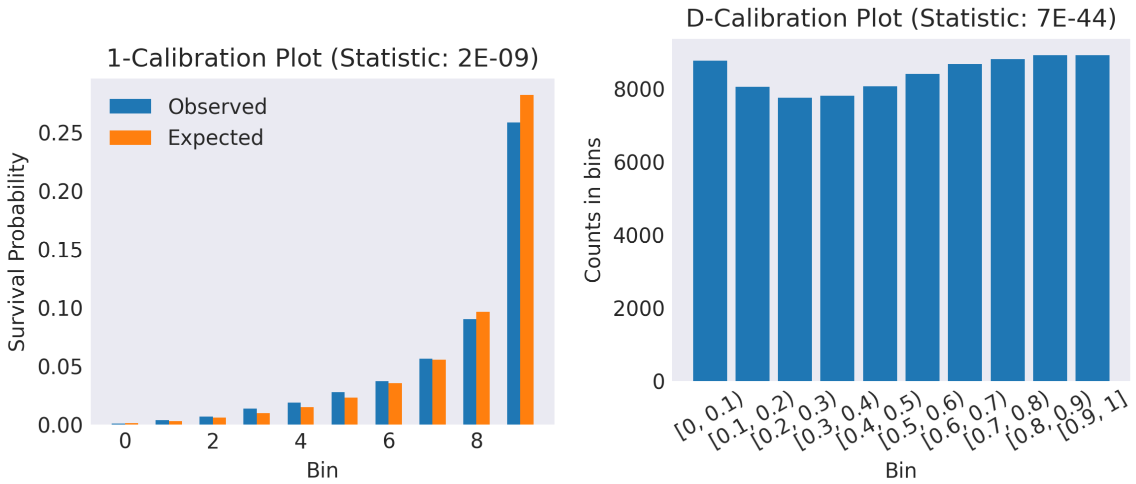


Figure 4.8: **1- and D-Calibration plots**. This figure summarizes sizes of bins used to calculate the test statistics for 1-calibration and D-calibration for our best-performing model—N-MTLR using **Detailed+Seq** features.

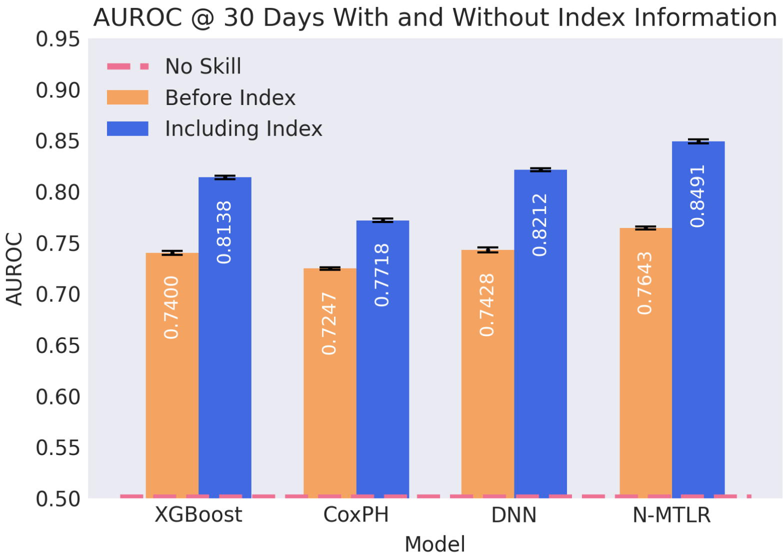


Figure 4.9: **AUROC evaluations of different using different models with and without `b4idx` features**. **Bold** indicates the highest score within model (column). XGBoost and CoxPH models used **AggSeq+Detailed** features. DNN and N-MTLR models used **Seq+Detailed** features.

Chapter 5

Discussion

Feature Sets and Model Performance

Five sets of features are compared in this study: **LACE**, **Bare** clinical features, **Detailed** clinical features, **AggSeq** representations (based on summed embeddings of recently acquired medical codes) and raw strings containing the patient's sequence of previous medical codes (**Seq**). Models able to utilize the **Seq** strings showed better performances than those that did not, even in the absence of **Bare** and **Detailed** clinical features. Adding **Bare** features to models already utilizing **Seq** features did not noticeably improve performance (AUROC 0.8460 versus 0.8467)—indicating that much of this information (such as age and sex) is already encoded in an individual's medical event history. All sets of features showed stronger predictive power than the LACE score across all evaluation metrics. However, the LACE score was an important tabular input feature for the XGBoost models, even in the presence of **Detailed** manually engineered features or the 100-element **AggSeq** vector containing information about recent medical events. This demonstrates that important medical information can indeed be contained in simple heuristics, despite achieving only mediocre performance as a single predictor.

Within a Medical Code Embedding dictionary trained on co-occurrence data, codes that point to similar medical events will be close in vector space. While this idea is intuitive, one may wonder whether the creation of these embedding vectors is adequately helpful in the context of this study. The MCE dictionary is utilized in this project in two ways—one, to generate the **AggSeq** patient representation (which can be paired with any machine learning model)—and two, to initialize the sequence models that learn target-informed representations of patients directly from medical codes. The addition of **AggSeq** features to the **Detailed** feature representation improves performance, implying that representational gaps exist between the 38 hand-calculated features chosen to represent diverse medical history information. Also, with even the most conservative estimates of performance, if **AggSeq** features are paired with the **Bare** features available within an EHR, the performance at least matches the **Detailed** feature set performance (which also includes **Bare** features—see Section 3.3.3)[1]. In conclusion, the results of this study imply that the MCE **AggSeq+Bare** features can either replace **Detailed** features and achieve the same performance on all-cause readmission prediction tasks, or that **AggSeq** features can supplement the **Detailed** features and improve all-cause readmission prediction performance[2].

Another merit of the MCE dictionary (especially to generate tabular **AggSeq** features) lies in its simplicity of use, which is makes it relatively easy to explain. If a doctor understands the intuition behind the distributional hypothesis (codes co-occurring are assigned similar sets of numbers), the intuition behind summing code vectors to represent an individual follows easily. If the resulting **AggSeq** features are still not penetrable enough (i.e., the user needs to know what each element of the 100-dimensional embedding vector indicates), Word2Vec embeddings can be generated with a non-negative constraint (as seen in 2016 by E. Choi [39]), such that the top k codes that have the largest values for the i^{th} coordinate of the embedding can be used to "explain" that element. A physician could then intuit what each

[1] The summation of one-hot vectors representing a patient's historical medical codes could be used in the place of the MCE-informed AggSeq representation. However, this would have a dimension of at least 3000 (one index for each medical code); using these sparse vectors can be computationally difficult and often lead to lower performance.

[2] Which one is better? The answer to this question depends on how a hospital administrator values small gains in performance against extra effort spent on engineering a more detailed representation.

element is “measuring” based on the most associated medical concepts—this allows for a level of decipherability not often seen with machine-generated features. Beyond informing the **AggSeq** patient representation, the Medical Concept Embedding dictionary also increases performance when used to initialize the embedding layer of a sequence model, contributing to the best performing model reported in this dissertation. Initializing machine learning models with baseline knowledge frequently leads to increased performance; models that use lower-dimensional inputs also train fewer parameters, which can decrease overfitting. Embedding dictionaries are especially helpful when the coding system contains multiple codes for the same concept. This is common both in natural language (with synonyms) and medical codes[3].

There are two general axes on which categorize the models used in this study: *lightweight* (LACE, XGBoost, and CoxPH) versus *heavyweight* (DNN and N-MTLR), and **estimates a single value** (LACE, XGBoost, DL) versus **estimates an ISD** (CoxPH and N-MTLR) (Figure 5.1, Table 4.1). The best lightweight model (XGBoost using **AggSeq+Detailed** features) performs admirably with an AUROC of 0.8138, out-performing the other lightweight model, CoxPH, by a significant margin for 30-day readmission prediction. Interestingly, the lightweight XGBoost also out-performs the heavyweight DNN approach when relying only on clinical features. XGBoost is highly regarded as a classification powerhouse (especially when using categorical features as input), so this result is not surprising. When comparing the two heavyweight models, N-MTLR significantly out-performs the DNN approach given the same model architecture (aside from the predictor module) when given **Seq** features. This speaks to the power of using a time-to-event target and/or modeling event probabilities at multiple timepoints when learning from sequential past events. Every machine learning model—even the lightweight models paired with only the barest clinical features—noticeably out-perform the **LACE** baseline. Readmission prediction models with a 30-day AUROC score of greater than 0.8 are generally considered

[3] There is a nearly analogous ICD-10-CM code (NACRS and DAD) for every ICD-9 code (Claims)

to be very good. All models but except LACE and Cox have the capacity meet this threshold using (at minimum) the **Detailed** and/or **Bare+AggSeq** features sets.

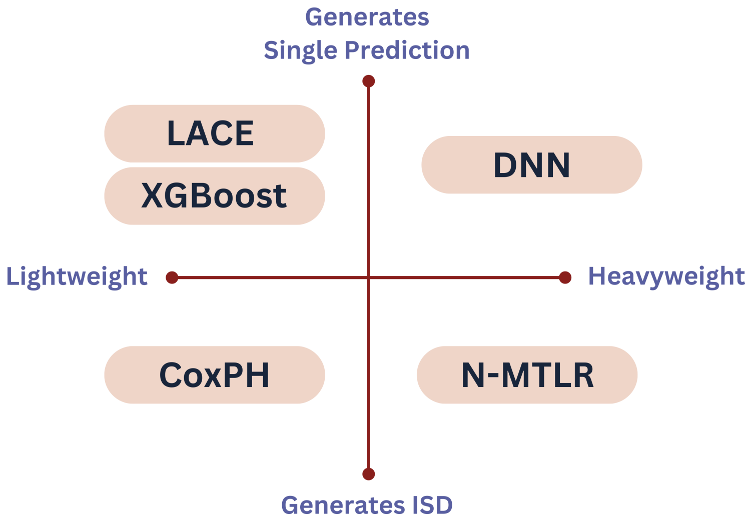


Figure 5.1: **Categorizing models used in this study.** Each model either relies on deep learning (heavyweight) or does not (lightweight) and either generates a single value (risk score or probability at timepoint) or a range of values over multiple timepoints.

The c-statistic/c-index/concordance, as used in this study, is (roughly) a time-agnostic version of the 30-day AUROC score. The former measures how well the model ranks patients by median survival time, and the latter measures how well the model ranks patients by their 30-day readmission probability. Similarly, the Integrated Brier Score measures the model's average brier score over time. We see that in both cases, the metric specific to the 30-day time-point indicates better performance than its time-independent counterpart. This likely implies that the time-specific AUROC and Brier scores for our ISD models drop as time progresses. This makes sense given the days-until-readmission and censoring distribution of our dataset (Figure 4.3)—very little information about readmission events exist for the models to learn from nearing three years post-discharge, which may make the task of learning far-future prognosis more difficult. Therefore, readmission-free survival probabilities towards the tail end of the curve should be interpreted with appropriate caution.

Two calibration-specific measures were used to evaluate our models (1-calibration @ 30 days and D-calibration), but no combination of architecture and feature set produced a calibrated model according to either test. Despite the $p < 0.05$ (for 30-day 1-calibration especially) high or moderately-high agreement is shown between expected and observed bins (Figure 4.8). Hosmer-Lemeshow or χ^2 tests used for these calibration calculations can be too sensitive for a large-population-level dataset; it is possible that when evaluated on a sample of the test set, the models would be appear both 1 and D-calibrated.

N-MTLR consistently generates higher quality ISDs than CoxPH using the same input features, despite both learning from a time-to-event dataset. This can be partially explained by the neural layers that augment the MTLR module, allowing the learning of richer input feature representations—in contrast, CoxPH is restricted by the assumption that relationships between features are linear. Another major drawback of Cox models is the reliance on the proportional hazards assumption. This dictates that the impact of any particular feature does not change with the progression of time, therefore the survival curves for different patients will never cross. N-MTLR is not restricted by such an assumption. Figure 4.7 shows a situation where Cox ranks patient C the riskiest for readmission, A the second riskiest, and B the lowest risk. This ranking remains the same every time-point. Differently, N-MTLR understands that patient A may be highly likely to be readmitted early post-discharge, however, if they are not readmitted in this risky time-frame, their probability of event-free survival begins to plateau. At 30 days, N-MTLR ranks A the riskiest, then C, then B. The curves for A and C cross at around the 150 day mark, which changes the ranking to match Cox's. Thus, N-MTLR is able to rank patients more flexibly, likely leading to better AUROC, concordance, and other scores. This shows that using time-to-event data is not enough to achieve state-of-the-art performance—the model chosen must also be able to adequately represent patients' past and futures.

N-MTLR, allowed to learn from sequences of historical medical codes (**Seq** inputs), is the most promising model and feature combination explored in this study. At predicting the presence of a readmission by 30 days, this combination out-performs all other models (including the deep learning approach) trained **specifically** to predict 30-day readmissions. This can likely be explained by three factors. First, the survival dataset's time-to-event target provides granular information about post-discharge progression. The model can learn to predict a lower probability of readmission at 30 days for someone whose time-to-readmission is 600, compared to an individual with a time-to-readmission of 60. Second, specifically modeling probabilities at many future time-points (both before and after 30 days) may cause the model's 30-day readmission prediction to consistently fall closer to the actual observation. Third, the combination of accessing a patient's sequence of past events (including codes related to the passage of time) pairs uniquely well with an algorithm that models a sequence of event-free survival probabilities in the future. This favourable combination of sequential event-related inputs and modeling event probabilities at all future time-points could likely improve performance on other tasks as well: both medical (such as modeling cancer progression) and otherwise (e.g., predicting time until an industrial machine fails given past sequences of evaluations and repairs).

We evaluate the performance of all our models using features *from the entirety of the index admission and prior* (`incl_idx`) and features only using information from *at or before the time of admission* (`b4_idx`). With the `b4_idx` restriction, performance decreased regardless of model and granularity of features used. Our best model (N-MTLR) was able to achieve a AUROC of 0.7643 and an AUPRC of 0.1829 using **Detailed+Seq** features. The general decrease in scores may be based on a number of factors. One, information associated with the urgency of a hospital admission can be highly predictive for future hospitalizations, and is absent (except for capturing the emergent status of the hospital visit) in `b4_idx` features. Additionally, the *number* of features available to learn from was smaller—for example, numeric features characterizing hospital usage during the index admission (e.g., number of unique drugs prescribed whilst in the hospital) were foregone entirely in our **Detailed** `b4_idx` feature set. Two, for some patients, their index admission was their first and only hospitalization during the study period. This means that their `b4_idx` **Seq** and **AggSeq** representations only contained information from a single code ("`ADM`"), further decreasing the useful information. Despite this drop in performance, even our lightweight models (XGBoost AUROC: 0.7400, AUPRC: 0.1577) using `b4_idx` **AggSeq+Bare** features out-performs **LACE** (AUROC: 0.6587, AURPC: 0.0996), which relies on features from the entirety of the hospital episode. Additionally, even lightweight models trained on only **Bare** `b4_idx` features out-perform the `incl_idx` **LACE** score. This motivates the adoption of recent machine learning techniques in practical settings, as more sophisticated models can offset the drop in accuracy from temporally restricting feature inputs. This leads to more clinically useful models without compromising performance.

Many studies engage with problems similar to this one, but comparing performances between studies using restricted-access medical data is difficult. Even within studies that use EHR data from AHS, data extraction and pre-processing will almost necessarily differ in subtle ways. The years of the study period used is one variable, as is the target population exclusion criteria, along with the methods for selecting the index admissions or combining admissions into an index episode, and how readmission status and/or time-to-event targets are calculated. Further, studies may opt to predict readmissions at six months or one year, or use a different lookback time period (how much medical history consider as model input). Evaluation metrics reported can also vary wildly. All these factors make it difficult to make claims about a model architecture's merit in comparison to those already published. However, comparing against a common baseline (such as LACE) places new scores in the context of what is applied in practice, lending credence to our proposed approaches. Using a publicly available dataset such as MIMIC[4]) is one approach some authors choose to partially

[4] https://physionet.org/content/mimiciii/1.4/

overcome this problem.

The mean and standard deviation of each combination of model and feature-set were computed using external five-fold cross validation. Standard deviation and standard error are intuitive measures of spread, but bootstrapping is generally preferred to report a non-parametric 95% confidence interval for model performance. In our case, bootstrapping was not computationally feasible, as the MCE dictionary must be calculated for every sampled training set to avoid potential data leakage.

Data Sources and Target Definition

One of the strengths underlying this study is the data our machine learning models rely upon. The province of Alberta has a single payor, universally accessible, integrated health system, which enables the collection of comprehensive administrative data with minimal loss to follow-up. However, some limitations exist—most notably, linked administrative data is a less complete and less detailed information source than comprehensive electronic health records. Accordingly, information from the latter type of repository (such as narrative physician and allied health notes) may further improve prediction accuracy if incorporated. Revisiting the approaches in this study as Alberta continues robust data collection with ConnectCare may also lead to further performance improvements.

There exist many nuances in deciding how to define the adverse outcome of all-cause readmissions to the hospital. One regards the study population whose outcomes we are interested in predicting. As detailed in Section 3.1.2, we do not restrict our study population to a particular disease or population cohort. However, our index episode selection criteria does preclude hospital admissions on the basis of psychiatric admissions and routine admissions related to childbirth. Our models make predictions for all age ranges, covering both medical and surgical admissions, and is otherwise not limited to patients with a specific condition. As motivated in Chapter 2, all-cause readmission models may be more amenable to the general hospital setting.

Previous studies distinguish between “planned” vs. “unplanned” readmissions, or even “preventable” vs. “unpreventable” readmissions. There exists no agreed-upon algorithm for reliably determining readmissions that fall under the “preventable” category, although attributes of a preventable readmission may include proximity to the index admission, similarity to index admission in cause, and evidence of complications from care given in the index admission [94]. Given the degree of extra consideration this label requires to calculate (and the lack of guarantee the label would be correct), our readmission definition includes readmissions that may not have been “preventable”. In the Discharge Abstract Database (DAD), the code `ADMITCAT` (Admit Category) classifies the visit type as either “Elective” or “Urgent/Emergent”, which would allow an analyst to determine if a hospitalization was “planned” or “unplanned”. Unfortunately, this variable was not available to us at the time of initial data extraction, nor was it available in a follow-up data extraction in later months. Another way to determine whether a hospitalization was “planned” vs. “unplanned” is to look at emergent status—whether the patient arrived at the institution through the use of ambulatory services. This would be determined by searching the NACRS database for records of an emergent admission that coincided with the day of (or day previous to) the beginning of the hospitalization episode[5]. However, many individuals are triaged through the Emergency Department in serious condition after arriving at the hospital using their own means of transportation—if we were to restrict readmissions to those that arrived by ambulance, this would be failing to capture many unplanned and potentially life-threatening admissions.

Therefore, another limitation of this study is that our models may be learning associations between index admissions and planned follow-ups. In certain scenarios, this could lead to an overestimation of model performance on the cohorts we really care about—those who may benefit from targeted interventions. One example of this could be that patients with a certain condition, when admitted for a procedure, are

[5] This is indeed how we calculated the **A** in the LACE score – see Section 3.3.3.

recommended to visit the hospital for a follow-up procedure within six weeks. The model may be able to encode this relationship, and achieve a higher performance for the positive class because these individuals are included with those who are urgently readmitted. Regardless, our models' out-performance of the LACE baseline—which is (1) implemented in practice, (2) contains comorbidity information, and (3) is tested against our models using the same readmission definition—still indicates a positive delta of usefulness regardless of exact performance numbers. In the very most conservative of estimates, where our models only match LACE performance on the cohort we care about, our ISDs still provide more information for physicians than the LACE risk score. Further analysis, and access to the "ADMITCAT" code would be necessary to more completely understand this relationship.

Perhaps the most difficult peculiarity of our data to disentangle is that it only captures within-hospital deaths. This means that patients who were discharged and passed away are treated identically to those who were discharged and never readmitted. The former cohort are likely in dire condition in some way (which is likely captured in their medical history-based features), and the latter are likely associated with features that indicate relative health. The fact these two cohorts are so different yet are categorized in the same way could be hurting model performance by obscuring the boundary between health and sickness. One could expect that reformulating this problem as modeling time until readmission **or death** (as is seen often in the literature) would more effectively group patients, and could allow for further improvements at evaluation time.

Although our results can be considered generalizable to other single-payor, universally accessible health systems (such as those in other Canadian provinces), generalization beyond this setting should be performed with caution. Population size, available input features, and data quality will vary between institutions, as will medical code embedding dictionaries given differences in coding schema. These, plus differences in location-based demographics could cause a decrease in model performance.

Readmission-Prediction Models in Practice

For an AI system to be adopted in a medical context, health administrators must decide that the perceived benefits of adopting these systems outweigh potential drawbacks. These drawbacks can be numerous. Operationalizing these systems is expensive and often requires highly qualified personnel for implementation and maintenance. Once a system is in place, healthcare professionals must be trained to use it. The topic of blame surrounding medical AI causes uneasiness—if a medical mistake is caused by a machine learning system, the legality of fault is still murky, as the accountable parties have not yet been agreed upon. Perhaps most crucially, trust in AI and AI systems (from health practitioners and the public) has not had time to adequately develop, given the recency with which widespread AI adoption has erupted. The goal of this project is to propose tools and explore approaches that will help **tip the scale** in the favour of AI, and may one day inform medical decisions regarding rehospitalization management.

There are a number of attributes that make Individualized Survival Distributions favourable for a clinical setting (e.g., implemented within an electronic health record), beyond demonstrating improved performance by exploiting time-to-event targets. One is that the ISD stands alone: it is useful in the absence of information about the wellbeing of other patients for comparison. The same cannot be said for models that compute a risk score in isolation. Second, the information captured in an ISD is made available in a way that is visual and intuitive. A physician would not have to parse through numbers or text to understand the prognosis of an individual. Finally, survival curves provide insight into a far broader number of questions than other classes of models; questions such as "How many days may pass until patient x will be readmitted?", "What is the probability patient x will be readmitted in 30 days? What about 9 weeks?", "How risky is x for readmission **within 1 year** compared to patient y? How risky is x compared to y for a readmission **overall**?" can all be tackled with an ISD. The ISD is elegant in its ability to convey comprehensive, self-contained survival information hastily. Further, if the desired outcome is understanding how differing demographics or treatments may affect survival (a question more aligned with *survival analysis*), ISDs can be aggregated to simulate the information of a Kaplan Meier plot. A log-rank test could then similarly be applied to these aggregated survival curves to inspect group effects.

To implement any of our models within an EHR, we would need to link administrative data to create patient sentences, compute the numeric representations of the sentence components (e.g., diagnosis codes, procedure codes, etc), and build the prediction model *a priori*. Once the model is trained, we can make a prediction for a new patient by first converting his/her records into a sentence (thereby creating our **Seq** inputs), computing the **AggSeq** features of the patient (by using the previously obtained MCE dictionary), and computing the **Detailed** clinical features if required. To save on computation, a running **Seq** and/or **AggSeq** representation may be stored in a patient's EHR, and updated at the onset of new medical events. Note that all steps after model training can be automated in practice, and require the same privileges as the LACE model, as both rely on linked administrative data. To facilitate early care-trajectory planning, an initial prediction (either a 30-day readmission probability or an ISD) could be made on the day of admission using a `b4_idx` version of the model. Then, on the day of discharge (incorporating information from the duration of the hospital episode), a new prediction could be made, which would allow the attending physician to tweak the care strategy based on model outputs if appropriate. The MCE dictionary (Section 3.2.2) and prediction models should be occasionally re-trained to account for data drift and/or new medical codes entering common use.

Chapter 6

Conclusions

This study seeks to improve upon the individualized 30-day all-cause hospital readmission prediction task. We do this by 1) exploring methods of effectively representing a patient's sequential medical history using machine learning, and 2) utilizing algorithms that can model a patient's readmission-free survival prognosis using time-to-event data. An intentional problem-solving approach was followed throughout the course of this study; practical issues surrounding adoption and specific needs and desires of the medical-end user were considered, including the inclination towards models that predict readmission at the beginning of a hospital episode. Our proposed methods are tested and validated on retrospective population-wide linked administrative data from Alberta Health Services.

To represent a patient automatically, we consider aggregating vector embeddings representing various events (such as diagnoses, prescriptions, and procedures) from their medical histories. Pairing these computer-generated representations with simple clinical features showed potential for increased prediction accuracy over the combination of simple and manually-calculated features, and over the LACE model, a practical single-scalar heuristic implemented to predict readmissions in Alberta hospitals. We show further improved performance using deep-learning models that can learn directly from the sequence of historical medical codes. However, all aforementioned models learn to predict the probability of a patient experiencing a hospital readmission at or before 30 days, thereby viewing the problem as a binary classification task. Neural Multi-Task Logistic Regression (N-MTLR), a survival prediction algorithm, learns from time-to-readmission information, and makes readmission-free survival probability predictions at all future time-points. When allowed to learn representations of patients from longitudinal medical event histories, N-MTLR out-performs all models trained specifically on the task of predicting readmissions at 30-days, at the task of 30-day readmission prediction. Ultimately, we show that state-of-the-art 30-day all-cause readmission prediction performance (compared to gradient-boosting and deep-learning approaches) can be achieved through jointly learning from past sequential events and modeling prognosis across a sequence of future time-points.

There are many possible future directions for this work. One may involve validating N-MTLR and sequential medical code inputs with a publically available dataset to better understand where this approach sits in the context of available research. Another is addressing limitations by restricting our dataset to only *emergent* index admissions, and by incorporating deaths into the definition of our "adverse outcome of interest" alongside readmissions. Exploring the potential of longitudinal event-based features and N-MTLR for other problem settings (medical and otherwise) may also prove worthwhile. We note that deep-learning architecture and hyper-parameter tuning was not performed in this study—therefore, our numbers are a lower bound of what is possible using this approach. For example, an attention layer may further improve performance. A computationally-expensive grid-search would elucidate this gap. Finally, studying how physicians would interact with a tool like this, and how to best explain its use, would contribute valuable insights into barriers preventing the widespread adoption of medical artificial intelligence.

Bibliography

[2] S. Davis *et al.*, "Effective hospital readmission prediction models using machine-learned features," *BMC Health Services Research*, vol. 22, no. 1, p. 1415, 2022.

[3] P. Garrett and J. Seidman, *EMR vs EHR – What is the Difference?* en-US, Jan. 2011. [Online]. Available: https://www.healthit.gov/buzz-blog/electronic-health-and-medical-records/emr-vs-ehr-difference (visited on 04/09/2023).

[4] S. Phung, A. Kumar, and J. Kim, "A deep learning technique for imputing missing healthcare data," in *2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, IEEE, 2019, pp. 6513–6516.

[5] D. J. Cartwright, *Icd-9-cm to icd-10-cm codes: What? why? how?* 2013.

[6] B. Slinker and S. Glantz, "Multiple regression for physiological data analysis: The problem of multicollinearity," *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 249, no. 1, R1–R12, 1985.

[7] C. Van Walraven *et al.*, "Derivation and validation of an index to predict early death or unplanned readmission after discharge from hospital to the community," *Cmaj*, vol. 182, no. 6, pp. 551–557, 2010.

[8] M. E. Charlson, P. Pompei, K. L. Ales, and C. R. MacKenzie, "A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation," *Journal of chronic diseases*, vol. 40, no. 5, pp. 373–383, 1987.

[9] E. Mahmoudi, N. Kamdar, N. Kim, G. Gonzales, K. Singh, and A. K. Waljee, "Use of electronic medical records in development and validation of risk prediction models of hospital readmission: Systematic review," *bmj*, vol. 369, 2020.

[10] C. van Walraven, J. Wong, and A. J. Forster, "Lace+ index: Extension of a validated index to predict early death or urgent readmission after hospital discharge using administrative data," *Open Medicine*, vol. 6, no. 3, e80, 2012.

[11] J. D. Donzé *et al.*, "International validity of the hospital score to predict 30-day potentially avoidable hospital readmissions," *JAMA internal medicine*, vol. 176, no. 4, pp. 496–502, 2016.

[12] J. F. Graumlich, N. L. Novotny, and J. C. Aldag, "Brief scale measuring patient preparedness for hospital discharge to home: Psychometric properties," *Journal of Hospital Medicine: An Official Publication of the Society of Hospital Medicine*, vol. 3, no. 6, pp. 446–454, 2008.

[13] A. G. Au, F. A. McAlister, J. A. Bakal, J. Ezekowitz, P. Kaul, and C. van Walraven, "Predicting the risk of unplanned readmission or death within 30 days of discharge after a heart failure hospitalization," *American heart journal*, vol. 164, no. 3, pp. 365–372, 2012.

[14] E. F. Philbin and T. G. DiSalvo, "Prediction of hospital readmission for heart failure: Development of a simple risk score based on administrative data," *Journal of the American College of Cardiology*, vol. 33, no. 6, pp. 1560–1566, 1999.

[15] O. K. Nguyen *et al.*, "Predicting all-cause readmissions using electronic health record data from the entire hospitalization: Model development and comparison," *Journal of hospital medicine*, vol. 11, no. 7, pp. 473–480, 2016.

[16] C. A. Eastwood, J. G. Howlett, K. M. King-Shier, F. A. McAlister, J. A. Ezekowitz, and H. Quan, "Determinants of early readmission after hospitalization for heart failure," *Canadian Journal of Cardiology*, vol. 30, no. 6, pp. 612–618, 2014.

[17] R. Wallmann, J. Llorca, I. Gómez-Acebo, Á. C. Ortega, F. R. Roldan, and T. Dierssen-Sotos, "Prediction of 30-day cardiac-related-emergency-readmissions using simple administrative hospital data," *International journal of cardiology*, vol. 164, no. 2, pp. 193–200, 2013.

[18] P. Zhao, I. Yoo, S. H. Naqvi, *et al.*, "Early prediction of unplanned 30-day hospital readmission: Model development and retrospective data analysis," *JMIR medical informatics*, vol. 9, no. 3, e16306, 2021.

[19] M. Tonelli *et al.*, "Methods for identifying 30 chronic conditions: Application to administrative data," *BMC medical informatics and decision making*, vol. 15, no. 1, pp. 1–11, 2016.

[20] R. P. Yerex and Z. Terner, "A predictive model of patient readmission using combined icd-9 codes as engineered features," in *Federal Committee on Statistical Methodology Research Conference*, 2015.

[21] C. Sideris, N. Alshurafa, M. Pourhomayoun, F. Shahmohammadi, L. Samy, and M. Sarrafzadeh, "A data-driven feature extraction framework for predicting the severity of condition of congestive heart failure patients," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015, pp. 2534–2537.

[22] M. Grzyb *et al.*, "Multi-task cox proportional hazard model for predicting risk of unplanned hospital readmission," in *2017 Systems and Information Engineering Design Symposium (SIEDS)*, IEEE, 2017, pp. 265–270.

[23] T. Tran, T. D. Nguyen, D. Phung, and S. Venkatesh, "Learning vector representation of medical objects via emr-driven nonnegative restricted boltzmann machines (enrbm)," *Journal of biomedical informatics*, vol. 54, pp. 96–105, 2015.

[24] A. Rajkomar *et al.*, "Scalable and accurate deep learning with electronic health records," *NPJ digital medicine*, vol. 1, no. 1, p. 18, 2018.

[25] J. Jones, M. Pradhan, M. Hosseini, A. Kulanthaivel, M. Hosseini, *et al.*, "Novel approach to cluster patient-generated data into actionable topics: Case study of a web-based breast cancer forum," *JMIR medical informatics*, vol. 6, no. 4, e9162, 2018.

[26] S. Biswas, *Chatgpt and the future of medical writing*, 2023.

[27] K. Huang, J. Altosaar, and R. Ranganath, "Clinicalbert: Modeling clinical notes and predicting hospital readmission," *arXiv preprint arXiv:1904.05342*, 2019.

[28] S. B. Golas *et al.*, "A machine learning model to predict the risk of 30-day readmissions in patients with heart failure: A retrospective analysis of electronic medical records data," *BMC medical informatics and decision making*, vol. 18, no. 1, pp. 1–17, 2018.

[29] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Deepcare: A deep dynamic memory model for predictive medicine," in *Advances in Knowledge Discovery and Data Mining: 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part II 20*, Springer, 2016, pp. 30–41.

[30] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor ai: Predicting clinical events via recurrent neural networks," in *Machine learning for healthcare conference*, PMLR, 2016, pp. 301–318.

[31] K. Cho *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[32] Q. Suo *et al.*, "A multi-task framework for monitoring health conditions via attention-based recurrent neural networks," in *AMIA annual symposium proceedings*, American Medical Informatics Association, vol. 2017, 2017, p. 1665.

[33] P. Chakraborty *et al.*, "Blending knowledge in deep recurrent networks for adverse event prediction at hospital discharge," *AMIA Summits on Translational Science Proceedings*, vol. 2021, p. 132, 2021.

[34] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *Journal of the American Medical Informatics Association*, vol. 24, no. 2, pp. 361–370, 2017.

[35] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," *Advances in neural information processing systems*, vol. 29, 2016.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[37] Y. Choi, C. Y.-I. Chiu, and D. Sontag, "Learning low-dimensional representations of medical concepts," *AMIA Summits on Translational Science Proceedings*, vol. 2016, p. 41, 2016.

[38] B. Wang, Y. Sun, Y. Chu, D. Zhao, Z. Yang, and J. Wang, "Refining electronic medical records representation in manifold subspace," *BMC bioinformatics*, vol. 23, no. 1, p. 115, 2022.

[39] E. Choi *et al.*, "Multi-layer representation learning for medical concepts," in *proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1495–1504.

[40] C. Xiao, T. Ma, A. B. Dieng, D. M. Blei, and F. Wang, "Readmission prediction via deep contextual embedding of clinical concepts," *PloS one*, vol. 13, no. 4, e0195024, 2018.

[41] C. Pang *et al.*, "Cehr-bert: Incorporating temporal information from structured ehr data to improve prediction tasks," in *Machine Learning for Health*, PMLR, 2021, pp. 239–260.

[42] E. Steinberg, K. Jung, J. A. Fries, C. K. Corbin, S. R. Pfohl, and N. H. Shah, "Language models are an effective representation learning technique for electronic health record data," *Journal of biomedical informatics*, vol. 113, p. 103 637, 2021.

[43] J. Zhang, J. Gong, and L. Barnes, "Hcnn: Heterogeneous convolutional neural networks for comorbid risk prediction with electronic health records," in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, IEEE, 2017, pp. 214–221.

[44] Y. Cheng, F. Wang, P. Zhang, and J. Hu, "Risk prediction with electronic health records: A deep learning approach," in *Proceedings of the 2016 SIAM international conference on data mining*, SIAM, 2016, pp. 432–440.

[45] P Nguyen, T Tran, N Wickramasinghe, and S Venkatesh, "Deepr: A convolutional net for medical records (2016)," *ArXiv160707519 Cs Stat*,

[46] A. Garmendia, M. Graña, J. M. Lopez-Guede, and S. Rios, "Neural and statistical predictors for time to readmission in emergency departments: A case study," *Neurocomputing*, vol. 354, pp. 3–9, 2019.

[47] T. G. Clark, M. J. Bradburn, S. B. Love, and D. G. Altman, "Survival analysis part i: Basic concepts and first analyses," *British journal of cancer*, vol. 89, no. 2, pp. 232–238, 2003.

[48] H. Haider, B. Hoehn, S. Davis, and R. Greiner, "Effective ways to build and evaluate individual survival distributions," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 3289–3351, 2020.

[49] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, no. 2, pp. 187–202, 1972.

[50] F. A. McAlister, E. Youngson, J. A. Bakal, P. Kaul, J. Ezekowitz, and C. van Walraven, “Impact of physician continuity on death or urgent readmission after discharge among patients with heart failure,” *Cmaj*, vol. 185, no. 14, E681–E689, 2013.

[51] A. S. Mixon *et al.*, “Preparedness for hospital discharge and prediction of readmission,” *Journal of hospital medicine*, vol. 11, no. 9, pp. 603–609, 2016.

[52] J. M. Glasgow, M. Vaughn-Sarrazin, and P. J. Kaboli, “Leaving against medical advice (ama): Risk of 30-day mortality and hospital readmission,” *Journal of general internal medicine*, vol. 25, no. 9, pp. 926–929, 2010.

[53] I. K. Omurlu, M. Ture, and F. Tokatli, “The comparisons of random survival forests and cox regression analysis with simulation and an application related to breast cancer,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 8582–8588, 2009.

[54] J. P. Costantino *et al.*, “Validation studies for models projecting the risk of invasive and total breast cancer incidence,” *Journal of the National Cancer Institute*, vol. 91, no. 18, pp. 1541–1548, 1999.

[55] J. Wang, J. Sareen, S. Patten, J. Bolton, N. Schmitz, and A. Birney, “A prediction algorithm for first onset of major depression in the general population: Development and validation,” *J Epidemiol Community Health*, vol. 68, no. 5, pp. 418–424, 2014.

[56] E. L. Kaplan and P. Meier, “Nonparametric estimation from incomplete observations,” *Journal of the American statistical association*, vol. 53, no. 282, pp. 457–481, 1958.

[57] J. D. Kalbfleisch and R. L. Prentice, *The statistical analysis of failure time data*. John Wiley & Sons, 2011.

[58] N. Breslow, “Covariance analysis of censored survival data,” *Biometrics*, pp. 89–99, 1974.

[59] 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, 2011.

[60] S. Fotso, “Deep neural networks for survival analysis based on a multi-task framework,” *arXiv preprint arXiv:1801.05512*, 2018.

[61] Y. Wen *et al.*, “Time-to-event modeling for hospital length of stay prediction for covid-19 patients,” *Machine Learning with Applications*, vol. 9, p. 100 365, 2022.

[62] W. Sun *et al.*, “Improving ecg-based covid-19 diagnosis and mortality predictions using pre-pandemic medical records at population-scale,” *arXiv preprint arXiv:2211.10431*, 2022.

[63] Y. Feng, A. A. Leung, X. Lu, Z. Liang, H. Quan, and R. L. Walker, "Personalized prediction of incident hospitalization for cardiovascular disease in patients with hypertension using machine learning," *BMC Medical Research Methodology*, vol. 22, no. 1, pp. 1–11, 2022.

[64] S.-a. Qi *et al.*, "Personalized breast cancer onset prediction from lifestyle and health history information," *Plos one*, vol. 17, no. 12, e0279174, 2022.

[65] R. Sharma, H. Anand, Y. Badr, and R. G. Qiu, "Time-to-event prediction using survival analysis methods for alzheimer's disease progression," *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, vol. 7, no. 1, e12229, 2021.

[66] S. Kalmady *et al.*, "Improving the calibration of long term predictions of heart failure rehospitalizations using medical concept embedding," in *Survival Prediction-Algorithms, Challenges and Applications*, PMLR, 2021, pp. 70–82.

[67] C. I. for Health Information, "All-cause readmission to acute care and return to the emergency department," *Health System Performance.*, 2012.

[68] J LaPointe, *Strategies to reduce hospital readmission rates, costs*, 2019.

[69] N. Catalyst, "Hospital readmissions reduction program (hrrp)," *NEJM Catalyst*, 2018.

[70] C. Van Walraven, C. Bennett, A. Jennings, P. C. Austin, and A. J. Forster, "Proportion of hospital readmissions deemed avoidable: A systematic review," *Cmaj*, vol. 183, no. 7, E391–E402, 2011.

[71] M. P. A. Commission *et al.*, *Report to the Congress: promoting greater efficiency in Medicare*. Medicare Payment Advisory Commission (MedPAC), 2007.

[72] F. Jiang *et al.*, "Artificial intelligence in healthcare: Past, present and future," *Stroke and vascular neurology*, vol. 2, no. 4, 2017.

[73] A Goldfarb and F Teodoridis, "Why is ai adoption in health care lagging," *Brookings, Available Online: https://www. brookings. edu/research/why-is-ai-adoption-in-health-care-lagging/[Accessed 3 April 2022]*, 2022.

[74] M. A. Agarwal, G. C. Fonarow, and B. Ziaeian, "National trends in heart failure hospitalizations and readmissions from 2010 to 2017," *JAMA cardiology*, vol. 6, no. 8, pp. 952–956, 2021.

[75] P. R. Cronin, J. L. Greenwald, G. C. Crevensten, H. C. Chueh, and A. H. Zai, "Development and implementation of a real-time 30-day readmission predictive model," in *AMIA Annual Symposium Proceedings*, American Medical Informatics Association, vol. 2014, 2014, p. 424.

[76] E. Logue, W. Smucker, and C. Regan, "Admission data predict high hospital readmission risk," *The Journal of the American Board of Family Medicine*, vol. 29, no. 1, pp. 50–59, 2016.

[77] J. Benuzillo, W. Caine, R. S. Evans, C. Roberts, D. Lappe, and J. Doty, "Predicting readmission risk shortly after admission for cabg surgery," *Journal of cardiac surgery*, vol. 33, no. 4, pp. 163–170, 2018.

[78] L. Arbelaez Ossa, G. Starke, G. Lorenzini, J. E. Vogt, D. M. Shaw, and B. S. Elger, "Re-focusing explainability in medicine," *Digital Health*, vol. 8, 2022.

[79] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Transfer learning for time series classification," in *2018 IEEE international conference on big data (Big Data)*, IEEE, 2018, pp. 1367–1376.

[80] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Medical concept representation learning from electronic health records and its application on heart failure prediction," *arXiv preprint arXiv:1602.03686*, 2016.

[81] S. Lai, K. Liu, S. He, and J. Zhao, "How to generate a good word embedding," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 5–14, 2016.

[82] J. L. Saluk *et al.*, "The lace score as a tool to identify radical cystectomy patients at increased risk of 90-day readmission and mortality," *Current Urology*, vol. 12, no. 1, pp. 20–26, 2018.

[83] S. Damery and G. Combes, "Evaluating the predictive strength of the lace index in identifying patients at high risk of hospital readmission following an inpatient episode: A retrospective cohort study," *BMJ open*, vol. 7, no. 7, e016921, 2017.

[84] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[85] 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.

[86] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[87] G. W. Brier *et al.*, "Verification of forecasts expressed in terms of probability," *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[88] A. H. Murphy, "A new vector partition of the probability score," *Journal of Applied Meteorology and Climatology*, vol. 12, no. 4, pp. 595–600, 1973.

[89] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher, "Assessment and comparison of prognostic classification schemes for survival data," *Statistics in medicine*, vol. 18, no. 17-18, pp. 2529–2545, 1999.

[90] S.-a. Qi *et al.*, "An effective meaningful way to evaluate survival models," *arXiv preprint arXiv:2306.01196*, 2023.

[91] P. K. Andersen, J. P. Klein, and S. Rosthøj, "Generalised linear models for correlated pseudo-observations, with applications to multi-state models," *Biometrika*, vol. 90, no. 1, pp. 15–27, 2003.

[92] P. K. Andersen and M. Pohar Perme, "Pseudo-observations in survival analysis," *Statistical methods in medical research*, vol. 19, no. 1, pp. 71–99, 2010.

[93] R. B. D'Agostino and B.-H. Nam, "Evaluation of the performance of survival analysis models: Discrimination and calibration measures," *Handbook of statistics*, vol. 23, pp. 1–25, 2003.

[94] N. I. Goldfield *et al.*, "Identifying potentially preventable readmissions," *Health care financing review*, vol. 30, no. 1, p. 75, 2008.

[1] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.

Appendix A: 30-Day Readmission: Threshold-Dependent Metrics

For most binary classification algorithms, the predicted probability of event must be mapped to a class prediction. This is done by choosing a probability threshold τ, where probabilities above are mapped to the positive class, and those below are mapped to the negative class. AUROC, AUPRC, and Brier score at 30 days are all threshold independent metrics, meaning the score considers all values of τ. For other metrics, the τ value must be chosen. Youden's J statistic [1], calculated as $J_\tau = \mathbf{TPR}_\tau - \mathbf{FPR}_\tau$, provides an informed way to choosing τ. We can find τ_* fromt the training set, which indicates the cutoff that maximizes the distance between a ROC curve and the no-skill line:

$$\tau_* = \text{argmax}_\tau(J_\tau) \tag{A.1}$$

We then use this cut-off to calculate threshold-dependent metrics. Accuracy is the percentage of classifications that were made correctly:

$$\textbf{Accuracy}_{\tau_*} = \frac{TP_{\tau_*} + TN_{\tau_*}}{TP_{\tau_*} + FP_{\tau_*} + TN_{\tau_*} + FN_{\tau_*}} \tag{A.2}$$

Specificity, or the number of True Negatives correctly classified, is defined as:

$$\textbf{Specificity}_{\tau_*} = \frac{TN_{\tau_*}}{TN_\tau + FP_{\tau_*}} \tag{A.3}$$

Recall the definition of Precision and Recall/Sensitivity from Section 4.3.4. The F1-score summarizes precision and recall scores at a particular threshold:

$$\textbf{F1-score}_{\tau_*} = \frac{2 \times P_{\tau_*} \times R_{\tau_*}}{P_{\tau_*} + R_{\tau_*}} \tag{A.4}$$

Table A.1: **XGBoost threshold-dependent 30-day readmission prediction performances.** Boldface indicates the best performance over the metric across this table (XGBoost) and Table A.2 (Deep Neural Network). AUROC, AUPRC, and Brier scores (threshold-independent) are included for comparison.

	XGBoost: 30-Day Readmission Score (Std.Dev)			
	Bare	Detailed	AggSeq	AggSeq+Detailed
TN	60546.0 (1274.7)	65423.8 (362.4)	66035.8 (420.8)	**67427.6 (770.4)**
FP	19397.2 (1234.5)	14519.4 (365.8)	13907.4 (395.6)	**12515.6 (811.0)**
FN	1620.6 (87.1)	1648.4 (71.5)	1825.6 (14.6)	1681.4 (83.6)
TP	2653.0 (107.1)	2625.2 (18.0)	2448.0 (71.7)	2592.2 (64.1)
recall-sensitivity	0.6207 (0.021)	0.6144 (0.011)	0.5727 (0.008)	0.6067 (0.016)
specificity	0.7574 (0.016)	0.8184 (0.005)	0.8260 (0.005)	**0.8434 (0.010)**
precision	0.1205 (0.003)	0.1532 (0.002)	0.1497 (0.004)	**0.1720 (0.008)**
F1	0.2017 (0.003)	0.2452 (0.003)	0.2373 (0.005)	**0.2678 (0.009)**
accuracy	0.7504 (0.014)	0.8080 (0.004)	0.8132 (0.005)	**0.8314 (0.009)**
AUROC	0.7546 (0.005)	0.8002 (0.003)	0.7884 (0.002)	0.8138 (0.003)
AUPRC	0.1682 (0.004)	0.2274 (0.006)	0.2361 (0.004)	0.2877 (0.005)
brier	0.0453 (0.001)	0.0435 (0.001)	0.0436 (0.001)	0.0419 (0.001)

We report the results for True Negatives, False Positives, False, Negatives, True Positives, Recall/Sensitivity, Specificity, Precision, F1-score, and Accuracy, which are summarized for XGBoost (Table A.1) and Deep Neural Network architectures (Table A.2) using various feature sets. Using XGBoost with **AggSeq+Detailed** features lends the best performance in terms of True Negatives, False Positives. Using a DNN with only **Seq** features showed the best numbers for False Negatives, True Positives, and Recall/Sensitivity. DNNs trained on **Seq+Detailed** features gave the best threshold-independent (AUROC, AURPC, and Brier score) performances. Maximizing model usefulness in practice would require a domain-expert-informed cost matrix for True Positives, True Negatives, False Positives, and False Negatives, rather than relying on Youden's J-statistic.

Table A.2: **DNN threshold-dependent 30-day readmission prediction performances.** Boldface indicates the best performance over the metric across this table (DNN) and Table A.1 (XGBoost). AUROC, AUPRC, and Brier scores (threshold-independent) are included for comparison.

	DNN: 30-Day Readmission Score (Std.Dev)					
	Bare	Detailed	AggSeq	AggSeq+Detailed	Seq	Seq+ Detailed
TN	54836.6 (2167.8)	57303.4 (1204.1)	57051.6 (1482.2)	59626.0 (1149.6)	57200.6 (1509.9)	60047.6 (793.5)
FP	25106.6 (2189.7)	22639.8 (1221.6)	22891.6 (1461.3)	20317.2 (1125.0)	22742.6 (1454.6)	19895.6 (764.5)
FN	1516.2 (149.5)	1239.8 (83.8)	1157.8 (91.7)	1192.0 (84.3)	**1081.4 (36.0)**	1171.6 (64.1)
TP	2757.4 (132.0)	3033.8 (85.7)	3115.8 (107.1)	3081.6 (111.6)	**3192.2 (80.5)**	3102.0 (91.7)
recall-sensitivity	0.6454 (0.033)	0.7099 (0.018)	0.7291 (0.021)	0.7210 (0.020)	**0.7469 (0.010)**	0.7258 (0.015)
specificity	0.6860 (0.027)	0.7168 (0.015)	0.7136 (0.018)	0.7459 (0.014)	0.7155 (0.018)	0.7511 (0.010)
precision	0.0993 (0.004)	0.1183 (0.004)	0.1200 (0.004)	0.1318 (0.004)	0.1233 (0.004)	0.1349 (0.002)
F1	0.1719 (0.005)	0.2028 (0.006)	0.2060 (0.005)	0.2229 (0.005)	0.2117 (0.006)	0.2275 (0.003)
accuracy	0.6839 (0.024)	0.7165 (0.014)	0.7144 (0.016)	0.7446 (0.013)	0.7171 (0.017)	0.7498 (0.008)
AUROC	0.7147 (0.003)	0.7821 (0.004)	0.7948 (0.005)	0.8103 (0.005)	0.8114 (0.007)	**0.8212 (0.003)**
AUPRC	0.1330 (0.002)	0.1934 (0.005)	0.2222 (0.007)	0.2509 (0.006)	0.2406 (0.013)	**0.2666 (0.002)**
brier	0.0463 (0.001)	0.0444 (0.001)	0.0437 (0.001)	0.0427 (0.001)	0.0431 (0.001)	**0.0423 (0.001)**

Appendix B: AggSeq Feature Tuning

AggSeq features are created using a summation of MCE vectors corresponding to the k most recent codes in an individual's medical history. Internal cross-validation was used to determine the optimal k for each of the five outer cross-validation folds by examining the average internal-fold XGBoost (30-day readmission prediction) binary-cross-entropy loss. **AggSeq** vectors using $k = 5, 10, 15...100$, paired with **Bare** clinical features, were used as inputs. **Bare** features were included to encourage the model to find a k which captures information that is not already present in easily-available clinical features. The average Binary-Cross-Entropy losses from each set of inner folds are visualized in Figure B.1.

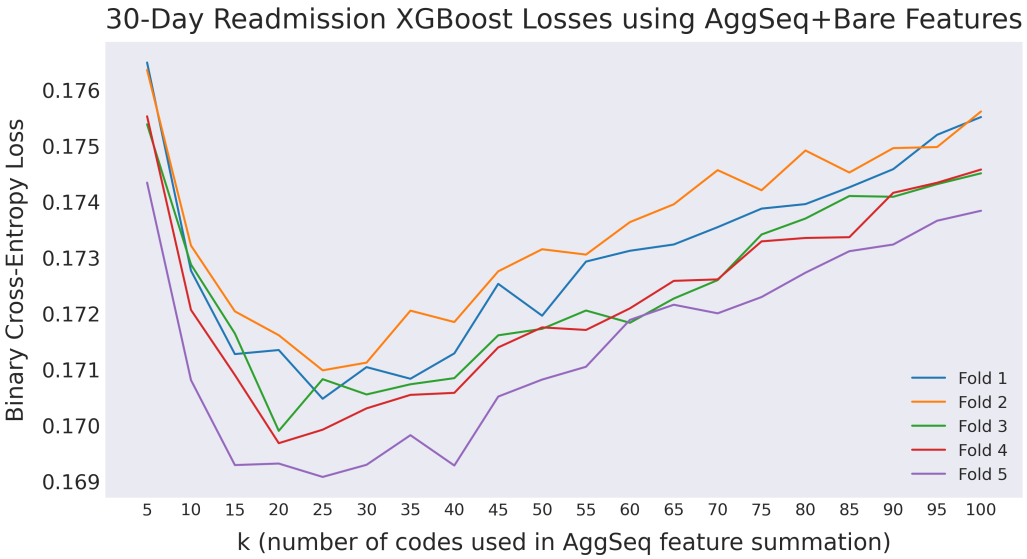


Figure B.1: **Average losses over k, used to determine the optimal k for each of the outer folds**.

Table B.1: **For each metric, evaluations using the data-set-wide k (according to loss), and the best and worst k according to that metric**.

	25CodesScore	BestNCodes	BestNCodesScore	WorstNCodes	WorstNCodesScore
logloss	0.17	25	0.17	5	0.176
TN	41616.04	10	41816.16	90	41144.28
FP	6349.88	10	6149.76	90	6821.64
FN	1212.4	35	1203.04	5	1350.36
TP	1351.76	35	1361.12	5	1213.8
recall-sensitivity	0.527	35	0.531	5	0.473
specificity	0.868	10	0.872	90	0.858
precision	0.176	25	0.176	100	0.163
F1	0.264	25	0.264	5	0.243
accuracy	0.85	10	0.853	90	0.841
AUROC	0.795	20	0.795	100	0.774
AUPRC	0.247	30	0.247	5	0.210
brier	0.043	30	0.043	5	0.045

The best k varies slightly depending on the input data—$k = 20$ for Folds 2 and 3, and $k = 25$ for Folds 1, 4, and 5. The general trend over all folds is the loss dropping sharply as k approaches 15-20, then the loss increasing steadily as k increases after 20-25. It is likely that low values of k do not capture enough information, and high values of k are too noisy to be useful. Dataset-wide, the optimal value of k is 25. The best and worst values of k for all threshold dependent and independent evaluation metrics are reported in Table B.1. A k within $[10, 35]$ generally leads to favourable results, while $k = 5$ or a k in $[90, 100]$ leads to poorer performance. However, when varying k, the absolute differences between the best and worst values is not large in an absolute sense.

Appendix C: Experiments with Un-Sampled Index Episodes

To define our dataset of index episodes, we sampled one hospital episode per-person. This is to ensure high-cost hospital users with many admission events are not biasing the evaluation. However, this is not making use of all the data available. In this section, we model 30-Day readmission risk, 30-day readmission probability, and individual survival distributions using an unrestricted version of the dataset. The choice not to sample index admissions changes the summary statistics, including the percentage of admissions associated with a readmission (previously 5.1%, now 10.3%). Table C.1 shows this and more. The average number of admissions within the index period per-patient is 1.377, with the highest number of unique admissions during this time being 31.

The procedure to generate these results is similar to what was reported in the main text. An admission's membership to one of the five folds was determined by patient ID to ensure no data leakage. The AUROC and Brier scores for each of the five model types (evaluated at 30 days) are reported in Tables C.2 and C.3 respectively, and compared with the corresponding results from the main text. Non-sampled AUPRC scores are not reported/compared with their sampled counterparts, as the no-skill classifier performance changes as the positive class percentages changes.

We see a slight decrease in AUROC performance overall when using the un-sampled dataset. For example, the AUROC of N-MTLR with **Bare+Seq** features significantly decreases from 0.8467± to 0.8310 ± 0.002, $p = 0.0002$. Brier scores using the

Table C.1: **Breakdown of attributes in sampled vs. un-sampled dataset versions**.

Summary Statistic	Sampled	Unsampled
Number Rows	421088	579817
Average Age (Years)	47	49
Average LACE Score	6.53	7.45
Average Index Length-of-Stay (Days)	7	8
Percentage Readmitted @ 30-Days	5.1%	10.3%
Percentage Uncensored	28.2%	41.5%
Average Days-to-Event	604	508

sampled version of the dataset are in the range of 0.04, whereas Brier scores using the non-sampled dataset range from 0.0897 (LACE model) to 0.0750 (N-MTLR with **Bare+Seq** features). This increase could be explained by a greater focus on the positive class by the model, given the higher number of readmissions. The N-MTLR model with **Bare+Seq** inputs has a concordance of 0.7688 ± 0.002, out-performing the sampled dataset model with a concordance of 0.7504 ± 0.001, $p = 0.00002$. The Bare+Seq N-MTLR model has an AUPRC@30 of 0.4208 ± 0.006, and IBS of 0.1562 ± 0.001. Notably, N-MTLR's L1-loss with pseudo observations using the non-sampled dataset is quite low at 709.1 ± 42.5 days, compared to 1104.9 ± 15.4 in the main text. This could partially be because the model has more data to learn from and can model time-to-event more accurately. However, it could also be because a smaller percentage of instances are predicted to have long time-to-event, because of the higher chance of readmission from repeated inclusions of high-risk users.

Table C.2: **AUROC performance for various feature sets when not restricting the dataset to one admission per-person**. "Sampled Index" is what was reported in the main text, "Unsampled Index" is new to this section. Bold indicates best performance across models.

	Features			**30-Day AUROC**				
	Clinical Features	AggSeq Features	Seq Features	Logistic Regression	XGBoost	CoxPH	DNN	N-MTLR
Sampled Index	-	-	-	0.6587±0.003		-	-	-
	Bare	✓	-	-	0.8025±0.005	0.7634±0.005	-	-
	Bare		✓	-	-	-	0.8174±0.006	**0.8467±0.004**
Unsampled Index	-	-	-	0.6600±0.002		-	-	-
	Bare	✓	-	-	0.7899±0.002	0.7514±0.003	-	-
	Bare	-	✓	-	-	-	0.8144±0.004	**0.8310±0.002**

Table C.3: **Brier score performance for various feature sets when not restricting the dataset to one admission per-person**. "Sampled Index" is what was reported in the main text, "Unsampled Index" is new to this section. Bold indicates best performance across models.

	Features			**30-Day Brier**				
	Clinical Features	AggSeq Features	Seq Features	Logistic Regression	XGBoost	CoxPH	DNN	N-MTLR
Sampled Index	-	-	-	0.0471±0.001	-	-	-	-
	Bare	✓	-	-	0.0428±0.001	0.0456±0.001	-	-
	Bare		✓	-	-	-	0.0426±0.001	**0.0402±0.001**
Unsampled Index	-	-	-	0.0894±0.001	-	-	-	-
	Bare	✓	-	-	0.0801±0.001	0.0852±0.001	-	-
	Bare	-	✓	-	-	-	0.0775±0.000	**0.0750±0.001**