Machine Learning for Healthcare: Model Development and Implementation in Longitudinal Settings

by

Erkin Ötleș

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2022

Doctoral Committee:

Professor Brian T. Denton, Co-Chair
Associate Professor Jenna Wiens, Co-Chair
Professor Amy Cohn
Professor Seth Guikema
Assistant Professor Karandeep Singh

Erkin Ötleș
eotles@umich.edu
ORCID iD: 0000-0003-3169-6832

DEDICATION

Ailem için.

ACKNOWLEDGMENTS

I owe a great deal to all of the people that have supported me on this journey.

I would first like to thank my doctoral advisors, Dr. Brian T. Denton and Dr. Jenna Wiens. I am immensely grateful for the opportunity to learn from both of you. Working with both of you has allowed me to catch a glimpse of what it means to be an outstanding leader and scientist. I enter the next phase of my training inspired to reach higher and work harder because of our time together. Thank you very much for everything.

Just as I have had exceptional advisors, I have also had fantastic committee members. Dr. Amy E. M. Cohn, Dr. Seth Guikema, and Dr. Karandeep Singh thank you very much for all the time and effort you have put into my training. You have all helped me grow as a researcher, and I greatly appreciate all your mentorship.

The University of Michigan is a phenomenal place to pursue physician-scientist training. This is partly due to the excellence of the training I have received in medicine and engineering. However, the people here have made the most significant impact. "Non-traditional" MD-PhDs can be challenging from an administrative perspective, so I would like to especially thank the Department of Industrial and Operations Engineering administration and the Medical Scientist Training Program office. The work of Dr. Brian T. Denton, Dr. Marina A. Epelman, Dr. Seth Guikema, and Dr. Ronald J. Koenig was instrumental in laying out the path for my training here. To follow this path, I had to rely on the skills and support of Ms. Grethcen Aland, Ms. Liz Bowman, Dr. Kathleen L. Collins, Ms. Akosua Dow, and Ms. Justine Hein. Thank you all so much for your guidance and support.

I have been blessed to train under phenomenal mentors and alongside spectacular colleagues. Both the college of engineering and medical school are filled with people that I owe gratitude to, below are only a small subset of these people. From the medical school, I would thank my doctoring mentors, Dr. Ross Kessler and Dr. Denise Zhao, as well as my doctoring group,[1] the CSTAR team, namely Dr. Brian C. George and Dr. Andrew E. Krumm, and the MICHAMP group, specifically Dr. Brahmajee K. Nallamothu. From the college of engineering, I would like to thank

[1]Drs. Advani, Cooley, Fernandez, Klueh, O'Donohue, Otte, Rainey, Sessine, Shulkin, Vijayakumar

my IOE cohort[2] and MLD3 lab mates[3] . Additionally, I want to thank my MSTP cohort[4] , what a good group of people to "watch the ship go down with."

Aside from my academic community, I have been fortunate to form close friendships with many wonderful people. Dr. Jacob Abou-Hanna, Mrs. Livia Abou-Hanna, Mr. James Basnett, Dr. Jack Buchanan, Dr. Molly Cory, Ms. Quyn Do, Dr. Sam Kosinski, Dr. Sean Miller, and Dr. Lorena Tagle, we are all overdue for a raclette dinner. Ms. Bruni Bazeti, Dr. Mark Dulchavsky, Dr. Jonathan McBride, and Dr. Laura O'Donahue, we are also overdue on dinner, but let's try to avoid jinxing another major political event. To the DarkAero crew, Messrs. Keegan, River, and Ryley Karl, your work inspires me to take my engineering to the next level.[5] Friendsgiving crew, Dr. Dan Belongia, Dr. Matt Lammers, Mrs. Sarah Karl, Dr. Jenn MacLure, Mr. Ricky Myers, Dr. Alyssa Thorpe, Dr. Glenn Thorpe, Dr. Morgan Weber, here's to many more adventures together. To Ms. Diana Chu and Mr. Ben Grzenia, I pose the following questions: "Where to next? Istanbul or Hong Kong?" And Mr. John B. Cheadle, I'm eternally grateful for the "Maggie's Farm" CD you[6] slid under my dorm room door sophomore year.

Somewhere between friends and family is my Turkish-American community. To Miss. Melis Başkaya, Dr. Mustafa Başkaya, Mr. Pars Başkaya, Dr. Pelin Cengiz, Dr. A. Mert Kartal, Miss. Mira Kartal, Mrs. Ceyda Onaran, and Mr. Tolga Han Unal, spending time with you bridges worlds. I have had the pleasure to learn so much from each of you.

Finally, this would not have been possible without the love and support of my family. I could not ask for better in-laws than Mrs. Sharon Kunkel and Dr. Steven Kunkel. Thank you both for your unwavering support. To my extended family: thank you for energizing and believing in me. Once more, now in Turkish. *Geniş aileme: bana inandığınız ve bana enerji verdiğiniz için teşekkür ederim.*

To my *Anne*, Sevtap Karaköy Ötleş, *Baba*, Dr. Zekai Ötleş, and my brother Arel Savaşkan Ötleş, a lifetime of "thank you"s will never be sufficient. You built a good life for us in a new world, far from family and friends. I will never be able to repay your sacrifices and hard work. But I hope that I can make you proud.

Most importantly, Steph and Evin, thank you! Steph, you have been my biggest advocate and supporter through this journey. I could not have done this without you. Evin, you probably do not realize it yet, but you have sacrificed too. Being your father is the most fulfilling thing I have ever done. *Thank you!*

[2]Drs. to be: Chung, DeRoos, Ghuge, Li, Otero León, Panesar, Seyedsalehi, Shen, Swanson, Tabattnon, Yue & COL Coxen, PhD

[3]Drs. to be: Chang, Jabbour, Kamran, Krishnamoorthy, Lee, Rubin-Falcone, Tang, Tjandra & Drs. Fox, Oh, Wang

[4]Drs. to be: Cuesta, Dulchavsky, Graniel, Henn, Huang, Kim, Park, Semack, Valesano

[5]Plus our group chat with Messrs. John B. Cheadle and Wyatt Karl keeps me sane.

[6]and Messrs. Seth Eatman, Michael Lambeth, and Patrick Grunewald

PREFACE

Medicine is the science of uncertainty and the art of probability.

WILLIAM OSLER

All models are wrong, but some are useful.

GEORGE BOX

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF PROGRAMS

PROGRAM

LIST OF APPENDICES

LIST OF ACRONYMS

ADT admission, discharge, and transfer

AUROC area under the receiver operating characteristic curve

c-index concordance index

CDI *Clostridioides difficile* infection

C. difficile *Clostridioides difficile*

CNN convolutional neural network

DBMS database management system

DGP data generation process

EBM evidence based medicine

ECE expected calibration error

EHR electronic health record

ETL extract, transform, and load

FDA Food and Drug Administration

FHIR Fast Healthcare Interoperability Resources

HIT health information technology

HL7 Health Level Seven

HMM hidden Markov model

IQR interquartile range

IT information technology

LIS laboratory information system

LR lab results

MIMIC-III Multi-parameter Intelligent Monitoring for Intensive Care - III

ML machine learning

MSK Memorial Sloan Kettering

MUMPS Massachusetts General Hospital Utility Multi-Programming System

MUSIC Michigan Urological Surgery Improvement Collaborative

NOCD non-organ confined disease

NRT near real-time

OI occupational injury

PCa prostate cancer

PCR polymerase chain reaction

POP proportion of patient-pairs

RDW research data warehouse

RNN recurrent neural network

ROC receiver operating characteristic

RTW return to work

SGD stochastic gradient descent

SQL Structured Query Language

T2 Temporal Transformer

ABSTRACT

Despite great promise, developing and implementing machine learning (ML) models for healthcare remains a challenging engineering task. The progression of disease generates complex longitudinal data that can be difficult to harness when developing models. Additionally, the practice of medicine is inherently dynamic, meaning that implemented models must be responsive to changes.

This dissertation aims to address some of these challenges. In the first part, we focus on the issues surrounding the development of models for use in the setting of occupational injuries. This field has typically focused on developing models that predict injured patients' return to work dates using information collected around the time of their injury. We demonstrate that a reformulated model using longitudinal observations has better predictive performance than a baseline representative of the existing approaches.

Parts two and three focus on the implementation of ML models. In the second part, we investigate the phenomena of *prospective performance degradation*. Although ML models experience degradation over time, the amount of degradation expected and the mechanisms through which degradation occurs are unclear. We introduce methods to formally quantify this degradation. Additionally, we present techniques to isolate the leading causes of this degradation, splitting *temporal shift* (changes in patients and practice) from information technology (IT) *infrastructure shift* (differences in the data pipelines serving retrospective model development and prospective implementation). These techniques and ancillary analyses allow model developers to debug models to improve prospective model performance.

In the third part, we focus on the problem of updating risk stratification models that have been integrated into clinical practice. Model developers may seek to maintain or improve ML model performance over time. Thus, model developers might *update* models as part of their regular maintenance. We focus on how updated models may change the risk stratification of patients, leading to poor clinician-model team performance. We propose a new rank-based compatibility measure for assessing risk stratification model updates. In addition to describing the behavior of this measure, we also introduce a technique for model developers to generate updated models that balance high rank-based compatibility against discriminative performance. Altogether, this work provides model developers with methods to analyze and develop updates for risk stratification models that support clinical decision making.

CHAPTER 1

Introduction

Machine learning (ML) holds great promise for advancing the practice of medicine. Modern evidence based medicine (EBM) practice depends on synthesizing copious amounts of information across large populations of patients. [1, 2] ML as a field provides a set of techniques to build data-driven prediction models to fulfill the goals of EBM. Difficult medical information synthesis tasks, such as detecting patients at risk for uncommon conditions may be aided by using ML models. [3] There are over 100 Food and Drug Administration (FDA) certified ML systems supporting physicians in a variety of tasks, ranging from electrocardiogram analysis to mammogram breast cancer detection. [4] Additionally, health systems and health information technology (HIT) vendors are developing and implementing ML systems that do not require FDA certification. [5, 6] These systems are meant to inform physicians by providing risk estimates that can be incorporated into medical decision making.

The development and implementation of ML systems for use in healthcare has not been without issue. [5, 7] *Development* is the set of processes involved creating an ML model. One of the first issues is access to datasets. [8] Having obtained data, model developers may realize that healthcare data, like healthcare itself, is complicated. [9] Processing and transforming data for use in ML model development requires a special mix of clinical and technical expertise. [10] Additionally, time plays a critical role in the practice of medicine, as the temporal ordering of information is often a key indicator in diagnostic and therapeutic decision making. [11–13] As such, models may need to respect the temporal nature of the data and healthcare processes. After being developed, models must be carefully internally and, in some settings, externally validated to assess if they will provide benefits to patients, physicians, or healthcare systems. [5, 10] However, external validation may be challenging due to restrictions in data sharing. [14–16]

Implementation is the set process of necessary to integrating and utilize an ML model in clinical care. The implementation process may begin once a model is validated. Implementation raises a host of issues not typically confronted during model development. The technical work needed to implement models often requires cobbling together disparate HIT systems, such as databases, web services, and electronic health record (EHR) interfaces. [17] Additionally, implementation

requires special attention to human factors and systems design. [18–21] These models are not used in a vacuum; developers must carefully consider users and their workflows. Finally, there is the issue of monitoring and maintaining these ML systems. As healthcare systems change, ML systems may experience performance degradation due to changes in patient populations or medical practice. [7, 22–25] Thus developers may need to update their models over time. Despite their promise, successfully developing, implementing, and periodically updating ML models for healthcare is a challenging engineering task.

The longitudinal nature of healthcare makes it an especially daunting application area. Many of the decision tasks being conducted have a strong dependence on time; for example, the temporal ordering of symptoms differentiates diseases. [11] Additionally, treatment delivery times can impact a variety of patient outcomes for diseases, ranging from traumatic injuries to cancer. [12, 13] Moreover, the practice of medicine itself is inherently dynamic: physician behavior, patient risk factors, and disease characteristics all change over time. [26–28]

This dissertation aims to address some of these challenges. In **Chapter 2**, we assess the value of incorporating longitudinal observations into models developed to predict return to work (RTW) for patients experiencing occupational injuries (OIs). In **Chapters 3** and **4**, we switch our focus to implementation issues that arise during the longitudinal use of models. Over time, environmental and IT infrastructure issues may cause models to experience performance degradation. To counter this, model developers may seek to update models in use. In **Chapter 3** we develop techniques to investigate the causes of model performance degradation observed over time after implementation. And in **Chapter 4**, we develop methods that enable updating models that have been integrated into clinical workflows. **Figure 1.1** shows an overview of the ML model development and implementation life cycle as well as the foci of each chapter. In the following section, we provide more details about these chapters and discuss the challenges they tackle. This introductory chapter is concluded with an enumeration of our technical contributions.

1.1 Challenges & Opportunities

ML techniques hold great promise to improve the practice of medicine. However, successfully developing and implementing healthcare ML systems in longitudinal settings is a demanding engineering and clinical task. These systems must benefit patients, physicians, and healthcare systems while respecting privacy, cost, HIT infrastructure, and workflow constraints. This dissertation examines problems across the spectrum of healthcare ML model development and implementation, with the following application areas: occupational health, infectious disease, and hospital early warning systems. We now catalog some of the challenges and opportunities associated with developing and implementing healthcare ML models.

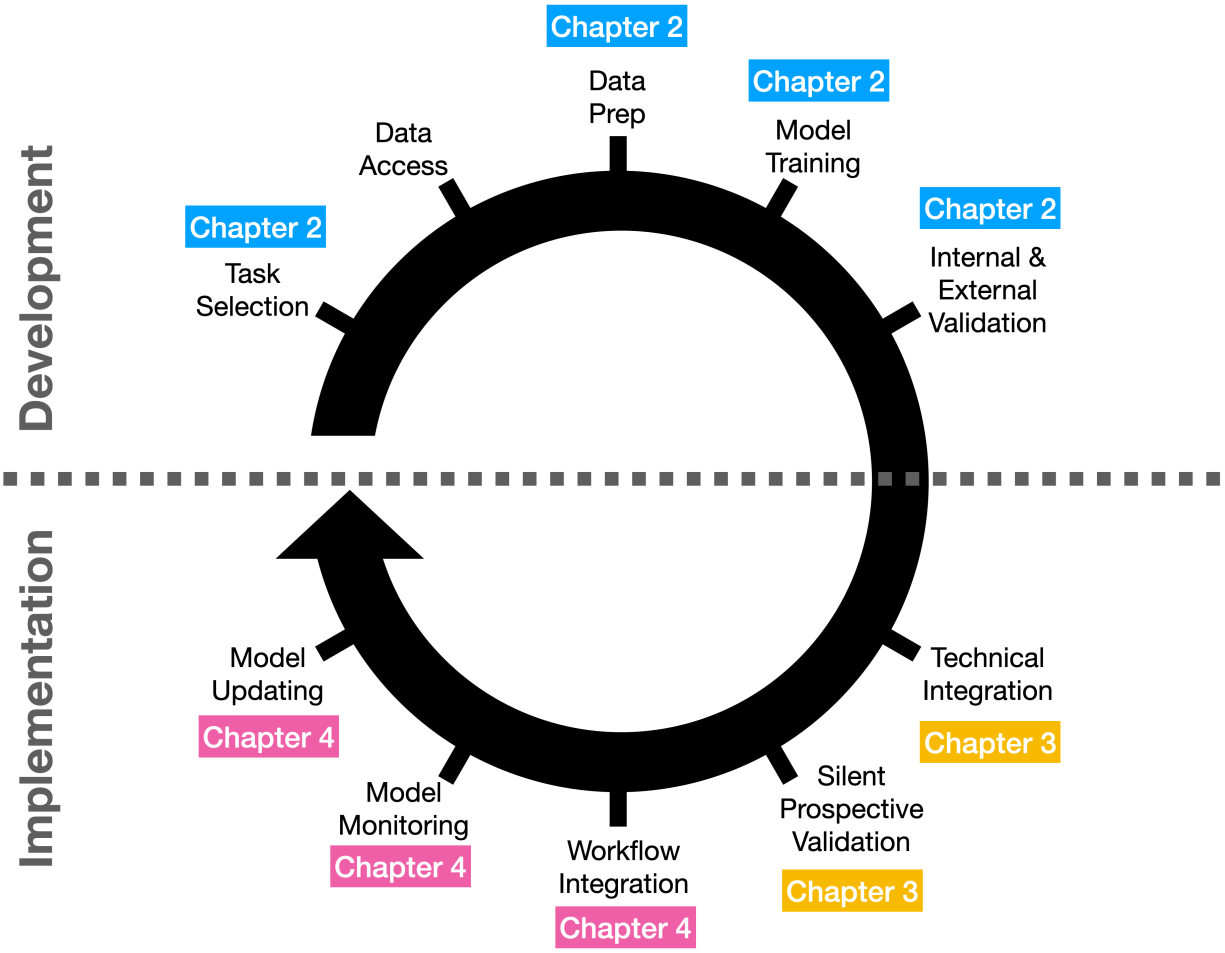


Figure 1.1: Development & Implementation Life Cycle. We contextualize this thesis by placing the foci and contributions of each technical chapter on the main components of the ML model development and implementation life cycle.

Model Development for Occupational Injuries

First, we focus on the development of ML models in the context of longitudinal data. In **Chapter 2**, we develop prediction models to monitor patients recovering from occupational injurys (OIs). OIs cause an immense burden on the U.S. population. Prediction models could help focus resources on patients at greatest risk of a delayed RTW. RTW depends on factors that develop over time, such as the diagnoses and treatments that are rendered over the patient's recovery process. However, existing methods only use information collected at the time of injury. This presents an opportunity to investigate a new modeling approach for RTW prediction. Thus, we explore the performance benefits of dynamically estimating RTW, using longitudinal observations of diagnoses and treatments collected beyond the time of initial injury.

We characterize the difference in predictive performance between an approach that uses infor-

mation collected around the time of initial injury (baseline model) and a proposed approach that uses longitudinal information collected throughout the patient's recovery period (proposed model). To control the comparison, both models use the same deep learning architecture and differ only in the information used. We utilize a large longitudinal observation dataset of OI claims and compare the performance of the two approaches in terms of daily prediction of future work state (working vs. not working).

Experimental results show that the proposed model outperforms the baseline model on this task in terms of both discriminative and calibration performance. These results highlight the value of using longitudinal observations to produce RTW predictions. This approach may enable physicians and workers' compensation programs to manage large populations of injured workers more effectively.

Implementation

Implementing ML models into clinical care is challenging. During model implementation, the goal is to use models to estimate unknown information that can be used to guide various healthcare processes. This exposes models to the transient behaviors of the healthcare system. Over time we expect the model's performance to change. Even though the model in use is not changing, the healthcare system is, and these changes in the healthcare system may reflect new patterns that the model was not trained to identify.

We contrast this with the fact that the model in use may also change over time. Although this dissertation focuses on static models (that may be updated by model developers), some models are inherently dynamic. These models change their behavior over time. Employing updating and dynamic models means that model performance is expected to change over time. This change would be driven by new model behaviors and changes in the healthcare system.

We now provide some examples:

- A model flags patients based on their risk of developing sepsis. There is an increase in the population of patients admitted with respiratory complaints due to a viral pandemic. This change in patient population leads to a massive increase in the number of patients the model flags, and the overall model performance drops because these patients do not end up experiencing sepsis. [7] This is an example of a static model being impacted by the changes in the healthcare system over time.

- A model identifies physicians who could benefit from additional training. The model uses a limited set of specially collected information. [29] Model developers create a new model version that utilizes EHR data. After implementation, the updated model identifies physi-

cians with better accuracy. This is an example of a static model being updated to improve performance over time.

Integration into clinical care requires the model to be connected to systems that can present it with data in real-time. We refer to these systems as *infrastructure*. Infrastructure are the systems (primarily IT systems) needed to take data recorded as a part of clinical care operations and present it in a format accessible to ML models. This infrastructure determines the availability, format, and content of information. Although data may be collected in the same source HIT system (*e.g.*, an EHR system), the data may be passed through a different series of extract, transform, and load (ETL) processes (sometimes referred to as *pipelines*) depending on the data use target.

Once integrated into clinical care, ML models may need monitoring and updating. For example, developers may want to incorporate knowledge about a new biomarker that changes how a disease is diagnosed and managed. Model developers may thus consider *updating* models as a part of their regular maintenance.

This maintenance is complicated because models do not operate in a vacuum. In many application areas users interact with models and learn to about their behavior over time. [30] In safety-critical applications like healthcare, models and users may function as a team. [31–33] The user and model each individually assess patients. The decision maker, usually the user, considers both assessments (their own and the model's) and then makes a decision based on all available information. The performance of this decision is the user-model *team performance*.

Part of the user's decision-making will involve integrating the model's assessments with their own. After using the model repeatedly over time, the user will develop knowledge on the behavior of the model's predictions. We term this knowledge *user expectations*. User expectations represent the user's anticipated correct behavior of the model's assessment. If the model is suddenly replaced with an updated one, then user expectations may be subverted. This is problematic when:

- The user expected an assessment to be correct, and now the updated model produces an incorrect assessment. In this case, the user might have chosen to act on an incorrect assessment.

- When the user expected an assessment to be incorrect, and now the updated model produces a correct assessment. In this case, the user might ignore a correct assessment.

Over time the user will come to recognize this behavior and have their knowledge match the updated model. However, during the intervening adaptation time, the user-model team may have its overall performance suffer. [31–33]

Compatibility measures quantify how much an updated model continues to produce the correct behavior exhibited by an original model. This can be contrasted with performance measures

that focus on the correct behavior of one model. Compatibility measures assess two models and measure the correct behavior of the updated model in the scenarios where the original model had behaved correctly. One interpretation of a compatibility measure is the probability that the updated model has the correct behavior, given that the original model had the correct behavior. We expect that with high compatibility team performance will suffer less immediately following a specific model update.

Risk stratification models produce continuous values, usually between 0 and 1, to rank patients by their risk of being affected by a specific condition in the future. Risk stratification models may be used to classify patients into different classes (*e.g.*, low- vs. high-risk). This may be done by employing a *decision threshold*, which is a number in the range of values produced by the risk stratification model. Any patient with a risk estimate below this decision threshold will be classified as low-risk, and patients with estimates above will be classified as high-risk.

Performance Degradation Observed After Model Implementation

We first focus on the problem of static models performing worse than expected when applied in real-time. It is widely accepted that performance may degrade over time due to changes in care processes and patient populations. However, the extent to which this occurs is poorly understood. In **Chapter 3**, we seek to prospectively characterize the changes to model performance over time after model implementation.

We compare the 2020-2021 prospective performance of a patient risk stratification model for predicting healthcare-associated infections to a 2019-2020 retrospective validation of the same model. We define the difference in retrospective and prospective performance as the *prospective performance gap*. We estimate how two major sources of dataset shift contribute to the prospective performance gap: i) *temporal shift*, changes in clinical workflows and patient populations, and ii) HIT *infrastructure shift*, changes in access, extraction, and transformation of data.

In terms of discriminative performance, the observed prospective performance gap was primarily due to infrastructure shift and not temporal shift. So long as we continue to develop and validate models using data stored in large research data warehouses, we must consider differences in how and when data are accessed, measure how these differences may negatively affect prospective performance, and work to mitigate those differences.

Updating Implemented Models

In **Chapter 4**, we examine updating and its impact on the expectations of clinical users. More specifically, we focus on how updated models may change the risk stratification of patients, leading to poor clinician-model team performance. *Compatibility measures* quantify, in part, the impact

a model update may have on clinician expectations. In the setting of patient risk stratification, existing compatibility measures depend on a single model decision threshold; however, clinicians often differ in their belief about the most appropriate decision threshold. As a result, existing measures cannot be directly applied to settings where the model is used to produce a ranking of patients based on estimated risk (*e.g.*, without a set decision threshold). We propose a new rank-based compatibility measure in light of these and other limitations.

We characterize the proposed compatibility measure in terms of the discriminative performance of the original and updated models, develop bounds, and show the central tendency of rank-based compatibility. These bounds demonstrate that it is possible to achieve very high rank-based compatibility while also maximizing the discriminative performance of the updated model. However, theory and empirical studies on the Multi-parameter Intelligent Monitoring for Intensive Care - III (MIMIC-III) 48-hour mortality prediction task suggest updated model selection based on improved discriminative performance may not achieve the highest level of compatibility. We introduce a new loss function based on our proposed rank-based compatibility measure that can be used for updated model selection. This loss function encourages the selection of updated models that balance improvements in discriminative performance against higher levels of compatibility. Additionally, we present a real-world case study focused on prostate cancer outcome prediction; in this case study, we show how model developers can use our proposed compatibility measure to understand the implications of model updating on clinician expectations. Altogether, this work provides model developers with techniques to analyze and develop updates for risk stratification models used in healthcare.

1.2 Contributions

This thesis presents new methods to develop and implement ML models for use in healthcare. We summarize the main contributions of each chapter below.

Chapter 2. In this chapter, we present the development of a risk prediction model that uses longitudinal data. It is focused on the application area of RTW prediction for workers experiencing OIs. The main contributions found in this chapter are as follows:

- *RTW as an ML for healthcare application area.* We provide an introduction to the problem of RTW, reviewing the existing literature and state of the art. We note that the most widely used models are proprietary and have not been described in the literature. In addition to reviewing existing techniques, we identify sources of data and potential implementation targets.

- *Reformulation of RTW prediction as dynamic work status prediction.* We introduce the first,

to our knowledge, reformulation of the RTW prediction problem in a longitudinal setting. Existing approaches predict the RTW using data collected at the time of the initial OI. As patients recover, physicians and the health system collect a great deal of additional information. Instead of generating a single static prediction, we repeatedly predict if the patient will be at work (their *work status*) in the future as new data are collected.

- *Recurrent neural network (RNN) implementation and accompanying training procedure.* To produce updated predictions in response to inputs collected over time we use a recursive model that stores information about all of the longitudinal observations collected about a patient (*i.e.*, their *history*). We use RNNs to encode this information over time and to derive predictions regarding the patient's future work status.

- *Longitudinal observation performance evaluation.* We evaluate the predictive performance of our proposed approach to RTW prediction. Additionally, we compare the proposed approach to a baseline representing a strong static prediction model that uses the information collected near the initial time of OI. This enables us to evaluate the performance benefits of utilizing longitudinal observation data compared to an optimistic hypothesis of the function of proprietary models.

- *Additional studies using simpler model architectures.* We investigate if longitudinal information still benefits predictive performance if simpler model architectures are employed. Although limited compared to the proposed RNN model, these results suggest that longitudinal observations still provide a benefit.

The work presented in **Chapter 2** has been accepted for publication by the Journal of the American Medical Informatics Association. It has been presented in part at the 2019 INFORMS Annual Meeting and the 2019 Machine Learning for Healthcare Conference. The methods described in this chapter are covered under a USPTO patent application.

Chapter 3. We present a careful evaluation of a prospectively implemented risk prediction model. Through this evaluation, we isolate the causes of performance degradation when transitioning models from retrospective development and validation to prospective implementation. The main contributions from this chapter are as follows:

- *Formalization of the notion of the prospective performance gap when validating ML-based models in clinical care.* We provide a formal definition of the prospective performance gap. Using this, model developers can calculate the amount of performance degradation observed when transitioning into model implementation from model development.

- *Formulation of the relationship between the prospective performance gap, temporal shift, and infrastructure shift.* We introduce and formally define two constituent components of the prospective performance gap. The first component captures changes in the population and care processes, *temporal shift*, and the second represent differences in the data infrastructure used for development and implementation, *infrastructure shift*.

- *Characterize the differences between a retrospective pipeline and a prospective pipeline and the resulting impact on the performance gap.* We provide the first, to our knowledge, rationale and example for why infrastructure shift may contribute to the prospective performance gap.

- *Quantifying how much of the performance gap can be attributed to temporal shift and infrastructure shift.* We expand the formal analysis of the prospective performance gap to isolate its components.

- *Develop methods and approaches to identify contributors to the performance gap.* These new methods enable model developers to determine which features contribute most to temporal and infrastructure shift. Features associated with significant contributions to infrastructure shift may be targeted by model developers to be fixed in model updates.

- *Highlight approaches for mitigating the effects of differences in retrospective versus prospective data infrastructure on the performance gap.* By focusing on our model implementation, we provide examples of improvements that model developers may consider to ameliorate the effects of infrastructure shift.

The results of the work in **Chapter 3** were presented at the 2021 Machine Learning for Healthcare Conference and published in the Proceedings of Machine Learning Research. [17]

Chapter 4. We propose and analyze a new rank-based compatibility measure to fill in the gaps associated with existing compatibility measures that assume a single decision threshold. This new rank-based compatibility measure is designed for evaluating updates to risk stratification models that do not depend on a single decision threshold. This measure may be used for updated model selection as an additional criterion focused on modeling user expectations. Alternatively, we show how model developers may incorporate it into model development. The main contributions presented in this chapter are as follows:

- *To the best of our knowledge, we introduce the first rank-based compatibility measure based on the concordance of risk estimate pairs.*

- *We characterize the extent to which the new compatibility measure may vary over all potential model updates.* This helps to establish the relationship between the discriminative performance of the original and updated models and the new rank-based compatibility measure. In addition to providing a direct connection between model discrimination performance and rank-based compatibility, we also introduce several ancillary measures to examine the characteristics of risk stratification model updates. The ancillary measures also contextualize and compare the rank-based compatibility values produced for updates considered to serve as secondary criteria for model selection among models of similar discriminative performance.

- *We provide bounds on the rank-based compatibility, which provide insights about optimistic and pessimistic outcomes of potential updates.* Additionally, the bounds show that as the discriminative performance of the models increases, the lower bound of rank-based compatibility increases. We also show that rank-based compatibility exhibits a central tendency. Common model development approaches may provide many more updated models with rank-based compatibility values towards the center of the bounds. Thus, while some rank-based compatibility arises from maximizing the area under the receiver operating characteristic curve (AUROC) of the updated model, additional search procedures may be necessary to find a model with desired rank-based compatibility.

- *We introduce a custom loss function that incorporates ranking incompatibility which can be used to engineer model updates with improved rank-based compatibility characteristics.* We show that utilizing the incompatibility loss during updated model training results in higher rank-based compatibility on held-out data. This higher rank-based compatibility comes at a small cost in terms of discriminative performance.

- *Using MIMIC-III, we present empirical results that show the updated models with larger rank-based compatibility values can be generated using incompatibility loss.* In addition to examining the rank-based compatibility observed through standard model selection, we analyze the impact of incorporating incompatibility loss as an alternative model selection criterion. This experiment shows that candidate update models built using standard training procedures provide a limited range for rank-based compatibility, which can be overcome by using a new loss function that incorporates ranking incompatibility.

- *We present a real-world use using the rank-based compatibility measure to understand the potential impact of updating a risk stratification model currently used for predicting prostate cancer outcomes.*

We presented early versions of this work at the 2021 INFORMS Annual Meeting and the 2021 INFORMS Healthcare Meeting. We plan to submit the methods portion of this work to a refereed

Conference. We plan to submit the case study and some of the more general findings to an archival journal.

Careful development and implementation are critical for ML models used in healthcare, especially in longitudinal settings. While there are many issues to be addressed in this domain, this dissertation makes an effort to advance the state of the art in ML model development and implementation using real-world use cases drawn from occupational health, infectious diseases, hospital early warning systems, and cancer. **Chapters 2**, **3**, and **4** describe the technical details of our contributions. The final chapter, **Chapter 5**, summarizes our findings and discusses future directions building on the work presented in this dissertation.

CHAPTER 2

Dynamic Prediction of Work Status for Workers with Occupational Injuries

2.1 Introduction

Occupational injuries (OIs) cause an immense burden on the U.S. population and economy. Millions of workers are injured annually, leading to pain, emotional suffering, and economic hardship. In addition to resulting in time away from work, OIs increase medical expenditures and shorten lifespans; furthermore, they disproportionately affect minorities. [34–41] OIs have far-reaching economic consequences due to decreases in corporate productivity and are high costs to government organizations. [34–36] As in other facets of medicine, timely and clinically appropriate intervention is critical to the injured worker's healing and recovery. [42–44] In occupational medicine, the primary clinical outcome is return to work (RTW).

The RTW process, like most medical episodes, is complex. [45] It requires individual medical management by highly trained physicians; additionally, injuries are often reviewed for treatment utilization by reviewers, or *recovery managers*, who oversee thousands of simultaneous cases on behalf of workers' compensation programs. [46] The current state of the art for injury recovery prediction are models and guidelines used at the onset of the injury. [47–49] Payers often use these models to estimate a worker's RTW date. Predicted RTW duration is a clinical and administrative tool ingrained into the occupational health framework. [50] The most prevalent modeling techniques for this approach are Cox proportional hazards models, time to event models that estimate the probability that a worker will return to work in a given period. [51–54] These models estimate RTW based on information at the time of a worker's injury. They provide guidance on the expected resources needed for a worker's recovery and enable stratification of the injured worker population. While these models assist initial triage of resources for injured workers, their utility decreases over time as they fail to account for the diagnoses and treatments workers receive throughout their recovery. To our knowledge, longitudinal data like insurance claims streams are not currently used to generate or update RTW predictions.

We investigate the predictive performance benefits of using longitudinal observations collected during a workers' compensation case to support decision-making over worker recovery. The use of longitudinal observations has been shown to improve performance in the prediction of cardiovascular events and in many other healthcare setting. [55, 56] However, to the best of our knowledge, this has not been characterized for the prediction of RTW. In this work, we measure the difference in predictive performance between the current approach to RTW prediction (baseline model), which only uses information collected near the time of injury, to an approach that uses longitudinal observations (proposed model) collected throughout a worker's recovery.

To do this, we present a new model to predict the RTW of injured workers dynamically. The proposed model reframes the prediction of RTW into a dynamic prediction task. For injured workers, it seeks to learn the relationship between observations collected daily and the worker's future *work status*, *i.e.*, whether the worker is working or not. To evaluate whether longitudinal observation data collected beyond the first week of injury can help predict work status, we developed a model capable of analyzing longitudinal observations inputs. We trained this proposed model with the entire history of longitudinal observations available in the training dataset. Given daily longitudinal observations, the model will return future work status predictions. Although the predictions are dynamic, the underlying model parameters are static.

We compare the performance of this proposed model against a baseline model, which is representative of current RTW prediction approaches. To assess the benefit of the proposed approach, we use a large claims dataset from the state of Ohio's workers' compensation program to develop the models. [49] Both models are implemented as RNNs to learn this relationship. [57–60] We evaluate the predictive performance difference between these two approaches using a held-aside portion of the claims dataset.

2.2 Contributions

The main contributions from this work are as follows:

- *RTW as an ML for healthcare application area.* We provide an introduction to the problem of RTW, reviewing the existing literature and state of the art. We note that the most widely used models are proprietary and have not been described in the literature. In addition to reviewing existing techniques, we identify sources of data and potential implementation targets.

- *Reformulation of RTW prediction as dynamic work status prediction.* We introduce the first, to our knowledge, reformulation of the RTW prediction problem in a longitudinal setting. Existing approaches predict the RTW using data collected at the time of the initial OI. As

patients recover, physicians and the health system collect a great deal of additional information. Instead of generating a single static prediction, we repeatedly predict if the patient will be at work (their *work status*) in the future as new data are collected.

- *RNN implementation and accompanying training procedure.* To produce updated predictions in response to inputs collected over time we use a recursive model that stores information about all of the longitudinal observations collected about a patient (*i.e.*, their *history*). We use RNNs to encode this information over time and to derive predictions regarding the patient's future work status.

- *Longitudinal observation performance evaluation.* We evaluate the predictive performance of our proposed approach to RTW prediction. Additionally, we compare the proposed approach to a baseline representing a strong static prediction model that uses the information collected near the initial time of OI. This enables us to evaluate the performance benefits of utilizing longitudinal observation data compared to an optimistic hypothesis of the function of proprietary models.

- *Additional studies using simpler model architectures.* We investigate if longitudinal information still benefits predictive performance if simpler model architectures are employed. Although limited compared to the proposed RNN model, these results suggest that longitudinal observations still provide a benefit.

The work presented in this chapter has been accepted for publication by the Journal of the American Medical Informatics Association.[1] It has been presented in part at the 2019 INFORMS Annual Meeting and the 2019 Machine Learning for Healthcare Conference. The methods described in this chapter are covered under a USPTO patent application.

2.3 Problem Setup & Related Work

The prediction of RTW for an OI is fundamental to decision-making by employers, occupational health physicians, and recovery managers – all of whom share the common goal of minimizing the employee's absence. Disability management is a human resources process conducted by many employers who recognize it as a critical component of workplace productivity. [61–63] On an individual basis, if the predicted RTW duration is short, then minimal personnel shifting needs to occur. On the other hand, with a longer prediction, employers face more operational decisions,

[1]This publication was co-authored by Erkin Ötleș, Jon Seymour, Haozhu Wang, and Brian T. Denton. EÖ led the core study design, data analysis, and manuscript preparation. JS, HW, and BTD assisted in study design refinement, data interpretation, and manuscript revisions. EÖ was primarily responsible for all of the core contributions presented in this chapter.

including whether to hire temporary workers or to offer the injured employee modified duty during the recovery.[64] On an aggregate basis, actual RTW performance against predicted RTW forms an essential metric for many businesses. [65] Questions that an employer may seek to use an RTW model are: Will the employer need to replace the worker on a temporary or permanent basis? Is modified duty a worthwhile option for this worker? Does the organization measure up to RTW benchmarks?

RTW predictions are related to the expert prognoses generated by occupational health physicians, who are often asked to supply absence notes for injured patients. [50, 66] Importantly, RTW patients are often seen by general primary care physicians. [67] As such, RTW predictions are used as a part of treatment guidelines for non-specialist physicians to benchmark occupational injuries. [68] A question that a physician may use an RTW model to answer: How long is this patient's absence from work expected to be?

Recovery managers, typically working on behalf of insurance organizations, are often assigned to cases based on the RTW prediction. Workers with longer predicted RTW durations are usually associated with more severe or complex injuries; that case is often directed to a more experienced manager. The RTW prediction is used to manage expectations and to dictate operational processes across clinical and corporate stakeholders. A recovery manager may use an RTW model to answer: How should this case be triaged? Should I alert other stakeholders that the RTW duration has exceeded the prediction?

Due to the close relationship between RTW prediction and treatment, predictive models are bundled with treatment and resource management guidelines. [69, 70] In this chapter, we focus our work on the task of RTW prediction and leave additional guidance for future work. From the existing RTW literature, it is essential to note that the state art in OI modeling has several potential avenues for further exploration. The first is that published models are, to the best of our knowledge, based on a static time-to-event prediction of RTW, designed for usage only at the time of injury, and incapable of handling newly observed information. [51–54] The second is that models are traditionally made for specific diseases with custom collected data. [51, 53, 54, 71–74] The current literature presents a gap to be explored. Specifically, what is the value of producing dynamic RTW predictions using longitudinal observations from claims data?

For an extended discussion of the RTW literature, please see **Appendix Section A.1.1**.

Problem Statement

To address this question, we reformulate OI modeling as a dynamic prediction task, where the prediction of a worker's RTW is made sequentially over the time horizon of their injury. These repeated predictions would be based on observational data commonly available to decision-makers

like physicians and workers' compensation programs. For example, each day, new claims observations may be fed to a model, which returns the likelihood that the injured worker would be at work a week in the future.

Stated more formally, we seek to learn a model, $f(\cdot)$, that when given a sequence of diagnoses and treatments observations, $\mathbf{x}_{i,t}$, over time, t, for a given worker injury, i, produces an estimate of the likelihood of return to work within a defined period, $\Pr(y_{i,t} = 1)$.

Sequence-to-Sequence Learning

The problem we consider is a *sequence-to-sequence* learning task, where a model captures the mapping between a given sequence of observations and a sequence of predictions. Markov chain-based models have successfully been used for sequence-to-sequence learning tasks. [75–78] However, we would like the model to learn to use the longitudinal observations directly (*e.g.*, no grouping or curation of diagnoses or treatments), and we would like the proposed model to learn a representation of the accumulated observations (or history).

RNNs, a type of deep neural network, are naturally well suited for this task. They can handle sequences with long-range time dependencies [57–60] and learn representations for high-cardinality categories (*e.g.*, diagnoses and treatment codes) with minimal modification. [79–83] We cover the rational for the use of RNNs in **Appendix Section A.1.1**. Additionally, we note that other types of deep neural networks, such as convolutional neural networks (CNNs) or transformers, which have become widely used in sequence-to-sequence learning tasks. [58, 84] Although these other models may be applied to sequential data, we focus on an RNN based architecture as it was the most common at the time.

2.4 Methods

We assess the value of utilizing longitudinal observations by reframing RTW prediction as a dynamic task. Our proposed model reframes the RTW prediction problem to produce future work status predictions using observations of diagnoses and treatments collected over time. We compare this to a baseline model that only uses information collected around the time of injury.

2.4.1 Approach

Let the set of worker injuries be denoted by I, with i denoting an individual worker. Time was discretized using a fixed time-step duration $\delta \in \mathbb{R}^+$, with δ set to 1 day for this study and $t = 1, 2, ..., T_{max}$ where T_{max} is the maximum number of days of a worker injury case. We limited

case durations to the typical cut-off for maximal medical improvement ($T_{max} = 365$ days). [85] This discretization and transformation are further described in **Section 2.6**. Moreover, **Figure 2.1** depicts an example of an injury transformation.

Each injury, i, had two types of data collected. The first type was characteristic data, which includes all time-invariant demographic data (*e.g.*, biological sex and job classification). The second type was longitudinal observations, which includes time-stamped information collected over time (*e.g.*, procedure information). For every injury, i, we created a characteristic vector, $\mathbf{c}_i$, of size $d_{\mathbf{c}}$, to represent time-invariant information that was known before the time of injury. We created an observation vector, $\mathbf{o}_{i,t}$, of size $d_{\mathbf{o}}$ for every injury, i, at every time-step t; these vectors represent information collected each day of an injured worker's recovery. Characteristic and observation vectors both contain information encoded as either real numbers and or as integers (for categorical data). Missing observations were denoted with a special missing value (see **Appendix Section A.1.3.1** for more detail). We let $\mathbf{o}^W_{i,t}$ denote the work status of an injured worker over time; $\mathbf{o}^W_{i,t} = 1$ indicates "working" status and $\mathbf{o}^W_{i,t} = 0$ represents "not working" status.

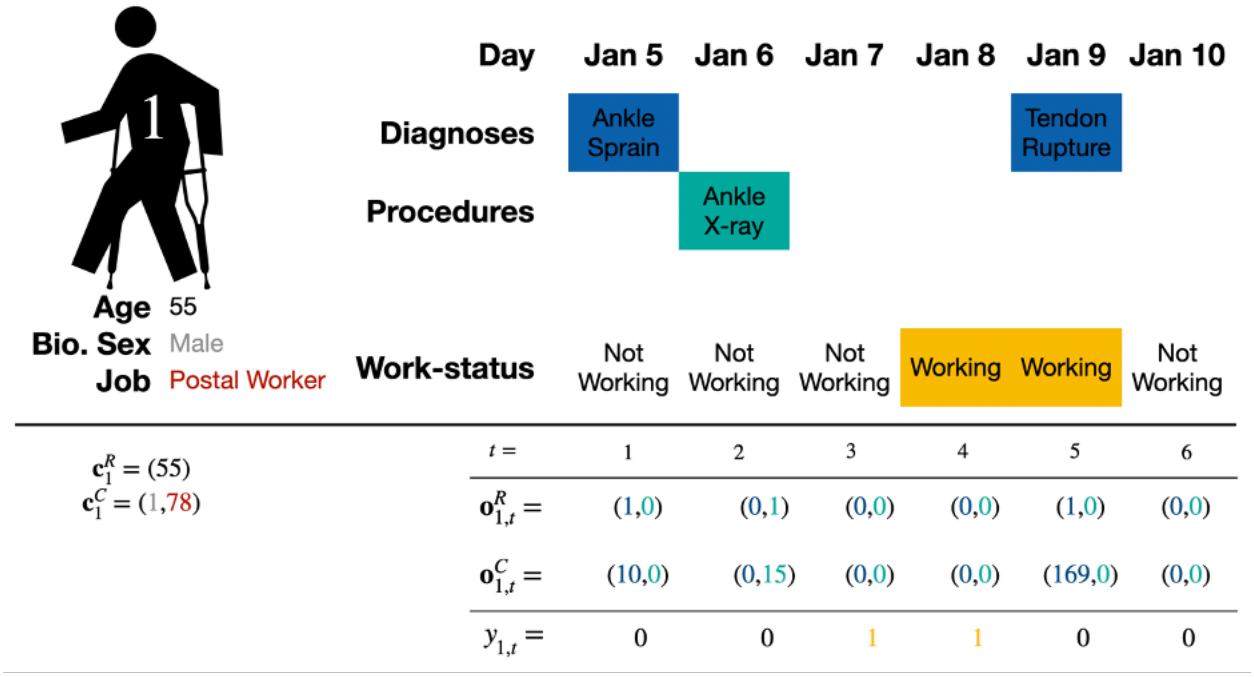
Characteristic and observation vectors were used to generate the model's input features and output labels. Model input features $\mathbf{x}_{i,t}$ denote the vector of injured worker's characteristics and observations over all time-steps $\mathbf{x}_{i,t} = (\mathbf{c}_i, \mathbf{o}_{i,t}) \ \forall \ i \in I, t \in T$. An example calculation of $\mathbf{x}_{i,t}$ is depicted and explained in **Figure 2.1**. Readers may find additional details on data variables in **Appendix Section A.1.2.1**. The model output label, the future work status, denoted as $y_{i,t}$ was also indexed in terms of injuries and time-steps. For this work, each $y_{i,t}$ was related to the observed work status. We define $y_{i,t} = \mathbf{o}^W_{i,t+\phi}$, where ϕ is termed the offset, a positive integer value for the number of time-steps in the future we would like to predict work status.

2.4.2 Proposed Model

Both the proposed and baseline model utilize an RNN architecture that operate upon learned embeddings of the longitudinal observations. This overall architecture is broken down into three subcomponents that each have a role. We now describe these subcomponents in more detail.

At every time-step, the model uses all the observed longitudinal observation information collected on an injured worker to estimate the probability that they will be at work in the future $\Pr(y_{i,t} = 1)$. We denote the overall model as $f(\cdot)$ and the model's parameters as θ, formally $f : (\mathbf{x}_{i,1}, ..., \mathbf{x}_{i,t}) \rightarrow \Pr(y_{i,t} = 1) \in [0, 1]$. The model is composed of three functions, the input encoder $f_{in}(\cdot)$, the history encoder $f_{mid}(\cdot)$, and the output estimator $f_{out}(\cdot)$. Each function is described in more detail below. The parameters of the overall model, $f(\cdot)$, θ is the combination of the parameters of these functions θ_{in}, θ_{mid}, and θ_{out}.

The model does not directly use $\mathbf{x}_{i,t}$ to predict $\Pr(y_{i,t})$; instead, instead, it uses two inter-



$\mathbf{c}_1^R = (55)$
$\mathbf{c}_1^C = (1,78)$

$t =$	1	2	3	4	5	6
$\mathbf{o}_{1,t}^R =$	(1,0)	(0,1)	(0,0)	(0,0)	(1,0)	(0,0)
$\mathbf{o}_{1,t}^C =$	(10,0)	(0,15)	(0,0)	(0,0)	(169,0)	(0,0)
$y_{1,t} =$	0	0	1	1	0	0

Figure 2.1: Example Occupational Injury Timeline and Corresponding Characteristic and Observation data. In this example, worker injury 1, a 55-year-old male postal worker, is injured on January 5th. This information is encoded in the real characteristic vector, $\mathbf{c}_1^R = (55)$, which contains the age information, and the categorical characteristic vector, $\mathbf{c}_1^C = (1, 78)$, which encodes biological sex (male = 1) and job code (postal worker = 78). His injury case runs until the last observed date, January 10th. Diagnoses and procedures are observed throughout. This information is encoded in daily observation vectors. On the first day of the worker's injury, January 5th, the real observation vector, $\mathbf{o}_{1,1}^R = (1, 0)$, contains information regarding the number of diagnoses and procedures observed for the injured worker at $t = 1$ (1 diagnosis and 0 procedures). The categorical observation vector $\mathbf{o}_{1,1}^C = (10, 0)$, encodes diagnosis (ankle sprain = 10) and the no procedures observed token (0). The input vector at $t = 1$, $\mathbf{x}_{1,1}$ is the concatenation of the observation vectors at that time and the characteristic vectors $(55, 1, 78, 1, 0, 10, 0)$. The model will then map the input vector to the output, $y_{1,1}$, which is the work status 1 day in the future ($\phi = 1$ for this example).

mediaries. These intermediaries are lower-dimensional approximations: the encoded observation vector, $\tilde{\mathbf{x}}_{i,t}$, and the encoded history vector, $\tilde{\mathbf{h}}_{i,t}$. The encoded feature vector, $\tilde{\mathbf{x}}_{i,t}$, is a transformation of $\mathbf{x}_{i,t}$ that replaces the categorical integer values with real-valued embeddings. [80, 82, 83] We compute $\tilde{\mathbf{x}}_{i,t}$ using $f_{in}(\mathbf{x}_{i,t})$ which uses the parameters θ_{in}.

Similarly, the encoded history vector, $\tilde{\mathbf{h}}_{i,t}$, approximates the whole history of the injury's observations, $\mathbf{x}_{i,1}, ..., \mathbf{x}_{i,t}$. The encoded history vector, $\tilde{\mathbf{h}}_{i,t}$, is a real-valued vector of size $d_{\tilde{\mathbf{h}}}$ that is updated by the middle function, $f_{mid}(\cdot)$. This recursive function takes the current time-step's encoded input, $\tilde{\mathbf{x}}_{i,t}$, along with the encoded history from the previous time-step, $\tilde{\mathbf{h}}_{i,t-1}$. It returns an updated encoded history for the current time-step, $\tilde{\mathbf{h}}_{i,t}$. It uses the parameters θ_{mid}.

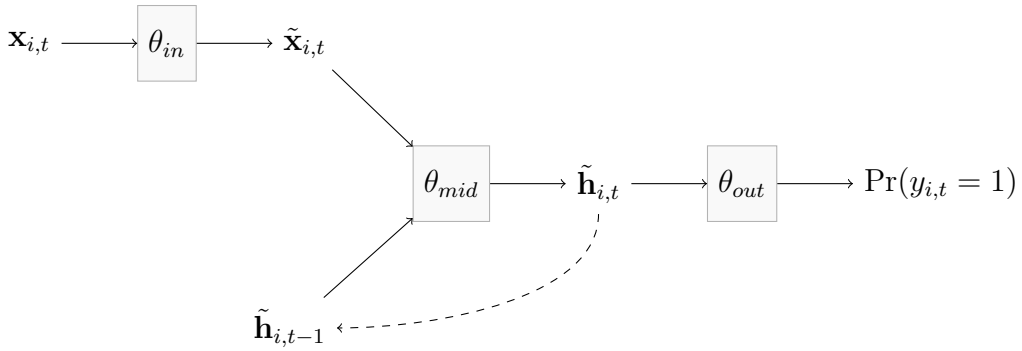


Figure 2.2: Proposed Model Diagram. Longitudinal observations, $\mathbf{x}_{i,t}$, and the prior history encoding, $\tilde{\mathbf{h}}_{i,t-1}$, are used as inputs at every time-step, t. Observations are encoded with θ_{in} to learn representations of high-cardinality categories ($\tilde{\mathbf{x}}_{i,t}$). Using θ_{mid} these representations are then encoded into the current history, $\tilde{\mathbf{h}}_{i,t}$. Finally, θ_{out} utilizes the current history to predict the injured worker's likelihood of being at work in the future, $\Pr(y_{i,t} = 1)$.

The encoded history vector, $\tilde{\mathbf{h}}_{i,t}$, represents the injury's entire history including the current time-step. Thus, it can be used to estimate a worker's health outcome. The out function, $f_{out}(\cdot)$, controls this mapping which. This function takes in $\tilde{\mathbf{h}}_{i,t}$ and returns a probability estimate of the worker being at work in the future. The out function $f_{out}(\cdot)$ is parameterized by θ_{out}. We can then use the probability estimate produced by f_{out} to estimate the outcome label by employing a threshold, τ. For example:

$$\hat{y}_{i,t} = \begin{cases} 1 \text{ if } \Pr(y_{i,t}) \geq \tau \\ 0 \text{ otherwise} \end{cases}$$

In summary, the prediction process from inputs to estimated probability of being at work in the future is:

$$\tilde{\mathbf{x}}_{i,t} = f_{in}(\mathbf{x}_{i,t}) \tag{2.1}$$

$$\tilde{\mathbf{h}}_{i,t} = f_{mid}(\tilde{\mathbf{x}}_{i,t}, \tilde{\mathbf{h}}_{i,t-1}) \tag{2.2}$$

$$\Pr(y_{i,t} = 1) = f_{out}(\tilde{\mathbf{h}}_{i,t}) \tag{2.3}$$

This recurrent approach yields a model that maps to clinical decision-making. Note, although the model updates the history encoding in response to observations collected across time, the underlying parameters of the model remain static across time-steps. A model block diagram is depicted in **Figure 2.2**. Because the overall model processes information through a series of sub-models consisting of one or more neural network layers, the entire model may be trained end-to-end via back-propagation. [58]

2.4.3 Baseline Model

Industry standards for predicting return to work are based on regression models that predict case duration given information about a worker collected early in their recovery process. This information includes characteristic data that was known at the time of injury, such as age, biological sex, job class, and comorbidities. Additionally, longitudinal information, such as initial diagnoses, collected early in the recovery process, the first week or so, may also be used. All of this information is then passed into a model that returns the estimated RTW date. ODG and MDGuidelines have deployed these models in their web-based subscription services for treatment guidance and resource management. [69, 70] The model sold by ODG was developed with the same dataset we used for this study. However, these models are proprietary and not available for use without purchase. [68]

As such, we sought to create a baseline model analogous to the industry-standard proprietary models. [49] Our approach was to modify the proposed model only to utilize the information available to the industry-standard proprietary models. This meant that the baseline model functions identically to the proposed model for the first week. However, after the first week, the baseline model receives no new longitudinal information. All daily inputs contain only the known characteristic information. This setup was employed at both training and testing time.

This process was carried out through the used of "missing observations" tokens where $t > 7$. We provide more technical information about these tokens along with high cardinality categories in **Appendix Section A.1.3.1**.

2.5 Experiments & Results

We now present experiments that focus on understanding the behavior and benefits of the proposed model for use in the reformulated RTW prediction task. We first catalog our main experimental questions. We then provide an overview of the dataset and experimental setup employed to answer these questions. A description of the dataset then follows this. We finally present our main experimental results, followed by subpopulation analyses.

Questions. These experiments seek to answer two related questions. The first relates to the primary study objective, which is to assess the value of longitudinal observations. The second is an initial investigation of potential bias in the proposed model. These questions are:

1. *Does the additional longitudinal observations lead to improved predictions of future work status?* (**Section 2.5.3**, **Figure 2.3**)

2. *How does the proposed model perform on subpopulations of interest?* (**Section 2.5.4**, **Figures 2.2** and **2.4**)

2.5.1 Data & Experimental Setup

Data. We utilized the Peers Health Ohio Workers' Compensation Dataset for this work. This dataset contains longitudinal workers' compensation claims information for over 1.2 million workplace injuries collected in Ohio from January 2001 to October 2010. For each injury record, demographic information describes the age, sex, and job type of the worker at the time of their injury. This demographic information is accompanied by time-stamped longitudinal information representing diagnoses and procedures. (Procedures are activities rendered by healthcare providers to improve a worker's health, like physical rehabilitation.) Finally, for each injury record, the dates of a worker's departure and return to work are recorded (an injury record may contain multiple depart from and return to work dates). This work was conducted with approval from the University of Michigan Institutional Review Board. Peers Health provided the data underlying this study.

Model Development. Based on preliminary experiments, we sample $300,000$ injuries to achieve a suitable trade-off between model training time and predictive performance. We exclude all injuries with case durations of less than seven days, as a predictive model would provide marginal utility for these cases. For our study, we set the offset to one week (or seven days, thus $\phi = 7$) to predict work status for 1 week in the future. Note, when $t + \phi$ surpasses the last observed $o^W_{i,t}$ value, the last observed $o^W_{i,t}$ value is filled forward. Injury cases are only considered to have reached completion once the injured worker has reached their maximal recovery or has transitioned to long-term disability at the 365-day cutoff we employ above. [85] Thus, the work status observed on their last day is likely to be their lasting work status.

We split the dataset evenly between training, validation, and test datasets ($1/3$, $1/3$, $1/3$, respectively). The size of the sub-models, f_{in}, f_{mid}, and f_{out}, and the activation functions for all their layers were model hyperparameters. Additional hyperparameters included the width and depth of each sub-model, drop-out rate applied to the inputs and between layers, learning rate, and layer activation functions. Thus, they were selected as a part of the hyperparameter search process. We used *hyperband*, selecting hyperparameters that yielded the best performance in terms of validation AUROC. [86] For full details on the possible values for each hyperparameter, please see **Section A.1.3.1**.

After finding the best hyperparameters, the final model training was conducted using a combined dataset consisting of both the training and validation datasets. Training was conducted using stochastic gradient descent (SGD) with early stopping based on the validation AUROC. Finally,

the held-out test dataset was used to evaluate the performance of our proposed model against the baseline model.

To compare between the baseline model to our proposed model, we sought to ensure that they had the same overall capacity and used the same training procedure. As such, we used the same framework and searched over the same hyperparameters; and only limited the baseline so that it only used information typically used for the proprietary models. This setup replicates the data used to create the Cox proportional hazards models traditionally used for this task with an added benefit; the baseline model can learn from the initial observation data and the observation timing. By using the same architecture, searching over the same hyperparameter space, and only using the first seven days of observations, we sought to create a strong baseline representing the best possible performance of existing proprietary models. The restriction to the first seven days is an optimistic interpretation of existing proprietary models, as they generally only use information collected at the time of injury. By comparing our proposed method against the baseline model, we can estimate the potential improvement of using longitudinal observations over the current industry approach of using information collected around the time of injury.

We implemented the entire data preprocessing, model training, and evaluation pipeline using python 3.6.9, using the TensorFlow docker container (tag:latest-gpu-jupyter accessed on June 9th 2021) running on an Ubuntu 18.04 workstation with an Intel Xeon 6146 CPU, 256 gigabytes of RAM, and an NVIDIA Titan V graphics card. [87] The proposed and baseline models were implemented using TensorFlow and Keras. [88–90] Additionally, we utilized the SQLite, SKLearn, NumPy, pandas, and tableone python packages. [91–96] We have released our data transformation code and model training python framework on GitHub. A U.S. Utility patent application covers the methods and approaches described above. [97]

Evaluation. We evaluated the daily predictions on the held-out test dataset of OIs. For each injury, all daily predictions, $\Pr(y_{i,t} = 1)$, were compared against the true label, $y_{i,t}$. We use this time-step-level approach (also known as time-horizon approach [5]) as model users can intervene on patients daily. We measured discriminative performance using the receiver operating characteristic (ROC) and the area under it (AUROC). Calibration performance was assessed using calibration curves and the expected calibration error (ECE). [98, 99]

To assess the variation in performance, we computed 90% confidence intervals for all curves and measures. Confidence intervals were generated using bootstrap sampling; in this procedure, the population of injuries in the test set was resampled with replacement 100 times to estimate model performance under varying distributions of injured workers.

We also evaluated several models using simpler machine learning architectures. We used L2-regularized logistic regression and random forest regression. We discuss these evaluations in **Ap-**

pendix Section A.2.

2.5.2 Dataset Characteristics

We first describe the characteristics of the dataset we used for this study. After applying our minimum case duration of seven days exclusion criteria, we had $294,103$ OI cases.

Table 2.1: Population characteristics. Demographic information (age and biological sex) is equally used between the baseline and proposed models. Both models do not use observations such as diagnoses and procedures equally, as the baseline model is limited to longitudinal observations within the first week of the injury. These observation characteristics are counted per worker for the baseline and proposed models.

Population Characteristics		n: 294,103
Demographic Characteristics		Entire Population
Age, Median (IQR)		35 (26, 45)
Biological Sex, n (%)		
n missing: 3,876		
Female, n (%)		92,674 (31.9)
Male, n (%)		197,553 (68.1)
Case Duration (days), Mean (Standard Dev.)		88.9 (111.7)
Observation Characteristics Per Worker	Baseline Model	Proposed Model
Number of Diagnoses, Median (IQR)	0 (0, 0)	1 (1, 2)
Number of Procedures, Median (IQR)	4 (2, 6)	5 (3, 10)

The median age of injured workers at the time of injury was 35 years old, with an interquartile range (IQR) between 26, 45 years old. Most of the workers were male, with only 31.9% having a biological sex of female. In total, these workers represented 595 different occupation classifications, with the five most common occupations being: city employees, restaurant workers, school district employees, nursing home workers, and automobile service workers (**Appendix Table A.4**). The median number of diagnoses observed per injured worker was 1 (IQR: 1, 2), and the number of procedures was 5 (3, 10). When limited to observing the first week of the worker's recovery, as in the case of the baseline model, the number of diagnoses observed was 0 (0, 0). The number of procedures was 4 (2, 6). See **Appendix Section A.1.3.1** for a discussion of this. The most commonly observed diagnoses and procedures are categorized in **Appendix Table A.5** and **Appendix Table A.6**. Since the RTW observations were not limited to the first week, the baseline and our proposed model observed 1.1 (standard deviation: 0.5) RTW events per injured worker. These numbers are also summarized in **Table 2.1**.

2.5.3 Model Performance

We now investigate the performance of the proposed and baseline models. By doing so, we seek to answer the question: *Does the additional longitudinal observations lead to improved predictions of future work status?*

When evaluated on daily predictions generated over the test dataset, our proposed model had an AUROC of 0.728 (90% confidence interval: 0.723, 0.734), compared to the baseline model's AUROC of 0.591 (0.585, 0.598). In terms of calibration, our proposed model had an ECE of 0.004 (0.003, 0.005) versus 0.016 (0.009, 0.018) for the baseline model. The values, along with ROC curves and calibration curves, are displayed in **Figure 3 2.3**. Despite under-estimation of RTW likelihood in low-likelihood cases, the proposed model displays better overall calibration (smaller expected calibration error) than the baseline model. The baseline model shows under-estimation of RTW likelihood in both low- and high-likelihood cases but also shows over-estimation in mid-likelihood cases.

From this experiment, we can see that the proposed model utilizing longitudinal observations outperforms the baseline model. This suggests that longitudinal observations may provide higher quality estimates of future work status.

2.5.4 Subpopulation Analysis

We now investigate our second question: *How does the proposed model perform on subpopulations of interest?* To answer this, we computed the performance of the proposed and baseline model on subpopulations of workers. These subpopulations were defined by stratifying the worker population by age and sex.

When examining the performance of our proposed model and the baseline model in subpopulations of injuries occurring in workers of different ages or sexes, their performance varies slightly. However, our proposed model always matches or exceeds the baseline model, **Figures 2.2** and **2.4**. There is only one subpopulation and performance measure where the proposed model does not demonstrate statistically significant performance. This is the ECE of workers aged less than 25 years old.

In all the other subpopulations, the proposed model outperforms the baseline model. It is important to note that the proposed model's performance varies across the subpopulations, but these differences do not appear to be very large or statistically significant. Although this analysis is not proof that the proposed model is not biased, it does show that we can expect the proposed model to work consistently and provide benefit over the baseline model across these subpopulations.

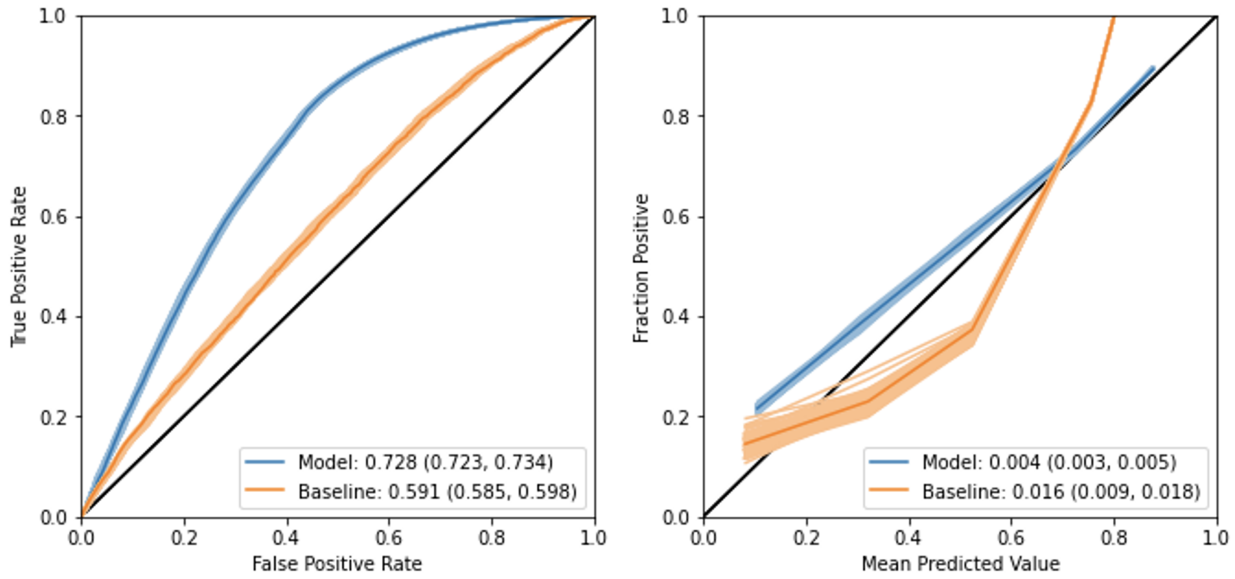
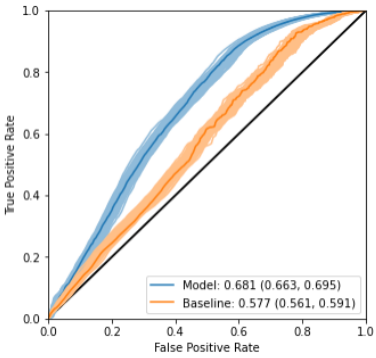
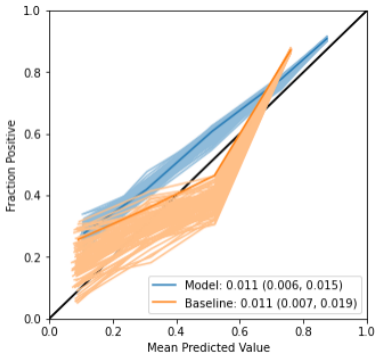
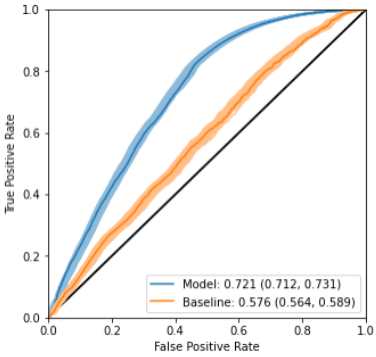
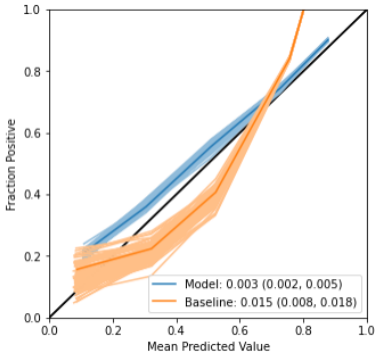
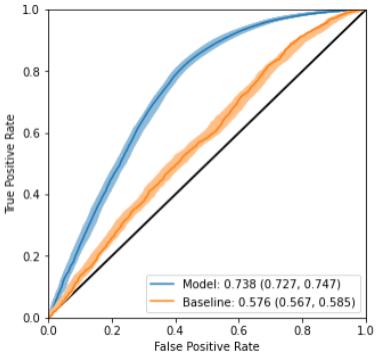
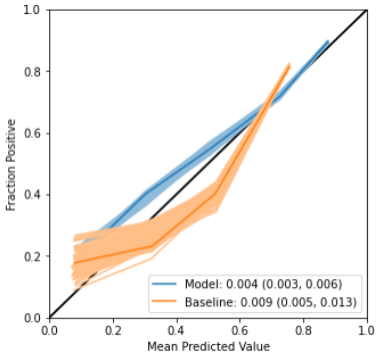
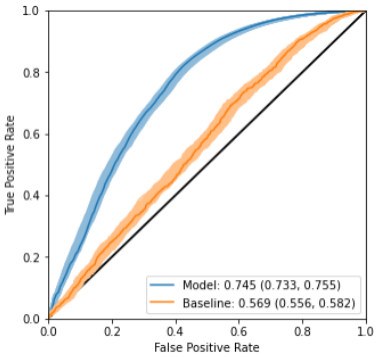
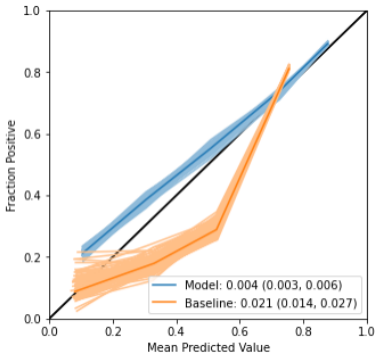
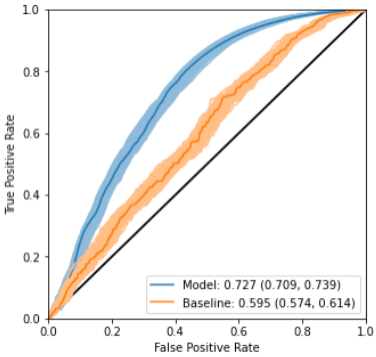
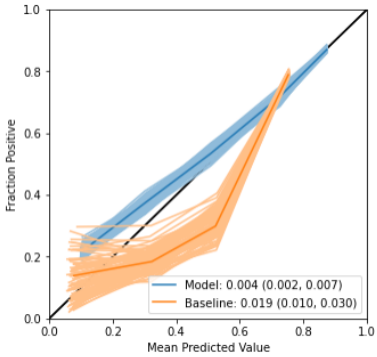
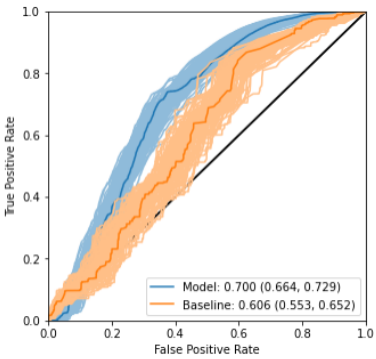
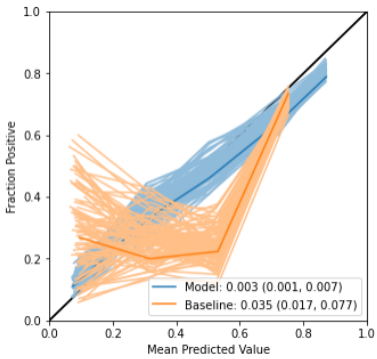


Figure 2.3: Predictive performance of the proposed model compared to the baseline model. In the left subfigure, discriminative performance is plotted in terms of the ROC, the proposed model is blue, and the baseline model is orange. The proposed model has a significantly better discriminative performance by dominating the ROC curve of the baseline model and having a larger area under the ROC curve, which is depicted in the legend. Quintile calibration curves for the proposed model (blue) and baseline model (orange) are displayed in the right subfigure. Despite under-estimation of RTW likelihood in low-likelihood cases, the proposed model displays better overall calibration (smaller ECE) than the baseline model. The baseline model shows under-estimation of RTW likelihood in both low- and high-likelihood cases but also shows over-estimation in mid-likelihood cases.

Table 2.2: Performance on Subpopulations of Injuries Based on the Age of Workers at Time of Injury. The discriminative performance is depicted with ROC curves and quantified in terms of AUROC. Calibration performance was assessed in terms of calibration curves and the ECE. All measures were bootstrap sampled at the injury level with replacement to create 90% confidence intervals, depicted as lighter curves and in parentheses. Generally, the findings from the entire population hold in each age subpopulation, with our proposed model outperforming the baseline model in terms of both discrimination and calibration. We observe the proposed model's worst discriminative performance in the subpopulation of workers under 25 years old. Additionally, this subpopulation has equivalent ECEs between the two models.

Age Range (years)	**n**	**Discrimination ROC Curve (AUROC)**	**Calibration Calibration Curve (ECE)**
< 25	$19,231$		
$25 - 35$	$27,431$		
$35 - 45$	$25,437$		

$45-55$	$17,657$		
$55-65$	$6,888$		
≥ 65	$1,313$		

2.5.5 Summary of Simpler Model Architecture Experiments & Results

The additional experiments utilizing simpler model architectures discussed in **Section A.2** reinforce the findings of the primary experiments. These simpler model architectures generally had worse discriminative performance than the proposed deep learning model. However, simpler architectures using longitudinal observations outperformed their respective baselines (without longitudinal observations). For example, the logistic regression model using longitudinal observations had an AUROC of 0.607 $(0.606, 0.607)$ compared to an AUROC of 0.581 $(0.580, 0.581)$ for the logistic regression model without longitudinal observations (see **Figure A.3** for full details).

Additionally, when we examined the importance of longitudinal observations, we saw that the longitudinal observation data played a prominent role in predicting future work status. We ob-

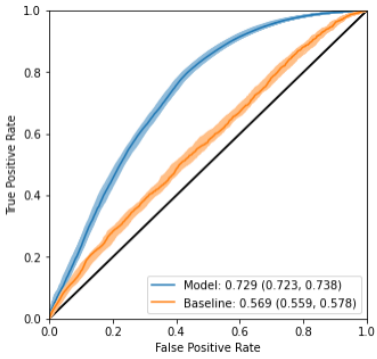
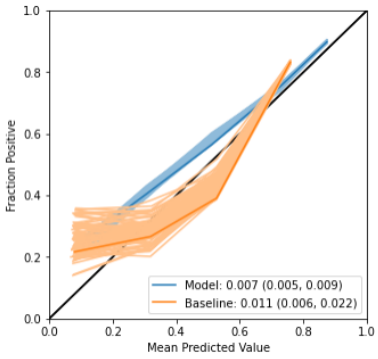
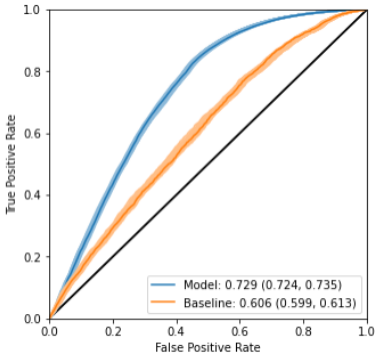
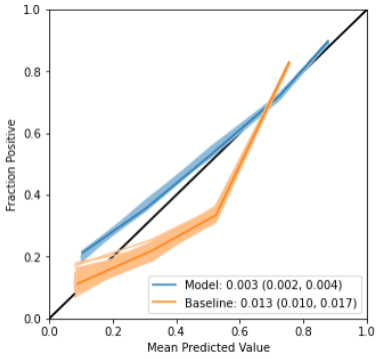
Biological Sex	n	Discrimination ROC Curve (AUROC)	Calibration Calibration Curve (ECE)
Female	30,819		
Male	65,932		

Figure 2.4: Performance on Subpopulations of Injuries Based on the Biological Sex of Worker. Discriminative Performance is depicted in terms of discrimination with ROC curves and quantified in terms of AUROC. Calibration performance was assessed in terms of calibration curves and the ECE. All measures were bootstrap sampled at the injury level with replacement to create 90% confidence intervals, depicted as lighter curves and in parentheses. The findings from the entire population hold in each sex subpopulations, with our proposed model outperforming the baseline model in terms of discrimination and calibration. Our proposed model has slightly worse calibration (in terms of ECE) when used for females than the males.

served that 9 out of the top 25 features of the logistic regression model corresponded to longitudinal observations. These features and their coefficients are displayed in **Appendix Table A.9**. Using *permutation importance*, we also observed the importance of longitudinal observations. Procedure Codes, a type of longitudinal observation, constituted the second largest group of features that impacted the discriminative performance of the logistic regression model, **Appendix Table A.5**.

2.6 TemporalTransformer Package

As mentioned above, the data used for the OI prediction is made up of a combination of characteristic information linked with longitudinal observations. Both the characteristics and longitudinal observations have elements that are very high cardinality. Similar dataset structures can be found across the domain of healthcare. This structure often necessitates additional aggregation and transformation mechanisms when preparing data for ML model development. Custom pipelines often are costly to develop, in terms of both time and money, may not be resilient to changes in datasets, and obscure many of the transformation assumptions. We designed a Python package to aid in preparing the data for this project. This package, Temporal Transformer (T2), is designed to ingest EHR or claims datasets and transform them for use with our proposed model (or other sequence modeling approaches).

T2 is a Python package (GitHub) that standardizes, simplifies, and speeds up the healthcare data preprocessing process. We designed T2 to make the data transformation process as easy as possible so that clinicians and researchers can develop models faster by spending less time prepping data. Several other packages have similar functionality to T2, namely FIDDLE and gpmodels. [100, 101] These packages were unavailable at the outset of this project.

To do this T2 extends the relational database structure often employed by analytical databases and research data warehouses (RDWs) tied to EHRs. Data tables and their configuration are passed to T2, which then automatically transforms the underlying data into a sequential format. This transformation is saved so that the exact same transformation can be applied to held-out partitions of the data or other datasets.

Table configuration helps to determine the relationship between tables and how data columns should be transformed. Tables have a tabular structure like standard SQL tables and are composed of one or more data columns. Additionally, tables must have an ID column, with each ID representing a unique entity (*e.g.*, patients or workers). If the table represents longitudinal observations, it must either have time-stamps or time-spans (start/stop times). The presence of temporal information informs T2 on which tables contain characteristic or longitudinal observation data. The ID column enables T2 to link information across tables. Table configurations denote the names and data types of all columns in a table. In **Appendix Section A.3.2**, we show the T2 configurations

used for the OI project.

In addition to loading the table data and table configurations, several other parameters such as the time-step duration and data filtration methods must be defined. The time-step duration is the time between successive predictions; in the case of the OI project, we set this to 1 day. The data filtration methods help to remove outliers numerical (*e.g.*, abnormally large or small values) and categorical data (extremely infrequent categories). We set the filtration parameter for the OI project to be 0, so no data was filtered.

Once these additional parameters are provided, T2 transforms the underlying dataset automatically. The general transformation process includes filtration, aggregation, and normalization. Extreme values are filtered out of the dataset according to the specified filtration process and parameter. Longitudinal observation data are then aggregated to correct temporal granularity. The functions used in aggregation depend on the data type of the column. Real valued columns are aggregated with summary statistics (*e.g.*, min, mean, median, max). Low cardinality category columns have their categories counted. And high cardinality categories are packaged into a list of all the distinct categories observed. For example, all the diagnoses a worker receives on a given day are concatenated into a list. Columns in characteristics tables do not go through the aggregation process. Normalization applies Z-score normalization to the transformed real valued and low cardinality categories. The high cardinality categories are ignored during this step. This process yields a transformed dataset and a transformation configuration information that can be saved and loaded.

The transformation process utilizes an in-memory database management system (DBMS) in order to achieve good run times. This DBMS stores all of the transformation steps for traceability of the transformation. Additionally, if persistence is desired or memory is an issue, DBMS can be run on disk. The transformed dataset may be exported out of the DBMS in several ways. The two most notable export formats are Tensorflow and Numpy.

The Tensorflow export utilizes Tensorflow's built in Dataset functionality. Additionally, it provides the initial ingestion layers for a TensorFlow model to be able to receive the transformed data. We used this option for the OI project and f_{in} was initialized automatically by T2. The benefit of this is that the list of high cardinality categories can efficiently be passed between the dataset and the model with no additional intervention needed on the part of the model developer. The Numpy export produces three-dimensional Numpy arrays packaged into a dictionary with the transformed column names as keys.

The full T2 configuration and transformation process used for the OI project can be found in **Program A.1**. Dataset partitions can be provided to T2; this ensures that transformation will be fit only using training data and that the held-out evaluation data will also be transformed in the same manner. By specifying the same underlying dataset, partitions, and configuration, we could use T2

to generate the transformed data on the fly for all of our model development. The training of the proposed and baseline models took full advantage of the TensorFlow export process using both the TensorFlow Dataset and the initial TensorFlow ingestion models. The simpler model architectures used the Numpy export process as the SKLearn models required Numpy arrays and could not take advantage of the TensorFlow ingestion models.

2.7 Discussion

We found that utilizing longitudinal observations improves the performance of RTW prediction compared to approaches that only use the information at the time of injury. Despite both models using the same RNN based deep learning architecture, the proposed model outperformed the baseline model in terms of discrimination and calibration. The baseline model is analogous to the current state of the art in RTW prediction, as it uses information collected at the time of injury to generate predictions. In contrast, the proposed model uses treatment information collected daily to update RTW predictions. The performance differences we observed between our proposed model and the baseline model show the potential practical benefit of reframing the RTW task as a dynamic prediction task.

Our proposed approach uses standard longitudinal data routinely collected by workers' compensation programs and exploits deep learning capabilities to build a dynamic model that outperforms methods that only use information collected around the time of injury. Our python framework, T2, transforms readily available injury claims data into sequences of daily observations. These observations encode characteristic information, like diagnoses and treatment codes, and are combined with longitudinal observation data (*e.g.*, worker demographics). The framework then trains an RNN based deep learning model to map these daily observations to the future work status of an injured worker. Thus, the learned model could be used to repeatedly generate RTW predictions given a sequence of longitudinally observed diagnoses and treatments.

Updating RTW in response to observed diagnostic and treatment information could be valuable for employers, physicians, and OI recovery managers. Existing RTW prediction models and treatment guidelines software have already been implemented into EHR systems. [43, 44] Our proposed approach may provide additional value as the dynamic assessment of the worker's future work status relates to how physicians and other clinicians assess injuries over time. Like the proposed model, physicians update their understanding of an injured worker's recovery and future recovery prognoses based on information they collect over time. Additionally, this formulation helps to monitor populations effectively. As near real-time observations are collected for individual injured workers, the proposed model can generate RTW estimates. RTW estimates can then be used by OI recovery managers to allocate treatment resources to injured workers. These estimates

can also be used by people with managerial responsibility for workforce coverage in industry organizations. Furthermore, this model may eventually be used to help answer "what-if questions"; using the model to assess the impact of potential treatment choices on work status could help support clinical decision making. Altogether, the dynamic prediction of work status may assist in managing OIs, ultimately positively impacting injured workers and organizations that support them (*e.g.*, workplaces and governmental organizations).

Implementation. To be useful, this dynamic model needs to be implemented within feasible workflows. We will briefly sketch potential implementations method that would enable predictions to be used by recovery managers. This implementation would utilize insurance claims data. A hosted model fed claims data automated or manually could provide predictions for recovery managers and employers. Another potential implementation mirrors a project implemented at Kaiser Permanente [43, 44] with direct integration of the proposed model into an EHR via Health Level Seven (HL7), Fast Healthcare Interoperability Resources (FHIR), or other application programming interfaces. Integration with an EHR would allow physicians and other clinicians to get real-time RTW predictions embedded directly in their clinical documentation and decision-making workflows. Transferring the model to the EHR setting would require careful validation and may require additional training with data collected directly from EHR systems. [17]

Both potential implementations raise many questions, ranging from privacy concerns to data infrastructure issues. [17] Of note, evaluation of OIs in terms of RTW depends on desired use case and implementation. We used a daily evaluation for this initial development study as it is the most plausible evaluation frequency. We present potential implementations not as finalized solutions but as ideas to inform future studies in this space.

Bias & Fairness. An essential set of issues that arise as we consider the translation of this model from "bench to bedside" are the issues of algorithmic bias and fairness. These must be carefully considered and studied before, during, and after any implementation of this work. [102] As noted in **Section 2.5.4**, our results show that the proposed model outperforms the baseline model for all age and sex subpopulations. This is an example of some of the analysis necessary, but not sufficient, to identify sources of algorithmic bias. Although this assessment was not the primary focus of this work, we present a brief discussion of some potential issues regarding bias and fairness of this proposed model.

One potential issue is the non-representativeness of the underlying claims data employed to develop the models. For example, undocumented workers may be underrepresented in this dataset. Generally, these workers are less likely to have their OIs properly documented, treated, and assigned to workers' compensation resources by employers. [103] Other socio-economic factors

obscured from claims data may also exert pressure on RTW decision making, for example, RTW duration has been shown to correlate with the size of an injured worker's family. [104] Blindly developing and implementing models may reinforce negative structures in society that harm vulnerable groups of people. As such, it could be problematic to implement the proposed model directly. Instead, we emphasize that these challenges are areas for careful future study, which should combine additional analytical work with further data collection and research.

Limitations. The main set of limitations pertains to the inaccessible baseline models and the deep learning architecture used for this study. To assess the proposed approach's improvement yields, we must compare it against a representative baseline. We trained a baseline analogous to proprietary models by limiting the data to the first week after injury. [49] We tried to ensure parity in capacity between our proposed model and the baseline model using the same framework and hyperparameter space. We believe this yielded a generous baseline, representing the predictive performance of using information collected around the time of injury. We note that this is not an attempt to measure the performance of existing proprietary models.

In addition, we employ deep learning approaches, which are powerful but problematic. In initial experiments our proposed had over 3 million parameters. This presents issues terms of complexity, power usage, and interpretability. [105–107] Work described in **Appendix Section A.2** examines longitudinal observations' impact on future work status prediction using simpler machine learning architectures. These results suggest that longitudinal observation data is vital in predicting future work status. Notably, these models demonstrate worse discriminative performance than the proposed model implemented with deep learning. Further study is needed to explore architecture tradeoffs fully. Although we observed performance degradation when using simpler model architectures, some of the benefits of longitudinal data are still realized under simpler architectures. There may be modeling approaches that provide similar performance benefits to deep learning with less complexity and more interpretability. [108–110] This could be a fruitful direction for future research.

Although the proposed model shows an improvement over the baseline there are several other limitations that must be headed when considering the generalizability and potential implementation of this model. Several of these limitations pertain to the dataset we used to create and validate our model. We utilized a large dataset from the state of Ohio's workers' compensation program, containing OIs and subsequent observations observed between 2001 and 2010. Using data from a single state limits the potential generalizability of the model to other regions, as some of the data collected are specially tailored to Ohio (*e.g.*, procedure codes specific to the state of Ohio's workers' compensation program).

Additionally, other U.S. states or regions outside of the U.S. may have a different composition

of occupations, injuries, and treatments. Moreover, diagnoses and treatments may have changed since the end of the data collection. For example, this dataset is unlikely to capture the recent shift away from opioid-based analgesics in treating pain. [46, 111] Despite validating the model in a single region, our work provides a valuable foundation for which to replicate our study for other regions.

Given the scope of this work, we focused entirely on utilizing retrospectively collected data. The proposed model would need to be studied with a prospective implementation to fully assess the utility of using longitudinal observations in real-world usage. Finally, using claims-based workers' compensation data provides a limited view of the recovery process, especially from the lens of algorithmic bias. Although our claims dataset contains time-stamped information regarding diagnoses and treatments, this is an incomplete depiction of recovery from OIs. For example, job type is a limited representation of the worker's occupation, and a great deal of recovery depends on psychosocial factors that are not explicitly captured through claims. [45, 112] With additional psychosocial information, the proposed framework would likely be able to create models with greater predictive performance that account for these factors.

Another limitation is that we do not fully understand how the benefits of longitudinal observations change over the duration of OI cases. Analysis like Oh et al. [110], Singh et al. [6], and Wong et al. [5] may show how much of an early warning benefit the proposed model provides over the baseline model.

To the best of our knowledge, our study is the first to evaluate the potential of dynamically predicting RTW for injured workers using longitudinal observations. Future work using other large claims or EHR datasets may address the abovementioned limitations.

Conclusion. In this chapter, we established the value of using longitudinal observations for the return to work prediction task by comparing approaches that use information collected in the first week of an occupational injury (OI) to longitudinal information collected throughout recovery. We proposed a new formulation for occupational injury (OI) prediction as a dynamic work status prediction task for this comparison. We utilized an approach that transforms longitudinal claims data into a sequence of observations. These longitudinal observations were fed to a recurrent neural network (RNN) based model to generate predictions about an injured worker's future work status. The model yields updated estimates as new longitudinal observations are collected daily. Thus, the longitudinal observation approach could help physicians and payers efficiently manage large populations and enable industrial organizations to better plan for their workforce needs. Supposing our initial findings are borne out through subsequent modeling and validation studies, dynamic prediction of return to work (RTW) may provide crucial support in clinical decision making, addressing a problem that plagues many insurers, governments, and workers.

CHAPTER 3

Mind the Prospective Performance Gap

3.1 Introduction

To date, the application of ML for patient risk stratification in clinical care has relied almost entirely on "retrospective" electronic health record (EHR) data. [113, 114] That is, researchers typically train and validate models using data sourced from a database *downstream* from data used in clinical operations (*e.g.*, a RDW or MIMIC-III). [113] These data are extracted, transformed, and stored to serve researchers without interrupting hospital operations. While critical to initial model development, model evaluation using such data may not represent prospective model performance in clinical practice. Importantly, it is the *prospective* or "real-time" model performance that ultimately impacts clinical care and patient outcomes. [115, 116] Although retrospective performance serves as an approximation of prospective performance, the validity of such an approximation relies on the assumption that the two datasets come from the same distribution (*i.e.*, the datasets have no significant differences in the relationships of covariates and outcome). However, many ML models are developed and validated with datasets that do not accurately represent their intended prospective use. [117] Without prospective evaluation, it is impossible to estimate *a priori* how a model will perform when deployed.

The need for prospective validation has been previously recognized in the context of screening for diabetic retinopathy. [118–120] However, these studies rely primarily on imaging data, so the difference in infrastructure for model development and deployment is minimal. Researchers have started reporting prospective performance with respect to models that rely on structured EHR data. For example, Kang et al. [121] prospectively compared a model to predict in-hospital resuscitation events with existing standards of care (*e.g.*, rapid response team activation). In addition, Brajer et al. [122] prospectively validated an in-hospital mortality prediction model. While these studies take an essential step towards model integration in clinical care, they do not specifically assess the root cause of discrepancies between prospective and retrospective performance.

To date, factors driving the differences between prospective and retrospective model perfor-

mance have primarily been attributed to clinical workflow changes [25, 123] or patient populations. [124, 125] For example, a global pandemic might lead to differences in personal protective equipment protocols. This change in gowning and gloving may impact infectious diseases within the hospital, which may affect model performance. However, such changes are difficult, if not impossible, to anticipate before an outbreak. [126, 127]

Here, we compare the effects of *temporal shift* (*i.e.*, changes due to differences in clinical workflows and patient populations) on model performance to another kind of shift: *infrastructure shift*. We define infrastructure shift as changes due to differences in the data extraction and transformation pipelines between retrospective and real-time prospective analyses. For example, some data available retrospectively may not be available prospectively because of the processing pipeline at one's institution (*e.g.*, vitals might be backdated by the clinical care team). Differences in how the data are sourced and preprocessed between retrospective and prospective pipelines may be more systematically addressed if recognized. However, it is currently unknown to what extent degradation in prospective performance can be attributed to changes in temporal shift vs. infrastructure shift.

In this chapter, we explore the prospective validation of a data-driven EHR-based patient risk stratification tool for predicting hospital-associated *Clostridioides difficile* infection (CDI) at University of Michigan Health, a large tertiary care academic health system. CDI is associated with increased length of stay, hospital costs and considerable morbidity and mortality. [128–131] The ability to accurately predict infections in advance could lead to more timely interventions, including patient isolation and antibiotic stewardship strategies, curbing the incidence and spread of disease. We measure the *prospective performance gap* between prospective and retrospective pipelines. More specifically, we *quantify* how much of the prospective performance gap can be attributed to temporal and infrastructure shifts.

3.2 Contributions

As the field of ML for healthcare advances and more models move from 'bench' to 'bedside,' prospective validation is critical. Most ML models are developed and initially validated using retrospective data. We explore the impact this disconnect can have on the prospective performance gap (*i.e.*, the difference between prospective performance and retrospective performance) through a case study in which we prospectively validated an EHR-based patient risk stratification model for CDI. Our contributions are as follows:

- *Formalization of the notion of the prospective performance gap when validating ML-based models in clinical care.* We provide a formal definition of the prospective performance gap.

Using this, model developers can calculate the amount of performance degradation observed when transitioning into model implementation from model development.

- *Formulation of the relationship between the prospective performance gap, temporal shift, and infrastructure shift.* We introduce and formally define two constituent components of the prospective performance gap. The first component captures changes in the population and care processes, *temporal shift*, and the second represent differences in the data infrastructure used for development and implementation, *infrastructure shift*.

- *Characterize the differences between a retrospective pipeline and a prospective pipeline and the resulting impact on the performance gap.* We provide the first, to our knowledge, rationale and example for why infrastructure shift may contribute to the prospective performance gap.

- *Quantifying how much of the performance gap can be attributed to temporal shift and infrastructure shift.* We expand the formal analysis of the prospective performance gap to isolate its components.

- *Develop methods and approaches to identify contributors to the performance gap.* These new methods enable model developers to determine which features contribute most to temporal and infrastructure shift. Features associated with significant contributions to infrastructure shift may be targeted by model developers to be fixed in model updates.

- *Highlight approaches for mitigating the effects of differences in retrospective versus prospective data infrastructure on the performance gap.* By focusing on our model implementation, we provide examples of improvements that model developers may consider to ameliorate the effects of infrastructure shift.

We do not present a new ML algorithm or architecture, but rather share novel insights gained through our experience developing and validating an EHR-based model for patient risk stratification. We highlight practical considerations for model developers moving from the 'retrospective' to 'prospective' setting. Given that the ultimate goal of ML for healthcare is to improve patient care, early considerations regarding prospective validation are critical to ensuring success.

The results of the this work were presented at the 2021 Machine Learning for Healthcare Conference and published in the Proceedings of Machine Learning Research[1] . [17]

[1]This publication was co-authored by Erkin Ötleş, Jeeheh Oh, Benjamin Li, Michelle Bochinski, Hyeon Joo, Justin Ortwine, Erica Shenoy, Laraine Washer, Vincent B. Young, Krishna Rao, and Jenna Wiens. EÖ , JO, and JW led the core study design. EÖ developed the prospective performance gap attribution methods. Data analysis was conducted by EÖ and JO. Manuscript preparation was conducted by EÖ and JW. All authors assisted in manuscript revisions. EÖ was primarily responsible for the formalization of the prospective performance gap, along with its separation into

3.3 Problem Setup & Related Work

Risk stratification model performance evaluation utilizes a dataset, $\mathcal{D}$, composed of model inputs and labels representing a population of patients. Model inputs, denoted as $\mathbf{X}$, are a matrix of real values with each row corresponding to a patient and each column corresponding to an input feature. Labels, $\mathbf{y}$, are a binary vector with each value corresponding to a patient's outcome. A risk stratification model, f, maps each row in $\mathbf{X}$ to a risk estimate, $\Pr(y = 1)$. This risk stratification model may be evaluated using a performance measure function, p, on the given dataset. We evaluate f applied to $\mathcal{D}$ using a performance measure function, denoted as $p : \mathcal{D} \rightarrow \mathbb{R}$. To simplify notation, we assume that the goal is to maximize p during model development (*e.g.*, models producing larger AUROC scores are preferred). This notation implicitly denotes two steps: one, the application of the model to inputs of the dataset to create risk estimates, and two, the evaluation of those risk estimates against the labels of the dataset. The data used for model development and validation is often collected retrospectively. We denote retrospective validation data as $\mathcal{D}_{ret}$; this held-out data is often used to estimate the expected performance of the model via $p(\mathcal{D}_{ret})$.

Although risk stratification models may be developed with retrospective data, they are often used prospectively. During implementation, they may be connected to data streams, like near-real-time EHR datasets; we label this data as $\mathcal{D}_{pro}$. As noted above, prospective performance, $p(\mathcal{D}_{pro})$, may be less than what was estimated by retrospective data $p(\mathcal{D}_{ret})$.

We seek an evaluation framework to quantify the differences in performance between ML models applied in real-time and the anticipated performance based on retrospective datasets. In framing this problem, we expect two sources of differences: one, the shift in the relationships between the features and labels over time due to changes in clinical workflows and patient populations; this is related to the concept of dataset shift. And two, the difference in the infrastructure for extracting data retrospectively versus prospectively. We now briefly discuss dataset shift and the differences in data infrastructure.

Dataset Shift

After models are developed, their performance tends to degrade over time. [24, 125] This problem is partly due to the healthcare environment's non-stationary behavior. [132, 133] Dynamic changes in the environment leading to changes in patient information are termed *dataset shift*, also known as *covariate shift* and *concept drift*. Data may be observed or collected over a time interval. We denote an example time interval a as T_a. The set of input and label data for T_a are denoted as $\mathbf{X}_a$ and $\mathbf{y}_a$ respectively.

temporal and infrastructure shifts, and developing methods to identify factors impacting the prospective performance gap. EÖ collaborated with other authors to present and formalize the other core contributions presented in this chapter.

Formally defined, dataset shift occurs when $\Pr(\mathbf{X}_a, \mathbf{y}_a) \neq \Pr(\mathbf{X}_b, \mathbf{y}_b)$ where T_a and T_b are distinct time intervals. Dataset shift occurs not only over the joint distribution $\Pr(\mathbf{X}, \mathbf{y})$, but also the covariate $\Pr(\mathbf{X})$ distribution, the class distribution $\Pr(\mathbf{y})$, the posterior distribution $\Pr(\mathbf{y}|\mathbf{X})$, and the conditioned covariate distribution $\Pr(\mathbf{X}|\mathbf{y})$; each reveals different facets of a potentially complex changes in the environment and patients we seek to risk stratify. Although dataset shift is recognized to impact the performance of models developed for medicine, it is unclear what the expected effect size should be and how to untangle its effects from differences in prospective and retrospective infrastructure. [132, 134–139]

Data Infrastructure

To highlight how infrastructure may impact observed performance, we now share important details regarding the data extraction and processing pipelines at University of Michigan Health. These pipelines were available for model development and prospective implementation and are summarized in **Figure 3.1**. Though some aspects (*e.g.*, the precise downstream research database) may be unique to our institution, many parts of these data pipelines are generally representative of infrastructure available across academic medical centers.

Retrospective Pipeline. Data used for model development and retrospective validation were extracted from a research data warehouse (RDW) at the University of Michigan. These data were extract, transform, and load (ETL) from University of Michigan Health's Epic EHR instance (Epic Systems Corporation, Verona, WI) and LIS (Soft Computer Consultants, Clearwater, FL). More precisely, the majority of EHR data were extracted nightly from the EHR's underlying Epic Massachusetts General Hospital Utility Multi-Programming System (MUMPS)-based Chronicles database and then transformed and loaded into our instance of Epic Clarity, a Structured Query Language (SQL) database. Then, a second ETL process was carried out, with data passed to a second SQL database, RDW. RDW is based on a health information exchange data model (initially developed by CareEvolution, Ann Arbor, MI). However, to support research operations, it has undergone significant additional custom design, development, and maintenance by the RDW development team at the University of Michigan. The timing of this second ETL process varied. However, the total delay between data being entered in the EHR and arriving in RDW typically ranged between one day and one week. In addition to this data pipeline, our EHR also passes hospital occupancy information directly to RDW via an ADT interface. Finally, RDW also captures information from the LIS using an LR interface. RDW is designed for large queries of transformed EHR data. Thus, we refer to these data as 'retrospective' since they were primarily used for retrospective analyses.

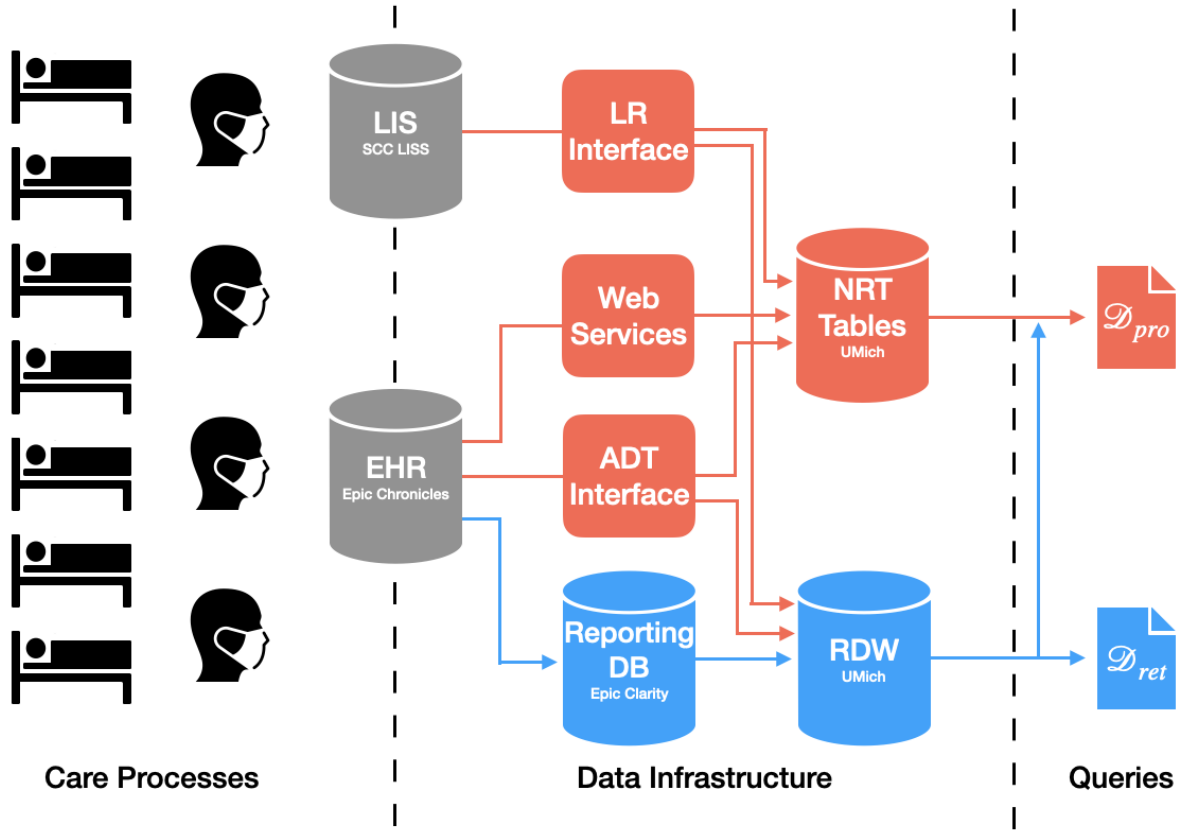


Figure 3.1: Prospective and Retrospective Pipelines. The information generated from the care process is documented in the EHR, produced by Epic Systems Corporation, and the laboratory information system (LIS), produced by Soft Computer Consultants (SCC). Data are extracted from these sources using two different pipelines. The near-real-time prospective pipeline for EHR data is primarily based on a web service architecture and has its information stored in near-real-time (near real-time (NRT)) database tables. It extracts data from the EHR more frequently, with less lead time and processing, allowing for the prospective implementation of predictive models (*i.e.*, it produces prospective datasets, D_{pro}). The bottom pipeline is a retrospective data pipeline that extracts data less frequently but with more curation and processing (*i.e.*, it generates large retrospective datasets, D_{ret}). Both pipelines rely on an lab results (LR) interface that passes information from the LIS and an admission, discharge, and transfer (ADT) interface that passes admission information from the EHR. Components in the pipeline that can be interacted with in near-real-time (*i.e.*, prospectively) are depicted in red. Components in which subsets of data require time to pass before having complete information (*i.e.*, retrospectively) are colored blue. The near-real-time query utilizes historical patient information; although, this information is technically collected via the retrospective pipeline, it is considered up-to-date when queried by the near-real-time query.

Prospective Pipeline. Not all data included in the retrospective model were available in near-real-time through the pipeline described above (*e.g.*, medications or laboratory values for current encounters). Thus, we developed a near-real-time prospective pipeline, which built upon the exist-

ing retrospective pipeline by adding daily updates (ETL) of data that were previously unavailable in real-time. We developed custom EHR web services to update the data necessary for model predictions. Specialized NRT database tables were created to access medications, vital sign measurements, and hospital locations, in near-real-time; *i.e.*, with a delay of less than an 8-hours. This maximum delay corresponds with the maximum duration for recording non-urgent information into the EHR (*i.e.*, the length of a typical nursing shift). In conjunction with the EHR web services, laboratory results and admission information are passed to the NRT tables using the aforementioned LR and ADT interfaces, respectively. Additionally, we continued to use components of the retrospective pipeline to extract historical patient data (*e.g.*, medications associated with previous encounters).

Overall, daily data extracts were inherently different from historical data and required careful validation to ensure queries were accessing the correct aspects of the EHR. Once extracted, we applied identical preprocessing steps. Using these daily data streams, we generated daily risk scores for all adult hospital encounters in our study cohort. Model results were generated daily and stored on a secure server. These scores were not made available to any clinical care team members and were only accessed by the authors.

3.4 Methods

In the context of inpatient risk stratification, we present an evaluation framework to quantify the differences in performance between ML models applied in real-time and the anticipated performance based on retrospective datasets. In framing this problem, we examine two major sources of differences: 1) the shift in the relationships between the features and labels over time due to changes in clinical workflows and patient populations (*i.e.*, temporal shift) and, 2) the difference in the infrastructure for extracting data retrospectively versus prospectively (*i.e.*, infrastructure shift).

To date, it is unknown to what extent differences in infrastructure contribute to the prospective performance gap relative to differences that arise due to temporal shift. Thus, to control for temporal shift and estimate the effect of infrastructure shift on the prospective performance gap, one can use their retrospective pipeline to query data for the prospective validation time period, in order to generate $\mathcal{D}'_{ret}$. Using the datasets $\mathcal{D}_{ret}$, $\mathcal{D}'_{ret}$, and $\mathcal{D}_{pro}$, we may estimate Δ_p^{infra} and Δ_p^{time}.

3.4.1 Estimating the Prospective Performance Gap

Let the labeled data extracted from retrospective and prospective pipelines be denoted as $\mathcal{D}_{ret}$ and $\mathcal{D}_{pro}$, respectively. We assume that we are given a predictive model, f, that maps a given data input matrix vector, $\mathbf{X}$, to a vector of estimated of patient risks, $Pr(y = 1)$. While $p(\mathcal{D}_{ret})$ often

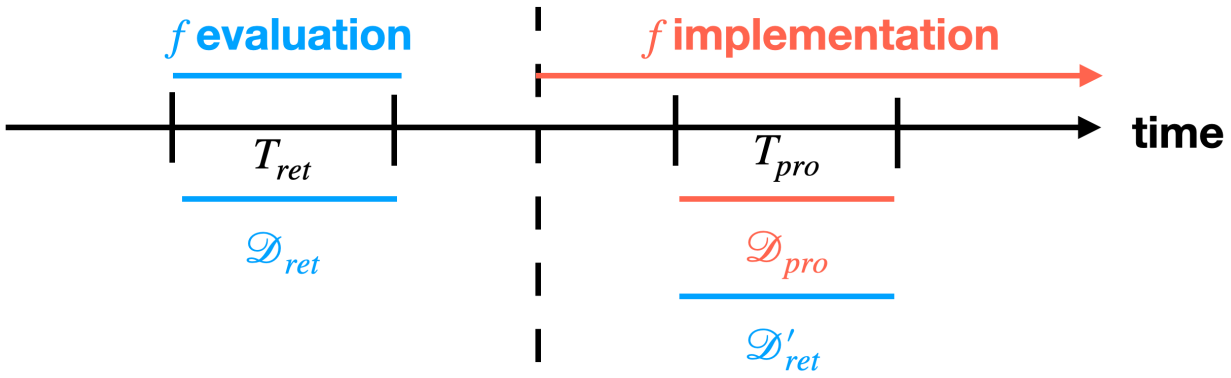


Figure 3.2: Retrospective Evaluation and Prospective Implementation Timeline. The dashed vertical line denotes the model implementation time, marking the start of the silent prospective deployment. Before implementation, the retrospective pipeline was used to retrospectively validate the model, f, applied to $\mathcal{D}_{ret}$. After the model implementation the model is applied to $\mathcal{D}_{pro}$ using the prospective pipeline. Once sufficient time has elapsed, the retrospective pipeline may be used to extract $\mathcal{D}'_{ret}$, data from the same period of time as the prospective dataset.

serves as an estimate for how the model will likely perform in the future, given differences between retrospective and prospective pipelines (discussed above) we do not necessarily anticipate $p(\mathcal{D}_{ret})$ to equal $p(\mathcal{D}_{pro})$. The difference between the retrospective and prospective model performance with respect to p is the *prospective performance gap*:

$$\Delta_p = p(\mathcal{D}_{ret}) - p(\mathcal{D}_{pro}). \tag{3.1}$$

When comparing model performance on retrospective data, $\mathcal{D}_{ret}$, to model performance on prospective data, $\mathcal{D}_{pro}$, there are two unique sources of potential differences: the shift in time period from T_{ret} to T_{pro} (*i.e.*, temporal shift) and the shift in pipelines (*i.e.*, infrastructure shift).

1. **Temporal Shift** arises due to changes in the underlying data generation process (DGP) *over time* (*e.g.*, the evolution of disease pathology, updates in clinical practice/workflows and changes in patient characteristics). If the retrospective and prospective pipelines were identical, then any difference between retrospective and prospective pipeline would be attributed to Δ_p^{time}, defined as:

   $$\Delta_p^{time} = p(\mathcal{D}_{ret}) - p(\mathcal{D}'_{ret}). \tag{3.2}$$

   We control for changes in infrastructure by re-extracting the data from the prospective period using the retrospective pipeline, $\mathcal{D}'_{ret}$.

2. **Infrastructure Shift** occurs when the data returned from the retrospective and prospective pipelines differ, after controlling for changes in time period. We calculate Δ_p^{infra} by compar-

ing performance on the prospective dataset, $\mathcal{D}_{pro}$, to performance on a retrospective dataset generated for the identical time period $\mathcal{D}'_{ret}$ (**Figure 3.2**). Once aligned in time, the only differences come about from the pipeline infrastructure used to create the datasets:

$$\Delta_p^{infra} = p(\mathcal{D}'_{ret}) - p(\mathcal{D}_{pro}). \tag{3.3}$$

The prospective performance gap from **Equation 3.1** can be broken into these two sources of differences:

$$\begin{aligned}
\Delta_p &= p(\mathcal{D}_{ret}) - p(\mathcal{D}_{pro}) \\
&= p(\mathcal{D}_{ret}) - p(\mathcal{D}_{pro}) + \Big(- p(\mathcal{D}'_{ret}) + p(\mathcal{D}'_{ret}) \Big) \\
&= p(\mathcal{D}_{ret}) - p(\mathcal{D}'_{ret}) + p(\mathcal{D}'_{ret}) - p(\mathcal{D}_{pro}) \\
&= \Delta_p^{time} + \Delta_p^{infra}.
\end{aligned}$$

3.4.2 Analyzing Sources of Infrastructure Shift

We may use $\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$ to pinpoint differences in infrastructure. This is because these two datasets represent the same population at the same point in time and only differ in the infrastructure used to collect the data.

First, we can focus on the difference in the estimated risk between the two datasets. Since performance measures like AUROCs summarize risk estimates across populations of patients, they may hide differences. Thus, we propose comparing the risk scores output by the retrospective versus prospective pipeline for every patient observed prospectively. These correspond to $\mathcal{D}'_{ret}$ and $\mathcal{D}_{pro}$, which share the same patients; thus, the model's output for both datasets can be compared directly. Score pairs can be found by aligning risk estimates for each patient using the prospective pipeline and the retrospective pipeline. Graphical or statistical methods may be used to compare these score pairs. For example, extremely discordant prospective and retrospective score pairs may be identified by selecting points far away from the best fit line.

To understand factors that could potentially be addressed with modifications to infrastructure, we may also compare the pair of feature vectors present for each patient in $\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$ by computing differences in feature inputs between the two datasets. The difference in the two data pipelines ($\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$) may be quantitatively assessed for every feature at the patient level. For example, features may be deemed discrepant if their prospective and retrospective values are not exactly equivalent.

Finally, large differences in features can result in minimal differences in estimated risk if the features that vary greatly are not deemed important by the model. Thus, we introduce a new

technique called feature swap analysis to determine which features contribute most to the observed infrastructure shift. Feature swap analysis calculates the effect of swapping out individual aspects of the prospective pipeline with the retrospective pipeline on overall prospective performance. For every feature, we compute the model's performance on a modified version of $\mathcal{D}_{pro}$ where the feature matrix $\mathbf{X}_{pro}$ has a column corresponding to the feature replaced with values from $\mathbf{X}'_{ret}$.

The feature swap importance of feature k, FSI_p^k, is calculated in the following manner.

$$FSI_p^k = p(\mathcal{D}_{pro}^k) - p(\mathcal{D}_{pro}). \tag{3.4}$$

$\mathcal{D}_{pro}^k$ is the modified version of $\mathcal{D}_pro$ where column k has been replace in $\mathbf{X}_{pro}$ with the value of the column k from $\mathbf{X}'_{ret}$. The larger FSI_p^k, the more impact feature k has on the infrastructure shift, making it a feature to target for debugging.

3.4.3 Analyzing Sources of Temporal Shift

To determine sources of model performance degradation due to population and workflow changes over time, we seek to uncover the impact of temporal shift by controlling for infrastructure shift sources. For example, sources of temporal shift can be interrogated by comparing the distribution of features between $\mathcal{D}_{ret}$ and $\mathcal{D}'_{ret}$.

3.5 Experiments & Results

For this experimental work, we leveraged a previously developed model designed to identify hospital-associated CDI in adult inpatients. This model and its development framework were retrospectively validated. [56, 110] We first list our experimental questions. We then describe how this framework was applied retrospectively to develop and validate the CDI model currently in silent prospective deployment. Finally, we provide results focused on addressing our main experimental questions.

Questions. These experiments seek to answer four related questions:

1. *What was the performance of the CDI model prospectively? And how does it compare to the expected retrospective performance?* (**Section 3.5.3**, **Figures 3.3**, **3.4**, and **3.5**)

2. *What is the prospective performance gap for the CDI model? What portion of the prospective performance gap is attributable to infrastructure shift? What portion of the prospective performance gap is attributable to temporal shift?* (**Section 3.5.4**, **Figure 3.6**)

3. *What are the contributing factors to the infrastructure shift of the CDI model?* (**Section 3.5.5**, **Figures 3.7**, and **Figures 3.8**)

4. *What are the contributing factors to the temporal shift of the CDI model?* (**Section 3.5.6**, **Table B.1**)

3.5.1 Data & Experimental Setup

We evaluated model performance over time, comparing the same model applied to i) retrospective data from 2019-2020, $\mathcal{D}_{ret}$, and ii) prospective data from 2020-2021, $\mathcal{D}_{pro}$. We analyzed the prospective performance gap that arises between these two datasets. We measured the gap in terms of the AUROC because optimizing for discriminative performance was the primary goal of prior work. However, calibration is known to be sensitive to temporal changes. [123, 140, 141] We also measured the prospective performance gap in the Brier score. [142, 143] However, since one aims to minimize the Brier score and the prospective performance gap assumes the goal is to maximize the performance measure, we take the negative of the Brier score when computing the gap. Confidence intervals for the prospective performance gap values were calculated using an empirical bootstrap where the samples ($1,000$ replications) were independently drawn for each data distribution.

When comparing model performance on $\mathcal{D}_{ret}$ to model performance on $\mathcal{D}_{pro}$, there are two unique sources of potential differences: the shift in time period from '19-'20 to '20-'21 (*i.e.*, temporal shift) and the change in pipelines (*i.e.*, infrastructure shift). Thus, to control for temporal shift and estimate the effect of infrastructure shift on the prospective performance gap; we used the retrospective pipeline to query data for the prospective validation time period, generating $\mathcal{D}'_{ret}$. We estimated Δ_p^{infra} and Δ_p^{time} using the datasets $\mathcal{D}_{ret}$, $\mathcal{D}'_{ret}$, and $\mathcal{D}_{pro}$, .

3.5.1.1 Study Cohort

The University of Michigan Institutional Review Board approved this retrospective and prospective cohort study. Our study population included all adult hospital patient encounters (*i.e.*, inpatient admissions) from January 2013 through June 2021 to University of Michigan Health. University of Michigan Health has over 1,000 beds and is the tertiary care academic health center associated with the University of Michigan. Because we were interested in primary, non-recurrent, hospital-associated CDI, we excluded encounters with a length of stay less than three calendar days and individuals who tested positive in the first two calendar days of the encounter or in the proceeding 14 days prior to the hospital encounter. [144]

3.5.1.2 Prediction Task

The task was formulated as a binary classification task where a patient encounter was labeled 1 if the patient tested positive for CDI during the encounter and 0 otherwise. The diagnosis of CDI was identified using a tiered approach, reflecting the institution's *Clostridioides difficile* (*C. difficile*) testing protocol when clinicians obtained stool samples for *C. difficile* based on clinical suspicion of active disease. First, samples were tested using a combined glutamate dehydrogenase antigen enzyme immunoassay and toxin A/B EIA (C. Diff Quik Chek Complete, Alere, Kansas City, MO). No further testing was needed if the results were concordant. If discordant, a secondary polymerase chain reaction (PCR) for the presence of toxin B gene (GeneOhm Cdiff Assay, BD, Franklin Lakes, NJ) was used to determine the outcome. That is, if positive by PCR, the encounter was considered a CDI positive case. We make predictions daily, intending to identify high-risk patients as early as possible during an encounter and prior to their diagnosis.

3.5.1.3 Model Development

Training Data. Our training cohort included patient admissions between 2013-2017 who met our inclusion criteria. When applying our inclusion criteria, we relied on patient class codes to identify hospitalized patient encounters. For each patient admission included in the training data, we extracted a binary classification label and information pertaining to a patient's demographics, medical history, laboratory results, locations, vitals, and medications. Once retrospective data were extracted, we transformed the data into d-dimensional binary feature vectors representing each day of a patient's admission (*i.e.*, an encounter-day). Features were transformed into a binary representation. Categorical features were transformed using one-hot encoding. Real-valued (numerical) features were split into quintiles and one-hot encoded. This is described in further detail in the feature preprocessing section of Oh et al. [110] and **Appendix Section B.1**.

Training Details. We employed a previously described and validated modeling approach. [56, 110] This validation was conducted at multiple institutions. In brief, we used a logistic regression model that uses a multitask transformation of the inputs to learn time-varying parameters. [109] The multitask regularized logistic regression model seeks to learn an encounter level label (*i.e.*, if the patient is ever diagnosed over their entire encounter). It does so by minimizing the cross-entropy loss at the encounter-day level. We subsampled encounter-days to reduce bias towards patient encounters with longer lengths of stay. This was done by randomly selecting 3 encounter-days per encounter (our inclusion criteria dictate that all encounters will have a length of stay of at least 3 days). This ensured that all encounters were represented by an equivalent number of encounter-days. Cross-validation folds were determined by year to partially account for dataset

shift. Hyperparameters were selected based on cross-validation across years in the training data optimizing for the AUROC. This approach is described in detail by Oh et al. [110] and Wiens et al. [56].

3.5.1.4 Model Validation

The model takes as input data describing each day of a patient's hospital encounter, extracted through either the retrospective or near-real-time prospective pipeline described above (*e.g.*, laboratory results, medications, procedures, vital sign measurements, and patient demographics) and maps these data to an estimate of the patient's daily risk of CDI. This estimate is updated throughout the patient's hospital encounter to help clinicians identify patients at risk of developing CDI over the remainder of the patient's hospital encounter.

Retrospective Validation. We validated the model on data on patient hospital encounters from the study cohort from 2018-2020. We extracted these data using the retrospective pipeline and identified hospitalized patients using patient class codes as we did in the training data (see above for inclusion criteria). In our primary analyses, we focus on performance during the more recent year, *i.e.*, '19-'20. For completeness, results from '18-'19 are provided in **Appendix Section B.4**.

We measured the AUROC and the sensitivity, specificity, and positive predictive value when selecting a decision threshold based on the 95^{th} percentile from '18-'19. In addition, we computed the Brier score by comparing the max probability of the outcome (*i.e.*, risk score) during a patient's visit with their actual outcome. We calculated empirical 95% confidence intervals on each test set using 1,000 bootstrap samples.

Prospective Validation. We applied the model prospectively to all hospital encounters from July 10th, 2020, to June 30th, 2021, estimating the daily risk of CDI for all patients who met our inclusion criteria. We relied on the hospital census tables instead of the patient class codes to identify our study population in real-time. The hospital census table tracked all hospitalized patients in real-time and enabled reliable identification of patients who were in the hospital for three calendar days or more.

We compared retrospective performance in '19-'20 to prospective performance in '20-'21. We evaluated model performance in terms of discrimination and calibration using the same metrics described above. In addition, to account for seasonal fluctuations in CDI rates [145], we further compared AUROC performance on a month-by-month basis. We compared monthly retrospective performance in '19-'20 to monthly prospective performance in '20-'21. Although encounters may span across multiple months, encounters were grouped into month-years based on the date of admission. Finally, given the large shift in care processes resulting from the onset of the COVID-19

pandemic, we conducted a separate follow-up analysis in which we compared model performance before and following March 2020 (**Appendix Section B.3**).

3.5.2 Study Cohort Characteristics

We now describe our study cohort. Our training cohort included 175,934 hospital encounters, in which 1,589 (0.9%) developed hospital-associated CDI. Feature extraction and processing resulted in 8,070 binary features. Our '19-'20 retrospective validation set ($\mathcal{D}_{ret}$) consisted of 25,341 hospital encounters, in which 157 (0.6%) met the CDI outcome. Prospectively in '20-'21, we identified 26,864 hospital encounters, in which 188 (0.7%) met the CDI outcome ($\mathcal{D}_{pro}$). Study population characteristics for both validation cohorts are reported in **Table 3.1**. During the prospective validation of the model, the prospective pipeline failed to run due to circumstances beyond the study team's control in 10 out of the 356 days. Specifically, from mid-December to February of 2021, an ADT data-feed issue led to a lag in processing some of the prospective data. Risk scores were not generated on days when the model failed to run.

Table 3.1: Yearly Cohort Characteristics. Retrospective and prospective cohorts from '19-'20, and '20-'21 each span from July 10th to June 30th of the following year. The cohorts have similar characteristics across years. For median values, we also present the interquartile range (IQR).

	'19-'20 ($\mathcal{D}_{ret}$) n=25,341	'20-'21 ($\mathcal{D}_{pro}$) n=26,864
Median Age (IQR)	59 (41, 70)	60 (42, 71)
Female (%)	51%	51%
Median Length of Stay (IQR)	5 (4, 9)	5 (4, 9)
History of CDI in the past year (%)	1.5%	1.4%
Incidence Rate of CDI (%)	0.6%	0.7%

3.5.3 Validation Results

We first investigated the performance of the model on the different data sources. We sought to answer the questions: *What was the performance of the CDI model prospectively? And how does it compare to the expected retrospective performance?* Using the experimental setup described above, we calculate the model performance in terms of both AUROC and the Brier score.

Applied to the '19-'20 and '20-'21 validation cohorts, the model achieved an AUROC of 0.778 (95% CI: 0.744, 0.815) and 0.767 (95% CI: 0.737, 0.801) and a positive predictive value of 0.036 and 0.026, respectively (**Figure 3.3**). Model calibration was fair across both '19-'20 and '20-'21

datasets, Brier scores: 0.163 (95% CI: 0.161, 0.165) and 0.189 (95% CI: 0.186, 0.191), respectively. Additionally, we break this analysis down by month for the AUROC. Monthly, prospective performance during '20-'21 did not differ significantly from the retrospective performance during '19-'20, except in March and May (**Figure 3.5**).

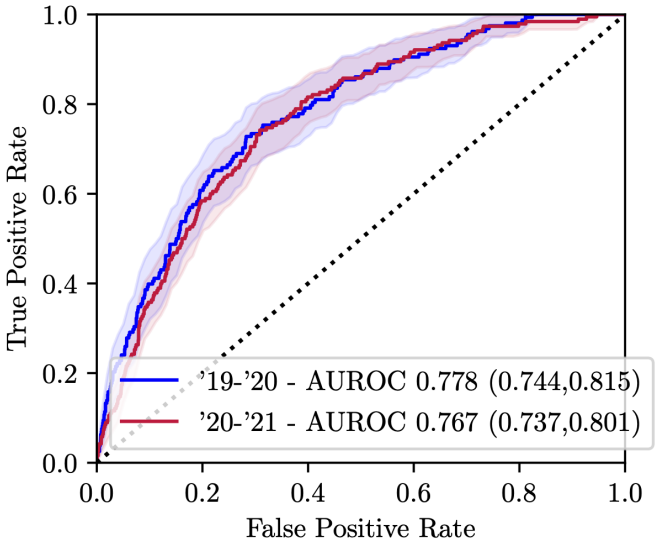


Figure 3.3: Risk Prediction Model Performance on '19-'20, and '20-'21 Validation Datasets. Compared with the model's retrospective validation period ('19-'20) performance, the model demonstrated slightly worse discriminative performance during its prospective validation period ('20-'21).

We observed that the prospective discriminative performance (AUROC) was slightly less than what would have been expected by the evaluation on retrospective data. This difference is not statistically significant. In calibration performance (Brier score), we observed that the model displayed worse performance on the prospective dataset than expected by retrospective evaluation. The calibration difference was statistically significant.

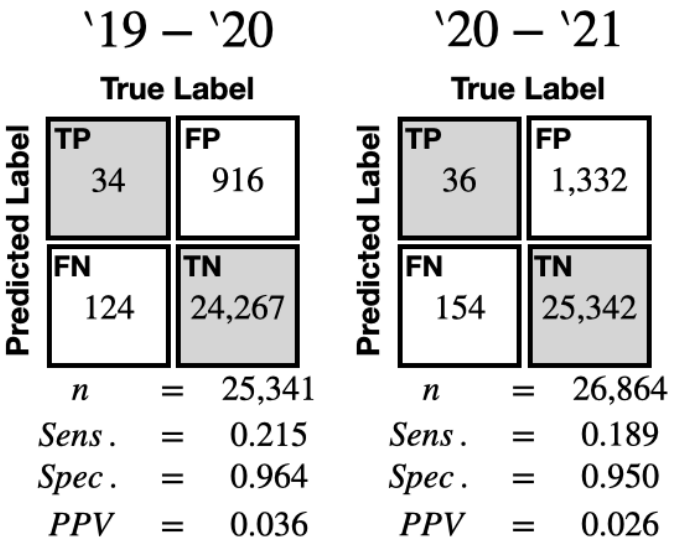


Figure 3.4: Risk prediction model performance on '19-'20, and '20-'21 validation datasets. Compared with the model's retrospective validation period ('19-'20) performance, the model demonstrated slightly worse sensitivity, specificity, and positive predictive value during its prospective validation period ('20-'21).

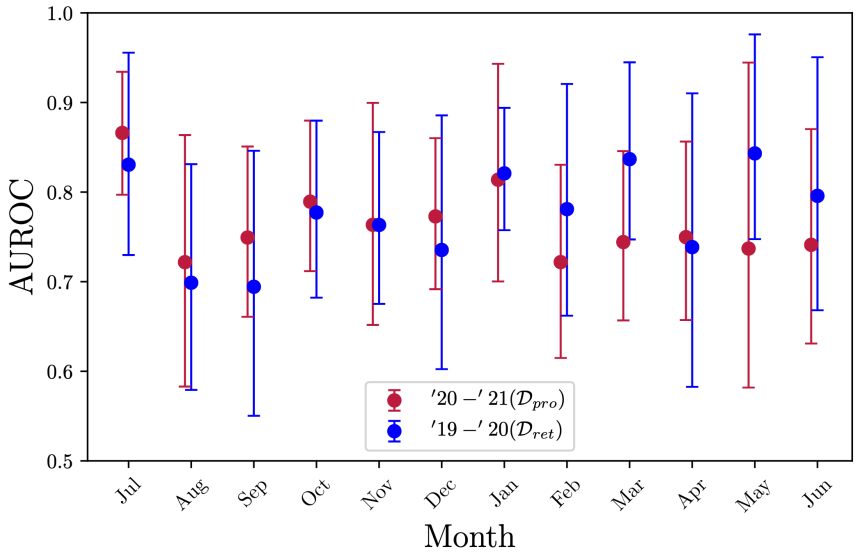


Figure 3.5: Monthly AUROC Performance. AUROC for '20-'21 prospective dataset and the '19-'20 retrospective dataset was broken down by month and bootstrap sampled $1,000$ times to generate empirical 95% confidence intervals. Performance fluctuates month by month, with the prospective pipeline generally outperforming or on par with retrospective performance with the exceptions of March and May.

Table 3.2: Model Performance Comparison. The prospective validation ran from July 10th, 2020 to June 30th, 2021 ('20-'21) and yielded dataset, $\mathcal{D}_{pro}$, and performance results. The '20-'21 retrospective dataset, $\mathcal{D}'_{ret}$, uses the retrospective pipeline to pull the same population observed in $\mathcal{D}_{pro}$. The retrospective '19-'20 retrospective dataset pulled data from July 10th, 2019 to June 30th, 2020 to have an equivalent annual comparison. We see a positive AUROC prospective performance gap and a negative Brier Score prospective performance gap indicating degraded prospective performance.

	'19-'20 Retrospective ($\mathcal{D}_{ret}$) n=25,341	'20-'21 Retrospective ($\mathcal{D}'_{ret}$) n=26,864	'20-'21 Prospective ($\mathcal{D}_{pro}$) n=26,864
AUROC (95% CI:)	0.778 (0.744, 0.815)	0.783 (0.755, 0.815)	0.767 (0.737, 0.801)
Brier Score (95% CI:)	0.163 (0.161, 0.165)	0.186 (0.184, 0.188)	0.189 (0.186, 0.191)

3.5.4 Prospective Performance Gap

We now turn our attention to the primary set of questions for this chapter: *What is the prospective performance gap for the CDI model? What portion of the prospective performance gap is attributable to infrastructure shift? What portion of the prospective performance gap is attributable to temporal shift?*

Overall, the prospective performance gap between $\mathcal{D}_{ret}$ in '19-'20 and $\mathcal{D}_{pro}$ in '20-'21 was $\Delta_{AUROC} = 0.011$ (95% CI: $-0.033, 0.056$) and $\Delta_{Brier} = 0.025$[2] (95% CI: $0.016, 0.110$). Applied to the re-extracted retrospective '20-'21 cohort ($\mathcal{D}'_{ret}$) the model achieved higher discriminative and calibration performance, AUROC=0.783 (95% CI: $0.755, 0.815$) and Brier score=0.186 (95% CI: 0.184, 0.188). Thus, according to **Equations 3.1, 3.2, and 3.3**, the prospective performance gap breaks down as follows:

$$
\begin{aligned}
\Delta_{AUROC} &= \text{AUROC}(\mathcal{D}_{ret}) - \text{AUROC}(\mathcal{D}_{pro}) &&= 0.011 \quad (95\%\ \text{CI: } -0.033,\ 0.056)\\
\Delta^{infra}_{AUROC} &= \text{AUROC}(\mathcal{D}'_{ret}) - \text{AUROC}(\mathcal{D}_{pro}) &&= 0.016 \quad (95\%\ \text{CI: } -0.022,\ 0.058)\\
\Delta^{time}_{AUROC} &= \Delta_{AUROC} - \Delta^{infra}_{AUROC} &&= -0.005 \quad (95\%\ \text{CI: } -0.051,\ 0.036)
\end{aligned}
$$

$$
\begin{aligned}
\Delta_{Brier} &= -(\text{Brier}(\mathcal{D}_{ret}) - \text{Brier}(\mathcal{D}_{pro})) &&= 0.025 \quad (95\%\ \text{CI: } 0.016,\ 0.110)\\
\Delta^{infra}_{Brier} &= -(\text{Brier}(\mathcal{D}'_{ret}) - \text{Brier}(\mathcal{D}_{pro})) &&= 0.002 \quad (95\%\ \text{CI: } -0.021,\ 0.064)\\
\Delta^{time}_{Brier} &= \Delta_{Brier} - \Delta^{infra}_{Brier} &&= 0.023 \quad (95\%\ \text{CI: } -0.003,\ 0.084)
\end{aligned}
$$

Figure 3.6 visualizes the breakdown of the AUROC (Δ_{AUROC}) prospective performance gap into $\Delta^{time}_{\text{AUROC}}$ and $\Delta^{infra}_{\text{AUROC}}$.

Regarding discriminative performance (AUROC), the differences in infrastructure pipelines be-

[2]Clarification: Reader may wonder why this isn't 0.026, this is simply due to rounding.

tween retrospective and prospective analyses had a larger impact on the prospective performance gap than temporal shift. However, the converse is true for calibration (Brier score) gap, where the shift from '19-'20 to '20-'21 had a greater impact on calibration performance. We note that only the prospective performance gap in terms of calibration is significant. All the discriminative prospective performance gap as well as all of the infrastructure and temporal shift have confidence intervals that overlap with 0.

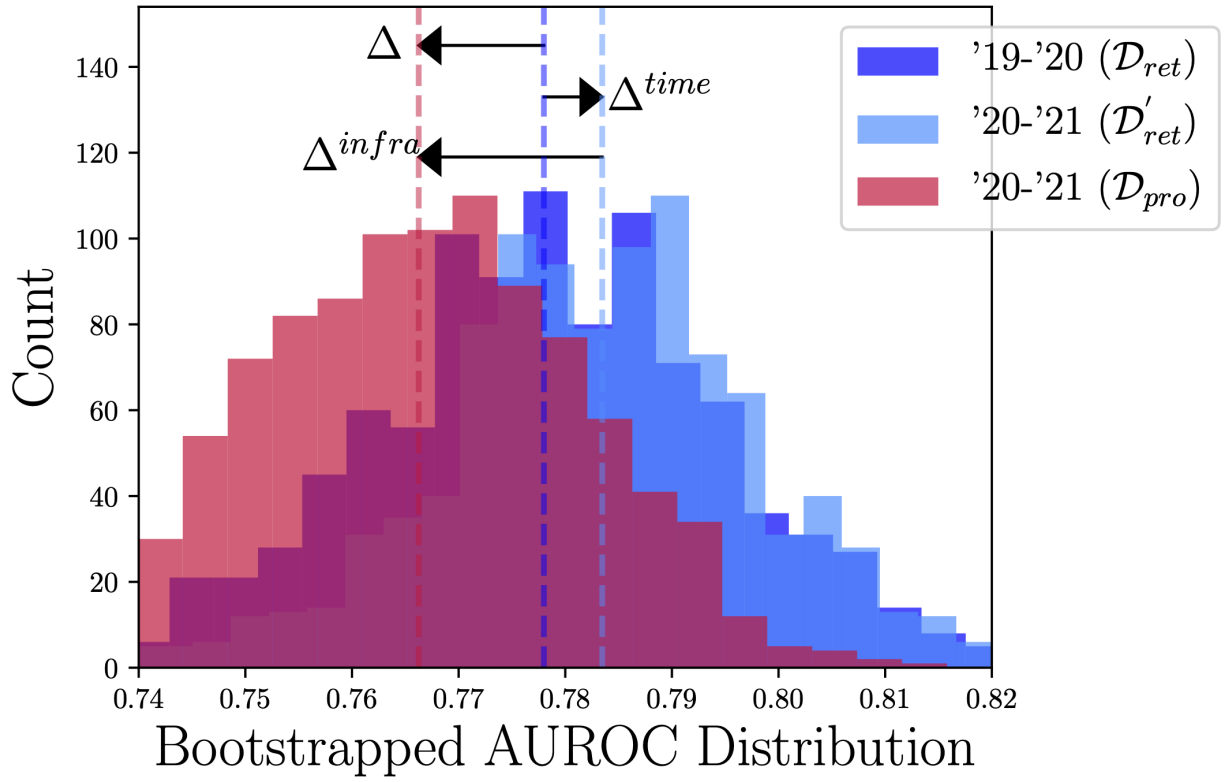


Figure 3.6: Relationship between prospective risk scores and retrospective risk scoring. The bootstrapped distribution of the AUROCs of the model applied to three different datasets are shown along with the prospective performance gap, Δ_{AUROC}, and its components, $\Delta^{infra}_{\text{AUROC}}$, and $\Delta^{time}_{\text{AUROC}}$. The overall gap is positive demonstrating discriminative performance degradation. This degradation is primarily due to the infrastructure shift since $\Delta_{infra} > \Delta_{time}$.

The infrastructure performance gaps indicate that the data extraction and processing pipeline differences led to a small (though not statistically significant) decrease in performance. When we compared the risk scores output by the model when applied to the retrospective versus prospective pipeline for every encounter in our '20-'21 cohort, we measured a correlation of 0.9 (**Figure 3.7**). 46 (0.2%) encounters had extreme score differences (greater than 0.5, denoted by the bounding dashed lines in the plot). 41 of these 46 encounters had a large number of days (more than 7 days for nearly all encounters) during which the prospective pipeline failed to run.

3.5.5 Sources of Infrastructure Shift Results

We now conduct a series of analyses to determine: *contributing factors to the infrastructure shift of the CDI model*.

We compared the risk scores output by the retrospective versus prospective pipeline for every encounter observed prospectively. These correspond to $\mathcal{D}'_{ret}$ and $\mathcal{D}_{pro}$, which share encounter-days; thus, the model's output for both datasets can be compared directly. Score pairs representing the maximum score were found for each encounter using the prospective and retrospective pipeline. These score pairs were graphed as a scatter plot and then were analyzed for their concordance in terms of Pearson's correlation coefficient and the slope of the best-fit line. Extremely discordant prospective and retrospective score pairs were identified by selecting points far away from the best fit line (*i.e.*, score pairs with a difference ≥ 0.5).

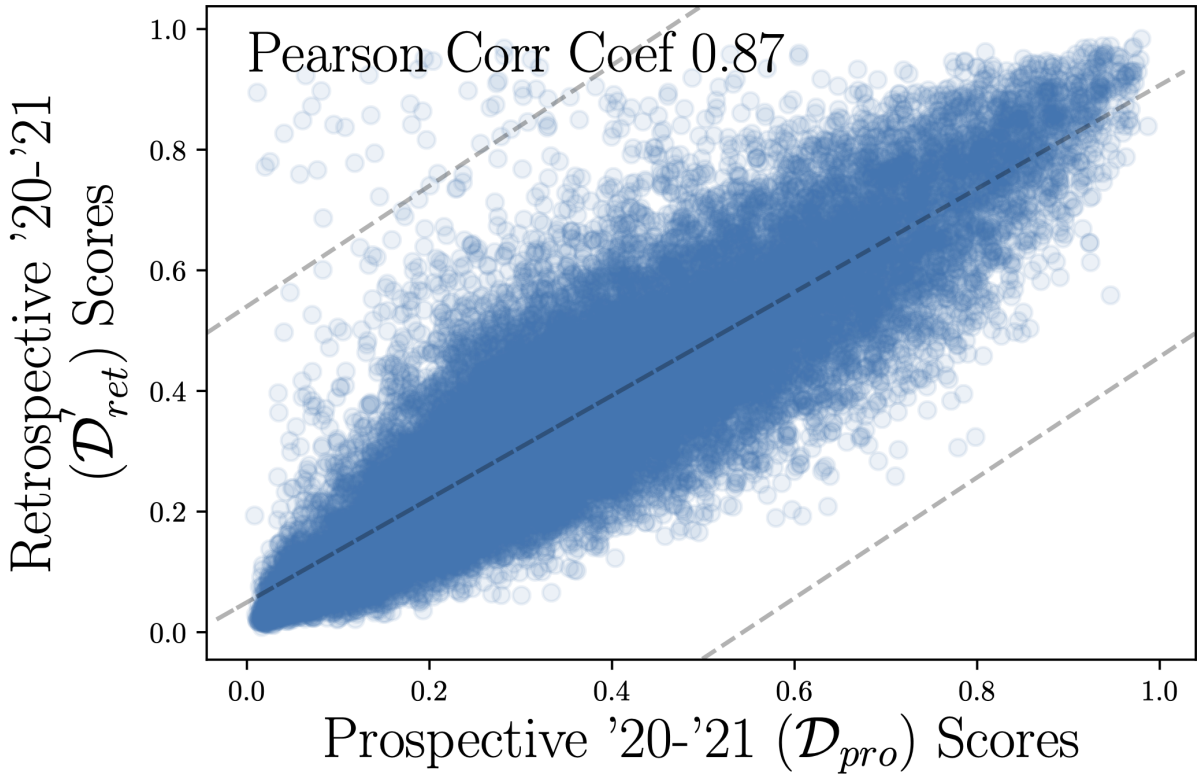


Figure 3.7: Infrastructure Performance Gap Scatter Plot. Risk scores generated by the '20-'21 prospective pipeline vs. '20-'21 retrospective pipeline are shown. Although highly correlated, some of the prospective and retrospective risk scores noticeably differ.

To understand factors that could potentially be addressed with modifications to infrastructure, we compared the pair of feature vectors present for each instance (a patient hospitalization encounter-day) in $\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$ by computing differences in feature inputs between the two

datasets. The difference in the two data pipelines ($\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$) was quantitatively assessed for every feature, at the encounter-day level. Since our model utilized a binary feature space, we deemed features discrepant at the encounter-day level if their prospective and retrospective values were not exactly equivalent. This can be extended to real-valued (numerical) features through either exact equivalency or by using ranges. To assess the impact of these features, we stratified features by the absolute value of their model coefficients and the proportion of discrepancies.

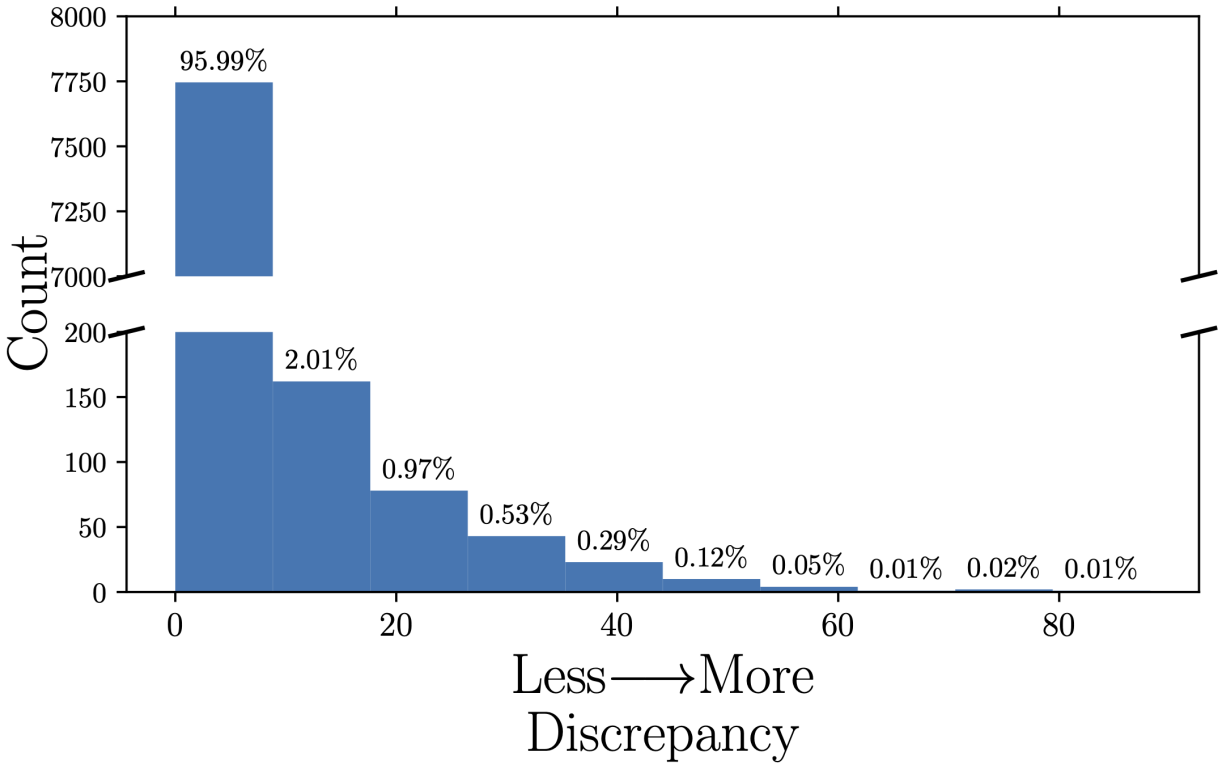


Figure 3.8: Infrastructure Performance Gap Analysis Feature Distribution. The distribution of features is based on how discrepant the features are (*i.e.*, percent of instances where a feature is discrepant between retrospective and prospective '20-'21). We see that although most features have low levels of discrepancy, there exists a subset of features whose values can vary greatly from the prospective to the retrospective pipeline.

Comparing the input features, we found that $6,178$ (77%) of the $8,070$ features had at least one instance (*i.e.*, encounter day) in which that feature differed across the two pipelines (**Figure 3.8**). However, only $1,612$ features (20%) differed in more than 1% of instances. However, not all features are equally important. To measure the actual impact of the discrepancies on model performance, we must look at the feature swap analysis.

We conducted a feature swap analysis on the feature groups defined in **Appendix Section B.1** using AUROC as the performance measure. Due to computational complexity, this analysis was

only conducted at the mid-point of our study and, as such, only uses data from July 10th to December 21st for both $\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$.

Applied to data from the first half of the prospective study period, the model achieved an AUROC performance of 0.769 on $\mathcal{D}_{pro}$. The AUROC after each feature group swap is displayed in **Table 3.3**. Hx: Medications, Idx: Medications, and Idx: In-Hospital Locations were the feature groups with the largest positive swap difference AUROC, corresponding to improved model performance when given feature information from the retrospective pipeline. In the case of Hx: Medications, one would think these features would be consistent prospectively and retrospectively. However, we use both retrospective and prospective pipelines to calculate prospective values. To obtain 90-day patient histories, we augment retrospective tables with prospective tables to fill the gap between when the data is logged in the EHR and when the data appears within RDW's tables. In addition, to identify which previous admissions were inpatient admissions, we use patient class codes, which are dynamic similar to laboratory results and medications. In addition to these more subtle changes, data that may be considered 'static' (*e.g.*, where a patient lives or BMI) is liable to change throughout a patient encounter as information is collected and updated by the clinical care team. The complete feature swap analysis is displayed in **Supplemental Table B.2**.

Table 3.3: Infrastructure Performance Gap Analysis - Feature Swap Performance. By swapping column values corresponding to feature groups between $\mathbf{X}_{pro}$ and $\mathbf{X}'_{ret}$ we were able to quantify the performance impact of differences in the infrastructure related to each feature group. Note, this analysis was conducted at an interim time-point of our study, as such only uses data from July 10th to December 21st for both $\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$. In addition to the feature group name, and the number of features in each feature group we display the AUROC on $\mathcal{D}_{pro}$ after the feature swap. Originally, we observed an AUROC of 0.769 on $\mathcal{D}_{pro}$, the final column displays the difference between this value after the swap and the original 0.769. We restrict this table to only positive differences, that is feature swaps that improve AUROC, all feature swap values are displayed in **Supplemental Table B.2**, **Appendix Section B.2**. Hx: Medications, Idx: Medications, and Idx: In-Hospital Locations had the largest positive swap difference in terms of AUROC, corresponding to improved model performance when given feature information from the retrospective pipeline.

Feature Group	AUROC After Swap	Difference
Hx: Medications	0.787	0.018
Idx: Medications	0.774	0.005
Idx: In-Hospital Locations	0.772	0.003
Hx: Previous Encounters (Length of Stay)	0.770	0.001
Demographics: Body Mass Index	0.770	0.001
Demographics: County & State	0.770	0.001
Idx: Colonization Pressure	0.770	0.001

Descriptions of feature groups can be found in **Table B.1**.

3.5.6 Sources of Temporal Shift Results

We round out our analyses by examining: *the contributing factors to the temporal shift of the CDI model*.

We identified sources of temporal shift by comparing the distribution of features between $\mathcal{D}_{ret}$ and $\mathcal{D}'_{ret}$. Specifically, for each binary feature we conducted a Z-test with a Bonferroni correction to test the difference in the proportion of times that feature was 'turned on' in one time period versus the other, controlling for differences in infrastructure. *E.g.*, was a particular medication used more frequently in one time period? We report the number of significant differences within each feature group (see **Appendix Section B.1** for feature grouping).

Table 3.4: Significantly Different Features Between '19-'20 & '20-'21 Study Populations. By feature group, lists the number of features that are significantly different between the '19-'20 study population and the '20-'21 study population. Significance was determined using a Z-test of the difference in proportions with a Bonferroni correction. One day was randomly sampled from each hospital encounter so that all feature instances are independent. Note, data that may be considered 'obviously static', like the location a patient lives (*i.e.*, County & State) may be updated throughout an encounter, leading to discrepancies between prospective and retrospective data.

Feature Group	Number of Significantly Different Features	Total Number of Features
Demographics	0	124
Hx: History of CDI	0	2
Hx: Diagnoses	1	983
Idx: Vital Sign Measurements	1	17
Idx: Admission Details	5	22
Hx: Previous Encounters	5	10
Idx: Laboratory Results	6	508
Idx: Colonization Pressure	7	10
Hx: Medications	23	2,731
Idx: In-Hospital Locations	30	932
Idx: Medications	38	2,731

Descriptions of feature groups can be found in **Table B.1**.

Comparing the feature distributions between and $\mathcal{D}_{ret}$ '19-'20 and $\mathcal{D}'_{ret}$ '20-'21 we noted significant differences in 116 (1.44%) of the features. Features pertaining to medications and in-hospital locations, had the largest fraction of differences. However, these categories also had a large number of overall features **Table 3.4**. Colonization pressure, patient history pertaining to number of previous encounters, and admission details had the greatest differences in fractions of differences within each category.

3.6 Iterative Debugging

After the initial publication of this work, we continued to refine the prospective infrastructure in preparation for a feasibility study. We discovered three additional bugs in the prospective data pipeline that contributed to differences between $\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$. These were:

1. Persistence issues with the logging of prospective scores,
2. Truncation of historical data lists, and
3. Inconsistent processing of diagnosis codes.

We now discuss these issues in further detail.

Persistence issues with the logging of prospective scores. The CDI model produces scores daily for all hospitalized patients. As mentioned above, we used near-real-time census tables to determine the current hospitalized patient population. Additionally, the daily scores produced by the model are not used directly. A cumulative mean was applied to the daily scores to smooth out any large daily changes. For evaluation, we calculated performance at the encounter level by taking the maximum score observed throughout the encounter. To facilitate the prospectively smoothing calculation, we stored a log with all the encounters' current and historical scores. For memory efficiency, we removed all encounters from the log when they were no longer observed in the census table.

We eventually determined that occasionally a patient encounter will be missing from the census table during the middle of their encounter. For example, a patient hospitalized from March 1st till the 21st might have been missing from the census table when we queried it on the 10th. This introduced an infrastructure discrepancy in two ways. First, was that that example encounter would be missing a prospective score on March 10th. Second, the cumulative mean would be reset starting March 11th for the prospective scores. It would be practically impossible for prospective and retrospective scores to match from March 11th to the 21st.

We are unsure of why this census table "encounter dropping" occurs. However, we posit that this might happen when a patient leaves their room for a procedure or test as these events are captured as separate "children" encounters of the "parent" hospital encounter. Given that the census table depends on a vendor-developed web service, it was not in our purview to modify it. We resolved this issue by increasing the length of time before encounters were removed from the score log. We now wait a week from the last census observation before we remove an encounter from the score log.

Truncation of historical data lists. Both data pipelines extract data from the SQL based RDW. SQL queries are run parameters that ensure the results conform to some standard, *e.g.*, rows no longer than $1,024$ characters. We noticed a difference in the maximum column lengths by comparing the raw data from the prospective and retrospective query results. The prospective query had its columns limited to 256 characters. Although most columns used significantly fewer characters, we noticed that columns containing lists of information tended to be truncated. For example, this discrepancy explains the differences we observed in historical medications. We resolved this issue by modifying the prospective SQL query's parameters to match the retrospective query's parameters.

Inconsistent processing of diagnosis codes. We also traced the feature extraction process, examining the transformation of raw data from the retrospective and prospective queries. Through this tracing, we noticed that diagnoses were not handled consistently by the prospective version of the feature extraction pipeline. Retrospectively, we implemented a procedure to group the diagnoses by their top-level International Classification of Disease (ICD)-9 code (e.g., using only the digits preceding the decimal point). Interestingly this bug does not seem to have induced a significant infrastructure gap. This may be due to the sparse nature of diagnoses.

3.7 Discussion

In healthcare, risk stratification models are trained and validated using retrospective data that have undergone several transformations since being initially observed and recorded by the clinical care and operations teams. In contrast, during deployment, models are applied prospectively to data collected in near-real-time. Thus, relying on retrospective validation alone can result in an overestimation of how the model will perform in practice. In this paper, we sought to characterize the extent to which differences in how the data are extracted retrospectively versus prospectively contribute to a gap in model performance. We compared the performance of a patient risk stratification model when applied prospectively from July 2020-June 2021 to when it was applied retrospectively from July 2019-June 2020. Overall, the prospective performance gap was small. However, differences in infrastructure had a greater negative impact on discriminative performance than differences in patient populations and clinical workflows.

To date, much work has focused on addressing changes in model performance over time due to temporal shift. [123, 141, 146, 147] In contrast, we concentrated on gaps due to differences in infrastructure. We relied on data extracted from a research data warehouse for model development and retrospective validation. In contrast, we leveraged data extracted from a combination of custom web services and existing data sources for near real-time prospective application of the model. Prospectively, we had to shift to using the hospital census tables to identify our study cohort (*i.e.*,

who was in the hospital) in real-time, partly because inpatient classification is dynamic and can shift over time. But even after accounting for differences in population, differences in how and when the data were sourced continued to contribute to a gap in performance. Our analysis pointed to two sources of inconsistencies between the retrospective and prospective pipelines: i) inaccurate prospective infrastructure and ii) dynamic data entry.

The first cause can be mitigated by revisiting the near-real-time extraction, transformation, and load processes (*e.g.*, rebuilding prospective infrastructure to pull from different components of the EHR). For example, our analysis identified discrepancies in patient location codes between prospective and retrospective datasets. While the EHR passed the same location information to both pipelines, the two pipelines transformed and served this information in an inconsistent manner. Thus, we can rebuild the prospective infrastructure using the same processing code as the retrospective infrastructure. The second cause is more difficult to address. The EHR is inherently dynamic as it serves many operational purposes [148]. For example, medication start and end dates can change over time as the desired treatment plan changes, and laboratory result names can change as initial results lead to further testing. In addition, specific aspects of features, such as laboratory result abnormality flags, can populate after the actual test results are filed (up to a day in our systems).

To mitigate the impact of these differences on the prospective performance gap, one can update the model to rely less on such dynamic elements of the EHR. For example, in our project, we substituted medication orders for medication administration. Although order time is available earlier than administration, orders are frequently cancelled or updated after a physician initially orders them. In contrast, medication administration information is more stable across time. Our findings underscore the need to build pipelines representative of the data available at inference time. The closer the retrospective representation of data are to data observed prospectively, the smaller the potential prospective performance gap.

Beyond differences in infrastructure, it is reassuring that changes in patient populations and workflows between time periods (*i.e.*, temporal shift) did not increase the gap in discriminative performance. On a month-by-month basis, the only significant differences in performance were during March and May, otherwise, the model performed as well, if not better, prospectively. Interestingly, predicting which patients were most likely to acquire CDI during the current hospital visit was significantly easier in March 2020 compared to March 2021. This discrepancy is likely due to significant operational changes at University of Michigan Health due to the onset of the COVID-19 pandemic. Comparing the expected feature vectors in '19-'20 vs. '20-'21, we noted significant differences in locations and admission types, changes likely attributed to the COVID-19 pandemic. For example, new patient care units were created for patients hospitalized for COVID-19 [149], and patient volume to existing units and services decreased significantly. [150–152] Additionally,

colonization pressure depends on locations. As such, we expect this to change with the distribution of patients in locations changing. This drastic change in the patient population may also explain the other changes in feature groups. While these changes made the problem easier during prospective validation (**Appendix Section B.3**), in-line with previous work, the calibration performance of the model was negatively impacted by the temporal shift. [123, 140, 141]

Limitations. This study is not without limitations. Aside from the limitations associated with studying a single model at a single center, there is another nuanced limitation that pertains to the timing of data. The age (*i.e.*, time between data collection and use for this analysis) of the retrospective data varied in our analysis. Some validation data had been present in RDW for over two years, while other data were populated far more recently. Data collected in large retrospective databases are always subject to change, but the likelihood of changes decreases over time as updates from clinical care and billing workflows settle. As we use data closer to the present (July 2021), it is possible that the data may continue to change. Thus, if we were to revisit the analysis in the future, the infrastructure gap could further increase. However, most updates to the retrospective data occur within 30 days of discharge, and thus we expect the impact on our results to be limited.

A note on statistical significance. While the performance gap we studied here was not statistically significant, it could result in significant performance degradation if left unaddressed. With this analysis, we aim to identify and mitigate differences that are well within our control, enabling the resolution of prospective performance gaps. Many issues arise when implementing models, and it behooves model developers to minimize all the potential sources of model performance degradation they can control. Temporal shifts that occur during implementation may be impossible to predict and mitigate in an *a priori* manner. Infrastructure shifts, by comparison, may be addressed by the efforts of model developers and system administrators. There exists an opportunity to eliminate any degradation caused by infrastructure shift. With the proper infrastructure, it should be possible to get the data presented by the prospective infrastructure to exactly match the data presented by the retrospective infrastructure.[3] In theory, this means that there would be no performance degradation due to infrastructure shift.

Rooting out infrastructure shift causes is of great practical concern in machine learning for healthcare. Predictive tasks are often difficult and minor improvements in model performance are often achieved through collection of additional data or with extensive model selection search procedures. Eliminating avoidable infrastructure shift performance degradation (even if small) may be a fruitful way to allocate the limited resources of model developers.

[3]If all data were accurately timestamped and immutable throughout all the infrastructure used then we would expect that there would be no way for infrastructure shift to occur. Alas, this does not reflect reality since entries are frequently updated retrospectively and are modified as they pass through IT systems.

Moreover, unaddressed infrastructure shifts present a "backdoor" for temporal shift vulnerabilities. Temporal shifts may be obscured or compounded by infrastructure shifts. For example, an infrastructure shift involving improper processing of diagnosis codes may become more problematic over time if the number of a patients with a particular important diagnosis code increases. Although isolating and mitigating the causes of infrastructure shift is an additional effort for model developers, it is likely worthwhile when implementing ML models for healthcare.

Conclusion. The prospective performance gap is due, in part, to the fact that we are trying to capture a moving target with a single snapshot. Existing EHR and associated database systems are primarily designed to support care operations. Therefore, they lack features to help develop and deploy real-time predictive models. EHR vendors are working to create tools to efficiently and effectively deploy ML models. [153] However, to the extent that we continue to develop models using data extracted from databases that are several steps removed from clinical operations, issues are likely to remain. While overwriting records and values may have little consequence for care operations, retrospective training is fraught with workflow issues. Mechanisms are needed to keep track of what the database looked like at every moment in time - à la Netflix's time machine. [154] However, in lieu of such solutions, thorough prospective validation and analysis can help bridge the gap, providing a more accurate evaluation of production model behavior and elucidating areas for improvement.

CHAPTER 4

Rank-Based Compatibility for Use in Updating Patient Risk Stratification Models

4.1 Introduction

As machine learning (ML) models become more commonplace in healthcare, there is a growing need to understand the impact of updating models in use by clinicians. Although model updating may lead to improved model performance, it may also affect clinician expectations, *i.e.*, how clinicians believe a model will perform given a set of patients to evaluate. Thus, updating can pose an issue when clinicians use models to augment their medical decision-making. [155] As such, it may not be sufficient to select updated models based on performance alone; when given multiple models with adequate discriminative performance model developers may want to choose the model that minimizes the disruption to clinical users. Thus, there is a need for effective tools to estimate how clinician expectations may influence the adoption of updated models without directly querying users. Fundamentally, we would like a way to answer this question: *if a user works with a model and then the model is updated, how different will the updated model's results be from the user's expected results*?

Compatibility measures seek to answer this question. Given an original model, a potential updated model, and an evaluation dataset, compatibility measures provide a sense of how much the clinician's mental model may be perturbed by switching to the updated model. Existing compatibility measures were primarily developed for supervised classification, with original and updated models being assessed in terms of the accuracy of their predicted labels. [31, 156] They can be modified for use in risk stratification settings by using decision thresholds to compare model categorization. However, this fails to capture essential differences between risk stratification models across decision thresholds or in environments where fixed decision thresholds cannot be employed. The existing compatibility measures are not directly comparable to discriminative performance measures (*e.g.*, area under the receiver operating characteristic curve (AUROC)) making model development trade-off decisions between compatibility and discriminative performance hard to

assess. Additionally, the existing compatibility measures may be sensitive to changes in model calibration that occur naturally over time. [157] Thus, there is a need for compatibility measures that do not depend on a threshold, especially in the context of evaluating updates to patient risk stratification models.

In light of this gap, we propose the first rank-based compatibility measure, which is based on existing approaches to measure concordance in ranking. Specifically, rank-based compatibility estimates the probability that both models will correctly rank a pair of discordantly labeled patients (a *patient-pair*) given the original model correctly ranked that patient-pair. Considering the ranking concordance between the output of two models, we can detect meaningful changes to risk stratification models when models are updated, which previously proposed measures fail to do. These measures can be compared directly with AUROC as both measure proportions of correctly ranked patient-pairs. Moreover, they may be more robust to calibration shifts, a commonly observed phenomenon in healthcare. [123, 140, 158] In sum, the new measure we propose is well suited for use in the evaluation of updates for models in healthcare as they are better able to detect changes in risk stratification models that may affect user expectations and subsequently affect trust in the updated models. We present a theoretical examination of this new measure showing its behavior in relation to the discriminative performance of original and updated models being considered. In addition to proposing and analyzing this new rank-based compatibility measure, we develop a related loss function that can be used to engineer model updates so that model developers can balance improvements in discriminative performance against compatibility. This work enables the evaluation and development of model updates that could lead to better clinician-model joint performance.

4.2 Contributions

We propose and analyze a new rank-based compatibility measure to fill in the gaps associated with existing compatibility measures that assume a single decision threshold. This new rank-based compatibility measure is designed for evaluating updates to risk stratification models and does not depend on setting a decision threshold. It may be used for updated model selection as an additional criterion focused on modeling user expectations or incorporated into model development.

We presented early versions of this work at the 2021 INFORMS Annual Meeting and the 2021 INFORMS Healthcare Meeting. We plan to submit the methods portion of this work to a refereed Conference. We plan to submit the case study and some of the more general findings to an archival journal.[1]

[1]The forthcoming works will be co-authored by Erkin Ötleș, Brian Denton, and Jenna Wiens. One of these pieces may also include authors representative of the MUSIC collaborative. EÖ was primarily responsible for all of the core contributions presented in this chapter.

The main contributions from this work are as follows:

- *To the best of our knowledge, we introduce the first rank-based compatibility measure based on the concordance of risk estimate pairs.*
- *We characterize the extent to which the new compatibility measure may vary over all potential model updates.* This helps to establish the relationship between the discriminative performance of the original and updated models and the new rank-based compatibility measure. In addition to providing a direct connection between model discrimination performance and rank-based compatibility, we also introduce several ancillary measures to examine the characteristics of risk stratification model updates. The ancillary measures also contextualize and compare the rank-based compatibility values produced for updates considered to serve as secondary criteria for model selection among models of similar discriminative performance.
- *We provide bounds on the rank-based compatibility, which provide insights about optimistic and pessimistic outcomes of potential updates.* Additionally, the bounds show that as the discriminative performance of the models increases, the lower bound of rank-based compatibility increases. We also show that rank-based compatibility exhibits a central tendency. Common model development approaches may provide many more updated models with rank-based compatibility values towards the center of the bounds. Thus, while some rank-based compatibility arises from maximizing the AUROC of the updated model, additional search procedures may be necessary to find a model with desired rank-based compatibility.
- *We introduce a custom loss function that incorporates ranking incompatibility which can be used to engineer model updates with improved rank-based compatibility characteristics.* We show that utilizing the incompatibility loss during updated model training results in higher rank-based compatibility on held-out data. This higher rank-based compatibility comes at a small cost in terms of discriminative performance.
- *Using MIMIC-III, we present empirical results that show the updated models with larger rank-based compatibility values can be generated using incompatibility loss.* In addition to examining the rank-based compatibility observed through standard model selection, we analyze the impact of incorporating incompatibility loss as an alternative model selection criterion. This experiment shows that candidate update models built using standard training procedures provide a limited range for rank-based compatibility, which can be overcome by using a new loss function that incorporates ranking incompatibility.
- *We present a real-world use using the rank-based compatibility measure to understand the potential impact of updating a risk stratification model currently used for predicting prostate cancer outcomes.*

We present several technical innovations: a new rank-based compatibility measure—the rank-based compatibility measure's relationships with discriminative performance and its concentrated distribution. And a differentiable incompatibility loss function to help engineer updates with desired rank-based compatibility characteristics. These technical innovations are aimed to aid model developers who wish to provide high-quality updates to risk stratification models in use by clinicians.

4.3 Problem Setup & Related Work

In this section, we provide background and setup for the problem of assessing risk stratification model updates in terms of user expectations. We start by defining notation, followed by a summary of the existing backwards trust compatibility measure.

In the context of learning patient risk stratification models, a patient i is represented by the tuple $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the feature vector and $y_i \in \{0, 1\}$ represents the binary label (*e.g.*, outcome). We are generally interested in risk stratification models, $f(\cdot)$, that output risk estimates, $\hat{p}_i \in [0, 1]$, which estimate $\Pr(y_i = 1|\mathbf{x}_i)$. These risk estimates can be converted to predicted labels, $\hat{y}_i = \mathbb{1}(\hat{p}_i > \tau)$, where τ is some model developer-defined decision threshold.

In this work, we seek to assess the impact on user expectations when updating from an original model, $f^o(\cdot)$, to an updated model, $f^u(\cdot)$. Note that the original and updated models are specific instantiations of the risk stratification models introduced above. They produce risk estimates denoted as $\hat{p}_i^o$ and $\hat{p}_i^u$, representing the $\Pr(y_i = 1\ |\mathbf{x}_i)$ estimated by the original and updated models. Given an original model in use, we would like to either directly train an updated model or select one from a set of candidate updates such that desired performance and compatibility characteristics are met. In this work, we are interested in evaluating moving from an original model currently in use to a candidate updated model. We refer to the combination of an original and updated model as a *model-pair*.

We will provide a background for discriminative and compatibility measures shortly; however, we will first give some additional definitions and notation. These definitions and notation can be used to describe the evaluation dataset and will aid our discussion of the existing literature and subsequent discussion. The original and candidate update risk stratification models will be evaluated on a held-out set of patients, denoted as I. This set of patients can be partitioned into two mutually exclusive subsets based on the label of the patient: 0-labeled patients, I^0, and 1-labeled patients, I^1. The size of these subsets of patients are denoted as n^0 and n^1, respectively, and their sum, n, is the cardinality of I.

We formalize the notion of a *patient-pair*, a pair of patients i and j that do not share the same label (*i.e.*, $i \in I^0$ and $j \in I^1$). The total number of patient-pairs, m, is the product $n^0 n^1$. We denote

the number of patient-pairs correctly ranked by the original and updated models as m^{o+} and m^{u+} respectively. Both m^{o+} and m^{u+} are integers taking on values between 0 and m inclusively.

Discriminative Performance

Discriminative performance measures a model's ability to separate patients with different labels. [159] AUROC is widely used to evaluate the performance of risk stratification models in healthcare as the class balance of a dataset does not unduly influence it. [160] AUROC is the probability of correctly ranking two patients with differing labels based on the risk estimates produced by the model. [161] The AUROC of a model, such as $f^o(\cdot)$, may be estimated by counting the number of patient-pairs ranked correctly by that model, m^{o+}, and then normalizing that value by the total number of patient-pairs. We define the AUROC metric for an evaluation dataset based on the above notation:

$$AUROC(f^o) = \frac{\sum_{i \in I^0} \sum_{j \in I^1} \mathbb{1}(\hat{p}_i^o < \hat{p}_j^o)}{m} = \frac{m^{o+}}{m} \tag{4.1}$$

The AUROC ranges between 0 and 1, naturally matching its probabilistic interpretation. Values corresponding to 0.5 correspond to essentially random ordering of patient-pairs.

The AUROC is related to the concordance index (c-index), which can be used to evaluate predictions against non-binary ordinal outcomes (*e.g.*, binned survival times). The c-index estimates the probability that a model will correctly rank a patient-pair in terms of a prognostic score (*e.g.*, estimated survival time) compared to actual survival time. [160] Although the c-index can be interpreted as a probability of correct ranking, it can also be thought of as a correlation measure and is directly related to Kendall-Goodman-Kruskal rank correlations. [162] In the non-binary ordinal case, the c-index is related to Kendall's-τ rank correlation; in the binary case the c-index is equal to the AUROC and related to the Wilcoxon-Mann-Whitney U statistic. [159]

Backwards Trust Compatibility

Researchers have recently proposed compatibility measures in the context of correct labels produced by classification models. [31, 33, 156] Here, we review the primary compatibility measure described in the literature: *backwards trust compatibility* ($\mathcal{C}^{\mathrm{BT}}$), which is defined based on the agreement between the true label and the predicted labels produced by the original and updated model. It measures the label agreement between the two models by counting the number of patients both labeled correctly and normalizing by the number of patients the original model correctly labeled. We now provide a definition of $\mathcal{C}^{\mathrm{BT}}$ that aligns with our notation:

$$\mathcal{C}^{\mathrm{BT}}(f^o, f^u) = \frac{\sum_{i \in I} \mathbb{1}(y_i = \hat{y}_i^o) \cdot \mathbb{1}(y_i = \hat{y}_i^u)}{\sum_{i \in I} \mathbb{1}(y_i = \hat{y}_i^o)} \tag{4.2}$$

Note that $\mathcal{C}^{\mathrm{BT}}$ calculation depends on the using evaluation set of patients, I, and that the $\mathcal{C}^{\mathrm{BT}}$ value ranges between 0 and 1. When the updated model fails to correctly label all of the patients labeled correctly by the original, then $\mathcal{C}^{\mathrm{BT}} = 0$. $\mathcal{C}^{\mathrm{BT}}$ is maximized to 1 when the updated model correctly labels all of the patients the original model correctly labeled. $\mathcal{C}^{\mathrm{BT}}$ is not symmetric, as $\mathcal{C}^{\mathrm{BT}}(f^o, f^u)$ does not necessarily equal $\mathcal{C}^{\mathrm{BT}}(f^u, f^o)$.

Using $\mathcal{C}^{\mathrm{BT}}$ provides model developers a measure of the degree to which clinicians might have their expectations met when an updated model is introduced into their workflows. However, this evaluation is only in terms of the correct labeling of patients, which presents problems when applied to risk stratification models designed to rank patients.

In the context of patient risk stratification models that output a continuous risk score or a ranking of patients the $\mathcal{C}^{\mathrm{BT}}$ requires thresholding predictions. However, many settings in healthcare do not use a strict decision threshold. Instead, risk stratification models may produce continuous risk estimates. Moreover, using a decision threshold may be driven by clinician opinions, resource constraints or other factors that may change over time or with respect to the state of the healthcare system. [163, 164] So evaluating a single decision threshold would provide limited utility.

Additionally, there is no direct relationship between $\mathcal{C}^{\mathrm{BT}}$ and the discriminative performance of the two models. Related work on the relationship between accuracy and discrimination by Cortes and Mohri [165] suggests that while there may be a positive correlation between the mean $\mathcal{C}^{\mathrm{BT}}$ and AUROC, the values for discriminative performance can be subject to significant variation. Altogether, these factors suggest that there is a need for a compatibility measure that functions without setting decision thresholds.

4.4 Methods

We first introduce the concept of our proposed rank-based compatibility measure, $\mathcal{C}^{\mathrm{R}}$, that does not depend on setting a decision threshold for the risk prediction models being considered. Inspired by the AUROC measure, it evaluates patient-pairs in terms of their correct ranking concordance between the original and updated model's risk estimates. Next, we examine the relationship between the discriminative performance of the two models. This discussion will naturally lead to the formal definition of $\mathcal{C}^{\mathrm{R}}$, followed by an enumeration of its properties. After discussing $\mathcal{C}^{\mathrm{R}}$'s properties, we will focus on developing methods to update models that emphasize high levels of $\mathcal{C}^{\mathrm{R}}$.

Table 4.1: Relationship between original and updated model discriminative performance, proportion of patient-pairs (POP) and count variables.

	Original Model Ranks Correctly	**Original Model Ranks Incorrectly**	
Updated Model Ranks Correctly	$\phi^{++} = \frac{m^{++}}{m}$	$\phi^{-+} = \frac{m^{-+}}{m}$	$AUROC(f^u) = \frac{m^{u+}}{m}$
Updated Model Ranks Incorrectly	$\phi^{+-} = \frac{m^{+-}}{m}$	$\phi^{--} = \frac{m^{--}}{m}$	$1 - AUROC(f^u)$
	$AUROC(f^o) = \frac{m^{o+}}{m}$	$1 - AUROC(f^o)$	1

4.4.1 Original & Updated Model Discriminative Performance

The relationship between the rank-based compatibility, $\mathcal{C}^{\mathrm{R}}$, and the traditional model performance measure of AUROC provides bounds for rank-based compatibility. This relationship arises because $\mathcal{C}^{\mathrm{R}}$ and AUROC both involve counting correct patient-pairs rankings (*i.e.*, if a patient-pair's ordering agrees with the ordering of the labels). To clarify these relationships, we introduce several ancillary rank-based compatibility variables. Four proportion of patient-pairs (POP) variables measure how the two models rank (correctly vs. incorrectly) patient-pairs. The POP variables follow the convention of ϕ^{ab}, where a represent how the original model ranks patient-pairs correctly $(+)$ vs. incorrectly $(-)$, and b represents the same information for the updated model. For example, the POP variable for patient-pairs *correctly* ordered by both models is denoted by ϕ^{++}, and the proportion of patient-pairs *incorrectly* ordered by both models is ϕ^{--}.

There are four POP variables in total, $\phi^{++}, \phi^{+-}, \phi^{-+}$, and ϕ^{--}. They all sum to 1 and have relationships with the AUROC of both models. For a given dataset, the observed AUROC of the original model should be equal to the sum of the two POP variables where the original model ranks correct ($a = +$), with $AUROC(f^o) = \phi^{++} + \phi^{+-}$. The AUROC of the updated model is equal to the sum of the two POP variables representing that the updated model ranks correctly ($b = +$) with $AUROC(f^u) = \phi^{++} + \phi^{-+}$. Each POP variables corresponds to a patient-pair count variable: m^{++}, m^{+-}, m^{-+}, and m^{--}. These variables follow the same convention as the POP variables. They are the un-normalized counts of the number of patient-pairs that each model ranked correctly or incorrectly. For example, m^{++} represents the number of patient-pairs that both models ranked correctly. The relationships between the POP variables, the count variables, and discriminative performances can be expressed in a tabular manner, as depicted in **Table 4.1**.

The count variables will be used to construct relationships between discriminative performance and rank-based compatibility. Additionally, the POP variables can be used as ancillary rank-based measures to understand how clinical users might be impacted by changing from the original to the

updated model. This will be discussed in detail in **Section 4.6**. Now focus on the definition of our proposed rank-based compatibility measure.

4.4.2 Rank-Based Compatibility Definition

The rank-based compatibility, presented in **Equation 4.3**, compares the ranking produced by the updated model against the rankings produced by the original model. Like the AUROC measure, the evaluation is conducted on patient-pairs. However, in contrast to the AUROC, it counts the number of patient-pairs that both models rank correctly and is normalized by the number of patient-pairs that the original model ranked correctly.

$$\mathcal{C}^{\mathrm{R}}(f^o, f^u) = \frac{\sum_{i \in I^0} \sum_{j \in I^1} \mathbb{1}(\hat{p}_i^o < \hat{p}_j^o) \cdot \mathbb{1}(\hat{p}_i^u < \hat{p}_j^u)}{\sum_{i \in I^0} \sum_{j \in I^1} \mathbb{1}(\hat{p}_i^o < \hat{p}_j^o)} = \frac{m^{++}}{m^{o+}} = \frac{\phi^{++}}{AUROC(f^o)} \tag{4.3}$$

Note, we arrive at the last term by dividing the patient-pair counts by the total number of patient-pairs, m. Like $\mathcal{C}^{\mathrm{BT}}$, $\mathcal{C}^{\mathrm{R}}$ requires the use of a set of evaluation patients, I. However, it operates on pairs produced by the mutually disjoint subsets I^0 and I^1. $\mathcal{C}^{\mathrm{R}}$ measures the concordance of ranking instances and ranges from 0 to 1. In contrast $\mathcal{C}^{\mathrm{BT}}$ measures concordance with respect to binary predictions. Additionally, $\mathcal{C}^{\mathrm{R}}$ may be more robust to model miscalibration because, unlike $\mathcal{C}^{\mathrm{BT}}$, the rankings are not dependent on the actual values of the risk estimates. Instead, $\mathcal{C}^{\mathrm{R}}$ focuses on the relative ranking of patients produced by the original and updated model.

4.4.3 Rank-Based Compatibility Bounds & Central Tendency

We start by stating some assumptions that we believe will hold for all original and updated risk stratification models being considered for use in healthcare. These assumptions are 1) both models will have a discriminative performance better than random and 2) that the updated model will have equivalent or better performance than the original model:

$$0.5 < AUROC(f^o) \leq AUROC(f^u) \leq 1. \tag{4.4}$$

4.4.3.1 Rank-Based Compatibility Bounds

Given values for $AUROC(f^o)$ and $AUROC(f^u)$, we can bound all POP variables. We will focus only on the POP variable that represents both models ranking patient-pairs correctly, ϕ^{++}, as it is the only one used directly in $\mathcal{C}^{\mathrm{R}}$. The bounds on ϕ^{++} are:

$$AUROC(f^o) + AUROC(f^u) - 1 \leq \phi^{++} \leq \ AUROC(f^o)$$

These bounds follow from the fact that the minimum value ϕ^{++} can take is the smallest proportion of correctly ordered instance-pairs by both models. Since the AUROCs of both models must be at least 0.5, the smallest this proportion is when there is minimal overlap in the set of correctly ordered patient-pairs for each model. This is the sum of the two AUROCs subtracted by 1. The maximal value for ϕ^{++} is determined by the smaller of the two model's AUROC's which is $AUROC(f^o)$. This yields the following bounds on the rank-based compatibility metric:

$$\frac{AUROC(f^o) + AUROC(f^u) - 1}{AUROC(f^o)} \leq \mathcal{C}^{\mathrm{R}}(f^o, f^u) \leq 1 \tag{4.5}$$

We produce a plot for the lower bound of the rank-based compatibility measure (**Figure 4.1**). For the regime of model updating that we are interested in (*i.e.*, **Assumptions 4.4**), only the lower bound of $\mathcal{C}^{\mathrm{R}}$ varies, increasing as the discriminative performance of the two models grow.

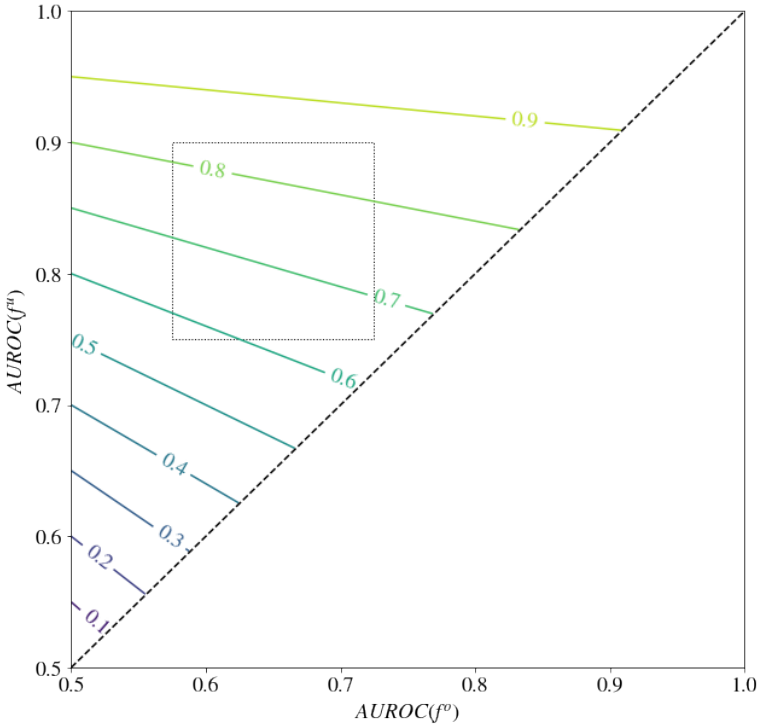


Figure 4.1: The lower bound of $\mathcal{C}^{\mathrm{R}}$ with respect to the discriminative performance of the original and updated models. The upper bound of rank-based compatibility is fixed at 1 when $0.5 < AUROC(f^o) \leq AUROC(f^u)$ (the triangular area above the dashed line). Note the lower bound increases as both models' performance increases. For context, we have boxed a region with dotted lines to demarcate a typical discriminative performance region. In this region, we would expect to observe $\mathcal{C}^{\mathrm{R}}$s no smaller than 0.5.

These bounds can be used to contextualize the $\mathcal{C}^{\mathrm{R}}$ of an update, as the range of $\mathcal{C}^{\mathrm{R}}$ changes depending on the model performances being considered. Additionally, they underscore a relation-

ship between the rank-based compatibility and the models' discriminative performance. The lower bound of $\mathcal{C}^{\mathrm{R}}$ increases with respect to the AUROC of the updated model.

For the model updating region we are interested, in the upper bound is always 1. Any updated model that correctly ranks all the patient-pairs that the original ranked correctly will produce $\mathcal{C}^{\mathrm{R}} = 1$ by setting $m^{++} = m^{o+}$. Ignoring the questions around the feasibility of constructing such updated models, we can see that there is always a possibility to construct an updated model with $\mathcal{C}^{\mathrm{R}} = 1$. This updated model would rank all of the original model's correctly ranked patient-pairs.

The combination of $\mathcal{C}^{\mathrm{R}}$'s increasing lower bound and its fixed upper bound for our region of interest suggests that it is theoretically possible to always find a candidate update model that leads to an increase in discriminative performance and high levels of rank-based compatibility (or even perfect rank-based compatibility) It is now natural to wonder if we might expect high levels of rank-based compatibility alongside the creation of updated models that maximize $AUROC(f^u)$. We revisit this later when we discuss our numerical experiments.

4.4.3.2 Central Tendency of Rank-Based Compatibility

Despite being informative in contextualizing the rank-based compatibility measure, the bounds in **Equation 4.5** provide a limited understanding of the behavior of $\mathcal{C}^{\mathrm{R}}$. The bounds show that $\mathcal{C}^{\mathrm{R}}$ increases with model AUROCs However, they do not explain how it would be distributed between the bounds. For updated models that are trained to minimize binary cross entropy loss, we hypothesize that the observed $\mathcal{C}^{\mathrm{R}}$ values will tend towards a value in the middle of the range. We present a brief analytical sketch of $\mathcal{C}^{\mathrm{R}}$'s behavior to explore this hypothesis.

Note, we do not seek to create a distribution for the $\mathcal{C}^{\mathrm{R}}$ generally (*e.g.*, for all data, for all models, and updating techniques); instead, we seek to build intuition for how $\mathcal{C}^{\mathrm{R}}$ may vary with both models' AUROC. This analytical approach is based on a combinatorial argument. We analyze the number of ways a given $\mathcal{C}^{\mathrm{R}}$ can occur given AUROCs for the original and updated models. This analysis is based on how each model ranks each patient-pair. A patient-pair's ranking for a given model is whether that model correctly ranks (*e.g.*, $\hat{p}_i < \hat{p}_j$ for the updated model) or incorrectly ranks that patient pair.

We can use the ranking of all patient-pairs to represent the behavior of original and updated models. All patient-pairs are distributed between two sets: correctly and incorrectly ranked. Suppose we constrain the distribution of patient-pairs between these two sets to align with the discriminative performance of the model being represented. In that case, we can then get a sense of the number of patient-pairs that both models rank correctly. This number is m^{++} and can be directly used to calculate the $\mathcal{C}^{\mathrm{R}}$ as per **Equation 4.3**. As mentioned in **Section 4.4.3.1**, m^{++} may range between $m^{o+} + m^{u+} - m$ and m^{o+}, corresponding to the bounds $\mathcal{C}^{\mathrm{R}}$ introduced in **Equation 4.5**.

Assuming models do not have any restrictions on how patient-pairs may be ranked we would

like to understand the behavior of $\mathcal{C}^{\mathrm{R}}$. To do this we will count the number of ways that each value of $m^{++} = k$, where $k \in \{m^{o+} + m^{u+} - m, ..., m^{o+}\}$, can be achieved given that each model meets a specific discriminative performance. This setup allows us to develop a closed-form expression for the number of combinations (or ways of ranking all patient-pairs) that yield $m^{++} = k$. This can be viewed as a measure of the size of the search space which we expect to be correlated wih the likelihood of selecting such a model.

The number of combinations is the numerator of the hypergeometric distribution with parameters related to the number of patient-pairs correctly ranked by the original and updated models. The number of patient-pairs that both models ranked correctly, $m^{++} = k$, is defined in relation to the number of total patient-pairs, m, the number of patient-pairs we are interested in selecting, m^{o+}, and the number of selections, m^{u+}. The number of combinations that produce a given $m^{++} = k$ is as follows:

$$|\{(m^{++} = k | m^{o+}, m^{u+})\}| = \binom{m^{o+}}{k}\binom{m - m^{o+}}{m^{u+} - k} \tag{4.6}$$

The location of the maxima and shape of this function provides us with a sense of the behavior of $\mathcal{C}^{\mathrm{R}}$ conditional on maintaining a fixed level of discrimination. We would expect this function's maxima to coincide with the mode of the corresponding hypergeometric distribution. For large values of m^{o+}, m^{u+}, and m we expect the mode of the hypergeometric distribution to be approximately equal to its mean. **Equation 4.6** has its maxima at $m^{++} = k^*$, where k^* is the value that provides the largest number of combinations.[2]This is:

$$k^* = \left\lfloor \frac{(m^{o+} + 1)(m^{u+} + 1)}{m + 2} \right\rfloor \approx \frac{m^{o+} m^{u+}}{m} \text{ for large } m^{o+}, m^{u+}, \text{ and } m.$$

We can then plot **Equation 4.6** to investigate the behavior of $\mathcal{C}^{\mathrm{R}}$ given $AUROC(f^o)$ and $AUROC(f^u)$. **Figure 4.2** shows the number of combinations for each $\mathcal{C}^{\mathrm{R}}$ value given original-updated model pairs. Each model pair had the same original model AUROC of 0.65, and the updated model AUROCs ranged between 0.65 and 0.95. Examination of these curves reveals several findings. First, the k^* for each model pair aligns with the AUROC of the updated model. Second, these curves exhibit a robust central tendency as the number of combinations decreases exponentially (note the logarithmic vertical axis) as $m^{++} = k$ diverges from k^*.

As expected, these curves cover the range between the bounds on $\mathcal{C}^{\mathrm{R}}$ for a given model pair. However, the number of combinations that produce the $\mathcal{C}^{\mathrm{R}}$ values at the upper and lower bounds are many orders of magnitude smaller than the maximal number of combinations. Additionally, we observe a difference in the number of combinations for the upper and lower bounds of a given

[2]This maxima is expressed in terms of m^{++}, which can be converted to be in terms of $\mathcal{C}^{\mathrm{R}}$ by dividing by m^{o+}. This maxima occurs at $\frac{m^{o+}}{m^{o+}}\frac{m^{u+}}{m} = \frac{m^{u+}}{m} = AUROC(f^u)$.

original and updated model pair. There are significantly more combinations that yield a $\mathcal{C}^{\mathrm{R}}$ equal to the lower bound than those that produce a $\mathcal{C}^{\mathrm{R}}$ equal to the upper bound.

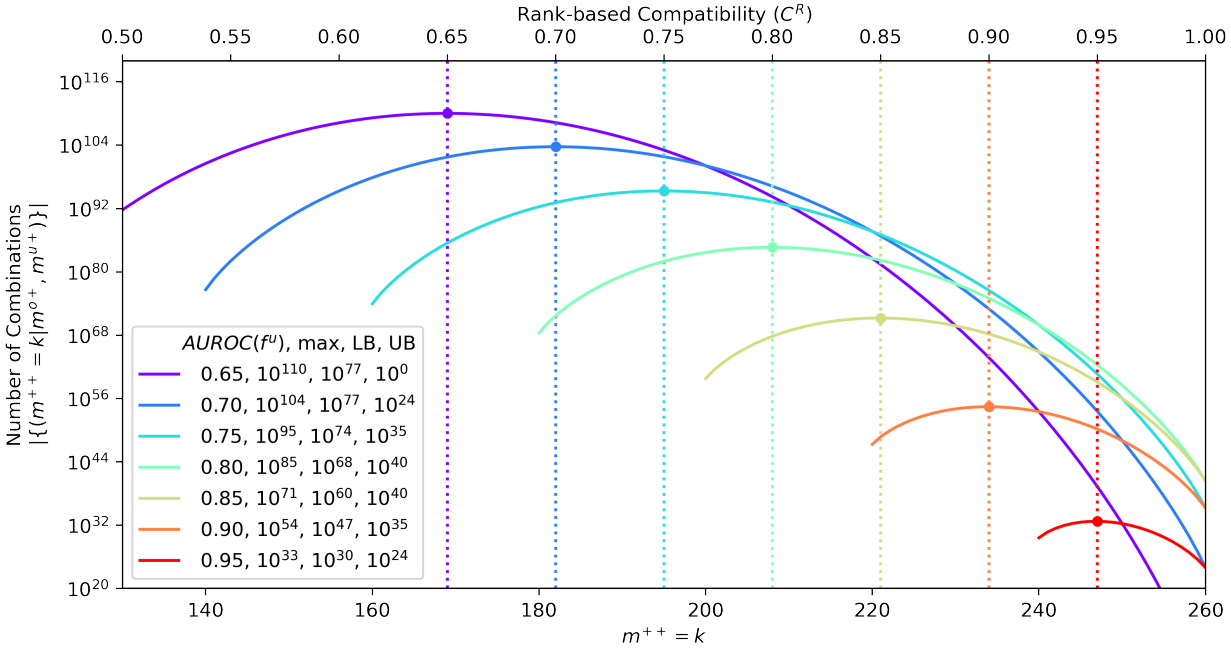


Figure 4.2: Central Tendency of $\mathcal{C}^{\mathrm{R}}$. We plot the combinations for each $\mathcal{C}^{\mathrm{R}}$ value given original-updated model pairs. Each model pair had the same original model AUROC of 0.65, and the updated model AUROCs ranged between 0.65 and 0.95. The updated model's AUROC is plotted as a vertical dotted line. We set $m = 400$. The $m^{++} = k^*$ value that achieves the largest number of combinations is plotted as a dot on the curves. This point aligns with $AUROC(f^o)$. Because all model-pairs have the same original model discriminative performance, the m^{++} values can be normalized by m^{o+} to provide the $\mathcal{C}^{\mathrm{R}}$ (top horizontal axis). These curves exhibit a strong central tendency as the number of combinations decreases increasingly (note the logarithmic vertical axis) as $m^{++} = k$ diverges from the k^*.

For this analysis, we assume that both models can select any subset of patient-pairs. This means that either model may select any patient-pair (or not). We note that this assumption may not entirely hold for all datasets and model types; for example, it may not be possible for a model to select one patient-pair and a not-select another patient-pair for a given dataset. Although employing this assumption limits our analysis's generalizability, we believe the analysis still provides utility by giving us a general sense of how $\mathcal{C}^{\mathrm{R}}$ may behave.

Additionally, although **Equation 4.6** relates to the hypergeometric distribution, we have intentionally avoided using this probability mass function for this analysis. This is because the true distribution for $\mathcal{C}^{\mathrm{R}}$ given the discriminative performances likely would not be directly modelled by the hypergeometric distribution. This is because the observation of each of the combinations being

counted in **Equation 4.6** likely do not all have the same probability. Furthermore, just because there are more ways to produce k does not mean it is more probable. For example, if m^{o+} is close to m, the number of combinations may be small, but this may be more probable. Finally, given specific data generation processs (DGPs) and model generating processes, we are more likely to observe some combinations than others. As mentioned above, some combinations may not even be possible.

This analysis is meant to illustrate a couple of crucial points about this problem. One, there are potentially many updated models to be explored that meet the given levels of original and updated model discriminative performance. This increases as we relax the search for updated models from a specific discriminative performance (*e.g.*, $AUROC(f^u) = 0.65$) to a range in discriminative performance (*e.g.*, $AUROC(f^u) \geq 0.65$). There are now many more combinations to be explored beyond what would be presented as a single curve on **Figure 4.2**. Additionally, this complicates any attempt to directly translate the notion of number of combinations into a sense of likelihood.

Two, the analysis also shows there some combinations achieve the bounds established above. For example, the upper bound of $\mathcal{C}^{\mathrm{R}} = 1$ can always be achieved in theory. This can be done by using the original model as the updated model or producing an updated model that produces the same correct patient-pair rankings. Three, more combinations will produce the lower bound than the upper bound. The number of combinations that produce a $\mathcal{C}^{\mathrm{R}} = 1$ is the smallest number of combinations observed in the whole range. Four, there is a central tendency in the number of combinations for $\mathcal{C}^{\mathrm{R}}$, as there are many more ways to produce a $\mathcal{C}^{\mathrm{R}}$ between the bounds than there are ways to produce $\mathcal{C}^{\mathrm{R}}$ values close to either bound. For **Equation 4.6**, this center is located at $\frac{m^{o+}m^{u+}}{m}$.

While we do not believe this specific center to hold for all DGPs and model updating procedures, we hypothesize that $\mathcal{C}^{\mathrm{R}}$ central tendency does. In **Section 4.5.2**, we investigate the central tendency of $\mathcal{C}^{\mathrm{R}}$ for original and updates trained using real data. The above analysis is still illuminating as it provides a way to estimate the relative number of combinations between different rank-based compatibility levels. There are many more ways for an updated model to achieve a moderate rank-based compatibility (near the value of the AUROC of the updated model) than a very high level compatibility (*e.g.*, above 0.95). This suggests that achieving very high levels of rank-based compatibility may be difficult without directed search efforts, which we discuss in the next section.

4.4.4 Training Model Updates Using Rank-Based Incompatibility Loss

Risk stratification models are often trained by minimizing the binary cross-entropy loss function, $\mathcal{L}^{BCE}$.

$$\mathcal{L}^{BCE} = -\frac{1}{n}\sum_{i \in I} y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i) \tag{4.7}$$

Minimizing $\mathcal{L}^{BCE}$ is equivalent to minimizing the log-loss and maximizing the log-likelihood. [58, 166, 167] Training with the $\mathcal{L}^{BCE}$ function attempts to optimize the discriminative performance of the model by reducing the probability estimates for 0-labeled patients and increasing them for 1-labeled patients. This in turn, indirectly optimizes the correct ranking of patient-pairs, the AUROC. [165] However, $\mathcal{L}^{BCE}$ only examines the relationship between a patient's label and the risk estimates produced by a model. To incorporate the rank-based compatibility, additional information in the form of the original model's risk estimate is needed.

Model developers may seek to generate updated models with high rank-based compatibility directly. Thus, we propose augmenting model update training to incentivize rank-based compatibility. This would allow model developers to avoid potentially fruitless searches over many updated models (*i.e.*, model selection) and more directly create updates that balance discriminative performance improvement and high rank-based compatibility.

To do this, we introduce a loss function called *rank-based incompatibility*, $\mathcal{L}^{R}$. Which we define as:

$$\mathcal{L}^{R} = 1 - \mathcal{C}^{\mathrm{R}}\,. \tag{4.8}$$

Rank-based incompatibility can be used as an additional loss term for model training. For example, suppose updated models were initially trained using binary cross entropy, $\mathcal{L}^{BCE}$. In that case, update training may be augmented to use a weighted combination of binary cross entropy and rank-based incompatibility:

$$\alpha\mathcal{L}^{BCE} + (1 - \alpha)\mathcal{L}^{R} \text{ where } \alpha \in [0, 1]. \tag{4.9}$$

The objective function presented in **Equation 4.9** enables us to train models by balancing the loss typically used to train risk stratification models ($\mathcal{L}^{BCE}$) with compatibility (in the form of $\mathcal{L}^{R}$). At the extremes, if $\alpha = 1$, then we train models by only focusing on minimizing $\mathcal{L}^{BCE}$, and if $\alpha = 0$, we only focus on reducing incompatibility (thus, maximizing compatibility). Varying α in the interval $[0, 1]$ provides a means to trade-off two important criteria.

The use of rank-based incompatibility has two stipulations. 1) Predictions produced by the original model must be incorporated into the loss function. 2) The exact function is non-differentiable due to the *ranking indicator function* ($\mathbb{1}(\hat{p}_i < \hat{p}_j)$). The first stipulation can be overcome by either embedding the original model into the loss function or calculating the original model's risk estimates ahead of the updated model training time. The second stipulation means rank-based in-

compatibility cannot be directly used in gradient based optimization procedures. We introduce a differentiable approximation of rank-based incompatibility ($\widetilde{\mathcal{L}^R}$).

This approximation replaces the ranking indicator function used to evaluate patient pairs with a *ranking sigmoid function*. The ranking sigmoid function utilizes a standard sigmoid function to operate on the estimated risk difference, $\hat{d}_{ji}$. [168] This is the difference in risk estimates produced for a patient pair, $\hat{d}_{ji} = \hat{p}_j - \hat{p}_i$, where $\hat{d}_{ji}$ naturally ranges between -1 and 1. A correct ranking corresponds to $\hat{d}_{ji} > 0$ and an incorrect ranking corresponds to $\hat{d}_{ji} < 0$. In order to align with the behavior of the ranking indicator function we encapsulate $\hat{d}_{ji}$ in a sigmoid function, which returns values between 0 and 1. This is the ranking sigmoid function and it is defined as follows:

$$\sigma(\hat{d}_{ji}) = \frac{1}{1 + \exp(-s \cdot \hat{d}_{ji})}$$

When $\hat{d}_{ji} > 0$, $\sigma(\hat{d}_{ji})$ will return a value between 0.5 and 1 and when $\hat{d}_{ji} < 0$, $\sigma(\hat{d}_{ji})$ will return a value between 0 and 0.5. Ideally, to match the behavior of the ranking indicator function the ranking sigmoid function would only return values of 0 and 1. We can drive these values closer to 0 and 1 using the spreading hyperparameter, s. Using s helps to ensure that differences close to 0 still get converted to values near 0 and 1. Note that using a sigmoid to overcome discontinuity in loss function is similar to work introduced to optimize for the AUROC directly. [169]

Using the ranking sigmoid function, we define the differentiable approximation of rank-based compatibility as follows:

$$\widetilde{\mathcal{C}^{\mathrm{R}}}(f^o, f^u) = \frac{\sum\limits_{i \in I^-} \sum\limits_{j \in I^+} \sigma(\hat{p}_j^o - \hat{p}_i^o) \cdot \sigma(\hat{p}_j^u - \hat{p}_i^u)}{\sum\limits_{i \in I^-} \sum\limits_{j \in I^+} \sigma(\hat{p}_j^o - \hat{p}_i^o)} \tag{4.10}$$

As mentioned, this $\mathcal{C}^{\mathrm{R}}$ approximation functions by converting the differences between the risk estimates for patients into values close to 0 and 1 in a differentiable manner. The exact $\mathcal{C}^{\mathrm{R}}$ cannot be used in settings dependent on gradient computation as it depends on the ranking indicator function, which is non-differentiable.

Ranking Function Visualization. Because approximate incompatibility loss operates on the set of all patient-pairs it isn't easy to visualize directly. Instead, we show two visualizations of the ranking indicator and ranking sigmoid functions. These visualizations serve as representations for the functionality of the exact and approximate loss functions for a single patient-pair. Both visualizations show the behavior of the *loss component*. The exact loss component is 1 minus the ranking indicator function, and the approximate loss component is 1 minus the ranking sigmoid function. The loss components simulate the behavior of $\mathcal{L}^R$ and $\widetilde{\mathcal{L}^R}$ for a single patient-pair.

The first visualization, **Figure 4.3**, shows the behavior of the ranking functions given a patient-pair with fixed risk for the 0-labeled patient ($\hat{p}_i = 0.25$). The second visualization, **Figure 4.4**, shows this behavior while varying the risk estimates for both the 0- and 1-labeled patients ($\hat{p}_i \in [0, 1]$ and $\hat{p}_j \in [0, 1]$). In both of these figures, we show the behavior of the approximate loss component for various s values and compare these to the exact loss component.

As s gets larger, we see that the ranking sigmoid function more closely matches the behavior of the ranking indicator function. This behavior is desirable until some point, as the gradient of the function becomes steeper and steeper. We have found that very large values of s (*e.g.*, $s \approx 100,000$) lead to numerical instability while training models with stochastic gradient descent. To avoid numerical instability issues, we set $s = 100$ for the experimental work using the approximate incompatibility loss. Experiments showing the robustness of the results to the setting of $s > 100$ can be found in **Section C.2.0.2**.

Note that the ranking sigmoid function is not the only differentiable alternative to the ranking indicator function. Another approach is to define a gradient for the ranking indicator function ilike the ReLU activation function. [58]

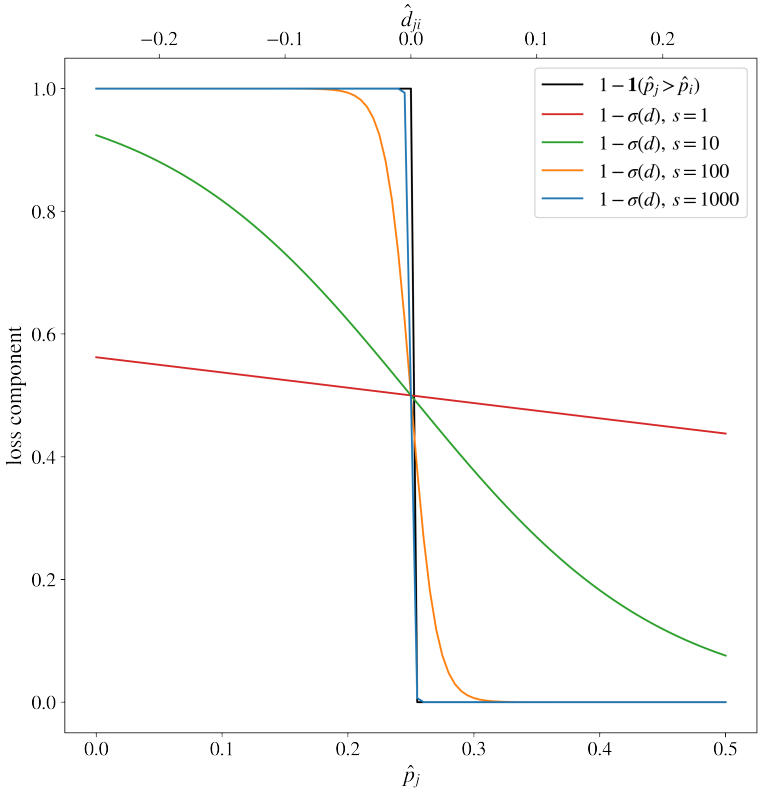


Figure 4.3: Visualization of the Behavior of Ranking Indicator Function and the Ranking Sigmoid Function Sweeping a Single Risk Estimate. We sweep the risk estimate for the 1-labeled patient ($\hat{p}_j \in [0, 1]$) and plot loss components. The approximate loss component is plotted at various s values using a fixed $\hat{p}_i$. For this comparison, we fix the risk estimate for the 0-labeled patient ($\hat{p}_i = 0.25$). We then plot the exact loss component as a reference against the approximate loss component. We evaluate the ranking sigmoid function at various s values. As s increases, the gradient for the ranking sigmoid function (and approximate incompatibility) becomes much steeper but more closely approximates the ranking indicator function (and exact incompatibility loss).

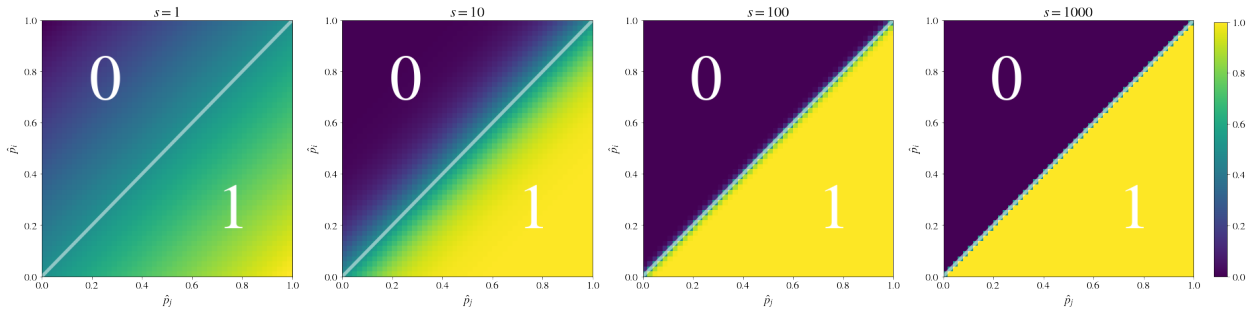


Figure 4.4: Visualization of the Behavior of Ranking Indicator Function and the Ranking Sigmoid Function Sweeping Both Patient Risk Estimates. Ranking sigmoid function at $s = 1, s = 10$, and $s = 100$ both $\hat{p}_i$ and $\hat{p}_j$ are varied between 0 and 1. We compare the exact loss component against the approximate loss component. We plot the approximate loss component value for patient-pairs i, j where $y_i = 0$ and $y_j = 1$. Their risk estimates $\hat{p}_i$, $\hat{p}_j$ are denoted on the horizontal and vertical axes, respectively. The area above the white line corresponds to an exact loss component value of 0. The area below corresponds to a value of 1. The color represents the approximate loss component, with blue representing values close to 0 and yellow representing values close to 1. As s increases, the gradient for this function (and approximate incompatibility) becomes much steeper but more closely approximates the exact incompatibility loss using the greater than operation.

4.5 Experiments & Results

We now present experiments that focus on understanding and engineering the behavior of model updates in terms of $\mathcal{C}^{\mathrm{R}}$ using a real-world dataset. We generated and analyzed model updates on the MIMIC-III mortality prediction dataset. After describing the dataset and model updating setup, we examine the $\mathcal{C}^{\mathrm{R}}$ observed for updated models created using standard updated model generation procedures. Additionally, we investigate the utility of engineering model updates such that $\mathcal{C}^{\mathrm{R}}$ is incorporated as a part of the updated model training process.

Questions. These experiments seek to answer two related questions:

1. *Does $\mathcal{C}^{\mathrm{R}}$ demonstrate a central tendency on model-pairs generated using standard update model generation when using real data?* (**Section 4.5.2**, **Figure 4.6**)

2. *Compared to standard update model generation and selection approaches, can we use the rank-based incompatibility loss, $\mathcal{L}^{R}$, to generate updates with better $\mathcal{C}^{\mathrm{R}}$? And can this be accomplished without a loss of AUROC?* (**Section 4.5.3**, **Figures 4.7** and **4.8**)

4.5.1 Data & Model Updating Setup

We simulate model updating using the MIMIC-III 48-hour mortality risk stratification dataset. We utilized this task as it is a widely available benchmark for healthcare machine learning and was employed by Bansal et al. [31]. All of the data were transformed using FIDDLE. [100] For details regarding the data inclusion and transformation, please see the procedures detailed by Tang et al. [100]. The only notable difference is that for computational efficiency, we reduce the number of features from $350,832$ to $35,000$ by random sampling.

This simulation was conducted by randomly splitting the MIMIC-III data into three disjoint datasets. Two of these datasets were allocated for model development and validation. The third dataset was reserved for held-out evaluation. Of the $8,577$ patients in the MIMIC-III mortality dataset, $1,000$ were allocated to the original model dataset. $5,000$ were assigned to the updated model dataset, and $2,577$ were held-out for the evaluation dataset. This distribution was chosen as it aligned with the distribution employed by Bansal et al. [31] and represents real-world model updating processes by enabling more data to be used for the updated model development. The two model datasets were used to develop and validate the original and updated models. The model datasets were each equally split (50/50%) into development and validation datasets. The dataset partitions and their sizes are depicted in **Figure 4.5**.

Original and updated models utilize a logistic regression architecture, which was implemented in TensorFlow, and training was conducted using SGD. [89] During the original model training validation loss was calculated using the original validation dataset, enabling the use of early stopping regularization. The updated models were initialized with the original model's weights and bias and then trained using the updated model development dataset. The updated model validation dataset was used for early stopping during updated model training and for updated model selection. The selected updated models were evaluated in terms of $\mathcal{C}^{\mathrm{R}}$ and AUROC on the held-out evaluation dataset. This procedure was replicated 40 times to understand how these results may vary. In **Program C.1**, we present an example python code that demonstrates dataset splitting and model-pair development.

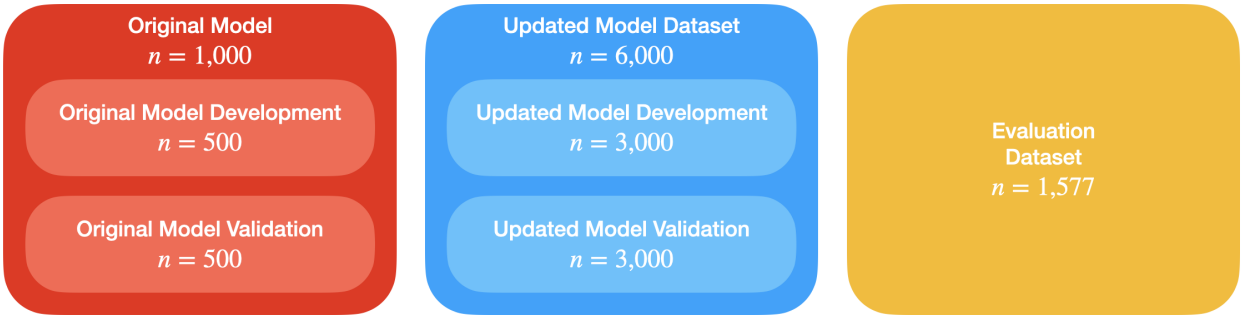


Figure 4.5: The MIMIC-III mortality data was partitioned into three datasets. Two of these datasets were allocated for model development and validation, one was held-out for evaluation. The model datasets were each split into development and validation datasets and were utilized for original and updated model development. Model-pairs were evaluated on the evaluation dataset.

4.5.2 Rank-Based Compatibility Central Tendency

We first investigated the central tendency posited by the theoretical analysis shown in **Section 4.4.3.2** is observed when generating realistic updates. We seek to answer the question of: *Does* $\mathcal{C}^{\mathrm{R}}$ *demonstrate a central tendency on model-pairs generated using standard update model generation when using real data?* Using the experimental setup described above, we used standard updated model generation procedures to create 150 updated models for each original model. These 150 updated models were created through a combination of dataset resampling, shuffling, and regularization weights. The updated model development dataset was either resampled (45 of the times) or shuffled (5 of the times) and then models were trained using binary cross entropy loss with one of three L2 regularization weights $\{0.1, 0.01, 0.001\}$. This of regularization range was selected using a preliminary scoping analysis, this is discussed in detail in **Appendix Section C.2.0.1**. This combination of dataset modification and regularization yielded $50 \times 3 = 150$ updated models in total.

Standard updated model generation was conducted as follows. All updated models were initialized using the weights and biases of the original models. They were then all trained using updated model development dataset using SGD to minimize binary cross entropy loss, $\mathcal{L}^{BCE}$. The training procedure was regularized using early stopping based on the validation AUROC.

We then examined the resultant $\mathcal{C}^{\mathrm{R}}$ distribution across the candidate update models. This distribution was assessed in terms of the empirical 95% confidence interval. We repeated this procedure for each replication. Each replication yielded new dataset splits, new original models, and new updated candidate models.

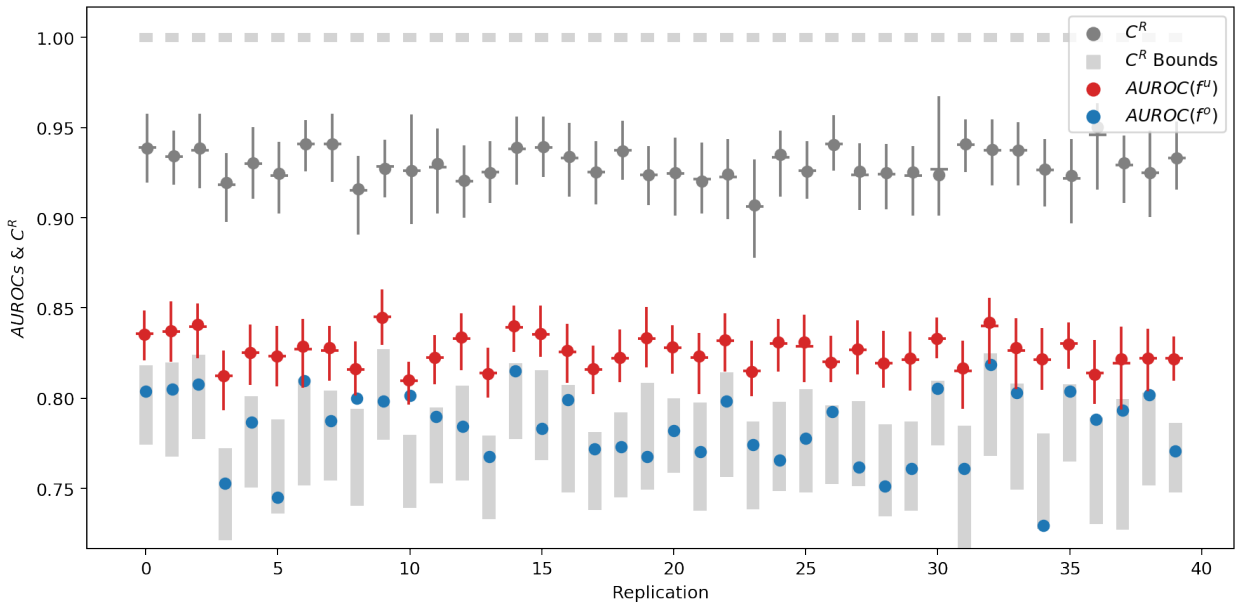


Figure 4.6: Central Tendency of $\mathcal{C}^{\mathrm{R}}$ For Model Updates on the MIMIC-III Mortality Task. The 95% confidence intervals for $\mathcal{C}^{\mathrm{R}}$ (gray) of the 100 candidate update models generated by standard update model generation. The AUROC of the original model (blue) and set of candidate update models (red, 95% confidence intervals) are plotted along with the upper and lower bounds for $\mathcal{C}^{\mathrm{R}}$. We note the restricted range of the confidence intervals for $\mathcal{C}^{\mathrm{R}}$ despite a large number of candidate models.

In addition to $\mathcal{C}^{\mathrm{R}}$, we calculated the AUROC of the original model, the distribution of the AUROCs for the candidate updated models, and the expected bounds of $\mathcal{C}^{\mathrm{R}}$. The results for all replications are depicted in **Figure 4.6**. This figure demonstrates that the observed $\mathcal{C}^{\mathrm{R}}$ values for the set of candidate updates occupies a small portion of the feasible range (between the lower[3] and upper bounds).

From this experiment, we see a central tendency in $\mathcal{C}^{\mathrm{R}}$ for updated models generated on real data using standard update model generation. This central tendency in $\mathcal{C}^{\mathrm{R}}$ means that model developers may be constrained if they wish to develop updated models that try to balance AUROC and $\mathcal{C}^{\mathrm{R}}$ using standard update generation procedures. We now turn our attention to evaluating the proposed model updating method via the weighted updated loss function.

4.5.3 Weighted Loss vs. Standard Updated Model Selection

We now investigate our second question: *Compared to standard update model generation and selection approaches, can we use the rank-based incompatibility loss, $\mathcal{L}^R$, to generate updates*

[3]Note, that the lower bound is presented as a range. This is because each candidate update model has a separate lower bound depending on its AUROC.

with better $\mathcal{C}^{\mathrm{R}}$? To answer this question, we compare the updated models generated using the standard updated model generation approach (described above) against updated models trained using the rank-based incompatibility loss function.

For each replication, we generated 150 models using the standard update model generation procedure. We refer to these as the "selection models." Using the same original model and data, we generated additional update models, we refer to these as the "engineered models." The engineered models were also initialized with the original model's weights and bias. However, they utilize the approximate rank-based compatibility loss ($\widetilde{\mathcal{L}^R}$) introduced in **Section 4.4.4** as a part of training. They were then trained using the weighted loss function on the updated model development dataset. The weighted updated loss function is based on **Equation 4.9**:

$$\alpha\mathcal{L}^{BCE} + (1-\alpha)\widetilde{\mathcal{L}^R}. \tag{4.11}$$

This weighted updated loss function includes a hyperparameter α that controls the trade-off between $\mathcal{L}^{BCE}$ and $\mathcal{L}^R$. When $\alpha = 1$, the weighted updated loss function equals $\mathcal{L}^{BCE}$ allowing us to replicate the standard model training procedure. When $\alpha = 0$, the weighted updated loss function equals $\widetilde{\mathcal{L}^R}$, and for $\alpha = 0.5$, the weighted updated loss function weights the $\mathcal{L}^{BCE}$ and $\mathcal{L}^R$ equally.

This procedure was conducted for each α in the set $\{0, 0.1, 0.2, ..., 0.9, 1\}$ and each L2 weight. Thus, there were 33 engineered models. Using the updated model validation dataset the best performing engineered model for each α was selected. In addition to using the weighted updated loss function, the engineered models had their early stopping criteria modified to align with their loss function, specifically, we used: $\alpha AUROC + (1-\alpha)C^R$.

To assess the impact of using the weighted updated loss function versus the standard update model generation procedure, we need a method to compare the candidate update models produced by both approaches. We do this by selecting a single selection model and comparing it directly against an engineered model. There are many potential ways to conduct this selection, a variety of which are cataloged in **Section C.3**. However, we focus on the one that aligns most with the standard update model generation procedure. This is a selection procedure based on AUROC. The candidate update model with the highest AUROC observed on the updated model validation dataset is the one that is selected. This selected model is then compared against each engineered model by calculating the difference in rank-based compatibility, $\Delta\,\mathcal{C}^{\mathrm{R}}$, and difference in AUROC, $\Delta AUROC$. These are defined as follows:

$$\Delta\,\mathcal{C}^{\mathrm{R}} = \mathcal{C}^{\mathrm{R}}(f^{\alpha}, f^{o}) - \mathcal{C}^{\mathrm{R}}(f^{s}, f^{o}) \tag{4.12}$$

$$\Delta AUROC = AUROC(f^{\alpha}) - AUROC(f^{s}) \tag{4.13}$$

Where f^{α} is the engineered updated model created using the weighted updated loss function with a trade-off weight of α and f^{s} is the selected candidate update model.

In **Figure 4.7** we show the $\mathcal{C}^{\mathrm{R}}$ and AUROC values calculated on the held-out evaluation data for all of the engineered models and for a subset of the selection models. This subset represents the selection models along the pareto frontier of the trade-off between $\mathcal{C}^{\mathrm{R}}$ and $AUROC$ (calculated using the updated model validation data). We also depict an example of how $\Delta\,\mathcal{C}^{\mathrm{R}}$ and $\Delta AUROC$ would be calculated between the engineered model where $\alpha = 0.5$ and the selected candidate update with the best AUROC.

For this example, we note that the circled engineered model induces a positive $\Delta\,\mathcal{C}^{\mathrm{R}}$ which denotes an increase in $\mathcal{C}^{\mathrm{R}}$, and a negative $\Delta AUROC$ which represents a reduction in AUROC. Although the $\Delta AUROC$ is negative, this does not mean that this updated model performs worse than the original model, which has an $AUROC = 0.805$. Instead, the engineered update ($AUROC = 0.848$) does not perform as well as the best-performing selection model ($AUROC = 0.855$).

As in the first experiment, we replicated this procedure 40 times and calculated 95% confidence intervals. These results are shown in **Figure 4.8**. From this figure, we can see that engineered models with more weight on incompatibility (lower α values) have higher compatibility and lower AUROCs. As α increases $\mathcal{C}^{\mathrm{R}}$ decreases, and AUROC increases. At α values ≤ 0.6 we see statistically significant increases in $\mathcal{C}^{\mathrm{R}}$ ($\Delta\,\mathcal{C}^{\mathrm{R}} > 0$). For some α values less than 0.6 we see statistically significant decreases in $AUROC$ ($\Delta AUROC < 0$ occurs at $\alpha \in \{0.0, 0.1, 0.2, 0.4\}$). These results pick the selection model based on the updated model validation AUROC. This selection procedure is most in line with standard model update generation procedures. In **Appendix Section C.3** we show that these results do not vary greatly for other selection procedures that may be used.

In sum, these results suggest that using the weighted updated loss function, we can generate updated models with larger $\mathcal{C}^{\mathrm{R}}$ values than would be observed through standard update generation procedures. It is important to note that this increase in $\mathcal{C}^{\mathrm{R}}$ appears to be accompanied by a cost in terms of the AUROC produced by the engineered models. In order to achieve the benefit of increased $\mathcal{C}^{\mathrm{R}}$, the resulting AUROC of the selected update model may be lower than an updated model generated through standard procedures.

This experiment suggests that updated models with improved rank-based compatibility can be trained using a weighted loss function incorporating incompatibility. Strong emphasis on compatibility may come at a cost in terms of discriminative improvement. However, this may be desirable in specific use-cases where compatibility is critical for good joint user-model performance. Using the incompatibility loss function may help model developers create updated models that balance user expectations and performance improvement.

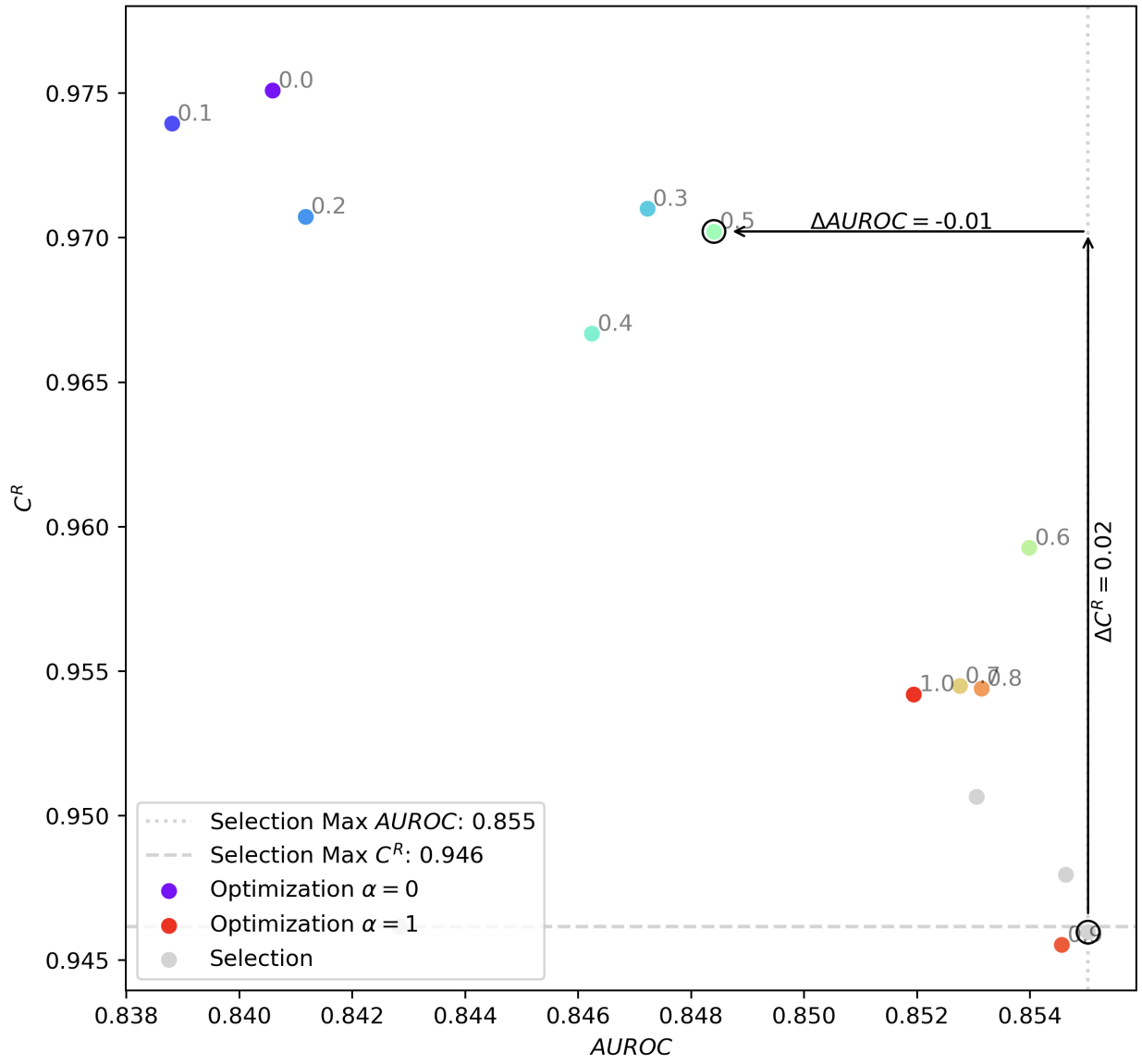


Figure 4.7: Example of Engineered Model vs. Selection Model Results. The AUROC and $\mathcal{C}^{\mathrm{R}}$ calculated on held-out evaluation dataset are reported for the engineered models and a subset of the selection models. In this example, we note that the circled engineered model ($\alpha = 0.5$) induces a positive $\Delta\,\mathcal{C}^{\mathrm{R}}$, which denotes an increase in $\mathcal{C}^{\mathrm{R}}$, and a negative $\Delta AUROC$ which indicates a reduction in AUROC.

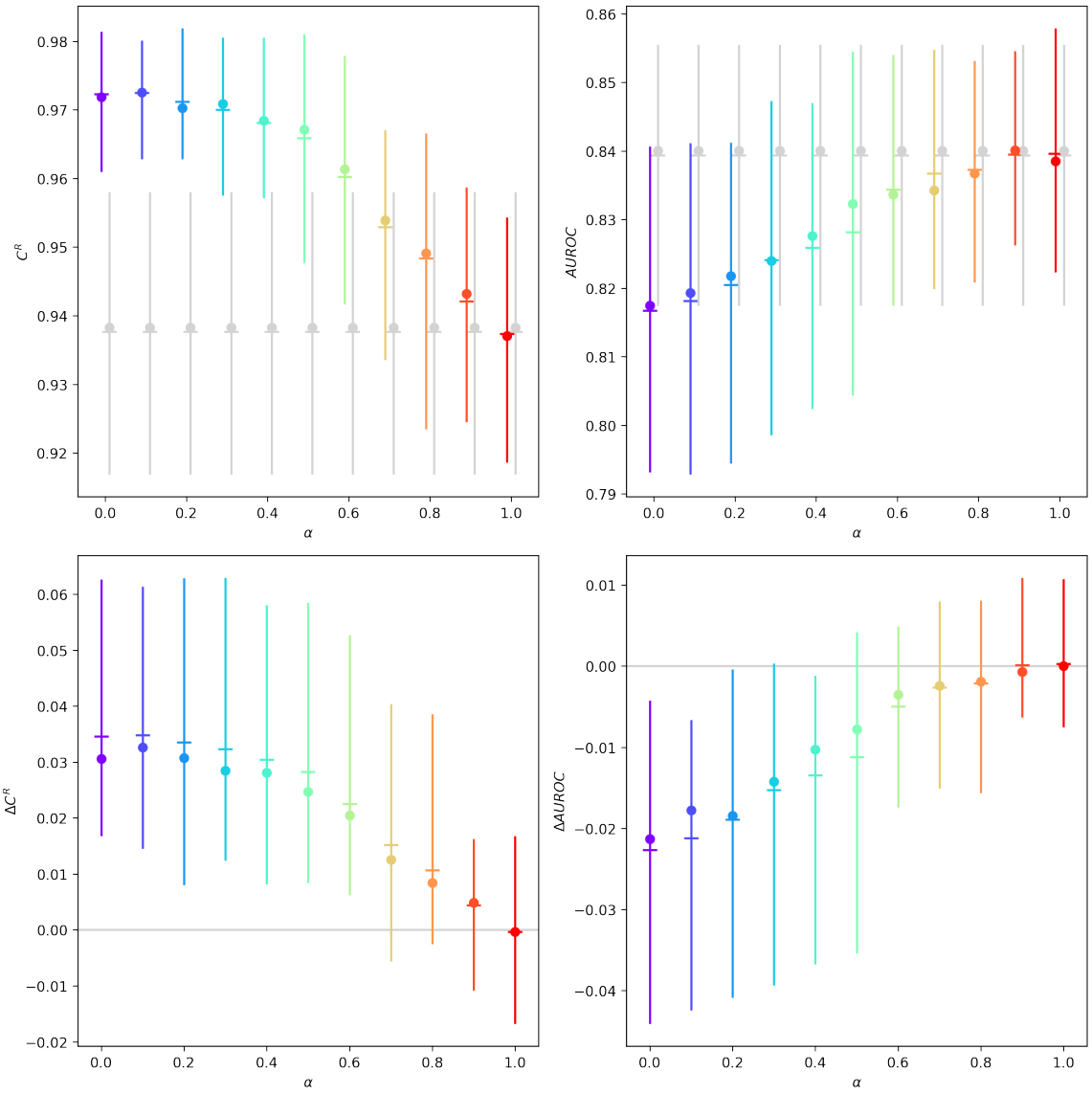


Figure 4.8: Engineered Models vs. Selection Models. Comparing engineered update models against selection models for 40 replications, each replication re-samples the mortality dataset and generates a new original model. 95% confidence intervals across all the replications are depicted for each α value (colored vertical bars), and the selection models (dashed gray lines) are depicted for $\mathcal{C}^{\mathrm{R}}$ (top left) and $AUROC$ (top right). These graphs show that engineered models with more weight on incompatibility (lower α values) have higher $\mathcal{C}^{\mathrm{R}}$ and lower AUROCs. As α increases $\mathcal{C}^{\mathrm{R}}$ decreases and AUROC increases. In the bottom graphs, we calculate $\Delta\,\mathcal{C}^{\mathrm{R}}$ (bottom left) and $\Delta AUROC$ (bottom right) for each replication and depict 95% confidence intervals. Using these graphs, we can assess statistical significance. At α values ≤ 0.6 we see statistically significant increases in $\mathcal{C}^{\mathrm{R}}$ ($\Delta\,\mathcal{C}^{\mathrm{R}} > 0$) and at α values of 0.0, 0.1, 0.2, and 0.4 statistically significant decreases in $AUROC$ ($\Delta AUROC < 0$).

4.6 Risk Stratification Model Updates for Prostate Cancer Case Study

Rank-based compatibility estimates the discrepancy between what clinicians expect and the updated model's risk stratification. To highlight its utility to model decision makers we present a case study focused on model updating of prostate cancer (PCa) prediction modeling. This case study extends a recent model development study and demonstrates how model developers may use rank-based compatibility and related measures to assess a potential model update.

This recent study examined an ML model (known as the Memorial Sloan Kettering (MSK) model) currently being used for non-organ confined disease (NOCD) prediction in patients with prostate cancer. Additionally, it introduced a new model (known as the Michigan Urological Surgery Improvement Collaborative (MUSIC) model). The MUSIC model exhibited better performance than the MSK model in terms of discrimination and calibration. Based on this information policymakers may recommend that urologists switch to the MUSIC model if they are currently using the MSK model. However, before making this recommendation, they may want to understand the impact this change will have on urologists' expectations.

We study this by evaluating NOCD risk stratification between the MSK and MUSIC models using the MUSIC validation dataset. This allows us to take the role of MUSIC administrators and understand the impact of recommending that all urologists in the collaborative switch to using the MUSIC model. Understanding the effects of this change is vital as there is a model performance improvement to be gained if the urologists were to switch from using the MSK model to the MUSIC model. This is because the MSK model has a lower AUROC of 0.68 (bootstrapped 95% confidence interval of: 0.66, 0.70), compared to the MUSIC model's AUROC of 0.74 (0.72, 0.76). This performance evaluation and dataset details were initially presented in a study by Ötleș et al. [170]. Using these AUROC, values we calculated bounds for rank-based compatibility. We then examined the potential update by calculating the rank-based compatibility on the MUSIC validation cohort.

The MUSIC validation cohort contained information from 2,911 patients collected from 41 urology practices across the state of Michigan. Each practice that contributed data to the MUSIC validation cohort had previously obtained an exemption or approval for participation from a local institutional review board. Using the previously reported AUROCs of the MSK and MUSIC, we could calculate that the rank-based compatibility would be between 0.62 and 1.00. The observed rank-based compatibility observed in the MUSIC validation cohort was 0.88 (0.87, 0.89). Confidence intervals were generated using 1,000-fold bootstrapping. The distribution of the discriminative performances of the models and the rank-based compatibility are displayed in **Figure: 4.9**.

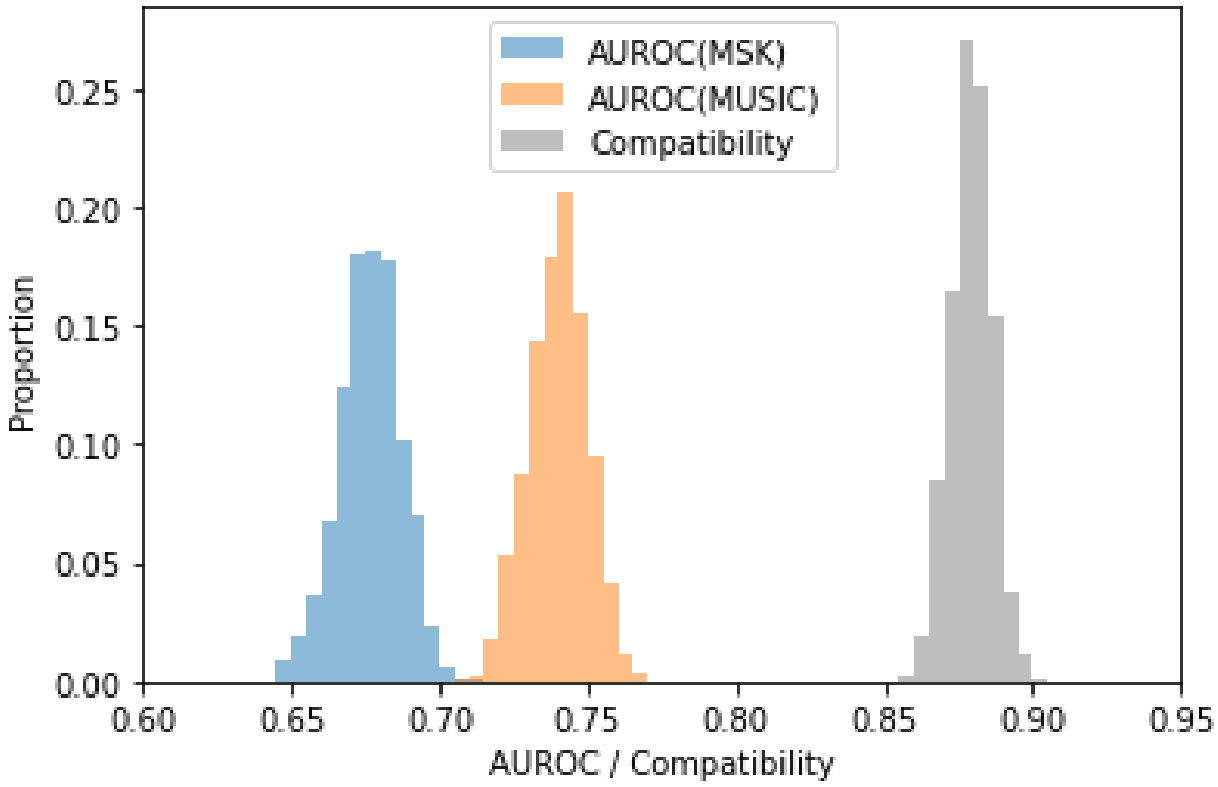


Figure 4.9: Original (MSK) and updated (MUSIC) model discriminative performance and rank-based compatibility for the non-organ confined disease prostate cancer risk prediction task. Left two histograms: discriminative performance of the MSK model currently in use is compared to the MUSIC model being considered as an update. Right histogram: rank-based compatibility of the potential update. The histograms were generated by bootstrapping with 1,000 replications.

We can utilize the bounds and the theoretical distributions to contextualize the observed values of the rank-based compatibility. We note that the rank-based compatibility falls in the upper half of its bounds.

Additionally, decision makers can use the observed rank-based compatibility and POP variable values when considering how to best facilitate the implementation of the updated model. POP values, along with the bootstrapped 95 percent confidence intervals, for this update are displayed in **Table 4.2**. ϕ^{++}, was 0.59, meaning that the majority of the patient-pairs were correctly ranked by both models, a desirable attribute. The remaining POP variables were as follows: $\phi^{+-} = 0.08$, $\phi^{-+} = 0.15$, and $\phi^{--} = 0.18$. Generally speaking, we would like ϕ^{+-} to be lower as it represents the proportion of patient-pairs that were originally ranked correctly but would be ranked incorrectly by the updated model, which is an unwanted attribute of a candidate update. Having a larger ϕ^{-+} may be desirable. This may mean that the updated model achieves a high level of discriminative performance by correcting ranking errors.

It is important to contrast ϕ^{++} and ϕ^{-+}, as they indicate how the updated models is deriving its AUROC. For this example, we see that the updated model's AUROC is primarily derived from ϕ^{++} as ($\phi^{++}/AUROC(f^u) \approx 0.594/0.74 = 0.81$), not from ϕ^{-+}, which is desirable. Finally, $\phi^{--} = 0.18$, this represents the proportion of patient-pairs that both models incorrectly rank. Lower is generally better for this value as it means that the updated model can fix ranking errors that the original model made.

Other than being able to comment generally on what values may be desirable, *e.g.*, high ϕ^{++}, it is hard to classify this update as good or bad at this juncture. This is due to the fact that we are only assessing one updated model due to the relative novelty of the rank-based compatibility measure and the related POP variables. Without a user-based study it is difficult to judge the quality of the update or the impact of the update on clinical workflows and patient outcomes. Additionally, as more model updates are evaluated, we may develop a sense of ranges for these values that correspond to good and bad updates (like we currently have with discriminative performance).

Although these measures do not give us a definitive absolute assessment of the potential MUSIC update, they do allow us to examine if certain urologists will have their expectations impacted more than others. We can calculate these update measures for each of the different urological surgery practices and assess the relative differences. This secondary analysis examining the performance, rank-based compatibility, and POP variables at the level of individual MUSIC practices can be seen in **Figure 4.10**.

In the **bottom left sub-figure of 4.10**, we see the rank-based compatibility value over the whole population of patients compared against the value observed by the individual practices. It is important to note that some practices that will see rank-based compatibility worse than the level presented at the population level. In these cases model administrators might choose to make a different decision regarding updating for these predictions. Or they may choose to use the update for these practices and counteract the problems by providing additional training or preparation for urologists at those practices. The four panels on the right depict the POP variables. As in the case of rank-based compatibility, we see that some practices would experience worse modification to expectations than the population level would suggest.

Table 4.2: Discriminative performance and POP variables for Prostate Cancer Risk Stratification Model Updating. The row and column sums may not exactly match due to rounding.

	MSK Model Ranks Correctly	**MSK Model Ranks Incorrectly**	
MUSIC Model Ranks Correctly	$\phi^{++} = 0.59\ (0.57, 0.62)$	$\phi^{-+} = 0.15\ (0.13, 0.16)$	$AUROC(MSK) = 0.74\ (0.72, 0.76)$
MUSIC Model Ranks Incorrectly	$\phi^{+-} = 0.08\ (0.07, 0.09)$	$\phi^{--} = 0.18\ (0.16, 0.19)$	$1 - AUROC(MSK) = 0.26$
	$AUROC(MUSIC) = 0.68\ (0.66, 0.70)$	$1 - AUROC(MUSIC) = 0.32$	1

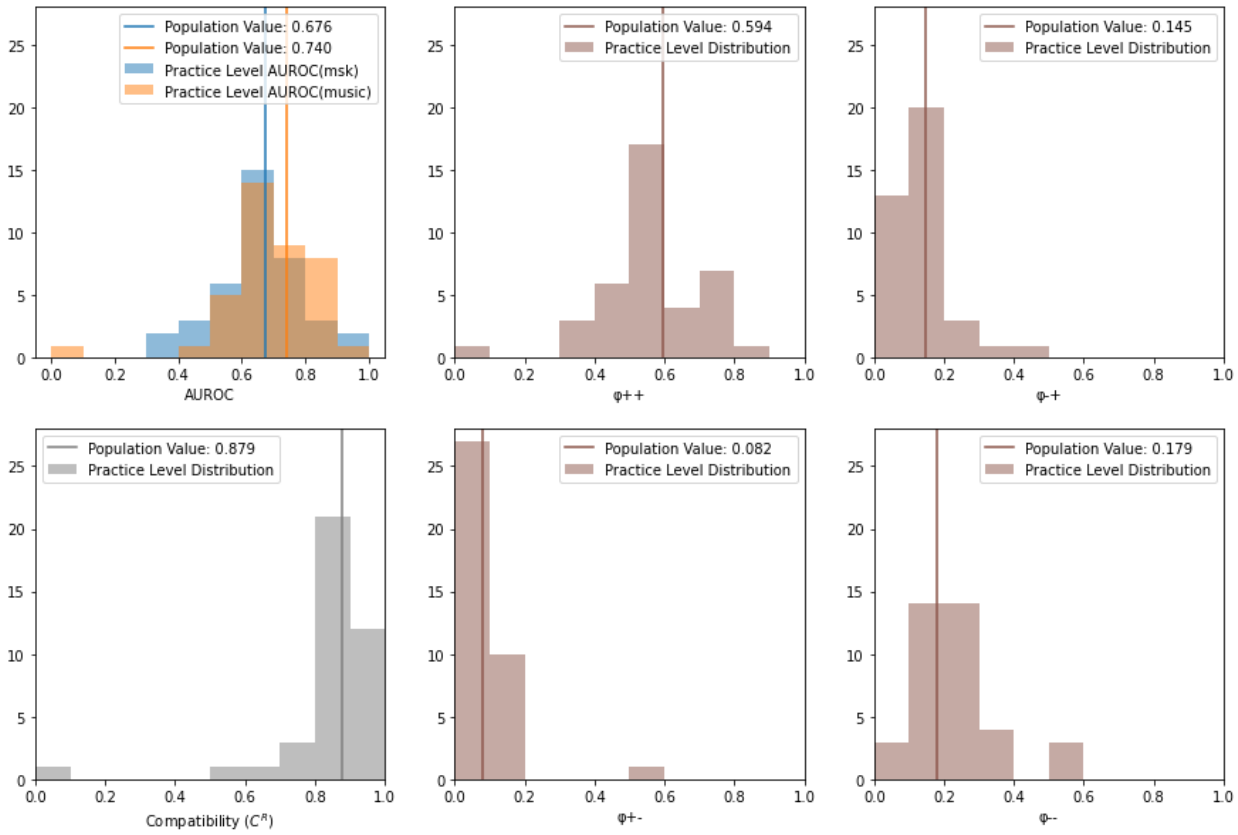


Figure 4.10: Performance, Rank-Based Compatibility, and POP Variable Values Calculated at the Individual Practice Level. In the top left, we see the discriminative performance for the two models compared in terms of the population level AUROC (vertical lines) and the distribution of the AUROCs observed by individual practices. In the bottom left panel, we depict the rank-based compatibility value for the whole population (vertical line) and observed by the individual practices (distribution). It is important to note that some practices that will see rank-based compatibility worse than the level presented at the population level. In these cases model administrators might choose to make a different decision regarding updating for these predictions. Or they may choose to use the update for these practices and counteract the problems with expectation by providing additional training or preparation for urologists at those practices. The four panels on the right depict the POP variables. As in the case of rank-based compatibility, we see that there are some practices that would experience worse modification to expectations than the population level would suggest.

4.7 Discussion

In this study, we propose the first rank-based compatibility measure. This rank-based compatibility measure functions by examining the concordance in ranking between a current model and a candidate updated model being considered for use. In addition to defining this rank-based compatibility

measure we show its connection to the discriminative performance of each of the original and updated models. This relationship suggests that increased rank-based compatibility accompanies improved discriminative performance, as the lower bound of rank-based compatibility increases as each model's discriminative performance increases. Despite the existence of this relationship, we show analytically and empirically that it is extremely unlikely to observe very high levels of rank-based compatibility through standard model development. As such, we introduce a new rank-based incompatibility loss function that can be incorporated into updated model development to control the trade-off between improvements in AUROC and against model rank-based compatibility.

We present these findings in the context of empirical work using the MIMIC-III mortality prediction task. We examined the rank-based compatibility and discriminative performance observed for standard updated model generation procedures. These results also suggest that standard approaches may provide limited rank-based compatibility benefits, further motivating the use of the incompatibility loss function. We then use the incompatibility loss function as a part of updated model development. These experiments show statistically significant improvements in rank-based compatibility. If rank-based compatibility is greatly emphasized over discriminative performance, then improvements may come at a cost by decreasing discriminative performance improvement.

In addition to the empirical work using MIMIC-III mortality, we presented a model-updating case study. This case study focused on models used to stratify patients with PCa based on their risk of having NOCD. This case study explores the implications of updating a risk stratification model by applying rank-based compatibility and related concepts.

The rank-based compatibility measure serves a different role than Bansal et al.'s [156] original backwards trust compatibility measure. Depending on the use case, one may want to use one or the other. Use cases that strongly depend on decision thresholds, such as sending a page when a patient scores above a specific cutoff, may have user mental models best represented by Bansal et al.'s [156] original compatibility measure. Alternatively, suppose a model is used without a fixed decision threshold, such as in a deterioration risk setting where the number of patients evaluated is tied to some resource constraint, *e.g.*, number of available ICU beds. In that case, this new rank-based compatibility measure may best represent clinician mental models.

Compatibility measures seek to measure the impact to user expectations given a change in model being used. User expectations can be considered a component of *trust*, which is "the attitude that [a model] will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability." [30] Trust plays an essential role in user's willingness to rely on models; it is formed through a dynamic process depending on the user, model, and situation in which they interact. [171, 172]

If an updated model with poor compatibility is implemented, users may have their expectations subverted initially. This may lead to poor team performance. [31–33] However, users will continue

to be informed over time, and they may adapt to the new model's performance. Although this adaption is expected, model developers may want to avoid the period with degraded team performance. This would be done through updated model selection focusing on high levels of compatibility.

Adaptation of user expectations over time is an example of a compensatory mechanism employed by users. [30] Users might employ a wide variety of other mechanisms. One notable example might be that users recognize an obvious failure mode of the model and can correct incorrect behavior for certain types of patients or scenarios. In this specific case, it may be beneficial to keep the updated model behavior consistent for given specific incorrect behavior. $\mathcal{C}^{\mathrm{R}}$ is not directly equipped to handle this scenario; however, the POP variables, specifically ϕ^{--}, may be employed in this case.

Compatibility measures cannot be considered a proxy for user trust, but it can be considered as representation of user's mental models which are built over time. We know that user's mental models are perturbed by updates, which can negatively impact the joint user-model team's performance. [31–33] Using our proposed incompatibility loss may help model developers create updated models that better match the mental models of clinician users. As long as incompatibility is not very heavily weighted, we can expect updated models to be generated with little to no cost in terms of improvement in discriminative performance.

There may be other ways to achieve high levels of team performance besides choosing updated models with high compatibility. The two primary mechanisms for this are education and collaboration. Education can take place after the updated model selection process, providing users with information about the differences between the original and updated models. Collaboration occurs during the model updating process and involves a dialogue between model developers and users. Users can provide model developers with their model behavior expectations and preferences. Model developers may then modify their updated model selection process to meet objectives aligned with user expectations and preferences. Additionally, model developers may inform users about changes. This process may be repeated until an updated model that satisfies users' needs is selected.

Limitations. This work is not without limitations. The first limitation relates to our analytical results, which assume that the discriminative performance is fixed and known for the evaluation dataset. While it might not be known precisely, we likely have a good sense of the model's discriminative performance based on validation data.

A related issue is the concept of dataset shift. This work's analytical and empirical findings may not hold exactly when updating in a regime in which dataset shift is occurring. The fundamental issue is the relationship between the features and the labels. If this relationship changes, both models' discriminative performances will suffer, which may affect rank-based compatibility.

Those changes may also lead to changes in clinician mental models. All together these changes make understanding the behavior of $\mathcal{C}^{\mathrm{R}}$ in settings with dataset shift difficult. This is an avenue of interest for future work.

Although we know the absolute scale of rank-based compatibility with 0 being perfect incompatibility and 1 being perfect compatibility we don't have a sense of what the numbers in between mean and how they compare across model updates. Ideally, we would like to have a sense of what is a good rank-based compatibility value, like we do with the AUROC measure (*e.g.*, a rule like: $AUROC(f) > 0.75$). This will likely come with further study of models being updated across different tasks. One advantage $\mathcal{C}^{\mathrm{R}}$ does present is that its improvements can be directly compared against improvements in AUROC by examining the POP variables.

Finally, while we discussed the different use cases for rank-based compatibility vs. Bansal et al. [31]'s backwards trust compatibility measure, we do not clearly understand clinicians' preferences. There may be update tasks that this measure is better suited for. We do not yet have a good understanding of the relationship between rank-based compatibility and user mental models. Additionally, rank-based compatibility captures the "global" user perspective. Modifying rank-based compatibility to focus on individual user perspectives may lead to better compatibility and parity with clinician expectations. We believe there is much work to do with this measure in terms of human user studies.

Conclusion. These limitations notwithstanding, we believe the rank-based measure and incompatibility loss presents a new way to think about model maintenance and updating models. Rank-based compatibility functions similar to the AUROC and extends the concept of compatibility measurement by considering the rank concordance between the output of two models. Additionally, rank-based compatibility has a direct relationship with AUROC and may be more robust to calibration shifts, a commonly observed phenomenon in healthcare. [123, 140, 158] This new measure is better suited for evaluating healthcare risk stratification models. In addition to proposing the new rank-based compatibility measure we develop a related loss function that can be used to engineer model updates, such that model developers may be able to balance improvements in discriminative performance against rank-based compatibility. In sum, this work enables the evaluation and development of model updates that would lead to better clinician-model joint performance.

CHAPTER 5

Summary

This dissertation addressed three problems across the spectrum of healthcare ML development and implementation. In **Chapter 2**, we assessed the value of longitudinal observations in relation to return to work (RTW) prediction. In **Chapter 3**, we formalized the concept of the prospective performance gap and its constituent components. Finally, in **Chapter 4**, we developed a rank-based compatibility measure and development procedure to assess and engineer updated risk stratification models. These projects advance our knowledge of ML applied for use in longitudinal healthcare settings.

We began this dissertation with a focus on ML development. *Development* is the set of processes conducted to build ML models. We targeted the occupational injury (OI) field, which uses models to predict Return to Work (RTW). The standard models used in this field are limited to making a single prediction about RTW at the time of initial injury, only using information known at that time. [47–49] The OI recovery process is dynamic, with additional information collected over time. From other healthcare ML tasks it is widely recognized that longitudinal observations improve predictive performance. [56] Thus, we assessed the value of longitudinal observations collected in the RTW setting by developing a new RTW model using claims data. We found that the inclusion of longitudinal observation data significantly improved the performance of RTW prediction.

We then switched our focus to ML implementation. *Implementation* consists of the processes related to initial model integration into clinical use, and the maintenance tasks necessary to keep the model functional over time. When models are implemented for prospective use in healthcare, they may demonstrate degradation in their predictive performance. [125] The inherently dynamic nature of the healthcare setting obscures the root causes of these prospective performance gaps. As such, we designed new methods to isolate causes of performance degradation associated with *temporal shift*, changes over time, from *infrastructure shift*, changes in deployment IT infrastructure. We applied these methods to a newly implemented CDI risk stratification model and observed that infrastructure changes contributed to performance degradation more than temporal changes. We then explored techniques to identify and mitigate specific infrastructure changes negatively impacting performance.

To overcome performance degradation due to temporal changes, model developers may update their models over time. However, once models are implemented, users begin to form expectations of the model's behavior, and updates may lead to changes that negatively impact user-model *team performance*. [31] Existing *compatibility measures*, which quantify the impact of a model update, are limited to a single decision threshold. This does not align well with model usage in healthcare, where there may be no threshold or several thresholds may be used simultaneously (*e.g.*, by different physicians). We designed a new rank-based compatibility measure and a method to create updated models that balances this measure against discriminative performance improvement. We showed how this measure might be used to assess new model updates being considered and also demonstrated the utility of our updated model creation method.

Altogether these studies tackled projects at the intersection of ML and healthcare; each underscored the vital role time plays in the context of medicine and modeling. We can utilize the findings from these studies to inform future work in this field. **Chapter 2** provided additional evidence for the utility of longitudinal observation information in healthcare ML model development. If possible, developers should seek to address predictive problems using updated information over time to represent the state of patients. **Chapter 3** demonstrated two related findings. First, although changes in patients and care practice may affect model performance over time, infrastructure differences also contribute to the prospective performance gap. Second, infrastructure differences may arise from temporal factors; without a keen understanding of clinical documentation processes, model developers may rely on information that may not be available in real-time. Finally, **Chapter 4** examined risk stratification model updating with a specific focus on rank-based compatibility. This work provides model developers with tools to assess and design updated risk stratification models that may be implemented over time.

Avenues of Future Work

This dissertation has many exciting potential avenues for future work. We provide a brief overview of possible extensions for each technical chapter.

Chapter 2. One of the most obvious avenues for future work is the continued assessment of the proposed model created for the reformulated RTW prediction task. To assess the potential value of this model, we must study it in a representative setting. This setting would involve integration with the workflows of recovery managers or physicians, as they are the key decision-makers in OI management. Such a study would ultimately entail a prospective pilot or clinical trial. In the lead-up, it would be necessary to validate the performance of the proposed model externally, using a separate dataset. For example, using workers' compensation claims from another geographic

region.

Another future direction is the continued development T2 framework we introduced. For example, this framework could be integrated with RDWs to aid model development. Wrapping T2 and its database bindings into a WYSIWYG user interface could enable physicians or other clinical users with limited technical skills to select and transform data. We could pair this infrastructure with auto-ML tools, allowing researchers to rapidly develop and test new ML models. Additionally, T2 could be expanded to enhance model portability and validation across different institutions. With the addition of model and data transformation packaging, entire pipelines could be ported relatively easily.

A third direction of future study is to explore incorporating other data. Significant factors in RTW include psychological and social state factors, which are missing from our proposed model. [51, 53, 54, 71, 72] Integration with EHRs and patient-reported outcomes may help provide this perspective.

Chapter 3. There remains a great deal of work to do in understanding and mitigating the impacts of infrastructure shift. As mentioned in **Chapter 3**, model developers and users may benefit from a "time-machine" like infrastructure. A time-machine like the one developed by Netflix [154] would allow healthcare ML model developers to access data in a manner that replicates the real-time or near-real-time data stream. This is a complex problem that depends on the foundational infrastructure of the HIT technology stack. Work may need to be done by EHR vendors: Epic or Cerner. Third-party vendors that develop data transfer, storage, and interoperability tools, like Redox, may also be able to address the gap. Successfully addressing infrastructure shift will involve careful re-design of data systems. Modern HIT systems must support both the operational work of medicine and the computational work needed to integrate ML properly into clinical care.

Aside from foundational infrastructure work, additional theoretical work focused on detecting data elements at risk for infrastructure and temporal shifts would be valuable extensions of this work. Detection may help model developers avoid certain features or enable them to be closely monitored. Additionally, this may help develop our understanding of the nature of these shifts and their behavior in different predictive tasks and settings. It will be necessary to build software packages to enable the careful custom analysis presented in **Chapter 3** to be replicated at scale.

Chapter 4. There are many potential extensions of our rank-based compatibility work. The most obvious next step would be to study how model updating impacts clinical user expectations. We may find that clinical users do not experience the same performance degradation when updates do not meet their expectations. A human subject study would also enable us to understand how well the rank-based compatibility measure aligns with real human expectations. It would be ideal to

study this in a clinical setting with clinical users. However, it may be dangerous to conduct this study in a manner that would impact clinical care. Thus a simulation or a survey methodology may need to be employed. Either way, this study would require many resources and careful design. Additionally, it would involve a time frame beyond the scope of this dissertation. This work might be suitable for a longer-term study supported by an NIH grant.

Another direction of future work would be to understand how model updates may be made over time. The model updating decision is not singular; instead, it can be thought of as a process where models are trained, implemented, and updated continuously over time. This process's goal is always to use the model that best supports the user-model team performance. Suppose model developers can update models over time. In that case, the decisions about which models to choose and when to update them will depend on compatibility and other factors, like performance and implementation costs. Further study is needed to uncover the best way to drive this decision-making process.

There are additional methodological extensions of this work to consider. The rank-based compatibility and its updating methodology are focused on global compatibility. By focusing on individual users and their experience with the model, there may be additional flexibility in generating compatible updated models.

Another exciting area of future direction is to clarify and optimize the role of model users, model implementers, and model developers. Currently, the roles and relationships between these parties are unclear. Clear governance structures and roles may help to improve model development and implementation processes. Moreover, it may also help ensure that models are designed and function in the desired manner. We present the task of monitoring model performance as an example.

We argue that the task of model performance is the responsibility of at least three parties, the model developers, the model implementers, and the model users. Each of these parties has differing access and perspectives on the implemented model. The model developer likely has the best initial understanding of the model's functionality and may be best suited for identifying different failure modes. For example, in retrospective validation studies, the model may show weak performance in specific subpopulations. Thus, the model's performance on these subpopulations should be monitored prospectively.

The model implementer may have the best understanding of the infrastructure on which the model runs prospectively. They may be responsible for run-time monitoring issues. For example, during technical integration testing, the model implementer may note that the model takes a long time to run. If infrastructure is modified this may mean that some patients may not receive predictions from the model because of extended model run times.

Finally, users should be encouraged to monitor model performance and report issues they ob-

serve. Clinical users have the greatest chance of catching model mistakes as they happen. Even if model users do not fully understand the underlying model, they should be able to raise patient safety issues or suggest ways for models to be improved. Ultimately, if users do not believe a model to be well suited for their clinical use, they may choose to ignore its predictions. In order to avoid this, we must address user performance concerns. We can enable this by building strong feedback loops from model users to model developers.

Conclusion

We look forward to building upon the work presented in this thesis; the three technical chapters provide a foundation to explore the intersection of ML for healthcare further. These chapters each contribute to the body of knowledge in this area. **Chapter 2** assessed the value of longitudinal observations in the context of RTW prediction. **Chapter 3** formalized the concept of the prospective performance gap and its constituent components. **Chapter 4** presented a rank-based compatibility measure and development procedure to assess and engineer updated risk stratification models. Although they focused on developing and implementing ML models for different clinical prediction tasks, they all share a particular focus on the longitudinal nature of medicine. We believe there is still much to explore about how healthcare ML models act and react to the inherently temporal nature of medicine. This work has the potential to increase the efficiency of healthcare systems, aid physicians, and most importantly, improve patient care.

APPENDIX A

Appendix For Chapter 2

A.1 Details for Main Study

In this section, we provide additional details for the main study presented in **Chapter 2**.

A.1.1 Problem Setup & Related Work

Return to Work Literature

As mentioned above, in the United States, OIs affect millions of patients annually and cost hundreds of billions of dollars. The actual burden of these injuries is likely to be significantly underestimated. [34–36, 173] In addition to physical symptoms, patients with OIs often experience complicating psycho-social issues, like depression However, these issues are rarely detected or treated. [174] Together, these factors incentivize patients, workplaces, physicians, and payers to understand the amount of time a patient will be away from work to help minimize it eventually.

There have been many retrospective studies that seek to identify factors affecting RTW. Significant factors include patient demographics, injury-related, professional, workplace-related, treatment, and psycho-social factors. [51, 53, 54, 71, 72] Examples of specific elements are highlighted in **Table A.1**. Predictors of shortened RTW duration include job control, work ability, perceived (good) health, and high socio-economic status. Some predictors of lengthened RTW duration include job strain, anxiety/depression, comorbidities, older age, and educational attainment. [73, 74]

These studies provide a view into how RTW is shaped by various factors across patients, workplaces, and injuries. However, the findings from these studies cannot easily be generalized across large populations of injured patients. This is due to 1) a specific focus on injury subpopulations and 2) the use of specially collected data. The focus on specific injury subpopulations, such as patients who experience a lumbar disc herniation, prevent findings from generalizing across the population of OIs. [72] Specially collected data, like many of the variables presented in **Table**

Table A.1: Factor groups and specific factors that are related to RTW duration.

Factor Group	Specific Examples
Patient Demographics	Age, Functional status, Medical comorbidities
Injury-Related	Injury severity, Body region affected, Number of hospitalizations, Work ability
Professional	Level of education, Type of work, Union membership, Compensation
Workplace-Related	Workplace arrangements, Physical demands, Perception of injury relatedness to work, Job control, Job strain
Treatment	Opioid prescriptions
Psycho-social	Self-efficacy, Recovery-expectations, Mental health comorbidities, Perceived health, Socio-economic status

A.1, must be collected from patients, providers, or workplaces with special research workflows. [51, 53, 54, 71, 72]

RTW modeling has traditionally taken the form of a time-to-event prediction task. Much of the modeling work done in this field treats RTW as a single event. In this setup, model developers seek to predict the time of RTW when a patient is initially injured. The most prevalent modeling technique used for this approach is the Cox proportional hazards model. There have been examples of time to RTW using hazard models, with slight modifications to predict the length of receiving benefits and to identify prolonged claims. [51–54]

ML techniques, like decision trees, naïve Bayes, and gradient boosted machines, have been used for problems related to RTW. They are not routinely used for the prediction of RTW, which is dominated by the time-to-event approach, and the major work is focused on ancillary prediction tasks. Two examples include appropriate rehabilitation intervention selection, where ML techniques out-perform clinicians, and classification of patient final disposition (*e.g.*, eventually return or never return). [175, 176]

Even though ML is not heavily utilized for RTW prediction, it has seen increased usage in the greater field of OI, specifically for use in automated injury coding. For example, ML models have been used in construction-related injuries to retrieve injury etiology from free-text reports automatically. [177] These models can potentially augment human-based injury surveillance systems, classify injuries and intervention categories, and guide prevention efforts and policy. [178–180]

From this existing literature, we note that RTW prediction has several potential avenues for

further exploration. First, models are generally made for specific diseases with custom collected data. Second, is that RTW models are generally based on static time-to-event prediction, designed for usage just at the time of injury, and incapable of handling newly observed information.

Modeling specific injuries through custom research databases helps physicians to refine their understanding of patient recovery from those injuries. However, it limits the overall utility of models. We seek to build a model that can be used for the multitude of OIs that patients experience, so we must employ a dataset representative of this variety. This dataset must be relatively universal in terms of its availability and its representation of patient injuries and recoveries. Gross et al. [175]'s work on ML-assisted rehabilitation intervention selection utilizes data from an administrative database. [175] Statewide administrative databases of workers' compensation claims represent a potential avenue for accessible and routinely collected data regarding patient injuries. [181] These databases have been shown to have high concordance with BLS occupational injury statistics and thus are a source of relatively high-quality large-scale data. [182]

RTW duration predictions made at the injury onset are helpful for patients, workplaces, physicians, and payers. This information helps set patients expectations, allows workplaces to plan, and helps physicians and payers categorize patients and plan for eventual resource usage. However, the value of this information degrades over time. Plans made with initial predictions must be updated without the guidance of validated models, and there are no tools to directly compare the trajectory of a patient currently recovering to that of historical patients.

We could alleviate the issues by employing RTW prediction models that update over time. Barriers to creating dynamic models for patient conditions have included small data-set sizes, methodological constraints, and insufficient hardware. However, these constraints have recently been overcome. [58] Recently, several related dynamic prediction models have been published, helping physicians to screen for traumatic brain injuries, assess risk factors for recovery from non-work-related injury, and predict the need for hospitalization in pediatric asthma exacerbations. [183–185]

Sequence-to-Sequence Learning Models

We seek to present a new approach to modeling RTW prediction that can be used dynamically, unlike the existing static time-to-event framework. This new approach would treat the input as a sequence of information and the outputs as a related sequence of information. The input sequence is all of the longitudinal observations collected on a patient. The output sequence can take several forms, either directly predicting the time-to-event of RTW or estimating probabilities of RTW at future time points. Thus, we have two sequences. The desired task is to sequentially predict the outcome sequence given the observation sequence, a sequence-to-sequence prediction task.

Markov chains are a well-studied modeling technique for sequences consisting of a series of

states, s_t, over time, t. A Markov Chain is a stochastic process that enforces a conditional distribution between the current state, s_t, and the future state s_{t+1}. It assumes that given the current state, s_t, the future state, s_{t+1}, is independent of all past states $s_0, s_1, ..., s_{t-1}$. This means that future states are only dependent on the current state. [77, 186, 187] Markov models are widely used in medicine due to their elegant structure and ease of clinical interpretability. They have been used to model immune responses, cancer outcomes, and to analyze the histories of patients with strokes. [188–192]

A notable extension of the Markov chain is that of the hidden Markov model (HMM), which enables the modeling of a sequence of observed signals generated from an unobserved (hidden) series of states. HMMs are frequently used to study sequences generated from systems with stochasticity. They have been used throughout medicine, from studying protein sequences, analyzing human movement, and predicting treatment decisions. [75, 193–195] For a more thorough review of the theory and application of Markov models to problems of longitudinal data analysis, see Bartolucci et al. [78].

Despite their wide use, Markov chains are limited by their underlying formulation, which restricts the sequential dependence of s_{t+1} to only s_t. More complicated processes can be transformed into a Markov chain formulation by redefinition of the states. [186] Thus, fixed-length histories can be embedded into the current state, allowing for the representation of history by state-space expansion. Time-homogeneity is often a fundamental assumption, as the probability of transitioning to s_{t+1} depends only on s_t, independent of the current time-step unless this dependence is represented using the state-space.

The observations for patients returning to work are of exceptionally high cardinality, as they include several types of categorical variables that may take on many possible values. An example of this is diagnosis. There are thousands of possible diagnoses for patients injured at work. Additionally, on any given day, patients may have zero, one, or more diagnosis codes assigned to them. Treatment is another high-cardinality category. The timing and order of treatments may impact the recovery of an injured patient. Thus, history beyond the current observation may prove important in modeling RTW. This high cardinality and history dependence make Markov models ill-suited for the task. We could reduce this cardinality by restricting the problem definition to a specific disease (*e.g.*, lower back sprain). Or we could use category groups to lower the cardinality of high cardinality categories. However, both these approaches demand a great deal of clinician time. Clinicians may need to manually group categories or suggest reasonable ways to reduce the scope. Additionally, this would counter our goal of creating a general model. Instead, we have turned to a series of techniques from deep learning to help address these challenges.

Deep Learning offers a set of alternatives to Markov models. Deep learning methods have gained popularity in recent years due to improved hardware performance, the ubiquity of large

datasets, and the availability of high-quality deep learning frameworks, such as TensorFlow and PyTorch. [88, 196] RNNs are a class of deep learning based sequential prediction modeling techniques. RNNs can selectively store information in a vector called the hidden state. They can pass the hidden state to future time steps and update it as needed. [57] According to Goodfellow "RNNs are useful when we believe that the distribution [of the outcome] may depend on a value of [an input] from the distant past in a way that is not captured by the effect of [a one-step transition]." [58] They may also be used for sequence labeling. [59]

Thus, RNNs have desirable properties that may increase potential model performance compared to other approaches. RNNs can model long-range dependencies as the hidden state generated can store any observed information instead of being limited to the previous time step. They can also express a larger hidden state space than Markov chain based models. [60] RNNs have been very successful in speech recognition and natural language processing. [197] While not as pervasive as Markov chain based models in medicine, they have been successfully used to predict heart failure onset and events in clinical event occurrence. [198, 199] One notable recent project utilizing RNNs, was a study conducted by the Google Deep Mind Health team that created a model to dynamically predict acute kidney injury in hospitalized patients. [200]

RNNs can become increasingly difficult to train due to error signals vanishing or exploding when conducting training. This problem is exacerbated by the length of time between signals. [201, 202] Though this is not a problem for every application of standard RNNs, there exist several modifications of standard RNNs to help overcome these issues: Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). LSTMs include alterations in how the hidden state is computed, allowing models to explicitly forget previous information and input new information. [203–205] GRUs share this modification; however, the forget/input operations are combined, reducing the number of parameters in the model, thus potentially making them easier to train. [206–208]

Deep learning allows models to learn feature representations as a part of model training. A deep learning feature representation technique called *word embeddings* enables models to naturally learn groups of category values when handling high cardinality categorical values. [79–83]

Categorical values are often embedded in a fixed-size vector, where the components are binary. One hot encoding is a particularly popular approach as it ensures that each category is treated independently. Unfortunately, these approaches scale linearly with the category dimension and do not allow for learning encoding of knowledge between categories. Word embeddings map category values to real-valued vectors, which can be updated throughout training. After training, similar categories will be closer to one another in this representation space than other categories. Once learned, embeddings can be reused and interrogated for their representational meaning.

A.1.2 Methods

A.1.2.1 Data Variable Details

Characteristic Data. Each injury, i, has a vector of characteristics, $\mathbf{c}_i$, composed of a real characteristic data vector $\mathbf{c}_i^R$ and a categorical data encoding vector $\mathbf{c}_i^C$. For each injury i, $\mathbf{c}_i^R$ represents a vector of real numbers of size $d_{\mathbf{c}_i^R}$; $\mathbf{c}_i^C$ represents a vector of positive integers encoding characteristic information from high-cardinality categorical data. As a part of the overall learning process, representations of these categories will also be learned. We have employed these high-cardinality encodings to represent job codes, of which there are hundreds. The choice to encode data as high-cardinality categories is left to the model developer; however, the number of features encoded in this manner is denoted as $d_{\mathbf{c}_i^C}$. The characteristic vector, $\mathbf{c}_i$, is the concatenation of $\mathbf{c}_i^R$ and $\mathbf{c}_i^C$ for each worker, $\mathbf{c}_i = (\mathbf{c}_i^R, \mathbf{c}_i^C)$.

Longitudinal Observation Data. For every injury i and time-step t there is a vector of longitudinal observations, $\mathbf{o}_{i,t}$. Three components may be represented by the longitudinal observation vector: a real observation data vector $\mathbf{o}_{i,t}^R$ of size $d_{\mathbf{o}^R}$, a high-cardinality category observation vector $\mathbf{o}_{i,t}^C$ of size $d_{\mathbf{o}^C}$ and the current work status value $\mathbf{o}_{i,t}^W$ (optional). To highlight the utility of other observational data, we chose not to include the current health-state in the observation data for this work. Note: the dimension of these observation vectors is constant across time. Like the high-cardinality characteristic data, representations of these categories will be learned through the training process. Examples of observational data encoded as high-cardinality categories are diagnoses and procedure codes. Each of these categories have thousands of unique values. [209] We denote vector $\mathbf{o}_{i,t}$ as the concatenation of $\mathbf{o}_{i,t}^R$ and $\mathbf{o}_{i,t}^C$ for each worker, $\mathbf{o}_{i,t} = (\mathbf{o}_{i,t}^R, \mathbf{o}_{i,t}^C)$.

A.1.2.2 Evaluation

area under the receiver operating characteristic curve: computed on a per prediction day basis. Experiments that reweighted predictions such that every injury contributed equally (i.e. each prediction is weighted $1/(\text{injury case duration})$) yielded similar results in preliminary experiments.

Expected calibration error: computed by calculating a calibration curve with 5 uniformly sized bins, which we then grouped the daily predictions into, and then calculated the fraction of those instances positive (future work status = Working). These were used to generate calibration curves. The mean squared error of these curves were used to compute their ECE.

Window-level Evaluation: The daily predictions were used to calculate performance measures. We use this window-level approach (also known as the time-horizon approach [5]) as model users

can intervene on patients daily. In **Figure A.1**, we show an example of how this calculation takes place.

	Patient (i)	Day/Timestep (t)	Future Work Status ($y_{i,t}$)	Estimated Probability ($\Pr(y_{i,t}=1)$)
	1	1	0	0.10
	1	2	0	0.13
	...	...	...	...
	1	55	1	0.69
	1	56	0	0.66
	2	1	0	0.17
	2	2	0	0.19
	...	...	...	...
	2	73	1	0.73
	2	74	1	0.74
	3	1	0	0.45
	3	2	1	0.51
	...	...	...	...
	3	29	1	0.79
	3	30	1	0.81
...	...	...	...	...
	99	1	0	0.10
	99	2	0	0.13
	...	...	...	...
	99	47	1	0.69
	99	48	0	0.66
	100	1	0	0.17
	100	2	0	0.19
	...	...	...	...
	100	101	1	0.73
	100	102	1	0.74
	101	1	0	0.45
	101	2	1	0.51
	...	...	...	...
	101	67	1	0.79
	101	68	1	0.81

Figure A.1: Example Window-Level Evaluation. All daily predictions for the patients in the evaluation dataset are aggregated into a vector (right-most column) and compared against the ground truth future work status vector (second column from the right). All daily predictions contribute equally to the calculation of the evaluation measures.

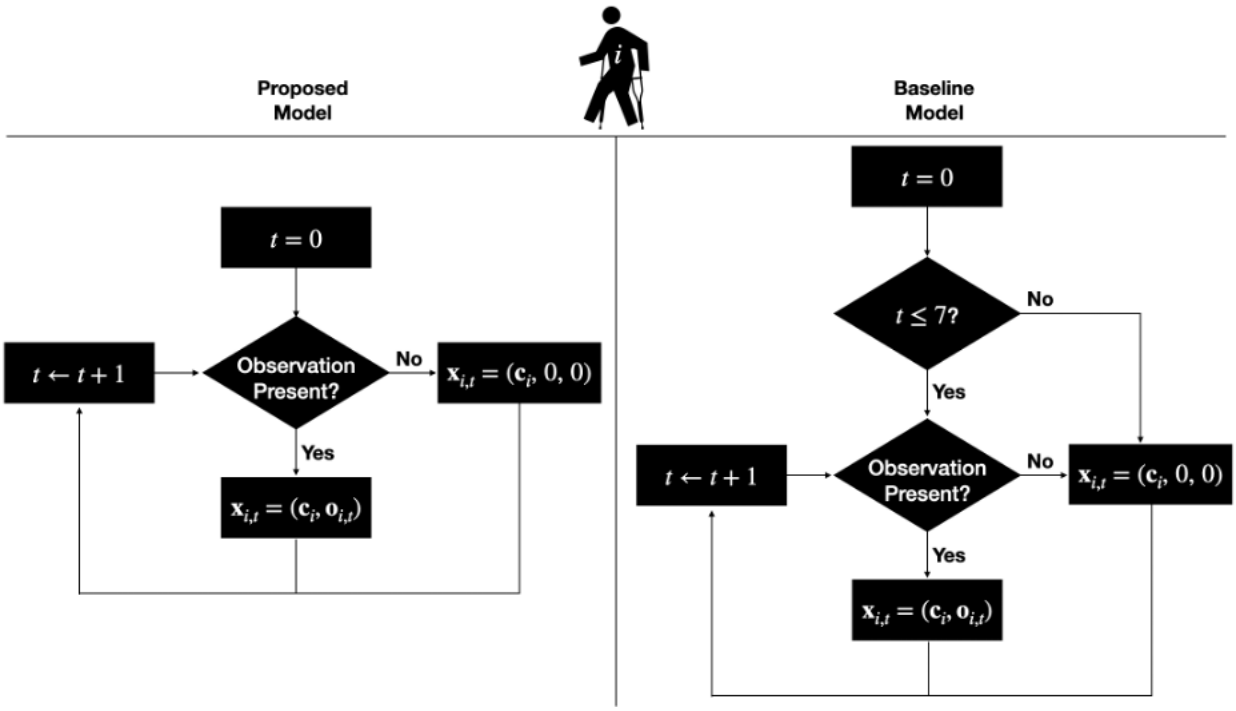


Figure A.2: Flowchart Showing the Relationship Between Time and Longitudinal Observations. For a given patient, i, at time-step, t, we generate $\mathbf{x}_{i,t}$ for every time-step using characteristic, $\mathbf{c}_i$, and observation data, $\mathbf{o}_{i,t}$. Let's say that categorical two values are observed over time representing diagnosis and procedure, $d_(\mathbf{o}^R) = 2$. We now highlight the difference between the proposed and baseline model. For the proposed models, the input if there were no observations observed on a given time-step, t, we have $\mathbf{o}_{i,t} = (0, 0)$, which represents two missing observations. Because the baseline model represents approaches currently in use, where predictions are based on information only available near the time of injury, we need some mechanism to replicate this. Thus, the baseline model has an additional check. If the time-step is after the 7th, day we automatically censor all the observations with the missing observation vector.

A.1.3 Experiments & Results

A.1.3.1 Experimental Setup

Data preprocessing: Observations (Diagnoses & Procedures). The raw injury claims data has information about diagnoses and information about procedures. Both sets of information have dates and codes denoting the procedure and diagnostic information observed for a worker's injury. In addition to the dates and procedure codes, the raw procedure data also contains procedure diagnosis codes. These diagnoses that justify the rationale for ordering (thus and billing for) the procedure code. When encoding the procedure information, we intentionally stripped the procedure diagnosis codes as these are unconfirmed diagnoses, and their addition adds computational complexity to the model.

We only used the diagnosis codes that came from the raw diagnosis data. These diagnosis codes may have different observation timestamps than the procedures. These codes generally are documented after procedure codes and usually after the first week of an injury occurring. This explains why we observed 0 for the number of diagnoses observations in the baseline model.

Data preprocessing: TemporalTransformer. Observation and Characteristic data were processed by the TemporalTransformer package (GitHub). The configuration used to process the injury claims data is shown in **Section A.3.2**. The transformation was conducted with the following settings: no data filtering (`fit_filter_percentile_cutoff = 0.0`) and a high-cardinality category channel size of 1 (`has_time_hdc_channels = 1`). This second setting dictated that if multiple of the same type of high-cardinality observations (e.g., multiple diagnoses) were observed on the same day only, 1 would be passed to the model. This observation was chosen randomly.

High-cardinality category embeddings. Embeddings were configured based on the training set. We mapped high-cardinality category values to integer indices (from 1 to n, where $n = cardinality(category)$). Two additional index tokens were added, 0 and $n + 1$, enhancing the observation representations. The 0 index token represents a missing observation (*e.g.*, no observation of that high-cardinality category at the time-step for the given injury). In addition to preserving the structure of inputs for the neural network, it also allows the proposed model to assign meaning to missing observations. The 0 token was used to censor observations beyond the seventh day for the baseline model. This usage of the missing observation token depicted in **Figure 4 A.2**.

The $n + 1$ index token represents an observation value of "other." During training, "other" may be a catchall for extremely infrequent category values. And during testing, category values that have never previously been observed will be replaced with "other." Since we did not use the filter setting of TemporalTransformer, this "other" token was not employed during training. It was used at inference time, potentially slightly negatively impacting performance

High-cardinality category embeddings: we utilized the quarter-power rule of thumb [83] to automatically size the real valued embeddings based on the cardinality of the category. The number of index tokens $(2 + cardinality(category))$ is the input dimension of the embedding. We raised this to the quarter power, rounded, and added 1 to determine the embedding's output dimension (the dimension of the embedding each category value maps to).

$$\text{output dimension} = \text{round}(\text{input dimension}^{1/4}) + 1$$

In **Table A.2**, we catalog the input and output dimensions of the embeddings used in our approach.

Table A.2: Model High-Cardinality Category Embedding Information. For each high-cardinality category, we show the observed input dimension (from the training dataset), the output dimension, and the total number of parameters needed for this embedding. The input dimension is $2 + cardinality(category)$. For example, in the training dataset, TemporalTransformer observed 564 unique job code values. It then added 2 special category index-tokens, for a total input dimension of 566. Using our modified quarter-power rule, TemporalTransformer determined the 6 as the ideal output dimension for the job code category. Multiplying the input dimension with the output dimension yields $3,396$, the total number of parameters needed to embed the job code category.

High-cardinality Category	**Input Dimension**	**Output Dimension**	**Embedding Parameters**
Job Code	566	6	$3,396$
Diagnoses Code	$1,679$	7	$11,753$
Procedure Code	$3,832$	9	$34,488$

Hyperparameters We conducted an extensive hyperparameter search using the training dataset for both the proposed and baseline models. We used the Hyperband hyperparameter search method. We optimized hyperparameter selection based on the AUROC on the development set. [86] For Hyperband search, we used a factor of 3 and a maximum of 8 training epochs for each trial. Both models had access to the same hyperparameter space, shown in **Table A.3**.

After hyperparameter selection, each model was then trained for up to 30 epochs (there was potential for early stopping as we set epoch improvement patience to 5). The best-performing model in terms of development dataset AUROC performance was saved at the end of each epoch. This procedure yielded the final proposed and baseline models used in the evaluation.

Table A.3: Hyperparameter Search Values. Several hyperparameter search values were layer dependent and conditional on other search values. These are denoted with *. For example, the size and activations of each layer in $f_{out}(\cdot)$ (the out sub-model) are determined independently after number of Out Layers is determined. The only limitation to this scheme was that the RNN layers did not have an activation search – they utilized the default Keras LSTM layer activation.

Hyperparameter	**Search Values**
Activations*	{linear, relu, elu, tanh}
Widths*	$\{16, 32, 64, 128\}$
Initial Dropout	$\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$
Hidden Layer Dropout	$\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$
Number of In Layers	$\{0, 1, 2\}$
Number of RNN Layers	$\{1, 2\}$
Number of Out Layers	$\{0, 1, 2\}$
Learning Rate	$\{1e-2, 1e-3, 1e-4\}$

A.1.3.2 Population Characteristics Details

Table A.4: Top 10 Most Frequent Job Codes. The entire dataset contained 595 unique job codes. The number of injuries with each job code is shown, along with the count column and percentage of injuries. Note that over 10% of the injuries occurred in workers with jobs tied to a municipality (city, county, and school districts). Restaurant, healthcare, and manufacturing jobs are among the top 10 most frequent job codes. Job Descriptions were found by searching for Ohio-specific NCCI or job codes. We conducted this search in June 2021.

Job Code Description	**Job Code**	**Count**	**%**
City employees*[*,1]	9431	13,604	4.63%
Restaurants[2]	9082	11,843	4.03%
Local school districts[*,3]	9434	9,675	3.29%
Nursing Or Convalescent Home–all Employees[4]	8829	9,569	3.25%
Automobile–service Or Repair Center & Drivers[5]	8380	9,011	3.06%
Restaurant: Fast Food[6]	9083	8,151	2.77%
County employees[*,7]	9430	6,802	2.31%
Machine Shop Noc[8]	3632	6,655	2.26%
Metal Stamped Goods Mfg. Noc[9]	3400	5,604	1.91%
Plastics Mfg.–laminated Molded Products Noc[10]	4484	5,517	1.88%

[*] Denotes "all employees & clerical, clerical telecommuter, salespersons, drivers."
[1] https://www.bwc.ohio.gov/downloads/blankpdf//oac4123-17-72appendixe.pdf
[2] https://www.workerscompensationclasscodes.com/2017/11/30/restaurant/
[3] https://www.bwc.ohio.gov/downloads/blankpdf//oac4123-17-34appendixa.pdf
[4] https://www.insurancexdate.com/class/OH/nM5X/nursing-or-convalescent-home-all-employees
[5] https://www.insurancexdate.com/classreport.php?search=8380&state=OH
[6] https://www.insurancexdate.com/class/OH/5g0y/restaurant-fast-food
[7] https://www.bwc.ohio.gov/downloads/blankpdf/oac4123-17-72appendixe.pdf
[8] https://www.insurancexdate.com/class/OH/rXM5/machine-shop-noc
[9] https://www.insurancexdate.com/class/OH/VglD/metal-stamped-goods-mfg-noc
[10] https://www.insurancexdate.com/class/OH/ggM9/plastics-mfg-laminated-molded-products-noc

Table A.5: Top 10 Most Frequent Diagnosis Codes. The entire dataset contained 2,292 unique diagnosis codes. The number of observations with the diagnosis code is shown, along with the count and percentage of all diagnosis observations (403,931 in total). Back sprains account for over 18% of all diagnoses, and hand wounds account for over 10%. Leg, knee, ankle, and foot sprains were also in the top 10 most frequent diagnosis codes. Diagnosis Code Descriptions were found by searching for www.icd9data.com for the ICD-9 diagnosis codes. [210] We conducted this search in June 2021.

Diagnosis Code Descriptions	**Diagnosis Code**	**Count**	**%**
Open wound of finger(s)[1]	883	35,859	8.77%
Sprain of lumbar[*,2]	847.2	24,096	5.89%
Sprains & strains of other and unspecified parts of back[3]	847	14,029	3.43%
Sprains & strains of unspecified site of shoulder & upper arm[4]	840.9	13,306	3.25%
Sprains & strains of sacroiliac region[5]	846	12,863	3.15%
Sprain of thoracic[*,6]	847.1	12,368	3.02%
Sprains & strains of ankle & foot[7]	845	11,296	2.76%
Sprains & strains of unspecified site of knee & leg[8]	844.9	10,480	2.56%
Open wound of hand except finger(s) alone[9]	882	10,016	2.45%
Contusion of knee[10]	924.11	9,912	2.42%

[*] Denotes sub-category of "Sprains and strains of other and unspecified parts of back."
[1] http://www.icd9data.com/2012/Volume1/800-999/880-887/883/default.htm
[2] http://www.icd9data.com/2012/Volume1/800-999/840-848/847/847.2.htm
[3] http://www.icd9data.com/2014/Volume1/800-999/840-848/847/default.htm
[4] http://www.icd9data.com/2014/Volume1/800-999/840-848/840/840.9.htm
[5] http://www.icd9data.com/2012/Volume1/800-999/840-848/846/default.htm
[6] http://www.icd9data.com/2014/Volume1/800-999/840-848/847/847.1.htm
[7] http://www.icd9data.com/2013/Volume1/800-999/840-848/845/845.htm
[8] http://www.icd9data.com/2014/Volume1/800-999/840-848/844/844.9.htm
[9] http://www.icd9data.com/2014/Volume1/800-999/880-887/882/default.htm
[10] http://www.icd9data.com/2014/Volume1/800-999/920-924/924/924.11.htm

Table A.6: Top 10 Most Frequent Procedure Codes. The entire dataset contained 5,003 unique procedure codes. The number of observations with the procedure code is shown, along with the count and percentage of all procedure observations (4,482,683 total). Physical medicine codes account for over 28% of all procedure. Office visits, emergency department visits, and chiropractic procedure codes are also present in the top 10 most frequent procedure codes. Procedure Code Descriptions were found by searching for www.aapc.com for the CPT procedure codes. [211] We conducted this search in June 2021.

Procedure Code Description	**Procedure Code**	**Count**	**%**
Under Physical Medicine and Rehabilitation Therapeutic Procedures[1]	97110	445,020	9.93%
Special Procedure Code Character "N"[2]	N	303,371	6.77%
Under Supervised Physical Medicine and Rehabilitation Modalities[3]	97014	293,818	6.55%
Under Established Patient Office or Other Outpatient Services[4]	99213	243,133	5.42%
Under Chiropractic Manipulative Treatment Procedures[5]	98940	241,370	5.38%
Under Supervised Physical Medicine and Rehabilitation Modalities[6]	97010	205,972	4.59%
Under Physical Medicine and Rehabilitation Therapeutic Procedures[7]	97140	170,421	3.80%
Under Constant Attendance Physical Medicine and Rehabilitation Modalities[8]	97035	165,007	3.68%
Under New or Established Patient Emergency Department Services[9]	99283	150,718	3.36%
Under Established Patient Office or Other Outpatient Services[10]	99212	113,283	2.53%

[1] https://www.aapc.com/codes/cpt-codes/97110
[2] No token provided by dataset – this was processed as a separate procedure code as we did not restrict inputs to a specific code set
[3] https://www.aapc.com/codes/cpt-codes/97014
[4] https://www.aapc.com/codes/cpt-codes/99213
[5] https://www.aapc.com/codes/cpt-codes/98940
[6] https://www.aapc.com/codes/cpt-codes/97010
[7] https://www.aapc.com/codes/cpt-codes/97140
[8] https://www.aapc.com/codes/cpt-codes/97035
[9] https://www.aapc.com/codes/cpt-codes/99283
[10] https://www.aapc.com/codes/cpt-codes/99212

A.2 Simpler Model Architectures

The deep learning architecture we explore in this study is capable of learning complex temporal relationships. However, this complexity comes at a cost in terms of interpretability. Specifically, it can be difficult to understand the importance of features. Using a simpler model architecture, we can examine the importance of the individual features. This feature examination enables us to tease apart the contribution of characteristic vs. longitudinal observation features.

A.2.1 Methods

This section examines two simpler model architectures: 1) L2-penalized logistic regression and 2) random forest regression.[212, 213] Logistic regression presents one of the most straightforward architectures possible, allowing us to directly examine coefficient values to understand the importance of various features. Random forest regression represents a middle ground with additional complexity. A sense of its behavior can be established by directly examining some of the component trees of the random forest ensemble.

Training these models requires a deviation from the formulation presented in the main methods. The primary deviation is the removal of the encoded history vector $\tilde{\mathbf{h}}_{i,t}$. Building this encoded history introduces significant complexity to the model, reducing our ability to examine the direct impact of individual features. Instead, we use a simpler architecture that directly maps a modified encoding of the worker's input features, $\tilde{\mathbf{x}}'_{i,t}$, to the probability of the worker's future work status, $y_{i,t}$, formally: $\Pr(y_{i,t} = 1) = f_{additional}(\tilde{\mathbf{x}}')$. This architecture modifies how we encode the input feature vector and abandons the encoded history vector $\tilde{\mathbf{h}}_{i,t}$. To create the modified input feature encoding $\tilde{\mathbf{x}}'_{i,t}$ we re-scale real values to range between 0 and 1 (converting them from their standard normally scaling). We modify category encoding so that all categories are encoded with one-hot encoding (we contrast this with $\tilde{\mathbf{x}}_{i,t}$, which employs word embeddings for high-cardinality categories).

The changes to the input feature encoding and removal of the encoded history vector changes enable us to understand the impact of specific features more directly. However, these changes may come at a cost to predictive performance.

A.2.2 Experiments & Results

Questions. We examine two related questions with these simpler model architectures.

1. *Does the utilization of longitudinal observations by the proposed model provide any benefits in the prediction of future work status?* (**Section A.2.2.2**, **Figure A.3**)

2. *Which features are the most important for the simpler model architectures utilizing longitudinal observations?* (**Section A.2.2.3**, **Table A.9**, **Figure A.4** and **Figure A.5**)

A.2.2.1 Experimental Setup

We use the same train/validation/test dataset split and the same offset value of 7 days ($\phi = 7$). We then trained L2-penalized logistic regression and random forest models using the training dataset. The validation dataset was used to aid in the hyperparameter grid search. The hyperparameter search space for both model architectures is displayed in **Table A.7**. As in the primary methods, we train "baseline" versions of these models, which only utilize characteristic data and do not include longitudinal observations.

Table A.7: Simple Model Architecture Hyperparameter Search Values.

Model Architecture	Hyperparameter	Search Values
Logistic Regression	C	$\{1E-4, ..., 1E4\}$
Random Forest	Max Depth	$\{2, 5, 7, 9\}$
	N Trees	$\{5, 10, 15, 20\}$

A.2.2.2 Performance

The predictive performance for the best models found for both model architectures are displayed in **Figure A.3**. We note that the discriminative performance of all the simpler model architecture is smaller than the model proposed in the main methods. Additionally, we note that for each model architecture, the models utilizing the longitudinal observations outperform the baseline models, which only use the static characteristic data.

The performance results for these simpler model architectures also suggests that the longitudinal observation information may be valuable for predicting future work status. However, this analysis does not directly tell us the value of the longitudinal data.

A.2.2.3 Feature Importance

To understand the importance of the features, we conduct two additional analyses using the simpler model architectures. The first involves examining the learned model, and the second consists of a feature evaluation technique.

We examine the learned models, $f_{additional}$ in terms of their direct inputs, the modified input feature encoding, $\tilde{\mathbf{x}}'_{i,t}$, composed of $4,868$ encoded features. For the logistic regression architec-

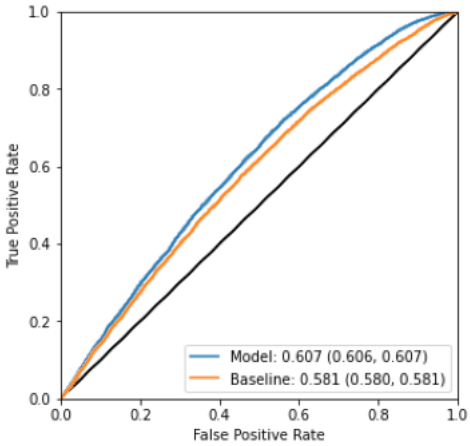
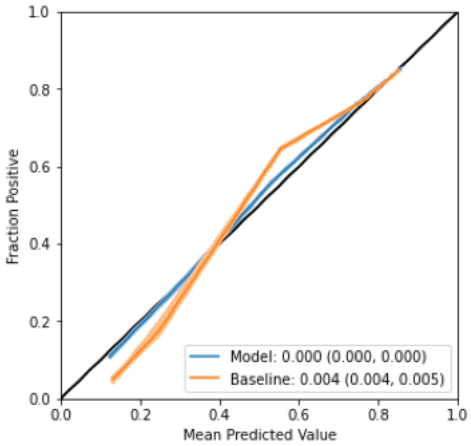
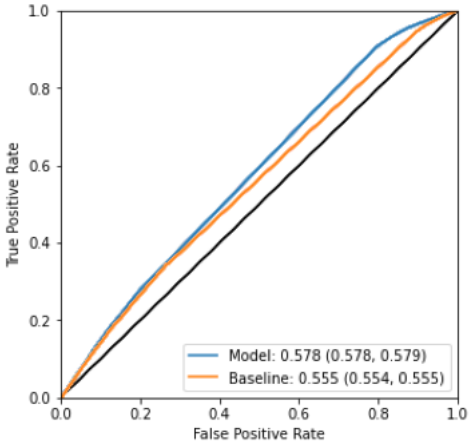
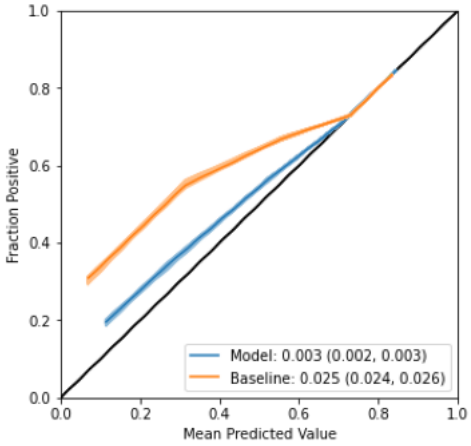
Model Architecture	Discrimination ROC Curve (AUROC)	Calibration Calibration Curve (ECE)
Logistic Regression		
Random Forest		

Figure A.3: Performance for Simpler Model Architectures.

ture, the top 25 encoded features, in terms of absolute coefficient value, are displayed in **Table A.9**. Since all the encoded feature values are restricted between 0 and 1, we can compare these values to determine their relative weight in generating the predictions. Of these 25 top encoded features, 9 are longitudinal observation encoded features. Since a substantial portion of the top 25 encoded features are longitudinal observations, this further suggests the value of longitudinal observations.

We can conduct a similar examination of the model learned with the random forest architecture. Instead of examining the model's coefficients, we plot one of the regression trees in **Figure A.4**. In this tree, encoded Procedure Codes features play an important role. The first node of both example trees splits based on the encoded feature Procedure Code = Empty (*e.g.*, if a procedure code was observed or not). Many other Procedure Code splits occur further down the example trees. These nodes all correspond to splits using longitudinal observation features. The relative simplicity of these model architectures allows us to easily observe the relative importance of the longitudinal observations.

Table A.9: Coefficient Values for Logistic Regression Model. These coefficients correspond to the values used by $f_{additional}$ operating on the modified input feature encoding, $\tilde{\mathbf{x}}'_{i,t}$. Bolded lines denote features that are longitudinal observations.

Encoded Feature Name TableName/FeatureName_Value	**Coefficient**	**Absolute Coefficient**
characteristic_dem/age	−13.33810	13.33810
characteristic_dem/job_4233	−4.51670	4.51670
characteristic_dem/job_6102	−4.31268	4.31268
samples_pr/count	**−4.27146**	**4.27146**
characteristic_dem/job_3000	−4.25290	4.25290
characteristic_dem/job_6005	−4.00636	4.00636
characteristic_dem/job_7771	−3.83329	3.83329
characteristic_dem/job_2913	−3.83070	3.83070
characteristic_dem/job_2131	3.23588	3.23588
samples_dx/count	**−3.11486**	**3.11486**
samples_pr/concat_cd_w0105	**−3.03534**	**3.03534**
characteristic_dem/job_2063	−2.96478	2.96478
samples_pr/concat_cd_99232	**−2.94987**	**2.94987**
samples_pr/concat_cd_99231	**−2.94126**	**2.94126**
characteristic_dem/job__empty_	−2.90068	2.90068
characteristic_dem/job_2211	2.82609	2.82609
samples_pr/concat_cd_w0120	**−2.73202**	**2.73202**
characteristic_dem/job_1642	2.72051	2.72051
characteristic_dem/job_2000	−2.62837	2.62837
samples_pr/concat_cd_90718	**2.44883**	**2.44883**
samples_pr/concat_cd_90471	**2.44060**	**2.44060**
samples_pr/concat_cd_w0179	**−2.40318**	**2.40318**
characteristic_dem/job_3152	2.39431	2.39431
characteristic_dem/job_5905	2.36187	2.36187
characteristic_dem/job_8102	−2.34343	2.34343

This simplicity enables direct examination of these models. Additionally, it allows relatively fast inference (compared to the proposed model). This speed and the removal of the history encoding enables an additional avenue for feature evaluation called permutation importance. Permutation importance allows model developers to understand the importance each features being used by

comparing the model's performance on the test dataset against its performance on the same dataset where the feature of interest has randomly been permuted among all the instances.[214]

For the permutation importance analysis, we conducted the permutations on the un-encoded features (at the level of $\mathbf{x}_{i,t}$ as opposed to the modified input feature encoding $\tilde{\mathbf{x}}'_{i,t}$). This enables us to understand the impact of categorical variables as a single unit. We present plots of permutation importance in terms of the difference in discriminative performance (AUROC) with 10 replications in **Figure A.5**. We can see that for the logistic regression model, the top three most important features at the level of $x_{i,t}$ are Job Code, Procedure Code, and Age, with a mean difference in AUROC values of 0.049 (standard deviation: $1.9E-4$), 0.031 ($1.0E-4$), and 0.021 ($1.4E-4$) respectively. Of these, the Procedure Code is a longitudinal observation variable. For the random forest, the top three most important features are Procedure Code 0.033 ($1.0E-4$), Age 0.030 ($2.3E-4$), and Job Code 0.011 ($5.1E-5$).

These experiments on the simpler model architectures help to underscore the importance of the longitudinal observations for this predictive task. Their results reinforce that the differences in predictive performance observed between the proposed and baseline model arise due to the longitudinal observations' value.

A.3 Data Definition and TemporalTransformer Configuration

A.3.1 Data Definition

We briefly describe the longitudinal claims dataset we employed for this study. Although the data arrived in a single `.csv` file, it is best represented by a relational database structure. We converted the data to a series of tables and then used our TemporalTransformer package to process the data in preparation for ML tasks. The tables are now described in the following paragraphs.

`characteristic_dem`[1]**:** describes non-longitudinal demographic information about each injured worker. Contains one row per patient. Has four columns: `i_id`, `age`, `sex`, and `job`. `i_id` is a unique key that identifies each injured worker. `age` is an integer value. `sex` is a categorical variable representing male, female, and other. `job` is a categorical variable representing job type (see **Table A.4** for additional description).

`samples_dx`: describes longitudinal observation information (samples) about the diagnoses each injured worker received over time. Contains many rows per patient, each row represents a diagnosis being given to an injured worker. Has four columns: `i_id`, `i_st` (start time), `i_et` (end time), and `dx`. `i_id` connects the observation to the injured worker. `i_st` represents the date at which the diagnosis is given. `i_et` is the day after the date of the diagnosis being given.[2] `dx` is a categorical variable representing ICD-9 diagnosis codes (see **Table A.5** for an additional description).

`samples_pr`: describes longitudinal observation information (samples) about the procedures each injured worker received over time. Contains many rows per patient, each represents a procedure being given to an injured worker. Has four columns: `i_id`, `i_st`, `i_et`, and `cd`. `i_id`, `i_st`, and `i_et` are all defined as per above. `cd` represents CPT procedure codes (see **Table A.6** for an additional description).

A.3.2 TemporalTransformer Configuration

Below is the data transformation configuration provided to Temporal Transformer. Although TemporalTransformer now has the functionality to handle timestamp representations of time, the initial

[1] characteristic is misspelled in some of our pre-processing code as "characterisitic". This misspelling is corrected in the main text and supplemental.

[2] The TemporalTransformer framework allows for observations to span across time-points (*e.g.*, across days), by setting the `i_et` (end time) to the day after `i_st` (start time), we represent an observation that occurred on the day of the start time.

version of TemporalTransformer we used for this project used integer representation of time. As such, we recorded observation timestamps to integers representing the number of days since injury. Note, TemporalTransformer now has a time discretization parameter `dt`. However, at the time of this work `dt` was a default parameter set to 1 that was not exposed to end-users.

`ds` is a dataset is an object which contains each of the separate datasets (e.g., train, development, test). This dataset can then be used with TemporalTransformer's Prepper module to automatically build a model with the proper in a $f_{in}(\cdot)$ and $f_{out}(\cdot)$ components given some user definition for the history encoder ($f_{mid}(\cdot)$).

```
'''
Example python code
Provided under a polyform strict license

given data:
partition_data { lists of lists (lol), where each list has injury id (i_id)
    and dataset partion name (e.g. train/dev/test)
characteristic_dem { lol, where each list has i_id, age, sex, and job code
sample_dx { lol, where each list has i_id, observation start time (i_st),
    observation end time (i_et), and the diagnosis code
sample_pr - lol, where each list has i_id, i_st, i_et, and the procedure code
'''
From TemporalTransformer import Hopper
From TemporalTransformer import Prepper


h = Hopper.dbms(verbose=False)

h.set_partitions(partition_data)

tc = Hopper.table_config("characteristic_dem",
                         ["age", "sex", "job"],
                         ["real", "ldc", "hdc"],
                         has_times=False,
                         primary_key=True)
h.create_fvm_with_data(tc, characteristic_dem)


_tc = Hopper.table_config("samples_dx",
                         ["dx"],
                         ["hdc"],
                         has_times=True,
                         primary_key=False)
h.create_fvm_with_data(_tc, sample_dx)


_tc = Hopper.table_config("samples_pr",
                         ["cd"],
                         ["hdc"],
                         has_times=True,
                         primary_key=False)
h.create_fvm_with_data(_tc, sample_pr)

h.dew_it(fit_normalization_via_sql_qds=False,
         default_first=0,
         fit_filter_percentile_cutoff=0.0)

p = Prepper.tf_prepper(h)

p.fit(offsets=[7],
      label_fns=["samples_ws/avg_ws", ],
      partition="train",
      ignore_fns=["samples_ws/avg_ws", "samples_ws/count"],
      has_time_hdc_channels=1)

ds = p.transform_to_ds()
```

Program A.1: Temporal Transformer Configuration Code. Used to prepare and transform longitudinal claims data for use with RNNs.

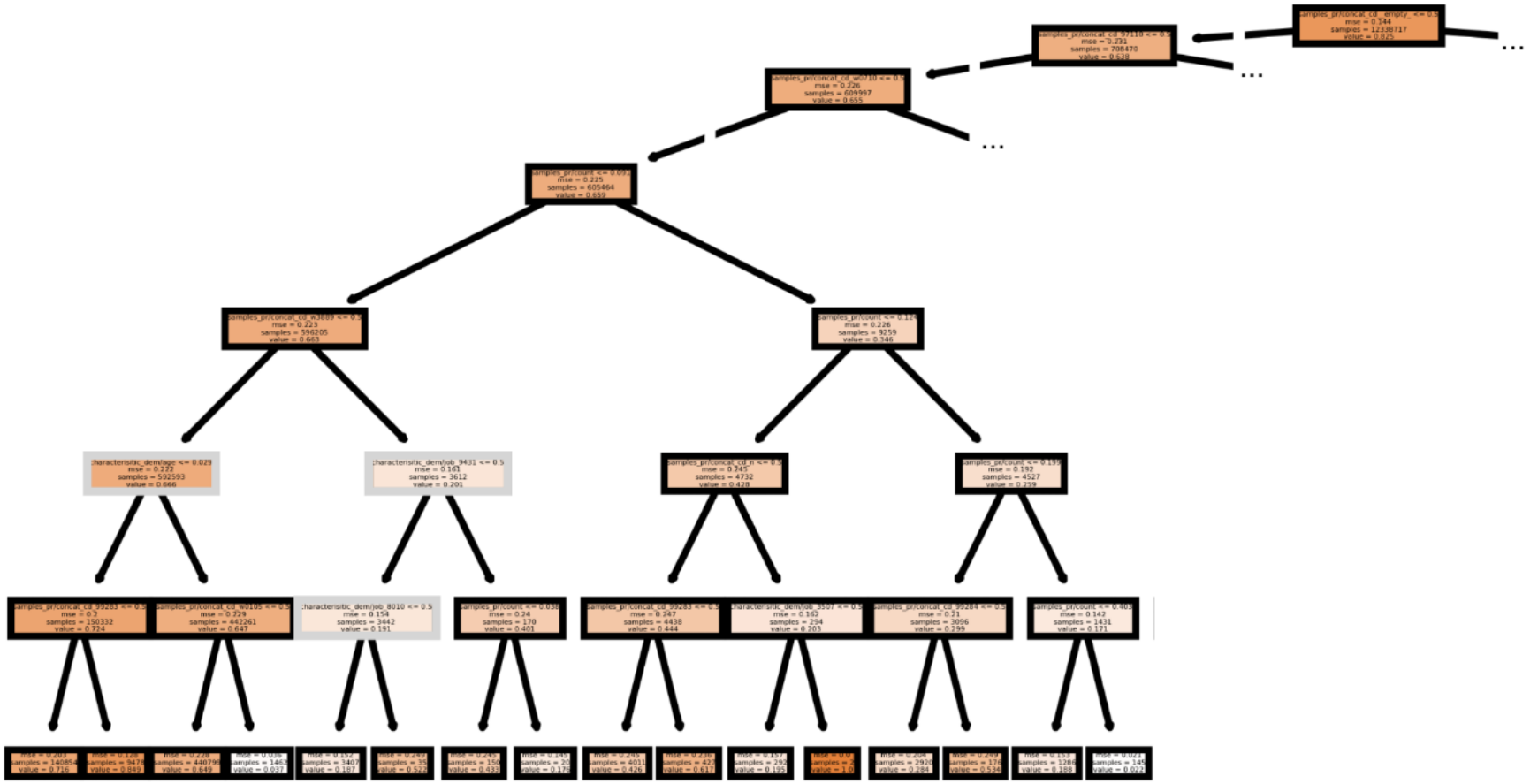


Figure A.4: Example Portion of a Regression Tree from the Learned Random Forest Model. Tree nodes correspond with splits on the modified input feature encoding, $\tilde{\mathbf{x}}'_{i,t}$. Split nodes bordered with black represent splits utilizing longitudinal information, and gray represents non-longitudinal information.

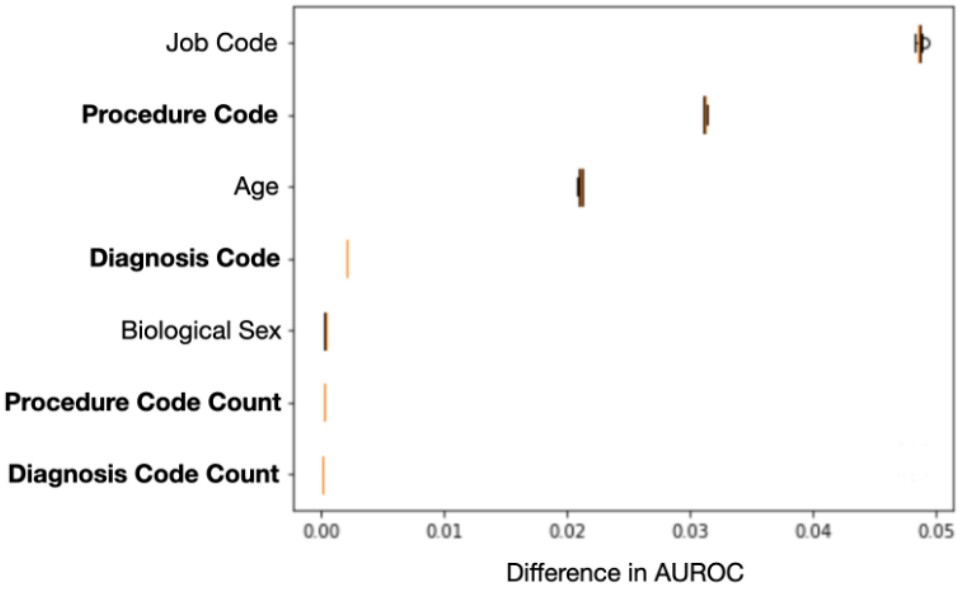
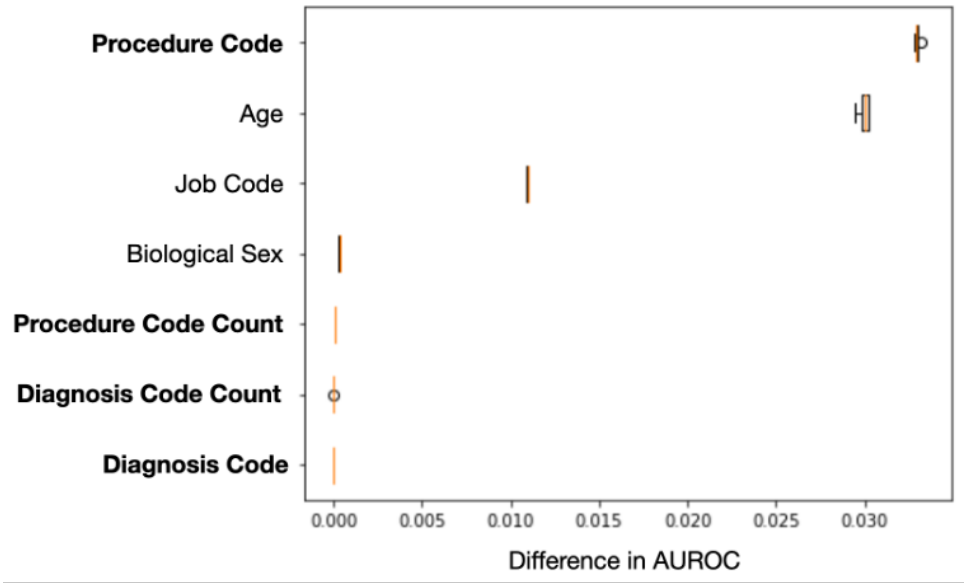
Model Architecture	Permutation Importance Difference in AUROC
Logistic Regression	
Random Forest	

Figure A.5: Permutation Importance For Simpler Model Architectures. The difference in AUROC was plotted for 10 replications of permutation importance for each of the $x_{i,t}$ level features. Bolded features correspond to longitudinal observation information.

APPENDIX B

Appendix For Chapter 3

B.1 Feature Groups

Feature groups are a grouping of features that have a shared meaning or source. We utilize them to pinpoint sources of discrepancies between the prospective and retrospective pipelines. Feature groups may be composed of other feature groups, **Table B.1** displays this hierarchical aspect of feature groups. For example, feature group "Idx: Admission Details" contains the feature groups "Idx: Admission Type" and "Idx: Insurance Type".

Table B.1: Feature Groups and Their Descriptions. Feature groups are hierarchical, with two major categories **Demographics** and **Clinical Characteristics**. Demographics are generally static patient-level attributes. Clinical characteristics are dependent on time and may change throughout an encounter. Clinical characteristics may also be broken into two major categories, based on which encounters the information is tied to: *historical encounters* or *index encounters*. Historical encounter information may be denoted in the main text with a "Hx" prefix and represents information collected in the encounters leading up to the current encounter. This history look-back is limited to 90 days. The index encounter information pertains to the current encounter and may be denoted with the "Id" prefix. Descriptions of each feature group are provided, along with the number of features included in this feature group. Various levels of feature group hierarchical structure are employed depending on the analysis.

Feature Group	Number of Features
Demographics	124
Age	5
Gender	2
Race	8
Marital Status	2
County & State	102

Continued on next page

Table B.1 – *Continued from previous page*

Feature Group	Number of Features
Body Mass Index	5
Clinical Characteristics	
Historical Encounters (Hx)	
History of CDI	2
Previous Encounters (stats)	10
Number of Previous Encounters	3
Length of Stay	7
Diagnoses (Diagnosis-Related Group/ICD9/ICD10)	983
Medications	2,731
Medication	1,886
Ingredient	620
Class	225
Index Encounter (Idx)	
Admission Details	22
Admission Type	3
Patient Type	12
Insurance Type	6
Emergency Visit	1
In-Hospital Locations	932
Vital Sign Measurements	17
Laboratory Results	508
Medications	2,731
Medication	1,879
Ingredient	629
Class	223
Colonization Pressure	10
Unit-based	5
Hospital-wide	5

B.2 Infrastructure Performance Gap Analysis

Table B.2: Infrastructure Performance Gap Analysis - Full Feature Swap Performance. By swapping column values corresponding to feature groups between swap analysis between $\mathbf{X}_{pro}$ and $\mathbf{X}'_{ret}$ we were able to quantify the performance impact of differences in the infrastructure related to each feature group. Note, this analysis was conducted at an interim time-point of our study. As such, it only uses data from July 10th to December 21st for both $\mathcal{D}_{pro}$ and $\mathcal{D}'_{ret}$. In addition to the feature group name, and the number of features in each feature group we display the AUROC on $\mathcal{D}_{pro}$ after the feature swap. Originally, we observed an AUROC of 0.769 on $\mathcal{D}_{pro}$, the final column displays the difference between this value after the swap and the 0.769. Hx: Medications, Idx: Medications, and In-Hospital Locations were the feature groups that had the largest positive swap difference in terms of AUROC, corresponding to improved model performance when given feature information from the retrospective pipeline.

Feature Category	AUROC After Swap	Difference
Hx: Medications	0.787	0.018
Idx: Medications	0.774	0.005
Idx: In-Hospital Locations	0.772	0.003
Hx: Previous Encounters (Length of Stay)	0.770	0.001
Demographics (Body Mass Index)	0.770	0.001
Demographics (County & State)	0.770	0.001
Idx: Colonization Pressure	0.770	0.001
Demographics (Race)	0.769	0.000
Idx: Admission Details (Emergency Visit)	0.769	0.000
Demographics (Gender)	0.769	0.000
Hx: History of CDI	0.769	0.000
Idx: Admission Details (Patient Type)	0.769	0.000
Demographics (Age)	0.769	0.000
Demographics (Marital Status)	0.769	0.000
Idx: Admission Details (Admission Type)	0.769	0.000
Idx: Admission Details (Insurance Type)	0.769	0.000
Hx: Previous Encounters (Number of Previous Encounters)	0.769	0.000
Idx: Laboratory Results	0.768	-0.001
Hx: Diagnoses	0.767	-0.002
Idx: Vitals	0.766	-0.003

Descriptions of feature groups can be found in **Table B.1**.

B.3 Model Performance Pre vs During COVID-19

To measure the impact of COVID-19 on model performance, we look at monthly AUROC performance before and during COVID-19 in **Figure B.1**. We notice that performance pre-Covid is generally lower than during Covid except April, which has long error bars due to small cases. We hypothesize the improved performance during Covid may be due to a simplification of the task. Our task is to predict hospital-associated CDI. However, our method for distinguishing between hospital-associated vs community-associated/recurrent, hospital-associated is dictated by guidelines that may or may not always reflect ground truth. We hypothesize that the increased contact precautions led to fewer hospital-associated cases and relatively more community-associated/recurrent cases. The latter is easier to identify because it only requires identifying susceptibility versus susceptibility and exposure.

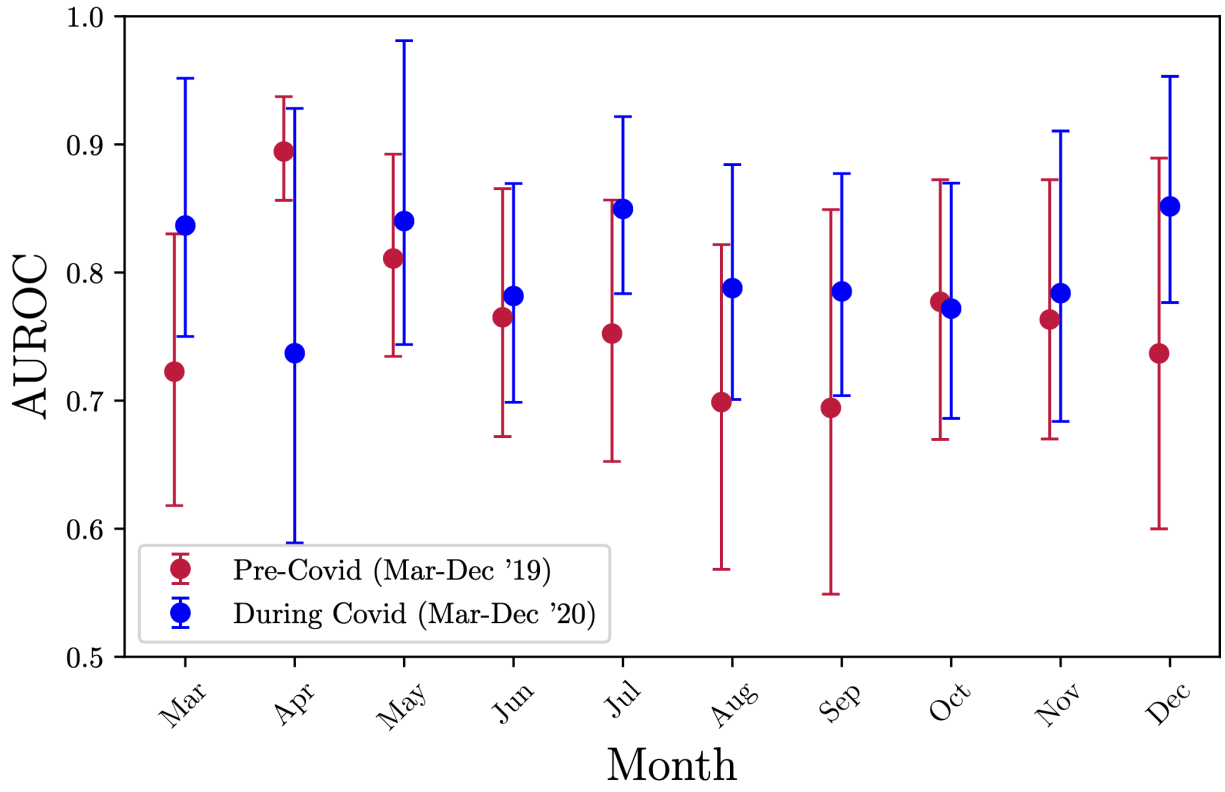


Figure B.1: Impact of COVID-19 Pandemic on Model Performance. Monthly performance of model in 2019 vs 2020 to show trends in performance before and during COVID-19. We see that performance is slightly higher during COVID-19.

B.4 '18-'19 Retrospective Validation

All validation datasets include encounters from July 10th to June 30th of the following year. After applying inclusion criteria, the '18-'19 retrospective validation set consisted of 26,450 hospital encounters. Population characteristics are detailed in **Table B.3**. It should be noted that the '18-'19 time period overlaps with the model validation period in 2018. This means that feature distributions from 2018 were used to help inform the decision to discard rare features. Applied to the retrospective validation data from '18-'19, the risk prediction model achieved AUROCs of 0.794 (95% CI: 0.767, 0.823) (**Figure B.2**). Selecting a decision threshold based on the 95th percentile of risk from the training set and applying on '18-'19 led to positive predictive values of 0.045, 0.036, and 0.027, respectively (**Figure B.3**). Monthly performance for '18-'19 is displayed in **Figure B.4**.

Table B.3: '18-'19 Cohort Characteristics.

	'18-'19 n=26,450
Median Age (IQR)	59 (41, 70)
Female (%)	51%
Median Length of Stay (IQR)	5 (4, 9)
History of CDI in the past year (%)	1.7%
Incidence Rate of CDI (%)	0.7%

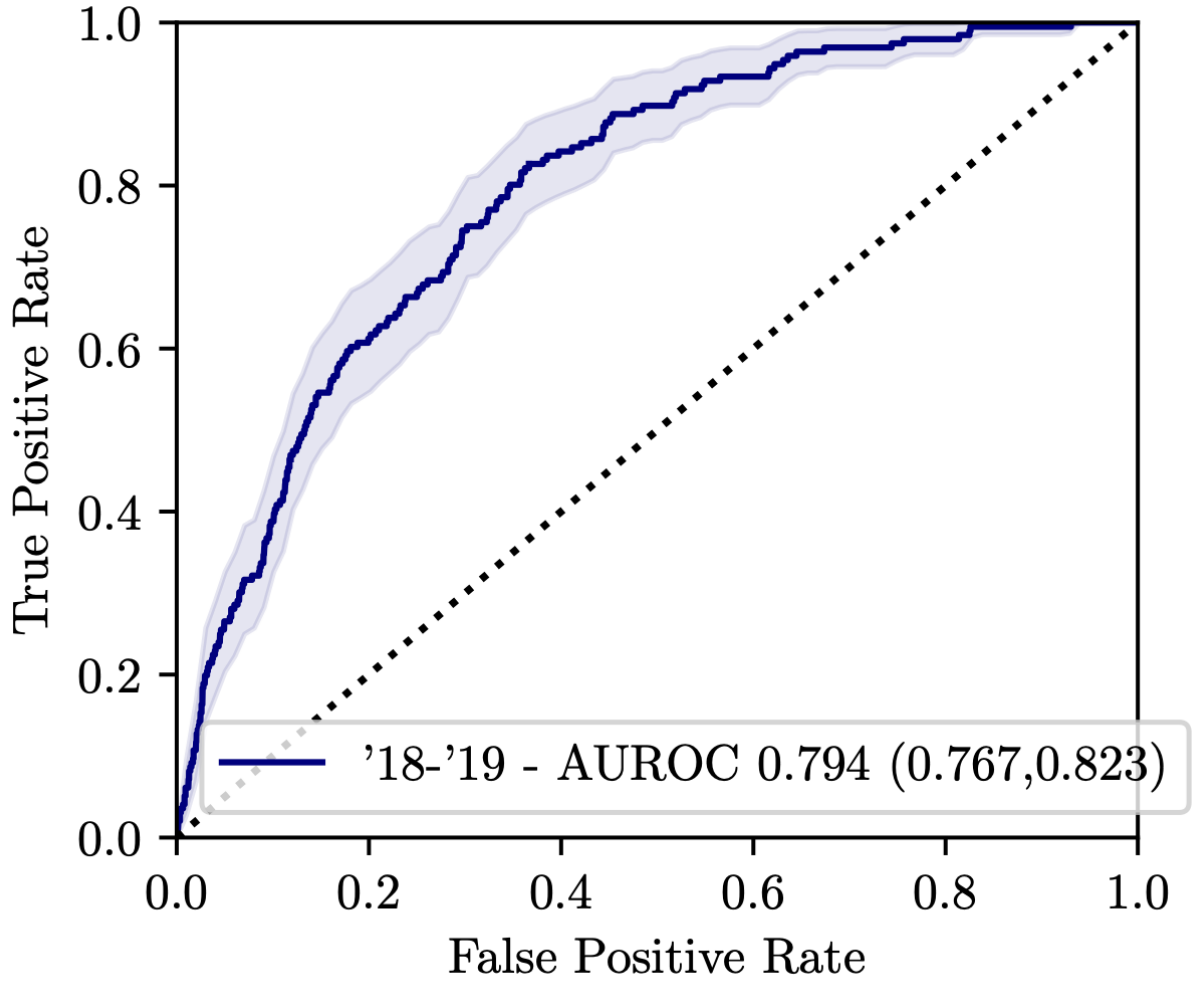


Figure B.2: Risk Prediction Model AUROC on the Retrospective ’18-’19 Validation Dataset.

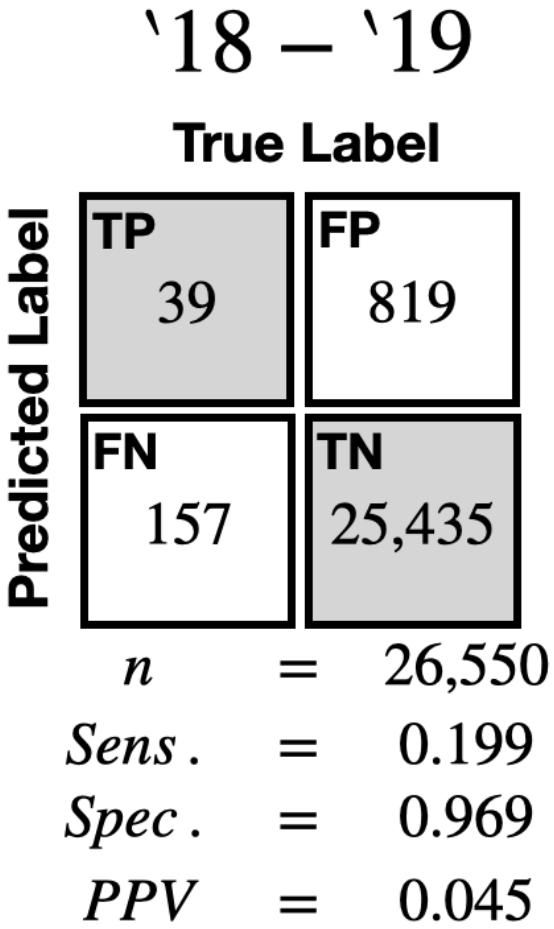


Figure B.3: Risk Prediction Model Confusion Matrix on the Retrospective '18-'19 Validation Dataset.

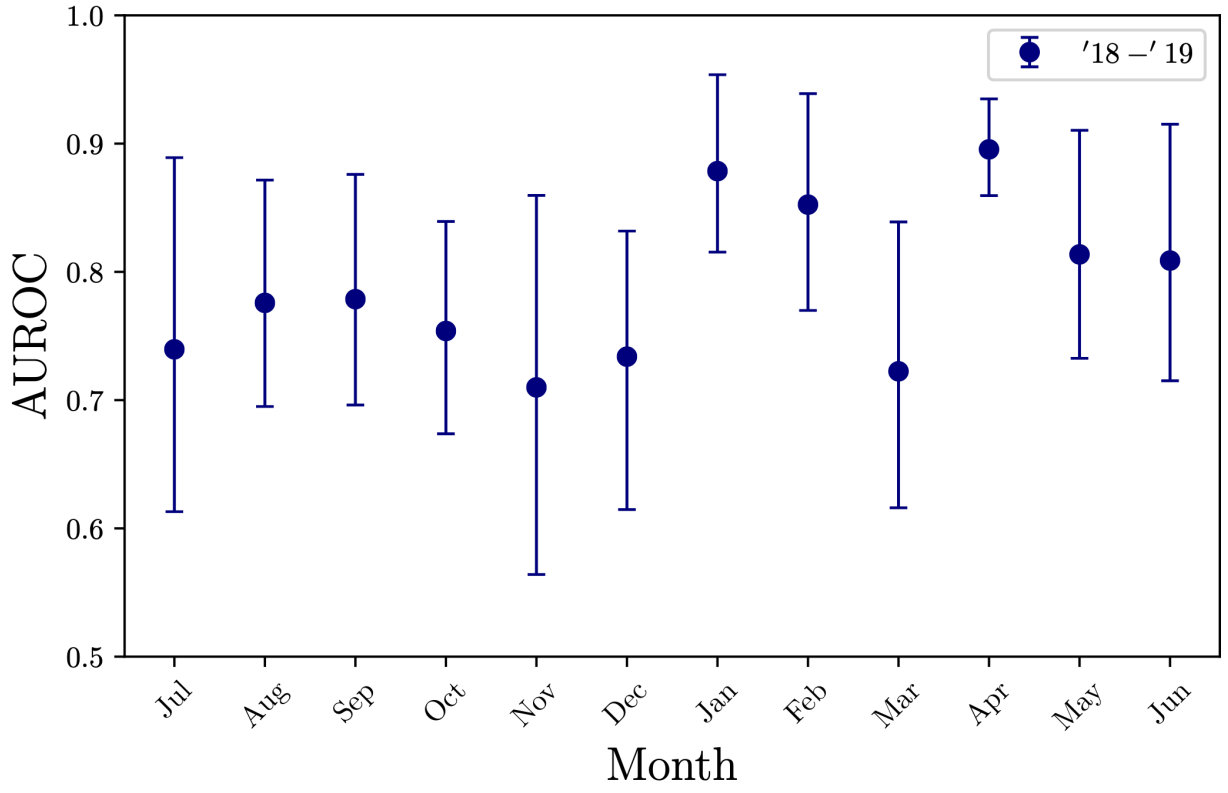


Figure B.4: Monthly AUROC Performance. AUROC for '20-'21 prospective dataset and the '19-'20 retrospective dataset broken down by month and bootstrap sampled $1,000$ times to generate 95% confidence intervals. We see that performance fluctuates month by month with higher performance in January, February, and April. There appear to be some monthly trends in performance across the years. Similar to '19-'20, we see lower, less variable scores in the later months of the year with higher and more variable scores in the earlier months.

APPENDIX C

Appendix For Chapter 4

C.1 Experiments & Results

C.1.1 Experimental Setup

C.1.1.1 Implementation in TensorFlow

All models were implemented as logistic regression models trained using stochastic gradient descent in TensorFlow. [88] Example code demonstrating the original and updated model training procedure is highlighted in **Program C.1**.

C.2 Results

C.2.0.1 Scoping Analysis

In order to guide selection of dataset size and L2 regularization weights for the main experimental setup we conducted a scoping experiment using the MIMIC-III dataset. We trained L2 regularized logistic regression models using a variety of dataset sizes, ranging from 100 to 3000 patients, and a variety of regularization weights $\{E-5, E-4, ..., E4, E5\}$ as well as no regularization. We measured the $AUROC$ performance of these models on a held-out validation dataset. This procedure was repeated five times in order to calculated the mean validation $AUROC$. Results are summarized in **Figure C.1**.

C.2.0.2 Hyperparameter Sensitivity Analyses

Due to the size of the MIMIC-III dataset experiments requiring many replications take a great deal of time. As such, we used another publicly available healthcare dataset to build original and updated risk stratification models. This dataset was the Kaggle Stroke Prediction dataset.

```
'''
Example python code
Provided under a *** license
'''
from sklearn import model_selection
from scipy import sparse
import numpy as np

_Xtr = sparse.load_npz('data/mortality/s_Xtr.npz').toarray()
_Xte = sparse.load_npz('data/mortality/s_Xte.npz').toarray()
_ytr = np.load('data/mortality/_ytr.npy')
_yte = np.load('data/mortality/_yte.npy')

size_o, size_ou_d = 0.15, 0.8
 = 0.8
X_o, X_u, y_o, y_u = model_selection.train_test_split(Xtr, ytr, train_size=size_o)
X_od, X_oe, y_od, y_oe = model_selection.train_test_split(X_o, y_o, train_size=size_ou_d)
X_ud, X_ue, y_ud, y_ue = model_selection.train_test_split(X_u, y_u, train_size=size_ou_d)

alpha_list = 0.5
size_od_prime, size_ud_prime = 200, 3000

res = []
for f_o_rep in tqdm(range(n_f_o_reps)):
    X_od_prime, _, y_od_prime, _ = model_selection.train_test_split(X_od, y_od,
                                                                    train_size=size_od_prime)
    X_ud_prime, _, y_ud_prime, _ = model_selection.train_test_split(X_ud, y_ud,
                                                                    train_size=size_ud_prime)

    f_o = OriginalLRModel()
    f_o.fit(X_od_prime, y_od_prime)

    p_hat_o_ud_prime = f_o.predict_proba(X_ud_prime)
    p_hat_o_ue = f_o.predict_proba(X_ue)
    p_hat_o_e = f_o.predict_proba(X_e)

    auroc_o_ue = metrics.roc_auc_score(y_ue, p_hat_o_ue)
    auroc_o_e = metrics.roc_auc_score(y_e, p_hat_o_e)

    f_u = UpdateLRModelExact(loss_function_weight=alpha,
                             incompatibility_loss_weight=1-alpha)
    f_u.fit(X_ud_prime, y_ud_prime, p_hat_o_ud_prime)

    p_hat_u_ue = f_u.predict_proba(X_ue)
    p_hat_u_e = f_u.predict_proba(X_e)

    auroc_u_ue = metrics.roc_auc_score(y_ue, p_hat_u_ue)
    auroc_u_e = metrics.roc_auc_score(y_e, p_hat_u_e)

    compat_ue = cm.np_pr_btc_score(y_ue, p_hat_o_ue, p_hat_u_ue)
    compat_e = cm.np_pr_btc_score(y_e, p_hat_o_e, p_hat_u_e)

    _res = {'f_o_rep': f_o_rep,
            'AUROC(f_o, ue)': auroc_o_ue,
            'AUROC(f_u, ue)': auroc_u_ue,
            'AUROC(f_o, e)': auroc_o_e,
            'AUROC(f_u, e)': auroc_u_e,
            'C^R(ue)': compat_ue,
            'C^R': compat_e,
           }
    res.append(_res)
```

Program C.1: Data Setup and Model-Pair Training.

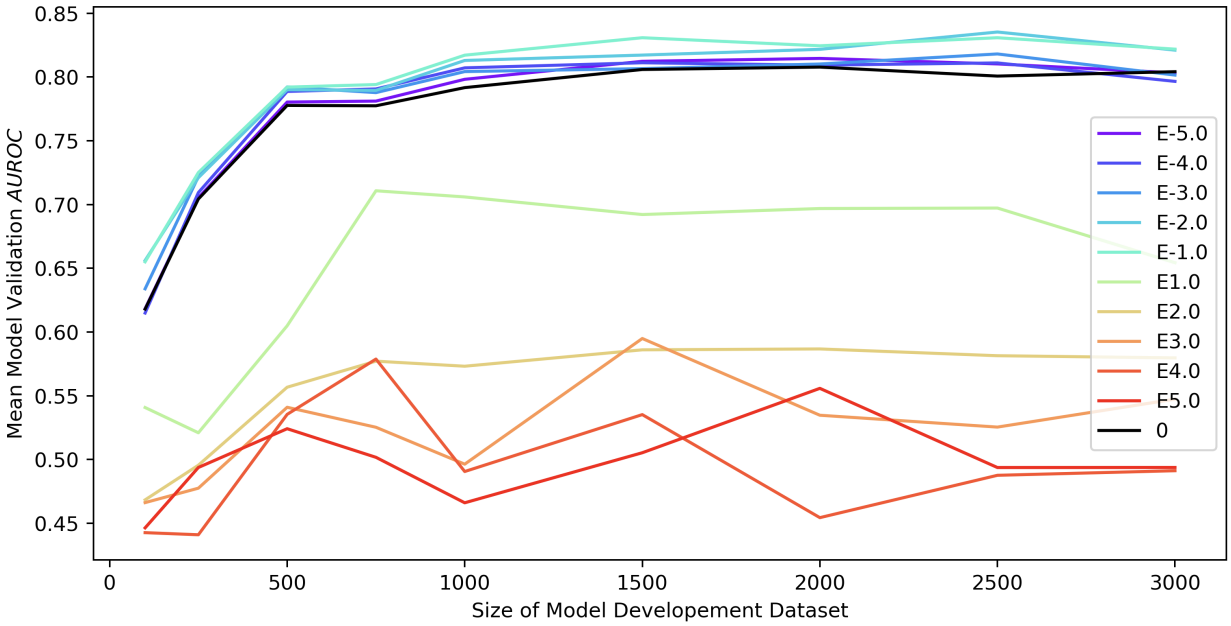


Figure C.1: Mean $AUROC$ Performance vs. Dataset Size and L2 Regularization Weight.

Using this dataset we get similar results, showing the utility of using the engineered update loss function.

Initialization. The primary experiment involves randomly initializing the weights ($\sim U[0, 0.01]$) each updated model being trained. This means that each of the updated models trained with the engineered update loss function (*e.g.*, at each α value) have different initial starting points. In order to assess the robustness of the engineered update loss function to initialization we conducted an additional experiment where we examined the impact of having all updated models for the same original model has the same initialization.

This experiment had 1 original model created, then the updated model creation process was replicated 100 times. This was done by randomly generating an initialization (all weights and the bias terms were randomly sampled uniformly between -0.1 and 0.1) and using this for all updates generated for the same original model. We controlled other hyperparameters (such as dataset sampling for the updated model, solver, epochs, *etc.*) by keeping them static for this experiment.

In **Figure C.2** we see that using the same initialization for all updated models yields similar results to the primary experiment, where updated models have different initialization.

Solver. The experiments presented in this work use the Adam solver. In order to assess the robustness of the engineered update loss function to choice of solver we conducted an additional experiment where we examined the impact of solver being used for updated model training.

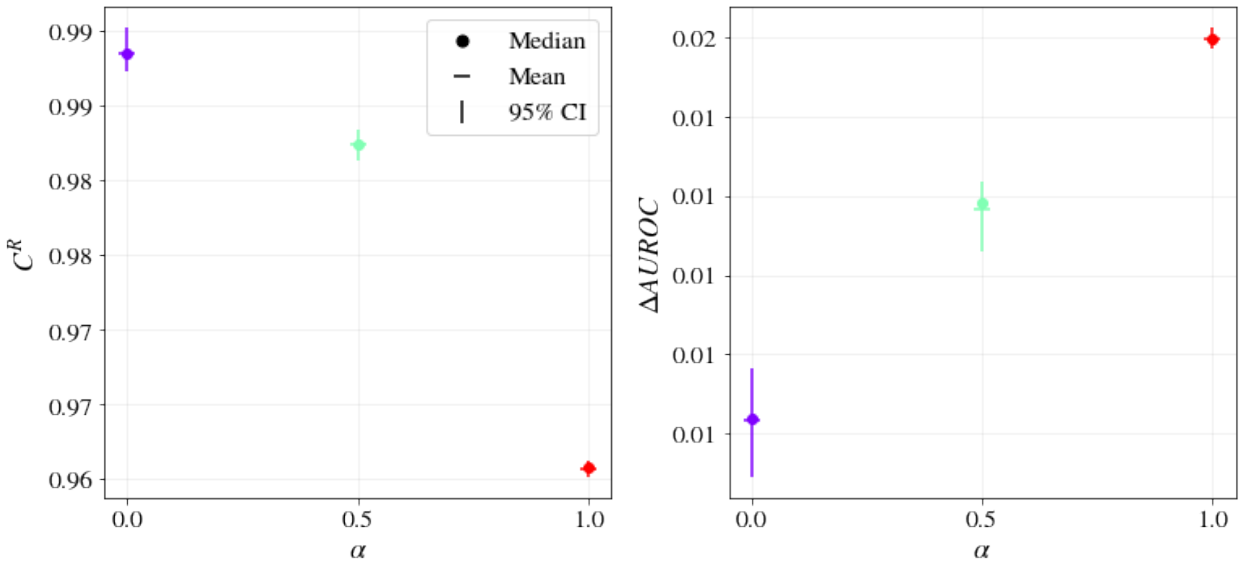


Figure C.2: Performance and $\mathcal{C}^{\mathrm{R}}$ of Engineered Model Updates with Same Initialization. We see behavior similar to what the primary experiment where models have different initialization.

This experiment had 1 original model created, then the updated model creation process was replicated 10 times. Updated model creation used the engineered update loss function with $\alpha = 0.5$. We controlled other hyperparameters (such as dataset sampling for the updated model, epochs, *etc.*) by keeping them static for this experiment, the only sources of variation were the ordering of data during training and updated model initialization. In addition to Adam two other solvers were tested: RMSProp and SGD.

In **Figure C.3** we see Adam and SGD yield results that are not statistically different for either AUROC of $\mathcal{C}^{\mathrm{R}}$. RMSProp produces a statistically significant lower value in terms of $\mathcal{C}^{\mathrm{R}}$ compared to Adam, but is not different in terms of AUROC.

Epochs. The experiments presented in this work use 100 epochs for the updated model training. In order to assess the robustness of the results presented we examined the impact of varying the number of epochs used to train updated models.

This experiment had 1 original model created, then the updated model creation process was replicated 10 times. Updated model creation used the engineered update loss function with $\alpha = 0.5$. For each replication we trained an updated model with the number of epochs varied between 1 and 10000 ($\{1, 10, 20, 50, 100, 200, 500, 1000, 10000\}$). We controlled other hyperparameters (such as dataset sampling for the updated model, solver, *etc.*) by keeping them static for this experiment, the only sources of variation were the ordering of data during training and updated model initialization.

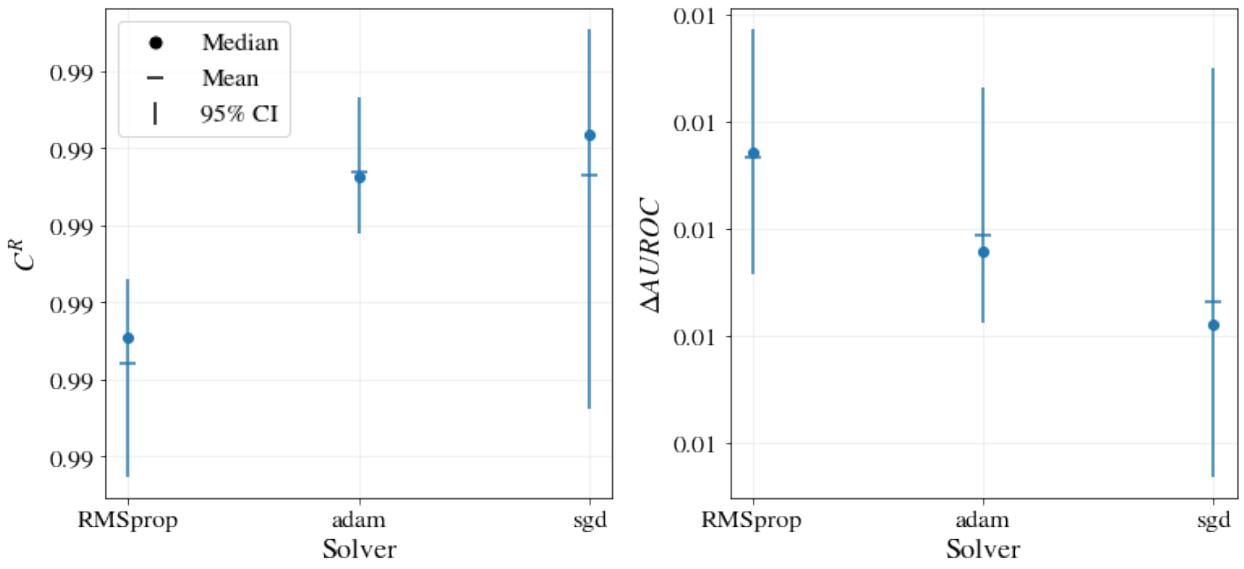


Figure C.3: Performance and $\mathcal{C}^{\mathrm{R}}$ of Engineered Model Updates with Same Different Solvers. We see Adam and SGD yield results that are not statistically different for either AUROC of $\mathcal{C}^{\mathrm{R}}$. RMSProp produces a statistically significant lower value in terms of $\mathcal{C}^{\mathrm{R}}$ compared to Adam, but is not different in terms of AUROC.

In **Figure C.4** we see variation in solutions up until a certain number of epochs. Once the updated model is trained with 50 epochs there appears to be little change in the resultant updated model.

Early Stopping The experiments presented in this work use 100 epochs with early stopping for the updated model training. Early stopping utilizes a patience parameter, which determines how many epochs it will permit observing increasing loss values before stopping. We utilize a patience value of 5. In order to assess the robustness of the results presented in relation to this patience value we varied the patience hyperparameter and assessed the impact to the updated model trained.

This experiment had 1 original model created, then the updated model creation process was replicated 10 times. Updated model creation used the engineered update loss function with $\alpha = 0.5$. For each replication we trained an updated model with the batch size varied between 8 and 128 ($\{8, 16, 32, 64, 128\}$). We controlled other hyperparameters (such as dataset sampling for the updated model, solver, *etc.*) by keeping them static for this experiment, the only sources of variation were the ordering of data during training and updated model initialization.

In **Figure C.5** we only see large variation in solutions when using patience$= 0$.

Batch Size The experiments presented in this work use a batch size of 32 for the updated model training. To test the sensitivity of the results presented to batch size 32 we varied the batch size

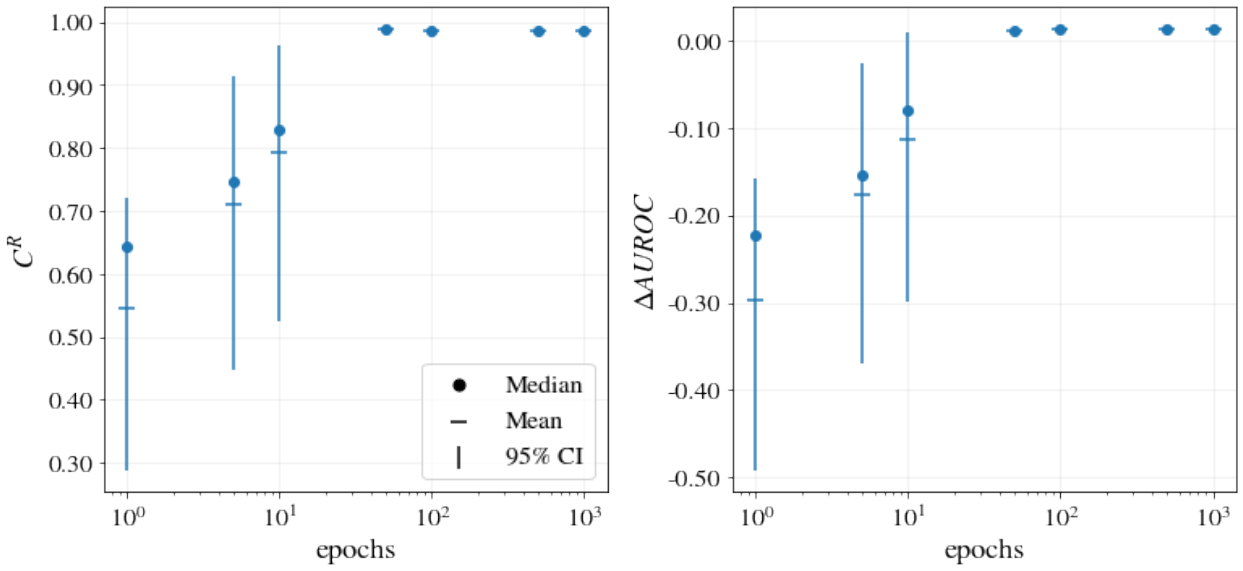


Figure C.4: Performance and $\mathcal{C}^{\mathrm{R}}$ of Engineered Model Updates with Respect to Epochs. We see variation in solutions up until a certain number of epochs. Once the updated model is trained with 50 epochs there appears to be little change in the resultant updated model.

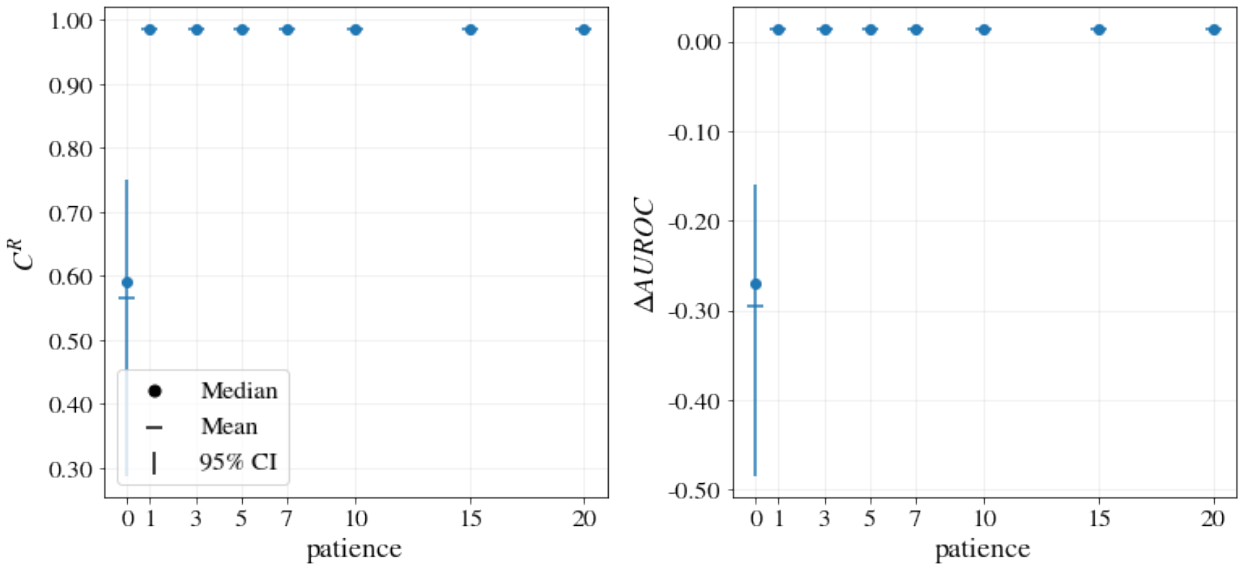


Figure C.5: Performance and $\mathcal{C}^{\mathrm{R}}$ of Engineered Model Updates with Respect to Patience. We only only see large variation in solutions when using patience$= 0$.

and assessed the impact to the updated model trained.

This experiment had 1 original model created, then the updated model creation process was replicated 10 times. Updated model creation used the engineered update loss function with $\alpha = 0.5$. For each replication we trained an updated model with the atch size varied between 0 and

20 ($\{0, 1, 3, 5, 7, 10, 15, 20\}$). We controlled other hyperparameters (such as dataset sampling for the updated model, solver, *etc.*) by keeping them static for this experiment, the only sources of variation were the ordering of data during training and updated model initialization.

In **Figure C.5** we see that both $\Delta AUROC$ and $\mathcal{C}^{\text{R}}$ vary as a function of batch size. $\Delta AUROC$ decreases as a function of batch size, whereas $\mathcal{C}^{\text{R}}$ increases. Batch size determines how many patient-pairs are evaluated in terms of ranking simultaneously. This then may make the $\widetilde{\mathcal{L}^{R}}$ component have lower values at training time when given larger batch sizes, this may in turn lead the model to emphasize better rank-based compatibility over discriminative performance.

The value we use, 32, may represent a "happy medium" between the competing effects. When training models using the engineered update loss function special attention should be paid to this parameter.

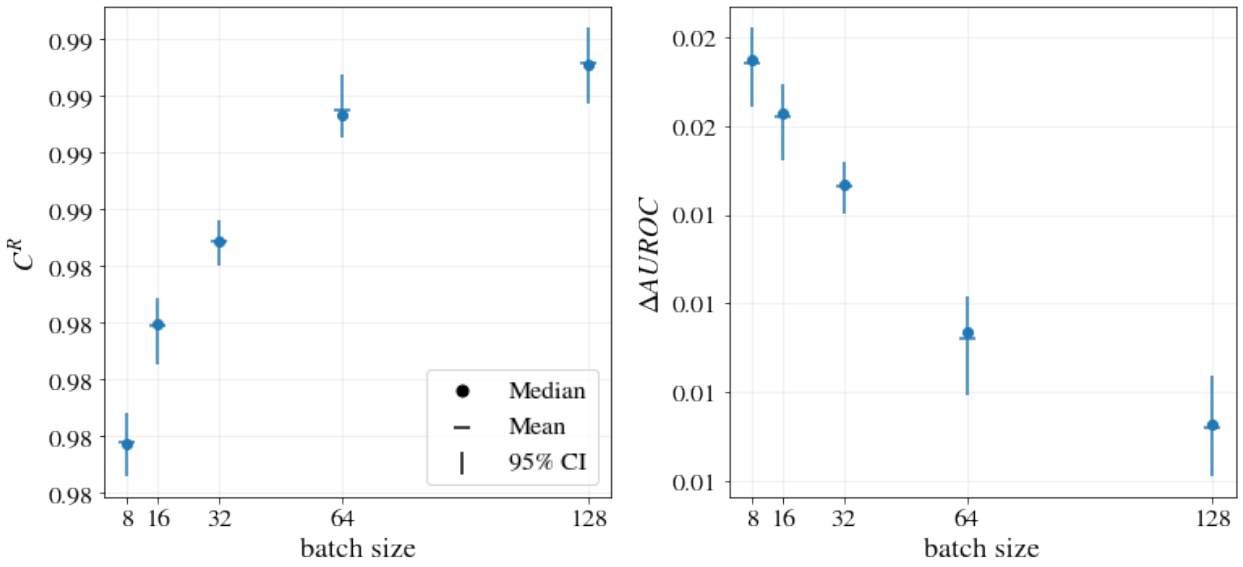


Figure C.6: Performance and $\mathcal{C}^{\text{R}}$ of Engineered Model Updates with Respect to Batch Size. $\Delta AUROC$ decreases as a function of batch size, whereas $\mathcal{C}^{\text{R}}$ increases.

Spreading Hyperparameter, s We employ a set spreading hyperparameter value of $s = 100$ for all our updated model training. In order to assess the robustness of the results presented we examined the impact of varying the s used to train updated models.

This experiment had 1 original model created, then the updated model creation process was replicated 10 times. Updated model creation used the engineered update loss function with $\alpha = 0.5$. For each replication we trained an updated model with s varied between 1 and 10^5 ($\{1, 10, 100, 10^3, 10^4, 10^5\}$). We controlled other hyperparameters (such as dataset sampling for the updated model, solver, *etc.*) by keeping them static for this experiment, the only sources of

variation were the ordering of data during training and updated model initialization.

In **Figure C.7** we see that both $\Delta AUROC$ and $\mathcal{C}^{\mathrm{R}}$ vary as a function of s. Small values of s produce larger $\Delta AUROC$ values and smaller $\mathcal{C}^{\mathrm{R}}$. Mid values of s produce larger $\Delta AUROC$ values and smaller $\mathcal{C}^{\mathrm{R}}$. Finally large values of s lead to large variation in $\Delta AUROC$ values and smaller $\mathcal{C}^{\mathrm{R}}$. s controls the steepness of the gradient used in the model training procedure. When the gradient becomes too steep the model training procedure may suffer from numerical instability.

The value we use, $s = 100$, may represent a "happy medium" between the competing effects, and we believe this may be a good value to use generally. However, depending on the use-case model developers may want to consider tuning this hyperparameter.

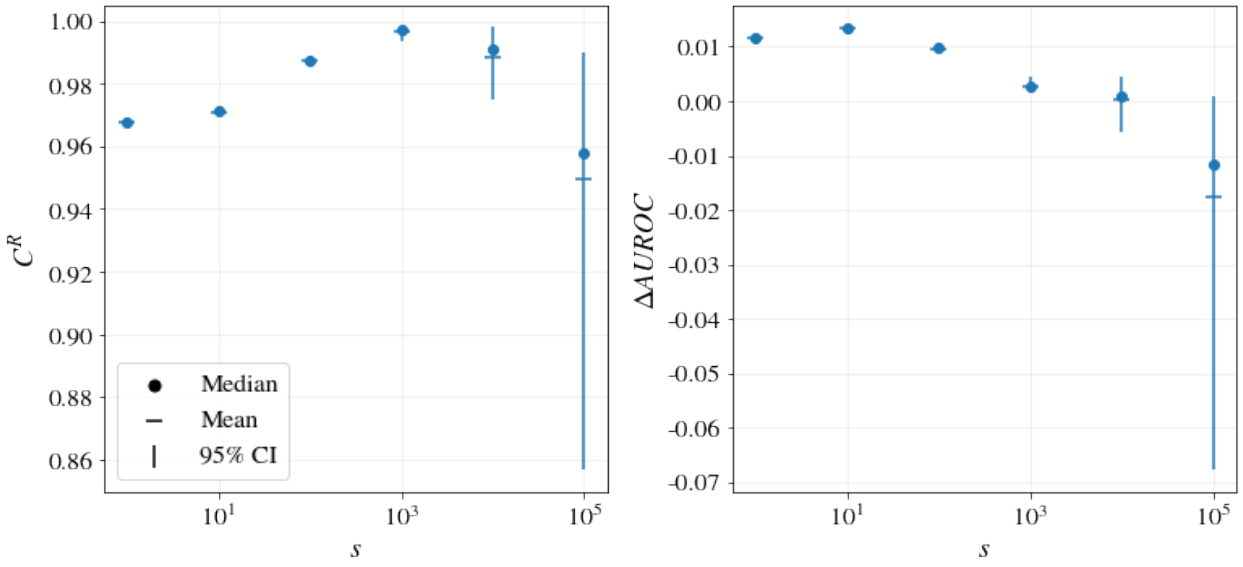


Figure C.7: Performance and $\mathcal{C}^{\mathrm{R}}$ of Engineered Model Updates with Respect to Spreading Hyperparameter, s. $\Delta AUROC$ decreases as a function of batch size, whereas $\mathcal{C}^{\mathrm{R}}$ increases.

C.3 Other Updated Model Selection Approaches

In addition to selecting a the selection model based on the best AUROC observed on the updated model validation dataset model developers may apply other selection criteria if they are seeking to produce models with high levels of $\mathcal{C}^{\mathrm{R}}$ in addition. These approaches include selection based on:

- best validation $\mathcal{C}^{\mathrm{R}}$
- best weighted combination of $\mathcal{C}^{\mathrm{R}}$ and C^R

In figure **C.8** we show the performance differences for these approaches. Additionally, we show the performance for an aggressive baseline that selects a baseline model for each the engineered

models by filtering out all models with AUROCs less than the engineered model and then selecting the best remaining model based on $\mathcal{C}^{\mathrm{R}}$.

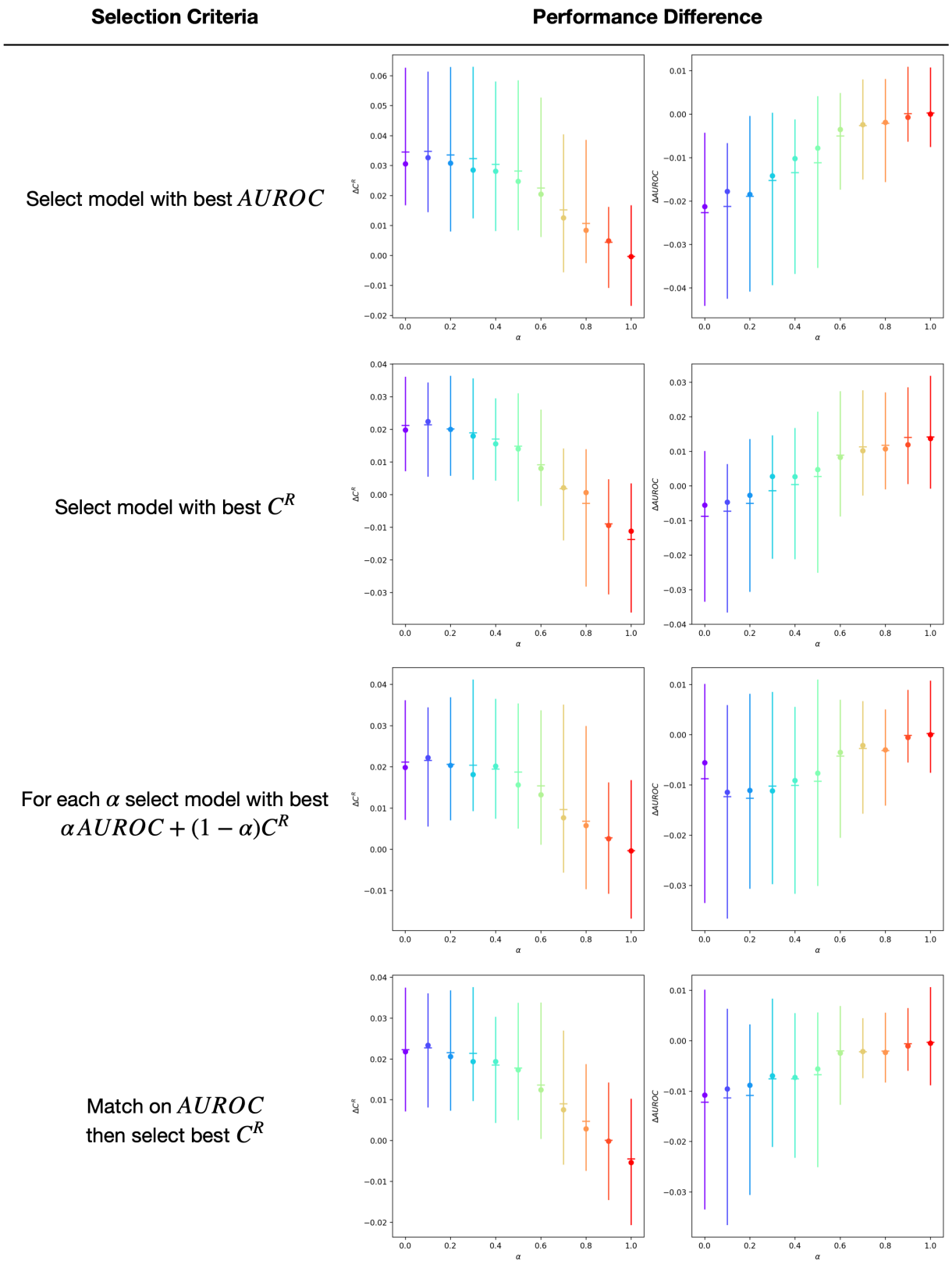


Figure C.8: Performance Differences Between Optimization and Selection.

BIBLIOGRAPHY

[1] Ania M. Jastreboff, Louis J. Aronne, Nadia N. Ahmad, Sean Wharton, Lisa Connery, Breno Alves, Arihiro Kiyosue, Shuyu Zhang, Bing Liu, Mathijs C. Bunck, and Adam Stefanski. Tirzepatide once weekly for the treatment of obesity. *New England Journal of Medicine*, 2022. ISSN 0028-4793. doi: 10.1056/nejmoa2206038.

[2] David L Sackett, William MC Rosenberg, JA Muir Gray, R Brian Haynes, and W Scott Richardson. Evidence based medicine: what it is and what it isn't. *British Medical Journal*, 312(7023):71–72, 1996. ISSN 0959-8138.

[3] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019. ISSN 0028-4793.

[4] Eric Wu, Kevin Wu, Roxana Daneshjou, David Ouyang, Daniel E. Ho, and James Zou. How medical ai devices are evaluated: limitations and recommendations from an analysis of fda approvals. *Nature Medicine*, 27(4):582–584, 2021. ISSN 1078-8956. doi: 10.1038/s41591-021-01312-x.

[5] Andrew Wong, Erkin Ötleș, John P. Donnelly, Andrew Krumm, Jeffrey Mccullough, Olivia Detroyer-Cooley, Justin Pestrue, Marie Phillips, Judy Konye, Carleen Penoza, Muhammad Ghous, and Karandeep Singh. External validation of a widely implemented proprietary sepsis prediction model in hospitalized patients. *JAMA Internal Medicine*, 2021. ISSN 2168-6106. doi: 10.1001/jamainternmed.2021.2626.

[6] Karandeep Singh, Thomas S Valley, Shengpu Tang, Benjamin Y. Li, Fahad Kamran, Michael W. Sjoding, Jenna Wiens, Erkin Ötleș, John P Donnelly, Melissa Y. Wei, Jonathon P. Mcbride, Jie Cao, Carleen Penoza, John Z Ayanian, and Brahmajee K Nallamothu. Evaluating a widely implemented proprietary deterioration index model among hospitalized covid-19 patients. *Annals of the American Thoracic Society*, 2020. ISSN 2329-6933. doi: 10.1513/annalsats.202006-698oc.

[7] Andrew Wong, Jie Cao, Patrick G. Lyons, Sayon Dutta, Vincent J. Major, Erkin Ötleș, and Karandeep Singh. Quantification of sepsis model alerts in 24 us hospitals before and during the covid-19 pandemic. *JAMA Network Open*, 4(11):e2135286, 2021. ISSN 2574-3805. doi: 10.1001/jamanetworkopen.2021.35286.

[8] Carl V. Asche, Brian Seal, Kristijan H. Kahler, Elisabeth M. Oehrlein, and Meredith Greer Baumgartner. Evaluation of healthcare interventions and big data: Review of associated

data issues. *PharmacoEconomics*, 35(8):759–765, 2017. ISSN 1170-7690. doi: 10.1007/s40273-017-0513-5.

[9] Kevin C. O'Kane and Richard J. Hildebrandt. An integrated health care information processing and retrieval system. In *Proceedings of the May 6-10, 1974, national computer conference and exposition on - AFIPS '74*. ACM Press, 1974. doi: 10.1145/1500175.1500193.

[10] Yun Liu, Po-Hsuan Cameron Chen, Jonathan Krause, and Lily Peng. How to read articles that use machine learning: Users' guides to the medical literature. *JAMA*, 322(18):1806–1816, 2019. ISSN 0098-7484. doi: 10.1001/jama.2019.16489. URL https://doi.org/10.1001/jama.2019.16489.

[11] 3rd Benich, J. J. and P. J. Carek. Evaluation of the patient with chronic cough. *Am Fam Physician*, 84(8):887–92, 2011. ISSN 1532-0650 (Electronic) 0002-838X (Linking). URL https://www.ncbi.nlm.nih.gov/pubmed/22010767.

[12] A.M.K. Harmsen, G.F. Giannakopoulos, P.R. Moerbeek, E.P. Jansma, H.J. Bonjer, and F.W. Bloemers. The influence of prehospital time on trauma patients outcome: A systematic review. *Injury*, 46(4):602–609, 2015. ISSN 0020-1383. doi: 10.1016/j.injury.2015.01.008.

[13] Marc A. Dall'Era, Peter C. Albertsen, Christopher Bangma, Peter R. Carroll, H. Ballentine Carter, Matthew R. Cooperberg, Stephen J. Freedland, Laurence H. Klotz, Christopher Parker, and Mark S. Soloway. Active surveillance for prostate cancer: A systematic review of the literature. *European Urology*, 62(6):976–983, 2012. ISSN 0302-2838. doi: 10.1016/j.eururo.2012.05.072. URL https://escholarship.org/content/qt4zm8z79g/qt4zm8z79g.pdf?t=o9umbp.

[14] Kushal T. Kadakia, Michael D. Howell, and Karen B. Desalvo. Modernizing public health data systems. *JAMA*, 326(5):385, 2021. ISSN 0098-7484. doi: 10.1001/jama.2021.12000.

[15] Robert M. Wachter and Michael D. Howell. Resolving the productivity paradox of health information technology. *JAMA*, 320(1):25, 2018. ISSN 0098-7484. doi: 10.1001/jama.2018.5605.

[16] Fahad Kamran, Shengpu Tang, Erkin Ötleș, Dustin S Mcevoy, Sameh N Saleh, Jen Gong, Benjamin Y Li, Sayon Dutta, Xinran Liu, Richard J Medford, Thomas S Valley, Lauren R West, Karandeep Singh, Seth Blumberg, John P Donnelly, Erica S Shenoy, John Z Ayanian, Brahmajee K Nallamothu, Michael W Sjoding, and Jenna Wiens. Early identification of patients admitted to hospital for covid-19 at risk of clinical deterioration: model development and multisite external validation study. *BMJ*, page e068576, 2022. ISSN 1756-1833. doi: 10.1136/bmj-2021-068576.

[17] Erkin Ötleș, Jeeheh Oh, Benjamin Li, Michelle Bochinski, Hyeon Joo, Justin Ortwine, Erica Shenoy, Laraine Washer, Vincent B. Young, Krishna Rao, and Jenna Wiens. Mind the performance gap: Examining dataset shift during prospective validation. *Proceedings of Machine Learning Research*, 2021.

[18] Onur Asan and Avishek Choudhury. Research trends in artificial intelligence applications in human factors health care: Mapping review. *JMIR Human Factors*, 8(2):e28236, 2021. ISSN 2292-9495. doi: 10.2196/28236.

[19] Pascale Carayon, Peter Hoonakker, Ann Schoofs Hundt, Megan Salwei, Douglas Wiegmann, Roger L Brown, Peter Kleinschmidt, Clair Novak, Michael Pulia, Yudi Wang, Emily Wirkus, and Brian Patterson. Application of human factors to improve usability of clinical decision support for diagnostic decision-making: a scenario-based simulation study. *BMJ Quality&Safety*, 29(4):329–340, 2020. ISSN 2044-5415. doi: 10.1136/bmjqs-2019-009857.

[20] John W Beasley, Richard J Holden, Erkin Ötleș, Lee A Green, Linsey M Steege, and Tosha B Wetterneck. It's time to bring human factors to primary care policy and practice. *Applied Ergonomics*, 85:103077, 2020. ISSN 0003-6870.

[21] Tosha Wetterneck, Richard Holden, John Beasley, and Erkin Ötleș. *Human Factors: Technical Series on Safer Primary Care*. Technical Series on Safer Primary Care. World Health Organization, Geneva, Switzerland, 2016.

[22] Ronald J. Koenig, Charles M. Peterson, Robert L. Jones, Christopher Saudek, Mark Lehrman, and Anthony Cerami. Correlation of glucose regulation and hemoglobin aicin diabetes mellitus. *New England Journal of Medicine*, 295(8):417–420, 1976. ISSN 0028-4793. doi: 10.1056/nejm197608192950804.

[23] Emily Jane Gallagher, Derek Le Roith, and Zachary Bloomgarden. Review of hemoglobin a1c in the management of diabetes. *Journal of Diabetes*, 1(1):9–17, 2009. ISSN 1753-0393. doi: 10.1111/j.1753-0407.2009.00009.x. URL https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1753-0407.2009.00009.x.

[24] Ben Van Calster, David J. Mclernon, Maarten Van Smeden, Laure Wynants, and Ewout W. Steyerberg. Calibration: the achilles heel of predictive analytics. *BMC Medicine*, 17(1), 2019. ISSN 1741-7015. doi: 10.1186/s12916-019-1466-7.

[25] Kim Luijken, Laure Wynants, Maarten Van Smeden, Ben Van Calster, Ewout W. Steyerberg, Rolf H.H. Groenwold, Dirk Timmerman, Tom Bourne, and Chinedu Ukaegbu. Changing predictor measurement procedures affected the performance of prediction models in clinical examples. *Journal of Clinical Epidemiology*, 119:7–18, 2020. ISSN 0895-4356. doi: 10.1016/j.jclinepi.2019.11.001.

[26] Ignazio Vecchio, Cristina Tornali, Nicola Luigi Bragazzi, and Mariano Martini. The discovery of insulin: An important milestone in the history of medicine. *Frontiers in Endocrinology*, 9, 2018. ISSN 1664-2392. doi: 10.3389/fendo.2018.00613. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6205949.

[27] Amedeo Lonardo, Simona Leoni, Khalid A. Alswat, and Yasser Fouad. History of nonalcoholic fatty liver disease. *International Journal of Molecular Sciences*, 21(16):5888, 2020. ISSN 1422-0067. doi: 10.3390/ijms21165888.

[28] S. Jain. Sepsis: An update on current practices in diagnosis and management. *Am J Med Sci*, 356(3):277–286, 2018. ISSN 1538-2990 (Electronic) 0002-9629 (Linking). doi: 10.1016/j.amjms.2018.06.012. URL https://www.ncbi.nlm.nih.gov/pubmed/30286823.

[29] Erkin Ötleș, Dan Kendrick, Quintin P Solano, Mary Schuller, Samantha L Ahle, Mickyas H Eskender, Emily Carnes, and Brian C George. Using natural language processing to automatically assess feedback quality: Findings from three surgical residencies. *Academic Medicine*, 2021. ISSN 1040-2446.

[30] J. D. Lee and K. A. See. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(1):50–80, 2004. ISSN 0018-7208. doi: 10.1518/hfes.46.1.50_30392.

[31] Gagan Bansal, Besmira Nushi, Ece Kamar, Dan Weld, Walter Lasecki, and Eric Horvitz. A case for backward compatibility for human-ai teams. *ICML Workshop on Human in the Loop Learning (HILL 2019)*, 2019.

[32] Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S. Weld, Walter S. Lasecki, and Eric Horvitz. Updates in human-ai teams: Understanding and addressing the performance/-compatibility tradeoff. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33: 2429–2437, 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33012429.

[33] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. Beyond accuracy: The role of mental models in human-ai team performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 2–11, 2019.

[34] J. Paul Leigh. Economic burden of occupational injury and illness in the united states. *The Milbank Quarterly*, 2011.

[35] 2018. URL https://injuryfacts.nsc.org/work/work-overview/work-safety-introduction/.

[36] U.S. Bureau of Labor Statistics. Injuries, illnesses, and fatalities. Report, U.S. Bureau of Labor Statistics,, 2019. URL https://www.bls.gov/iif/home.htm.

[37] L. I. Boden, P. K. O'Leary, K. M. Applebaum, and Y. Tripodis. The impact of non-fatal workplace injuries and illnesses on mortality. *Am J Ind Med*, 59(12):1061–1069, 2016. ISSN 1097-0274 (Electronic) 0271-3586 (Linking). doi: 10.1002/ajim.22632. URL https://www.ncbi.nlm.nih.gov/pubmed/27427538.

[38] Leslie I. Boden and Monica Galizzi. Economic consequences of workplace injuries and illnesses: Lost earnings and benefit adequacy. *AMERICAN JOURNAL OF INDUSTRIAL MEDICINE*, 1999.

[39] C. A. Okechukwu, J. Bacic, E. Velasquez, and L. B. Hammer. Marginal structural modelling of associations of occupational injuries with voluntary and involuntary job loss

among nursing home workers. *Occup Environ Med*, 73(3):175–82, 2016. ISSN 1470-7926 (Electronic) 1351-0711 (Linking). doi: 10.1136/oemed-2015-103067. URL https://www.ncbi.nlm.nih.gov/pubmed/26786757.

[40] X. S. Dong, X. Wang, J. A. Largay, and R. Sokas. Economic consequences of workplace injuries in the united states: Findings from the national longitudinal survey of youth (nlsy79). *Am J Ind Med*, 59(2):106–18, 2016. ISSN 1097-0274 (Electronic) 0271-3586 (Linking). doi: 10.1002/ajim.22559. URL https://www.ncbi.nlm.nih.gov/pubmed/26771100.

[41] S. A. Seabury, S. Terp, and L. I. Boden. Racial and ethnic differences in the frequency of workplace injuries and prevalence of work-related disability. *Health Aff (Millwood)*, 36(2):266–273, 2017. ISSN 1544-5208 (Electronic) 0278-2715 (Linking). doi: 10.1377/hlthaff.2016.1185. URL https://www.ncbi.nlm.nih.gov/pubmed/28167715.

[42] Yonatan Ben-Shalom, Steve Bruns, Kara Contreary, and David Stapleton. Stay-at-work/return-to-work: key facts, critical information gaps, and current practices and proposals. *Washington, DC: Mathematica Policy Research*, 2017.

[43] Robin Nagel, Steve Wiesner, and Jon Seymour. The electronic activity prescription tool (arx): Changing the work disability paradigm. *International Journal of Disability Management*, 7:40–61, 2012. ISSN 1833-8550. doi: 10.1017/idm.2012.9.

[44] S. Wiesner, J. Guerriero, and M. Garcia. From patient to productivity: Effectiveness of evidence-based guidelines in the clinical environment. *IBI Annual Forum*, 2016.

[45] Jesse E. Bible, Dan M. Spengler, and Hassan R. Mir. A primer for workers' compensation. *The Spine Journal*, 14(7):1325–1331, 2014. ISSN 1529-9430. doi: 10.1016/j.spinee.2014.01.030.

[46] Lyna Z. Schieber, Gery P. Guy, Puja Seth, and Jan L. Losby. Variation in adult outpatient opioid prescription dispensing by age and sex — united states, 2008–2018. *MMWR. Morbidity and Mortality Weekly Report*, 69(11):298–302, 2020. ISSN 0149-2195. doi: 10.15585/mmwr.mm6911a5.

[47] 2022. URL https://www.mcg.com/odg/odg-solutions/return-work-guidelines-modeling/.

[48] 2022. URL https://www.mdguidelines.com.

[49] Fraser Gaspar. Duration views methodology. Report, MDGuidelines, 2017. URL https://www.reedgroup.com/wp-content/uploads/2017/03/Duration-Views-Methodology.pdf.

[50] Kathryn Mueller, Doris Konicki, Paul Larson, T Warner Hudson, and Charles Yarborough. Advancing value-based medicine: why integrating functional outcomes with clinical measures is critical to our health care future. *Journal of occupational and environmental medicine*, 59(4):e57–e62, 2017. ISSN 1076-2752.

[51] W. H. Hou, J. Y. Tsauo, C. H. Lin, H. W. Liang, and C. L. Du. Worker's compensation and return-to-work following orthopaedic injury to extremities. *J Rehabil Med*, 40(6):440–5, 2008. ISSN 1650-1977 (Print) 1650-1977 (Linking). doi: 10.2340/16501977-0194. URL https://www.ncbi.nlm.nih.gov/pubmed/18509558.

[52] E. M. Haldorsen. The right treatment to the right patient at the right time. *Occup Environ Med*, 60(4):235–6, 2003. ISSN 1351-0711 (Print) 1351-0711 (Linking). doi: 10.1136/oem.60.4.235. URL https://www.ncbi.nlm.nih.gov/pubmed/12660369.

[53] S. Hogg-Johnson and D. C. Cole. Early prognostic factors for duration on temporary total benefits in the first year among workers with compensated occupational soft tissue injuries. *Occup Environ Med*, 60(4):244–53, 2003. ISSN 1351-0711 (Print) 1351-0711 (Linking). doi: 10.1136/oem.60.4.244. URL https://www.ncbi.nlm.nih.gov/pubmed/12660372.

[54] I. A. Steenstra, J. W. Busse, D. Tolusso, A. Davilmar, H. Lee, A. D. Furlan, 3rd Amick, B., and S. Hogg-Johnson. Predicting time on prolonged benefits for injured workers with acute back pain. *J Occup Rehabil*, 25(2):267–78, 2015. ISSN 1573-3688 (Electronic) 1053-0487 (Linking). doi: 10.1007/s10926-014-9534-5. URL https://www.ncbi.nlm.nih.gov/pubmed/25164779.

[55] Juan Zhao, Qiping Feng, Patrick Wu, Roxana A. Lupu, Russell A. Wilke, Quinn S. Wells, Joshua C. Denny, and Wei-Qi Wei. Learning from longitudinal data in electronic health record and genetic data to improve cardiovascular event prediction. *Scientific Reports*, 9 (1), 2019. ISSN 2045-2322. doi: 10.1038/s41598-018-36745-x. URL https://www.nature.com/articles/s41598-018-36745-x.pdf.

[56] Jenna Wiens, John Guttag, and Eric Horvitz. Patient risk stratification with time-varying parameters: a multitask learning approach. *The Journal of Machine Learning Research*, 17 (1):2797–2819, 2016. ISSN 1532-4435.

[57] Charles Elkan Zachary C. Lipton, John Berkowitz. A critical review of recurrent neural networks for sequence learning. *ArXiv*, 2015.

[58] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. ISBN 0262337371.

[59] A. Graves. Supervised sequence labelling with recurrent neural networks. *Supervised Sequence Labelling with Recurrent Neural Networks*, 385:1–141, 2012. ISSN 1860-949x. doi: 10.1007/978-3-642-24797-2. URL <GotoISI>://WOS:000304280700010.

[60] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv*, 2014.

[61] Emile Tompa, Claire De Oliveira, Roman Dolinschi, and Emma Irvin. A systematic review of disability management interventions with economic evaluations. *Journal of Occupational Rehabilitation*, 18(1):16–26, 2008. ISSN 1053-0487. doi: 10.1007/s10926-007-9116-x.

[62] William B Bunn, Robin S Baver, T Ehni, A Stowers, David D Taylor, Anita M Holloway, Duyen Duong, Dan B Pikelny, and David Sotolongo. Impact of a musculoskeletal disability management program on medical costs and productivity in a large manufacturing company. *Am J Manag Care*, 12:S27–32, 2006.

[63] Wayne N Burton and Daniel J Conti. Disability management: corporate medical department management of employee health and productivity. *Journal of Occupational and Environmental Medicine*, 42(10):1006–1012, 2000. ISSN 1076-2752.

[64] Niklas Krause and Thomas Lund. *Returning to work after occupational injury*. American Psychological Association, 2004. ISBN 1591470684.

[65] American College of Occupational and Environmental Medicine. Integrated health&safety index: Guide to a healthy&safe workplace. Report, American College of Occupational and Environmental Medicine,, 2017. URL https://acoem.org/Guidance-and-Position-Statements/Reference-Materials-Related-OEM-Documents/Integrated-Health-and-Safety-(IHS)-Index.

[66] ACOEM Guideline. Preventing needless work disability by helping people stay employed. *Journal of Occupational and Environmental Medicine*, 48(9):972–987, 2006.

[67] Maria G Michas and Carmine U Iacono. Overview of occupational medicine training among us family medicine residency programs. *FAMILY MEDICINE-KANSAS CITY-*, 40(2):102, 2008. ISSN 0742-3225.

[68] Teryl K Nuckols, Philip Harber, Yee-Wei Lim, Barbara O Wynn, Soeren Mattke, Rebecca Shaw, and Thomas M Wickizer. *Evaluating medical treatment guideline sets for injured workers in California*, volume 400. Minnesota Historical Society, 2005. ISBN 0833038354.

[69] 2021. URL https://www.mcg.com/odg/.

[70] 2021. URL https://www.mdguidelines.com/.

[71] F. J. Clay, S. V. Newstead, and R. J. McClure. A systematic review of early prognostic factors for return to work following acute orthopaedic trauma. *Injury*, 41(8):787–803, 2010. ISSN 1879-0267 (Electronic) 0020-1383 (Linking). doi: 10.1016/j.injury.2010.04.005. URL https://www.ncbi.nlm.nih.gov/pubmed/20435304.

[72] M. Papic, S. Brdar, V. Papic, and T. Loncar-Turukalo. Return to work after lumbar microdiscectomy - personalizing approach through predictive modeling. *Stud Health Technol Inform*, 224:181–3, 2016. ISSN 1879-8365 (Electronic) 0926-9630 (Linking). URL https://www.ncbi.nlm.nih.gov/pubmed/27225576.

[73] A. Gragnano, A. Negrini, M. Miglioretti, and M. Corbiere. Common psychosocial factors predicting return to work after common mental disorders, cardiovascular diseases, and cancers: A review of reviews supporting a cross-disease approach. *J Occup Rehabil*, 28(2):215–231, 2018. ISSN 1573-3688 (Electronic) 1053-0487 (Linking). doi:

10.1007/s10926-017-9714-1. URL https://www.ncbi.nlm.nih.gov/pubmed/28589524.

[74] J. Ervasti, M. Joensuu, J. Pentti, T. Oksanen, K. Ahola, J. Vahtera, M. Kivimaki, and M. Virtanen. Prognostic factors for return to work after depression-related work disability: A systematic review and meta-analysis. *J Psychiatr Res*, 95:28–36, 2017. ISSN 1879-1379 (Electronic) 0022-3956 (Linking). doi: 10.1016/j.jpsychires.2017.07.024. URL https://www.ncbi.nlm.nih.gov/pubmed/28772111.

[75] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 2001.

[76] Danilo Bolano, André Berchtold, and Gilbert Ritschard. A discussion on hidden markov models for life course data. *Sequence Analysis and Related Methods (LaCOSA II)*, page 241, 2016.

[77] Przemyslaw Dymarski. *Hidden Markov Models: Theory and Applications*. BoD–Books on Demand, 2011. ISBN 9533072083.

[78] Francesco Bartolucci, Alessio Farcomeni, and Fulvia Pennoni. *Latent Markov models for longitudinal data*. Chapman and Hall/CRC, 2012. ISBN 0429102577.

[79] T. Bai, A. K. Chanda, B. L. Egleston, and S. Vucetic. Ehr phenotyping via jointly embedding medical concepts and words into a unified vector space. *Bmc Medical Informatics and Decision Making*, 18, 2018. ISSN 1472-6947. doi: ARTN12310.1186/s12911-018-0672-0. URL <GotoISI>://WOS:000452837700001.

[80] Andrew L Beam, Benjamin Kompa, Inbar Fried, Nathan P Palmer, Xu Shi, Tianxi Cai, and Isaac S Kohane. Clinical concept embeddings learned from massive sources of medical data. *arXiv preprint arXiv:1804.01486*, 2018.

[81] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. *Advances in Neural Information Processing Systems 28 (Nips 2015)*, 28, 2015. ISSN 1049-5258. URL <GotoISI>://WOS:000450913101106.

[82] 2021. URL https://www.tensorflow.org/tutorials/text/word_embeddings.

[83] 2021. URL https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture.

[84] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems 30 (Nips 2017)*, 30, 2017. ISSN 1049-5258. URL <GotoISI>://WOS:000452649406008.

[85] Scott author Szymendera and issuing body Library of Congress. Congressional Research Service. *Workers' compensation : overview and issues*, volume Workers' compensation. Congressional Research Service, [library of congress public edition]. edition, 2018. URL https://purl.fdlp.gov/GPO/gpo112439.

[86] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 2018.

[87] Guido Van Rossum. *Python reference manual*. Network Theory LTD, 1995.

[88] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, and Matthieu Devin. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[89] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017. ISBN 1491962267.

[90] François Chollet. Keras: The python deep learning library. *Astrophysics Source Code Library*, page ascl: 1806.022, 2018.

[91] Mike Owens and Grant Allen. *SQLite*. Springer, 2010. ISBN 1430232250.

[92] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

[93] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science&Engineering*, 13(2):22, 2011. ISSN 1521-9615.

[94] Wes McKinney. Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, pages 51–56, 2010.

[95] Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, and Roger G Mark. tableone: An open source python package for producing summary statistics for research papers. *JAMIA open*, 1(1):26–31, 2018. ISSN 2574-2531.

[96] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct): 2825–2830, 2011.

[97] 2021.

[98] Jochen Bröcker and Leonard A. Smith. Increasing the reliability of reliability diagrams. *Weather and Forecasting*, 22(3):651–661, 2007. ISSN 1520-0434. doi: 10.1175/waf993.1.

[99] Nabeel Seedat and Christopher Kanan. Towards calibrated and scalable uncertainty representations for neural networks. *NeurIPS 2019*, 2019.

[100] Shengpu Tang, Parmida Davarmanesh, Yanmeng Song, Danai Koutra, Michael W Sjoding, and Jenna Wiens. Democratizing ehr analyses with fiddle: a flexible data-driven preprocessing pipeline for structured clinical data. *Journal of the American Medical Informatics Association*, 27(12):1921–1934, 2020. ISSN 1527-974X. doi: 10.1093/jamia/ocaa139.

[101] Sean Meyer. *Developing and Applying a Design Framework to Prepare Electronic Health Record Data for Time-Series Modeling*. Thesis, University of Michigan, 2021.

[102] Ravi B. Parikh, Stephanie Teeple, and Amol S. Navathe. Addressing bias in artificial intelligence in health care. *JAMA*, 322(24):2377, 2019. ISSN 0098-7484. doi: 10.1001/jama.2019.18058.

[103] Angela Stuesse. When they're done with you: Legal violence and structural vulnerability among injured immigrant poultry workers. *Anthropology of Work Review*, 39(2):79–93, 2018. ISSN 0883-024X. doi: 10.1111/awr.12148.

[104] Yonghua He, Jia Hu, Ignatius Tak Sun Yu, Wei Gu, and Youxin Liang. Determinants of return to work after occupational injury. *Journal of occupational rehabilitation*, 20(3):378–386, 2010. ISSN 1053-0487.

[105] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M. Rao, Troy D. Kelley, Dave Braines, Murat Sensoy, Christopher J. Willis, and Prudhvi Gurram. Interpretability of deep learning models: A survey of results. In *2017 IEEE SmartWorld, Ubiquitous Intelligence&Computing, Advanced&Trusted Computed, Scalable Computing&Communications, Cloud&Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 2017. doi: 10.1109/uic-atc.2017.8397411.

[106] Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning. *arXiv pre-print server*, 2020.

[107] Gary Marcus. Deep learning: A critical appraisal. *arXiv pre-print server*, 2018.

[108] Yolanda Hagar, David Albers, Rimma Pivovarov, Herbert Chase, Vanja Dukic, and Noémie Elhadad. Survival analysis with electronic health record data: Experiments with chronic kidney disease. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(5): 385–403, 2014. ISSN 1932-1864. doi: 10.1002/sam.11236.

[109] Hal Daumé III. Frustratingly easy domain adaptation. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2009.

[110] J. Oh, M. Makar, C. Fusco, R. McCaffrey, K. Rao, E. E. Ryan, L. Washer, L. R. West, V. B. Young, J. Guttag, D. C. Hooper, E. S. Shenoy, and J. Wiens. A generalizable, data-driven approach to predict daily risk of clostridium difficile infection at two large academic health centers. *Infect Control Hosp Epidemiol*, 39(4):425–433, 2018. ISSN 1559-6834 (Electronic) 0899-823X (Linking). doi: 10.1017/ice.2018.16. URL https://www.ncbi.nlm.nih.gov/pubmed/29576042.

[111] Ian Ayres and Amen Jalal. The impact of prescription drug monitoring programs on u.s. opioid prescriptions. *Journal of Law, Medicine&Ethics*, 46(2):387–403, 2018. ISSN 1073-1105. doi: 10.1177/1073110518782948.

[112] Sripriya Rajamani, Elizabeth S Chen, Elizabeth Lindemann, Ranyah Aldekhyyel, Yan Wang, and Genevieve B Melton. Representation of occupational information across resources and validation of the occupational data for health model. *Journal of the American Medical Informatics Association*, 25(2):197–205, 2018. ISSN 1067-5027. doi: 10.1093/jamia/ocx035. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6080809.

[113] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-Wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):160035, 2016. ISSN 2052-4463. doi: 10.1038/sdata.2016.35. URL http://europepmc.org/articles/pmc4878278?pdf=render.

[114] Alistair E W Johnson, David J Stone, Leo A Celi, and Tom J Pollard. The mimic code repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association*, 25(1):32–39, 2018. ISSN 1067-5027. doi: 10.1093/jamia/ocx084. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6381763.

[115] Matthew C Lenert, Michael E Matheny, and Colin G Walsh. Prognostic models will be victims of their own success, unless.... *Journal of the American Medical Informatics Association*, 26(12):1645–1650, 2019. ISSN 1527-974X. doi: 10.1093/jamia/ocz145.

[116] Andrew L. Beam and Isaac S. Kohane. Big data and machine learning in health care. *JAMA*, 319(13):1317, 2018. ISSN 0098-7484. doi: 10.1001/jama.2017.18391. URL https://jamanetwork.com/journals/jama/articlepdf/2675024/jama_Beam_2018_vp_170174.pdf.

[117] Bret Nestor, Matthew B. A. McDermott, Geeticka Chauhan, Tristan Naumann, Michael C. Hughes, Anna Goldenberg, and Marzyeh Ghassemi. Rethinking clinical prediction: Why machine learning must consider year of care and feature aggregation. *arXiv pre-print server*, 2018.

[118] Valentina Bellemo, Zhan W Lim, Gilbert Lim, Quang D Nguyen, Yuchen Xie, Michelle Y T Yip, Haslina Hamzah, Jinyi Ho, Xin Q Lee, Wynne Hsu, Mong L Lee, Lillian Musonda, Manju Chandran, Grace Chipalo-Mutati, Mulenga Muma, Gavin S W Tan, Sobha Sivaprasad, Geeta Menon, Tien Y Wong, and Daniel S W Ting. Artificial intelligence using deep learning to screen for referable and vision-threatening diabetic retinopathy in africa: a clinical validation study. *The Lancet Digital Health*, 1(1):e35–e44, 2019. ISSN 2589-7500. doi: 10.1016/s2589-7500(19)30004-4.

[119] Stuart Keel, Pei Ying Lee, Jane Scheetz, Zhixi Li, Mark A. Kotowicz, Richard J. Macisaac, and Mingguang He. Feasibility and patient acceptability of a novel artificial intelligence-based screening model for diabetic retinopathy at endocrinology outpatient services: a pilot study. *Scientific Reports*, 8(1), 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-22612-2. URL http://dro.deakin.edu.au/eserv/DU:30110170/kotowicz-feasibilityand-2018.pdf.

[120] Michael D. Abràmoff, Philip T. Lavin, Michele Birch, Nilay Shah, and James C. Folk. Pivotal trial of an autonomous ai-based diagnostic system for detection of diabetic retinopathy in primary care offices. *npj Digital Medicine*, 1(1), 2018. ISSN 2398-6352. doi: 10.1038/s41746-018-0040-6. URL https://doi.org/10.1038/s41746-018-0040-6.

[121] Michael A. Kang, Matthew M. Churpek, Frank J. Zadravecz, Richa Adhikari, Nicole M. Twu, and Dana P. Edelson. Real-time risk prediction on the wards. *Critical Care Medicine*, 44(8):1468–1473, 2016. ISSN 0090-3493. doi: 10.1097/ccm.0000000000001716.

[122] Nathan Brajer, Brian Cozzi, Michael Gao, Marshall Nichols, Mike Revoir, Suresh Balu, Joseph Futoma, Jonathan Bae, Noppon Setji, Adrian Hernandez, and Mark Sendak. Prospective and external evaluation of a machine learning model to predict in-hospital mortality of adults at time of admission. *JAMA Network Open*, 3(2):e1920733, 2020. ISSN 2574-3805. doi: 10.1001/jamanetworkopen.2019.20733. URL https://dx.doi.org/10.1001/jamanetworkopen.2019.20733.

[123] Lilian Minne, Saeid Eslami, Nicolette De Keizer, Evert De Jonge, Sophia E. De Rooij, and Ameen Abu-Hanna. Effect of changes over time in the performance of a customized saps-ii model on the quality of care assessment. *Intensive Care Medicine*, 38(1):40–46, 2012. ISSN 0342-4642. doi: 10.1007/s00134-011-2390-2. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3233667.

[124] R. Murphy-Filkins, D. Teres, S. Lemeshow, and D. W. Hosmer. Effect of changing patient mix on the performance of an intensive care unit severity-of-illness model: how to distinguish a general from a specialty intensive care unit. *Crit Care Med*, 24(12):1968–73, 1996. ISSN 0090-3493 (Print) 0090-3493. doi: 10.1097/00003246-199612000-00007.

[125] Samuel G. Finlayson, Adarsh Subbaswamy, Karandeep Singh, John Bowers, Annabel Kupke, Jonathan Zittrain, Isaac S. Kohane, and Suchi Saria. The clinician and dataset shift in artificial intelligence. *New England Journal of Medicine*, 385(3):283–286, 2021. ISSN 0028-4793. doi: 10.1056/nejmc2104626.

[126] Rafael Ortega, Mauricio Gonzalez, Ala Nozari, and Robert Canelli. Personal protective equipment and covid-19. *New England Journal of Medicine*, 382(26):e105, 2020. ISSN 0028-4793. doi: 10.1056/nejmvcm2014809.

[127] Andrew W. Artenstein. In pursuit of ppe. *New England Journal of Medicine*, 382(18):e46, 2020. ISSN 0028-4793. doi: 10.1056/nejmc2010025.

[128] Alice Y. Guh, Yi Mu, Lisa G. Winston, Helen Johnston, Danyel Olson, Monica M. Farley, Lucy E. Wilson, Stacy M. Holzbauer, Erin C. Phipps, Ghinwa K. Dumyati, Zintars G. Beldavs, Marion A. Kainer, Maria Karlsson, Dale N. Gerding, and L. Clifford Mcdonald. Trends in u.s. burden of clostridioides difficile infection and outcomes. *New England Journal of Medicine*, 382(14):1320–1330, 2020. ISSN 0028-4793. doi: 10.1056/nejmoa1910215.

[129] Fernanda C. Lessa, Yi Mu, Wendy M. Bamberg, Zintars G. Beldavs, Ghinwa K. Dumyati, John R. Dunn, Monica M. Farley, Stacy M. Holzbauer, James I. Meek, Erin C. Phipps,

Lucy E. Wilson, Lisa G. Winston, Jessica A. Cohen, Brandi M. Limbago, Scott K. Fridkin, Dale N. Gerding, and L. Clifford Mcdonald. Burden ofclostridium difficileinfection in the united states. *New England Journal of Medicine*, 372(9):825–834, 2015. ISSN 0028-4793. doi: 10.1056/nejmoa1408913.

[130] F. Barbut, L. Surgers, C. Eckert, B. Visseaux, M. Cuingnet, C. Mesquita, N. Pradier, A. Thiriez, N. Ait-Ammar, A. Aifaoui, E. Grandsire, and V. Lalande. Does a rapid diagnosis of clostridium difficile infection impact on quality of patient management? *Clinical Microbiology and Infection*, 20(2):136–144, 2014. ISSN 1198-743X. doi: 10.1111/1469-0691.12221. URL https://doi.org/10.1111/1469-0691.12221.

[131] Erik R. Dubberke, Dale N. Gerding, David Classen, Kathleen M. Arias, Kelly Podgorny, Deverick J. Anderson, Helen Burstin, David P. Calfee, Susan E. Coffin, Victoria Fraser, Frances A. Griffin, Peter Gross, Keith S. Kaye, Michael Klompas, Evelyn Lo, Jonas Marschall, Leonard A. Mermel, Lindsay Nicolle, David A. Pegues, Trish M. Perl, Sanjay Saint, Cassandra D. Salgado, Robert A. Weinstein, Robert Wise, and Deborah S. Yokoe. Strategies to prevent clostridium difficile infections in acute care hospitals. *Infection Control and Hospital Epidemiology*, 29(S1):S81–S92, 2008. ISSN 0899-823X. doi: 10.1086/591065. URL https://digitalcommons.wustl.edu/cgi/viewcontent.cgi?article=1819&context=open_access_pubs.

[132] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4): 1–37, 2014. ISSN 0360-0300. doi: 10.1145/2523813. URL http://eprints.bournemouth.ac.uk/22491/1/ACM%20computing%20surveys.pdf.

[133] Adriana Sayuri Iwashita and Joao Paulo Papa. An overview on concept drift learning. *IEEE Access*, 7:1532–1547, 2019. ISSN 2169-3536. doi: 10.1109/access.2018.2886026.

[134] Indrė Žliobaitė, Mykola Pechenizkiy, and João Gama. *An Overview of Concept Drift Applications*, pages 91–114. Springer International Publishing, 2016. ISBN 2197-6503. doi: 10.1007/978-3-319-26989-4_4. URL http://repositorio.inesctec.pt/bitstream/123456789/5348/1/P-00M-PR6.pdf.

[135] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2018. ISSN 1041-4347. doi: 10.1109/tkde.2018.2876857.

[136] João Gama and Pedro Pereira Rodrigues. *An Overview on Mining Data Streams*, pages 29–45. Springer Berlin Heidelberg, 2009. ISBN 1860-949X. doi: 10.1007/978-3-642-01091-0_2.

[137] João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. *Learning with Drift Detection*, pages 286–295. Springer Berlin Heidelberg, 2004. ISBN 0302-9743. doi: 10.1007/978-3-540-28645-5_29.

[138] João Gama, Ricardo Fernandes, and Ricardo Rocha. Decision trees for mining data streams. *Intelligent Data Analysis*, 10(1):23–45, 2006. ISSN 1088-467X.

[139] 2011. URL https://www.cs.waikato.ac.nz/~abifet/PAKDD2011/PAKDD11Tutorial_Handling_Concept_Drift.pdf.

[140] Sharon E Davis, Thomas A Lasko, Guanhua Chen, Edward D Siew, and Michael E Matheny. Calibration drift in regression and machine learning models for acute kidney injury. *Journal of the American Medical Informatics Association*, 24(6):1052–1061, 2017. ISSN 1067-5027. doi: 10.1093/jamia/ocx030. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6080675.

[141] Graeme L. Hickey, Stuart W. Grant, Camila Caiado, Simon Kendall, Joel Dunning, Michael Poullis, Iain Buchan, and Ben Bridgewater. Dynamic prediction modeling approaches for cardiac surgery. *Circulation: Cardiovascular Quality and Outcomes*, 6(6):649–658, 2013. ISSN 1941-7713. doi: 10.1161/circoutcomes.111.000012.

[142] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. ISSN 1520-0493.

[143] José Hernández-Orallo, Peter Flach, and César Ferri Ramírez. A unified view of performance metrics: Translating threshold choice into expected classification loss. *Journal of Machine Learning Research*, 13:2813–2869, 2012. ISSN 1533-7928.

[144] Centers for Disease Control Prevention. Identifying healthcare-associated infections (hai) for nhsn surveillance [internet]. *Centers for Disease Control and Prevention*, 2021.

[145] Luis Furuya-Kanamori, Samantha J. Mckenzie, Laith Yakob, Justin Clark, David L. Paterson, Thomas V. Riley, and Archie C. Clements. Clostridium difficile infection seasonality: Patterns across hemispheres and continents – a systematic review. *PLOS ONE*, 10(3):e0120730, 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0120730. URL https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0120730&type=printable.

[146] Andrew J. Vickers, Mathew Kent, and Peter T. Scardino. Implementation of dynamically updated prediction models at the point of care at a major cancer center: Making nomograms more like netflix. *Urology*, 102:1–3, 2017. ISSN 0090-4295. doi: 10.1016/j.urology.2016.10.049. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5376358.

[147] Sabrina Siregar, Daan Nieboer, Yvonne Vergouwe, Michel I.M. Versteegh, Luc Noyez, Alexander B.A. Vonk, Ewout W. Steyerberg, and Johanna J.M. Takkenberg. Improved prediction by dynamic modeling. *Circulation: Cardiovascular Quality and Outcomes*, 9(2):171–181, 2016. ISSN 1941-7713. doi: 10.1161/circoutcomes.114.001645.

[148] R. S. Evans. Electronic health records: Then, now, and in the future. *Yearbook of Medical Informatics*, 25(S 01):S48–S61, 2016. ISSN 0943-4747. doi: 10.15265/iys-2016-s006.

[149] 2021. URL https://www.uofmhealth.org/news/archive/202003/michigan-medicine-announces-covid-19-unit-new-paid-sick-time.

[150] Matthew N. Goldenberg and Vivek Parwani. Psychiatric emergency department volume during covid-19 pandemic. *The American Journal of Emergency Medicine*, 41:233–234, 2021. ISSN 0735-6757. doi: 10.1016/j.ajem.2020.05.088.

[151] Priya Venkatesan. The changing demographics of covid-19. *The Lancet Respiratory Medicine*, 8(12):e95, 2020. ISSN 2213-2600. doi: 10.1016/s2213-2600(20)30461-6.

[152] Laura E. Wong, Jessica E. Hawkins, Simone Langness, Karen L. Murrell, Patricia Iris, and Amanda Sammann. Where are all the patients? addressing covid-19 fear to encourage sick patients to seek emergency care. *NEJM Catalyst Innovations in Care Delivery*, 2020.

[153] 2017. URL https://healthitanalytics.com/features/epic-systems-machine-learning-is-the-ehr-usability-solution.

[154] Hossein Taghavi, Prasanna Padmanabhan, DB Tsai, Faisal Zakaria Siddiqi, and Justin Basilico. Distributed time travel for feature generation. *Netflix Technology Blog,*, Feb 12, 2016 2016. URL https://netflixtechblog.com/distributed-time-travel-for-feature-generation-389cccdd3907.

[155] Chris Paxton, Alexandru Niculescu-Mizil, and Suchi Saria. Developing predictive models using electronic medical records: challenges and pitfalls. In *AMIA Annual Symposium Proceedings*, volume 2013, page 1109. American Medical Informatics Association, 2013.

[156] Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S. Weld. Optimizing ai for teamwork. *arXiv pre-print server*, 2020.

[157] Ben Van Calster, Laure Wynants, Dirk Timmerman, Ewout W Steyerberg, and Gary S Collins. Predictive analytics in health care: how can we know it works? *Journal of the American Medical Informatics Association*, 26(12): 1651–1654, 2019. ISSN 1527-974X. doi: 10.1093/jamia/ocz130. URL https://academic.oup.com/jamia/advance-article-pdf/doi/10.1093/jamia/ocz130/29092183/ocz130.pdf.

[158] G. L. Hickey, S. W. Grant, G. J. Murphy, M. Bhabra, D. Pagano, K. Mcallister, I. Buchan, and B. Bridgewater. Dynamic trends in cardiac surgery: why the logistic euroscore is no longer suitable for contemporary cardiac surgery and implications for future risk models. *European Journal of Cardio-Thoracic Surgery*, 43(6):1146–1152, 2013. ISSN 1010-7940. doi: 10.1093/ejcts/ezs584. URL http://europepmc.org/articles/pmc3655624?pdf=render.

[159] Frank E Harrell Jr, Kerry L Lee, and Daniel B Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–387, 1996. ISSN 0277-6715.

[160] Frank E. Harrell. Evaluating the yield of medical tests. *JAMA: The Journal of the American Medical Association*, 247(18):2543, 1982. ISSN 0098-7484. doi: 10.1001/jama.1982.03320430047030.

[161] J A Hanley and B J Mcneil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982. ISSN 0033-8419. doi: 10.1148/radiology.143.1.7063747.

[162] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938. ISSN 0006-3444. doi: 10.1093/biomet/30.1-2.81.

[163] Jillian K. Gorski, Robert J. Batt, Erkin Ötleș, Manish N. Shah, Azita G. Hamedani, and Brian W. Patterson. The impact of emergency department census on the decision to admit. *Academic Emergency Medicine*, 24(1):13–21, 2017. ISSN 1069-6563. doi: 10.1111/acem.13103. URL https://onlinelibrary.wiley.com/doi/pdf/10.1111/acem.13103.

[164] Brian W. Patterson, Robert J. Batt, Morgan D. Wilbanks, Erkin Ötleș, Mary C. Westergaard, and Manish N. Shah. Cherry picking patients: Examining the interval between patient rooming and resident self-assignment. *Academic Emergency Medicine*, 23(6):679–684, 2016. ISSN 1069-6563. doi: 10.1111/acem.12895. URL https://onlinelibrary.wiley.com/doi/pdf/10.1111/acem.12895.

[165] Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization. *Advances in neural information processing systems*, 16, 2003.

[166] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006. ISBN 0387310738 (hd.bd.) 9780387310732. URL Publisherdescriptionhttp://www.loc.gov/catdir/enhancements/fy0818/2006922522-d.htmlTableofcontentsonlyhttp://www.loc.gov/catdir/enhancements/fy0818/2006922522-t.html.

[167] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012. ISBN 0262304325.

[168] Jun Han and Claudio Moraga. *The influence of the sigmoid function parameters on the speed of backpropagation learning*, pages 195–201. Springer Berlin Heidelberg, 1995. ISBN 0302-9743. doi: 10.1007/3-540-59497-3_175.

[169] Lian Yan, Robert H Dodier, Michael Mozer, and Richard H Wolniewicz. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 848–855, 2003.

[170] Erkin Ötleș, Brian T Denton, Bo Qu, Adharsh Murali, Selin Merdan, Gregory B Auffenberg, Spencer C Hiller, Brian R Lane, Arvin K George, and Karandeep Singh. Development and validation of models to predict pathological outcomes of radical prostatectomy in regional and national cohorts. *The Journal of Urology*, 207(2):358–366, 2022. ISSN 0022-5347.

[171] John Lee and Neville Moray. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35(10):1243–1270, 1992. ISSN 0014-0139. doi: 10.1080/00140139208967392.

[172] Kevin Anthony Hoff and Masooda Bashir. Trust in automation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 57(3):407–434, 2015. ISSN 0018-7208. doi: 10.1177/0018720814547570.

[173] P. A. Schulte, R. Pana-Cryan, T. Schnorr, A. L. Schill, R. Guerin, S. Felknor, and G. R. Wagner. An approach to assess the burden of work-related injury, disease, and distress. *Am J Public Health*, 107(7):1051–1057, 2017. ISSN 1541-0048 (Electronic) 0090-0036 (Linking). doi: 10.2105/AJPH.2017.303765. URL https://www.ncbi.nlm.nih.gov/pubmed/28520495.

[174] R. L. Franche, N. Carnide, S. Hogg-Johnson, P. Cote, F. C. Breslin, U. Bultmann, C. N. Severin, and N. Krause. Course, diagnosis, and treatment of depressive symptomatology in workers following a workplace injury: a prospective cohort study. *Can J Psychiatry*, 54(8):534–46, 2009. ISSN 1497-0015 (Electronic) 0706-7437 (Linking). doi: 10.1177/070674370905400806. URL https://www.ncbi.nlm.nih.gov/pubmed/19726006.

[175] D. P. Gross, J. Zhang, I. Steenstra, S. Barnsley, C. Haws, T. Amell, G. McIntosh, J. Cooper, and O. Zaiane. Development of a computer-based clinical decision support tool for selecting appropriate rehabilitation interventions for injured workers. *J Occup Rehabil*, 23(4):597–609, 2013. ISSN 1573-3688 (Electronic) 1053-0487 (Linking). doi: 10.1007/s10926-013-9430-4. URL https://www.ncbi.nlm.nih.gov/pubmed/23468410.

[176] K. S. Na and E. Kim. A machine learning-based predictive model of return to work after sick leave. *J Occup Environ Med*, 61(5):e191–e199, 2019. ISSN 1536-5948 (Electronic) 1076-2752 (Linking). doi: 10.1097/JOM.0000000000001567. URL https://www.ncbi.nlm.nih.gov/pubmed/30829888.

[177] A. J. P. Tixier, M. R. Hallowell, B. Rajagopalan, and D. Bowman. Application of machine learning to construction injury prediction. *Automation in Construction*, 69:102–114, 2016. ISSN 0926-5805. doi: 10.1016/j.autcon.2016.05.016. URL <GotoISI>://WOS:000380598400010.

[178] G. Nanda, K. M. Grattan, M. T. Chu, L. K. Davis, and M. R. Lehto. Bayesian decision support for coding occupational injury data. *J Safety Res*, 57:71–82, 2016. ISSN 1879-1247 (Electronic) 0022-4375 (Linking). doi: 10.1016/j.jsr.2016.03.001. URL https://www.ncbi.nlm.nih.gov/pubmed/27178082.

[179] K. Vallmuur, H. R. Marucci-Wellman, J. A. Taylor, M. Lehto, H. L. Corns, and G. S. Smith. Harnessing information from injury narratives in the 'big data' era: understanding and applying machine learning for injury surveillance. *Inj Prev*, 22 Suppl 1:i34–42, 2016. ISSN 1475-5785 (Electronic) 1353-8047 (Linking). doi: 10.1136/injuryprev-2015-041813. URL https://www.ncbi.nlm.nih.gov/pubmed/26728004.

[180] A. R. Meyers, I. S. Al-Tarawneh, S. J. Wurzelbacher, P. T. Bushnell, M. P. Lampl, J. L. Bell, S. J. Bertke, D. C. Robins, C. Y. Tseng, C. Wei, J. A. Raudabaugh, and

T. M. Schnorr. Applying machine learning to workers' compensation data to identify industry-specific ergonomic and safety prevention priorities: Ohio, 2001 to 2011. *J Occup Environ Med*, 60(1):55–73, 2018. ISSN 1536-5948 (Electronic) 1076-2752 (Linking). doi: 10.1097/JOM.0000000000001162. URL https://www.ncbi.nlm.nih.gov/pubmed/28953071.

[181] 2021. URL https://dwcdataportal.fldfs.com/ClaimsDataExtract.aspx.

[182] A. Oleinick and B. Zaidman. Methodologic issues in the use of workers' compensation databases for the study of work injuries with days away from work. i. sensitivity of case ascertainment. *Am J Ind Med*, 45(3):260–74, 2004. ISSN 0271-3586 (Print) 0271-3586 (Linking). doi: 10.1002/ajim.10333. URL https://www.ncbi.nlm.nih.gov/pubmed/14991853.

[183] H. O. Alanazi, A. H. Abdullah, and K. N. Qureshi. A critical review for developing accurate and dynamic predictive models using machine learning methods in medicine and health care. *J Med Syst*, 41(4):69, 2017. ISSN 1573-689X (Electronic) 0148-5598 (Linking). doi: 10.1007/s10916-017-0715-6. URL https://www.ncbi.nlm.nih.gov/pubmed/28285459.

[184] S. A. Christie, A. S. Conroy, R. A. Callcut, A. E. Hubbard, and M. J. Cohen. Dynamic multi-outcome prediction after injury: Applying adaptive machine learning for precision medicine in trauma. *PLoS One*, 14(4):e0213836, 2019. ISSN 1932-6203 (Electronic) 1932-6203 (Linking). doi: 10.1371/journal.pone.0213836. URL https://www.ncbi.nlm.nih.gov/pubmed/30970030.

[185] S. J. Patel, D. B. Chamberlain, and J. M. Chamberlain. A machine learning approach to predicting need for hospitalization for pediatric asthma exacerbation at the time of emergency department triage. *Acad Emerg Med*, 25(12):1463–1470, 2018. ISSN 1553-2712 (Electronic) 1069-6563 (Linking). doi: 10.1111/acem.13655. URL https://www.ncbi.nlm.nih.gov/pubmed/30382605.

[186] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014. ISBN 0124081215.

[187] Walter Zucchini, Iain L MacDonald, and Roland Langrock. *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC, 2017. ISBN 1315372487.

[188] J. R. Beck and S. G. Pauker. The markov process in medical prognosis. *Med Decis Making*, 3(4):419–458, 1983. ISSN 0272-989X (Print) 0272-989X (Linking). doi: 10.1177/0272989X8300300403. URL https://www.ncbi.nlm.nih.gov/pubmed/6668990.

[189] C. A. McGilchrist, C. W. Aisbett, and S. Cooper. A markov transition model in the analysis of the immune response. *J Theor Biol*, 138(1):17–21, 1989. ISSN 0022-5193 (Print) 0022-5193 (Linking). doi: 10.1016/s0022-5193(89)80175-4. URL https://www.ncbi.nlm.nih.gov/pubmed/2626064.

[190] B. L. Destavola. Testing departures from time homogeneity in multistate markov-processes. *Journal of the Royal Statistical Society Series C-Applied Statistics*, 37(2):242–250, 1988. ISSN 0035-9254. URL <GotoISI>://WOS:A1988P741100009.

[191] R. Kay. A markov model for analyzing cancer markers and disease states in survival studies. *Biometrics*, 42(4):855–865, 1986. ISSN 0006-341x. doi: Doi10.2307/2530699. URL <GotoISI>://WOS:A1986F624100013.

[192] P. L. Chen, E. J. Bernard, and P. K. Sen. A markov chain model used in analyzing disease history applied to a stroke study. *Journal of Applied Statistics*, 26(4):413–422, 1999. ISSN 0266-4763. doi: Doi10.1080/02664769922304. URL <GotoISI>://WOS:000080569100001.

[193] Tim Hunkapiller Pierre Baldi, Yves Chauvin and Marcella A. McClure. Hidden markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences of the United States of America*, 91, 1994.

[194] David R. Westhead and M.S.Vijayabaskar. *Hidden Markov Models*. Springer, 2017.

[195] Z. Huang, Z. Ge, W. Dong, K. He, and H. Duan. Probabilistic modeling personalized treatment pathways using electronic health records. *J Biomed Inform*, 86:33–48, 2018. ISSN 1532-0480 (Electronic) 1532-0464 (Linking). doi: 10.1016/j.jbi.2018.08.004. URL https://www.ncbi.nlm.nih.gov/pubmed/30138699.

[196] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[197] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan "Honza" Cernocky, and Sanjeev Khudanpur. Recurrent neural network based language model. *Interspeech*, 2010.

[198] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. *Proceedings of Machine Learning for Healthcare 2016*, 2016.

[199] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun. Using recurrent neural network models for early detection of heart failure onset. *J Am Med Inform Assoc*, 24(2):361–370, 2017. ISSN 1527-974X (Electronic) 1067-5027 (Linking). doi: 10.1093/jamia/ocw112. URL https://www.ncbi.nlm.nih.gov/pubmed/27521897.

[200] N. Tomasev, X. Glorot, J. W. Rae, M. Zielinski, H. Askham, A. Saraiva, A. Mottram, C. Meyer, S. Ravuri, I. Protsyuk, A. Connell, C. O. Hughes, A. Karthikesalingam, J. Cornebise, H. Montgomery, G. Rees, C. Laing, C. R. Baker, K. Peterson, R. Reeves, D. Hassabis, D. King, M. Suleyman, T. Back, C. Nielson, J. R. Ledsam, and S. Mohamed. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*, 572(7767):116–119, 2019. ISSN 1476-4687 (Electronic) 0028-0836 (Linking). doi: 10.1038/s41586-019-1390-1. URL https://www.ncbi.nlm.nih.gov/pubmed/31367026.

[201] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 1994.

[202] Yoshua Bengio Razvan Pascanu, Thomas Mikolov. On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[203] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[204] Fred Cummins Felix A. Gers, Jürgen Schmidhuber. Learning to forget: Continual prediction with lstm. *Neural computation*, 1999.

[205] Jürgen Schmidhuber Felix A. Gers, Nicol N. Schraudolph. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 2002.

[206] 2015. URL http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[207] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE Trans Neural Netw Learn Syst*, 28(10):2222–2232, 2017. ISSN 2162-2388 (Electronic) 2162-237X (Linking). doi: 10.1109/TNNLS.2016.2582924. URL https://www.ncbi.nlm.nih.gov/pubmed/27411231.

[208] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bagdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *ArXiv*, 2014.

[209] Maxim Topaz, Leah Shafran-Topaz, and Kathryn H. Bowles. Icd-9 to icd-10: evolution, revolution, and current debates in the united states. *Perspectives in health information management*, 10(Spring):1d–1d, 2013. ISSN 1559-4122. URL https://pubmed.ncbi.nlm.nih.gov/23805064https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3692324/.

[210] 2021. URL http://www.icd9data.com/.

[211] 2021. URL https://www.aapc.com.

[212] J.S. Cramer. The origins of logistic regression. *SSRN Electronic Journal*, 2003. ISSN 1556-5068. doi: 10.2139/ssrn.360300.

[213] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125. doi: 10.1023/a:1010933404324.

[214] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10): 1340–1347, 2010. ISSN 1460-2059. doi: 10.1093/bioinformatics/btq134. URL https://academic.oup.com/bioinformatics/article-pdf/26/10/1340/16892402/btq134.pdf.