DEEP LEARNING IN PERSONALIZED MEDICINE: APPLICATIONS IN PATIENT SIMILARITY, PROGNOSIS, AND OPTIMAL TREATMENT SELECTION

by
Beau Norgeot

DISSERTATION
Submitted in partial satisfaction of the requirements for degree of
DOCTOR OF PHILOSOPHY

in

Pharmaceutical Sciences and Pharmacogenomics

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

Approved:

DocuSigned by:
Atul Butte ______________________________ Atul Butte
51E283C7897F40E...
Chair

DocuSigned by:
Cucina, Russ ______________________________ Cucina, Russ
[illegible]74...

DocuSigned by:
Jinoos Yazdany ______________________________ Jinoos Yazdany
E1F7BDB9E03644B...

____

____
Committee Members

Acknowledgments

This short page will never be able to adequately express how grateful I am.

None of the work in the following pages would have happened without the incredible determination and support of my wife, Sara, who was willing to put her entire life on hold… for years…. while I pursued something that I thought "was really important".

My entire family, from mother to father(s), brothers and sisters, half, whole, and in-law, lent their support in many forms, assuring me that I was not, at least entirely, crazy for choosing to follow this path. Adoptive Uncle Dick kept me fed and looked out for me in many more ways than he needed to.

My primary mentor, Atul J. Butte, has taught me that research with real-world impact is more important than fancy research, that new research is only valuable as far as you can communicate its value to the rest of the world, and to always consider the biggest picture possible. Atul may not have taken me as a student if Ted Goldstein hadn't been willing to read my undergraduate thesis. Ted put academia in perspective for me and provided invaluable career coaching. Marina Sirota, welcomed me into the Lab, helped direct me through my rotation, focus me through quals, and acted as an experienced sounding board as I tried to figure out 'what next'. Jinoos Yazdany and Gabriela Schmajuk brought me into their research community; the mentoring that they have provided has been largely responsible for the successes, however modest, that I have had. Russ Cucina helped me to understand the larger healthcare landscape and that there's always time for coaching. Boris Oskotsky, Ben Glicksberg and Dima Lituiev kept me from coding alone, provided friendship, and made me a better informatician. I have benefited

tremendously from the conversation and insights from Esteban Burchard and Michael Keiser. My dear friend Timothy Patrick made all of this possible in so many ways. Last, but unquestionably not least, are my cohort members, friends, colleagues, and brothers-in-arms; Jasleen Sodhi and Julia Cluceru. Without the two of you to complain to, to bounce ideas off, to share dreams and commiserate with, I probably would have dropped out; because while science is wonderful, the path to a PhD is not.

Deanna Kroetz, Dexter Hadley, Idit Kosti, Milena Gianfrancesco, Dana Ludwig, Tom Peterson, Deb Datta, Vivek Rudrapatna, Charlotte Nelson, Matthew Kan, Sharmila Majumdar, and John Witte, as well as my entire PSPG cohort, the Burchard Lab, and the Keiser Lab shaped the thinking and experiences that have become my degree. There are many people who made really important contributions to my PhD whose names do not appear here.

From the outside, a PhD seems like an incredible individual accomplishment. From my own experience, the successes that I’ve had have had at least as much to do with the people that I’ve been fortunate enough to become involved with than any vision or talent that I have had.

Abstract

DEEP LEARNING IN PERSONALIZED MEDICINE: APPLICATIONS IN PATIENT SIMILARITY, PROGNOSIS, AND OPTIMAL TREATMENT SELECTION

Beau Norgeot

Two information technology revolutions are colliding in medicine. The first revolution has been the digitalization of health data, specifically Electronic Health Records (EHR). These records contain the details of who we are as patients, our ailments, treatments, and outcomes. Tragically, despite billions of dollars in investment from the US government, hardly any of this data is being utilized to better understand medicine or improve healthcare. This is largely because the data is voluminous, sparse, complex, and poorly formatted; making it unsuitable for traditional analytics methods. However the second revolution, modern Artificial Intelligence, specifically deep learning, provides tools, in the form of algorithms, to address exactly these problems. The primary difference between these modern algorithms and older ones is that the former are able to learn, more or less on their own, how to transform large complex data into a format that makes it easier to use and learn from.

In this dissertation, I have developed methods to apply deep learning to digital health data. Doing so, I have shown that we can predict the future health of individual patients with highly complex diseases, produced approaches to understand and leverage what these complex models are learning, and provided a framework for how healthcare systems of the near future could automatically learn to improve care daily.

For the first time in history, we are in a position to learn from the combined knowledge of tens of thousands of physicians and their experiences caring for hundreds of millions of patients. The

potential transformations to healthcare are difficult to fully fathom, but certainly include safer, more powerful and efficient medicine, and a rapid speed up in new medical discoveries and treatments. Despite the promise, we must proceed carefully, balancing the great need to collectively use our data for better medicine with the individual right to privacy.

Table of Contents

List of Figures

List of Tables

Chapter 1

Desideratta

1.1. Readme

This chapter serves as a brief summary of the complete dissertation. Here I will discuss the elements of the problem at hand and provide a concise description of each chapter to follow.

1.2. The Checklist

The problem, in a nutshell, is that human health is highly individual and complex. We finally have a large amount of detailed health information (EHRs) to study, but we lack good tools to fully utilize it to improve care or learn more about human health and disease. Additionally, these data points that we do have are made up of living people who have both legal and human rights to privacy, so we must proceed carefully and respectfully.

Given the above considerations, a checklist to launch this new field of Deep Learning Healthcare that would enable the systematic utilization of data on previous actions and outcomes to enable smarter ongoing choices, might look something like this:

1. Identify deep learning approaches to modeling individual patients, which are characterized by sparse, lumpy, longitudinal data
2. Apply those approaches to the simplest, safest task that addresses a meaningful clinical problem
3. Explain how the models are making decisions to foster trust and expand knowledge
4. Ramp up: Move beyond simple data to address deeper problems with greater granularity

1.3. The Following Chapters

Chapter 2 “Towards a Deep Learning Healthcare System” describes the vision of how the intersection of digital health data with modern AI will empower healthcare systems in the near future to automatically learn to improve decisions and care daily.

Chapter 3 “Modeling Longitudinal Electronic Health Records” presents the results of detailed algorithmic experiments to identify optimal deep learning approaches to represent longitudinal patient data.

Chapter 4 “Predicting the Future” applies the methodology learned in the previous chapter and applies it to the problem of forecasting future individual health outcomes for patients with Rheumatoid Arthritis, a complex auto-immune disease.

Chapter 5 “MI_CLAIM”, or Minimum Information for Clinical Artificial Intelligence Modeling studies, establishes a standard for designing, recording and reporting AI-based clinical informatics studies. This will, in turn, facilitate transparency and the establishment of trust and ultimately enable the utilization of such models in the clinical setting.

Chapter 6 “MAgEC”, Model Agnositc Effect Coefficients, offers a single, easily interpretable, method to analyze how any machine learning model (from logistic regression, to random forests, to deep neural networks) is making decisions. It provides global and local explanations and enables the direct comparison of what multiple different models have learned from the same data.

Chapter 7 “DeepMANN” is an inflection point, here we begin to move into data of greater complexity as I demonstrate that the combination of feature selection and relatively simple deep networks can be taught to identify clinical phenotypes from microarray data with accuracy that rivals or exceeds substantially more complex approaches.

Chapter 8 “Philter”, Protected Health Information Filter, opens up the potential to mine the richest information in the EHR, physician notes, by providing an algorithm that removes patient-specific information from each note, leaving only the relevant medical information about the patient’s condition, treatment, and response. 25,000 randomly selected notes (the largest such corpus in the world) were marked for PHI, word for word, by a team of trained expert annotators.

Chapter 9 “Deep Cumulative Dosage Information” details the process by which deep learning can be used in the Natural Language Processing space to automatically determine the total cumulative dose of a steroid that a patient has received directly from the EHR ‘Sig’ field. Cumulative steroid use is associated with several serious health conditions and is also an important surrogate outcome for auto-immune treatments.

Chapter 10 “Medical Research Topic Labeling” illustrates early successes of employing deep learning to automatically annotate complex medical datasets, making important information easier to find and reducing potential sources for labeling-error.

Chapter 11 “Conclusions” provides the summary and concluding thoughts on the overall paper.

Chapter 2

Towards A Deep Learning Healthcare System

2.1. Permissions

This Chapter was originally published in Nature Medicine (Nature Medicine volume 25, pages14–15 (2019) and is reproduced here with permission.

2.2. Call For A Deep Learning Healthcare System

We are currently witnessing two incredible information-technology revolutions colliding in medicine. Electronic health records (EHRs) are capturing the thoughts, orders, images, and outcomes of the best trained physicians. Advances in machine learning are beginning to supplement clinical medicine. But breakthroughs still remain fully unrealized because these revolutions are siloed. While the raw materials exist to learn from current actions and outcomes in medicine, they are not systematically utilized to improve the practice of medicine.

Nearly every other industry uses data on previous actions and outcomes to enable smarter ongoing choices. Amazon targets product recommendations based on similar customers' shopping patterns. Google updates its searches based on the outcomes of previous searches. Waze uses information on drivers traveling similar routes to optimize directions.

Why is medicine, as an industry, still left out?

The roadblocks to bringing medicine into the data-driven era are operational and cultural. While many have written about inefficiencies in the US medical system relative to rising healthcare spending[1] and the challenges in improving quality[2], the US medical system is a competitive one, meaning competitors are incented not to fully share data, pricing, and costs. There is a significant room for improvement, and potential to better use the data we do have.

While EHRs have known challenges[3,4], they now represent the legal medical record and are complete enough to enable another physician to completely care for a patient. This data is perhaps among the most expensive in the US, given that physicians are paid to enter much of it. Of course, EHR data must only be used in safe respectful ways, but it will be a tragedy if this data is not used to improve the practice of medicine.

Over ten years ago, Lynn Etheredge[5] and others[6] proposed the Learning Health System, where millions of EHRs could be used to inform medical practice and policy. But these visionaries were still proposing a system in which physicians mediate the learning. Now with nearly 80% of medication orders captured electronically and more than 1.7 billion prescriptions per year electronically tracked[7], combined with 98% of hospital systems now using EHRs[8], we can envision computer systems that learn how to improve the medical system by themselves.

It is now time to safely bring huge medical data repositories and advanced learning algorithms together with physicians to make a Deep Learning Healthcare System (see Figure 2.1). Deep learning (DL), the newest iteration of machine learning methodologies, is now performing at state-of-the-art levels in previously difficult tasks, including image analysis, language processing, information retrieval, and forecasting. DL is well suited for medical data as it can identify patterns in sparse, noisy data, and requires little input feature engineering. Current successes have shown performance that meets or surpasses experts, but perhaps more importantly, they can be run in real-time within or across entire hospital systems. We propose that future physicians will be armed with insights from models continuously trained and updated on real-world clinical data to make more accurate diagnoses and individually optimized treatment decisions.

Is there one optimal way to practice medicine? Imagine ten physicians faced with a single clinical conundrum (choice A, B, or C) on one patient. If these ten were provided with the maximum possible information about a patient in a clear format, from physical exam to preferences, the world’s literature, and data on similar patients, should all ten physicians reach the exact same choice for this clinical decision? We know today they probably would not, but shouldn’t they? If the answer is yes, then medicine is fundamentally machine learnable.

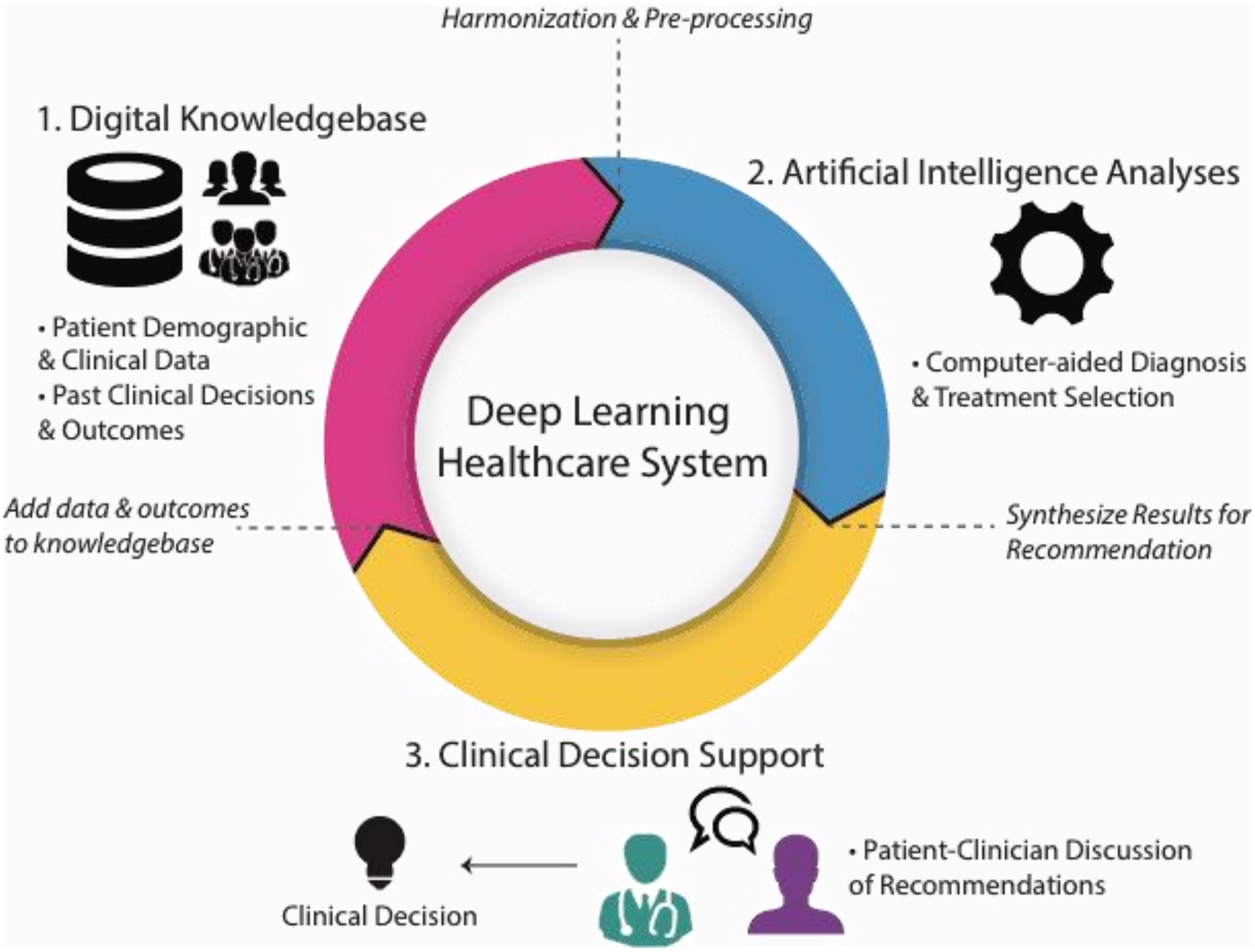

Harmonization & Pre-processing
1. Digital Knowledgebase
• Patient Demographic & Clinical Data
• Past Clinical Decisions & Outcomes
2. Artificial Intelligence Analyses
• Computer-aided Diagnosis & Treatment Selection
Deep Learning Healthcare System
Add data & outcomes to knowledgebase
Synthesize Results for Recommendation
3. Clinical Decision Support
Clinical Decision
• Patient-Clinician Discussion of Recommendations

Figure 2.1: A Deep Learning Healthcare System

Figure 2.1: (1) The EHR contains the record, in digital form, of the demographics, symptoms, vitals, labs, and diagnoses of each of the individual patients that have been seen as well treatment decisions made by their physician and often the resulting outcomes for the patient. (2) The amount of data that a physician must synthesize in one visit in order to make decisions is large and growing. Deep Learning can identify complex patterns in patients, treatments, and outcomes in an automated manner in near-real-time, distilling them into individualized recommendations for physicians based on real-world data. (3) Physicians can review the Deep-Learned recommendations, comparing it to their own knowledge and experience, then discussing the options with their patient before finalizing a decision. The system is iterative in nature, improving with every patient-physician interaction.

Many health conditions present in heterogeneous ways, making it challenging to establish an accurate diagnosis over time. The treatment regimens for many complex conditions require physicians to stay aware of the latest options and evidence. A Deep Learning Healthcare System would enable all physicians to practice at the same level of expertise as a panel of the very best physicians. Since deep learning models could be shared between hospitals without the privacy risks of sharing patient data, the potential is nearly limitless to create a new system of precision medicine learned from the decisions and outcomes of diverse physicians treating diverse patients.

Fogel & Kvedar[9] have insightfully noted that bringing AI to medicine will not sideline doctors, but will instead enhance their strengths. Physicians, empowered by patterns and evidence derived from large-scale real-world data, will be able to focus on the uniquely human elements of their profession for which they are best trained. Tasks which cannot be performed by a machine because they require emotional intelligence, such as asking careful questions of the patient to uncover more nuanced symptoms, and building trust through personal relationships by using human intuition, will still be unique qualifications of physicians to guide the implementation of the computationally optimized diagnoses and treatment plans of the future.

References

1. World Health Organization. Healthcare Expenditures per Country. 2016.
2. Organization for Economic Co-operation and Development. Life expectancy at birth. 2016.
3. Howe JL, Adams KT, Hettinger AZ, Ratwani RM. Electronic Health Record Usability Issues and Potential Contribution to Patient Harm. *JAMA.* 2018;319(12):1276-1278.
4. Wang MD, Khanna R, Najafi N. Characterizing the Source of Text in Electronic Health Record Progress Notes. *JAMA Intern Med.* 2017;177(8):1212-1213.
5. Etheredge LM. A rapid-learning health system. *Health Aff (Millwood).* 2007;26(2):w107-118.
6. Friedman CP, Wong AK, Blumenthal D. Achieving a nationwide learning health system. *Sci Transl Med.* 2010;2(57):57cm29.
7. Surescripts. 2017 National Progress Report. 2017.
8. Office of the National Coordinator for Health Information Technology. Hospitals Participating in the CMS EHR Incentive Programs. August, 2017.
9. Fogel AL, Kvedar JC. Artificial intelligence powers digital medicine. *npj Digital Medicine.* 2018;1(1):5.

Chapter 3

Modeling Longitudinal Electronic Health Records

3.1. Permissions

Portions of this chapter was originally published in NeurIPS 2018 and JAMA Network Open (JAMA Netw Open. 2019;2(3):e190606) and is available under a Creative Commons license.

3.2. Time aggregation and model interpretation for deep multivariate longitudinal patient outcome forecasting systems in chronic ambulatory care

Clinical data for ambulatory care, which accounts for 95% of the nations healthcare spending, is characterized by (relatively) small sample sizes of longitudinal data, unequal spacing between visits for each patient, with unequal numbers of data points collected across patients. While deep learning has become state-of-the-art for sequence modeling, it is unknown which methods of time aggregation may be best suited for these challenging temporal use cases. Additionally, deep models are often considered uninterpretable by physicians which may prevent the clinical adoption, even of well performant models. Using EHR data on a complex autoimmune disease from 2 hospital systems with highly diverse patient populations, we show that time-distributed-dense layers combined with GRUs produce the most generalizable models and we provide a framework for the clinical interpretability of the models.

3.2.1. Introduction

Chronic ambulatory disease care is associated with the overwhelming majority of death, disability, and healthcare spending Buttorff et al. (2017), for Medicare & Medicaid Services et al. (2016) in the United States. Successful predictive modeling in this setting has the potential to significantly improve clinical care, patient quality of life, and healthcare efficiency. Identifying the best methods and functions for time aggregation will be a critically important part of building models that perform well in these types of settings. However, even well performant models may not be sufficient to warrant the clinical adoption of artificial intelligence (AI) to longitudinal patient care. Deep learning has traditionally been met with resistance in the clinical community due to a general sentiment that the models function entirely as uninterpretable black boxes. While we agree that the use of black boxes in clinical care should be avoided wherever possible, we posit that deep time series models need not be black boxes at all.

By transferring and extending a traditional method to calculate variable importance in machine learning models known as Permutation Importance Scoring Breiman (2001) to deep time series modeling and by developing a method of visualizing the final model-learned patient representations as clusters we aim to show that it is possible to interpret the driving factors behind model predictions on both the patient and population levels. These methods can not only contribute to model interpretation but could be used in the future to generate and test medical and pharmaceutical hypotheses in this space in which individual progression and response to treatment for many of the diseases may not be well understood.

We selected Rheumatoid Arthritis (RA) as a use case. RA is a common (1% nationwide) complex chronic autoimmune disease with unknown causes along with highly individualized responses to therapeutics and disease progression. It is associated with significant morbidity and a high cost of care. Our goal was to examine the impact of time aggregation strategies on deep time series models which used a patient's history of labs, medications, and disease activity along with their current Machine Learning for Health (ML4H) Workshop at NeurIPS 2018. arXiv:1811.12589v1 [cs.LG] 30 Nov 2018 treatment plan and current clinical measurements to forecast whether a patient's disease activity would be controlled or uncontrolled at their next visit. The highest performing model was then examined for interpretation using the approaches described above.

3.2.2. Methods

Electronic Health Record (EHR) data were extracted from two rheumatology clinics with significantly different patient populations and provider treatment patterns, a University Clinic (UC) and a public Safety Net (SN). Patients from the larger UC cohort (n=578) were split into three groups [train (n=369), validation (n=93), test (n=116)] using stratified random sampling on the primary outcome which was the binary category of controlled or uncontrolled disease activity at their most recent clinical visit. Patients from the smaller SN cohort (n=242) were split into two groups, train (n=125) and test (n=117) using stratified random sampling as previously described. The patient data were grouped into three windows of one hundred days each (which corresponds to the median number of days between visits) to overcome unequal length of time between visits. Data within each window was further aggregated, se Relative Difference = (mean(permuted score) - original score) / (original score)

3.2.2.1. Population Differences

We generated a graphic, which we have called a Confusion Plot, by extracting the final dense representation learned by the fully trained model for each patient and plotting them using T-SNE, colored by outcome category, to assess the coherence of the representations learned by the model. We performed this experiment for each cohort and compared the results side by side to determine the differences in patient representation for the model in each patient population.

3.2.3. Results

3.2.3.1. Time Aggregation

Results from the time aggregation experiment can be found in Table 3.1. Using a Time Distributed Dense (TDD) layer as the input layer provides the single largest increase in predictive performance as seen by the relative difference between the TDD and Dense architectures. Using a convolutional layer with non-causal padding after the TDD provides a modest improvement over a dense layer, while using causal padding increases performance even further. The top performing architectures used a recurrent layer following the TDD with GRUs outperforming LSTMs.

Table 3.1: Time Aggregation Experiment Results

Function	Dense	TDD Dense	TDD GRU	TDD LSTM	TDD CNN	TDD Causal CNN
AUC	0.778	0.817	0.845	0.838	0.821	0.832
95% CI	[0.683, 0.864]	[0.731, 0.894]	[0.753 ,0.914]	[0.743 ,0.911]	[0.727, 0.897]	[0.740, 0.906]

3.2.4. Interpretability

3.2.4.1. Longitudinal Permutation Importance Scoring

More recent time windows are more important than more distant time windows. CDAI history is of greatest importance followed by steroid prescription (Prednisone, a reasonable surrogate for all steroids representing 55% of steroids prescribed in our data set). Changing a patient's previous DMARD treatment strategy at the visit prior to the most recent visit was of significant performance as were the presence or absence of certain specific DMARDs (See Figure 3.1).

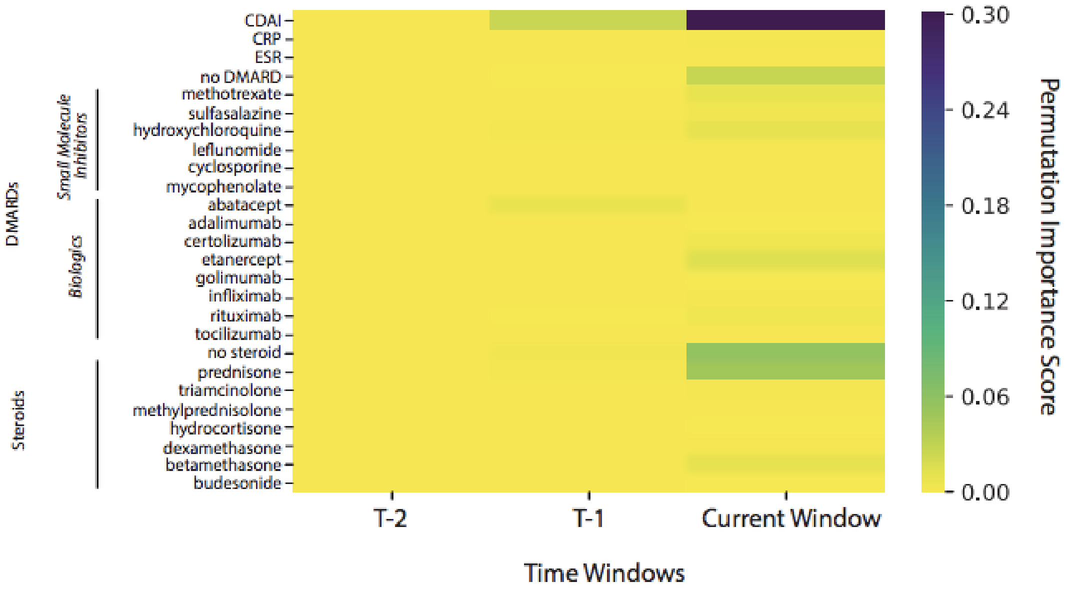


Figure 3.1: Permutation Importance Scores

3.2.4.2. Confusion Plot

In both the UC and SN populations the final patient representations that the model learned formed a one-dimensional manifold (a curve). The model is clear in both clinics about patients that will definitely be at one end of the disease activity spectrum or the other at their next clinical visit. However, the confusion plots are clearly different between the two populations. In the UC cohort (see Figure 3.2), the presence of Controlled and Uncontrolled patient representations mix in just one pocket of close proximity to each other in the middle of manifold, while within the SN Clinic cohort there are multiple of these highly proximus representations that begin to occur in pockets much closer to the tails.

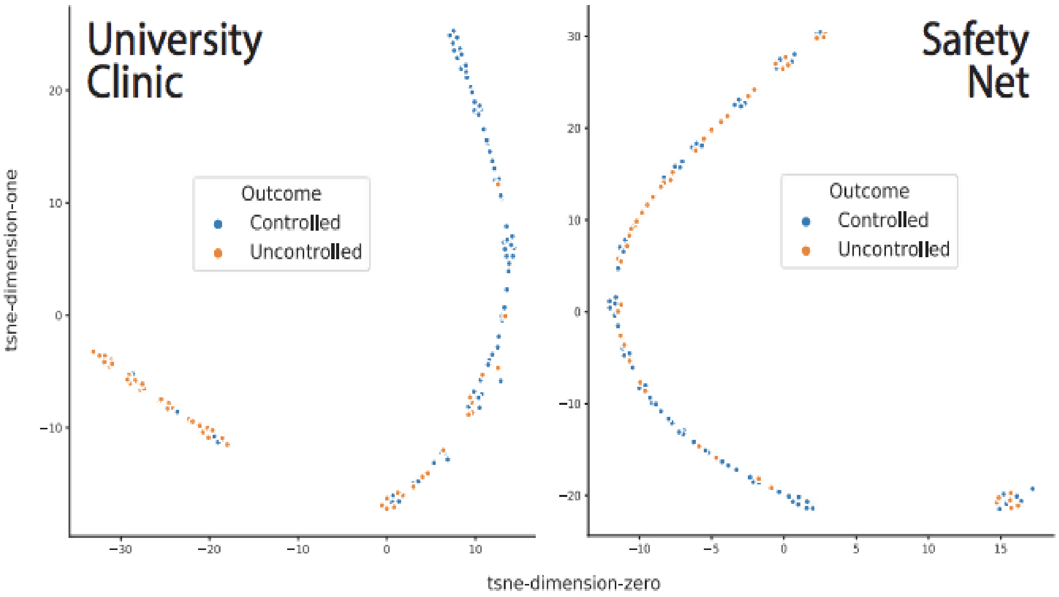


Figure 3.2: Confusion Plots

3.2.5. Discussion and Conclusions

In this study, we compared different time aggregation functions for ambulatory outcome forecasting and provided a framework for interpreting models in this setting. We found that using a time distributed dense layer (which uses the same function to re-weight input features across all time windows), followed by recurrent modeling of the re-weighted windows produced the best results. Longitudinal Permutation Importance Scoring reveals that newer time points are most important, and that recent disease activity scores are important but that quantitative inflammatory markers are not. Prescription of new steroids at the current visit, which we interpret to clinically indicate currently uncontrolled disease, and maintaining or switching to a new DMARD, we interpret to act as surrogate for the patient and physician believing that the current DMARD is working, are also important. There are some findings such as the influence of certain DMARD changes and steroids additions that require further examination to determine whether these finding may lead to new treatment strategies or are the result of confounding by indication. A potential limitation of the Longitudinal Permutation Importance Scores as implemented in this paper is a lack of directional effect between future disease activity and each variable within each window. This could be solved be reporting the changes in probability instead of the changes in auROC associated with each permutation, though additional considerations will need to be made for the continuous variables.

An additional straightforward application of directional longitudinal permutation importance scoring would be to permute medication choices to optimize probabilities for a successful outcome. Examining the Confusion Plots in the current study, the single mixed patient pocket at the UC seemed to indicate a natural transition between patients who have Controlled and Uncontrolled disease state at their next visit, while the multiple mixed pockets for the SN patients perhaps indicates that strong confounding factors are driving outcomes for many patients. Since the patients in the SN cohort, unlike those in the UC, are known to be predominantly non-White and be considerably less likely to have private insurance, these findings support suspected social determinants of health within the population that should be further examined.

In summary, Longitudinal Permutation Importance Scores can be extended from traditional machine learning approaches into longitudinal deep learning methodologies, providing insight into variable significance over time. Confusion Plots, which visualize model-learned dense patient representation vectors, can used to search for sub-cohorts, indicate the presence of potential confounding factors, and examine differences between subgroups and populations. Taken together, we found that longitudinal deep learning can be successfully applied to ambulatory disease forecasting and that the resulting models can be interpreted in a straightforward manner. We expect that these methods will be used to facilitate the adoption of deep learning in the field of clinical medicine.

3.3. Supplementary Methods

3.3.1. Data

3.3.1.1. Primary Cohort (UCSF)

In order to use real-world longitudinal patient data to build and evaluate our models, we utilized the resources made available by the UCSF Clinical Data Research Consultations Team. Each day the team extracts the EHR data from Epic Chronicles into the Epic Clarity relational database (RDB). A subset of that RDB is used to update the Epic Caboodle data warehouse. The Caboodle data is then de-identified using the Safe Harbor method: Private information such as names and addresses are removed. Key personal identifiers are replaced with randomly assigned surrogate identifiers. Dates are shifted by a random number of days from 0 to -364 so that the true date is known no more precisely than the year, however date shifts are kept consistent for each patient so that their chronology is accurately maintained. Ages are adjusted so that patients > 90 years old are presented as 90 years old. Once a month, the de-identified data is extracted to a set of delimited flat files, which our group stands back up into a SQL database. This database contains longitudinal information on over 900,000 individual patients dating from January 2014 to the present.

3.3.1.2. Replication Cohort (ZSFG)

The IRB for ZSFG did not require de-identification of patient records. EHR data were directly accessed using the eCW product "eBO reports" which runs on an IBM Cognos platform.

3.3.1.3. Variables Utilized in Model

Given the relatively small number of patients available and the complexity of time series models, we chose to only include variables with known clinical significance in the models. These variables are known to be associated with disease activity, however no study that we are aware of has shown them to be predictive of future disease activity. This approach reduces the risks associated with including all variables in the EHR, including under-fitting (inability to separate signal from noise) or over-fitting (confusing noise for signal). We include the following: prior CDAI, ESR and CRP, DMARDs (Supplementary Table 3.2), oral and injected glucocorticoids, autoantibodies (presence of rheumatoid factor [RF] and/or anti-cyclic citrullinated peptides [anti-CCPs]), and demographics (age, sex, race/ethnicity). Medication names were standardized by first using the R scripting library MetaMap[1] and then programmatically removing any remaining characters associated with delivery or dosage. Medication names were then mapped to the list of DMARDs. Steroids were included if their pharmaceutical class was labeled as "glucocorticosteroid" in the EHR and their route of administration was either oral or injection. All patient medications that did not map to either a DMARD or steroid were dropped. Most machine learning libraries, including the TensorFlow[2] library that we planned to use for modeling, do not accept string values within tensors. Therefore, we encoded medications using a dictionary mapping the drug name to a unique integer value (e.g., Methotrexate= 1) in each patients record. We chose to include only the first occurrence of each medication given the lack of reliable medication stop dates in the EHR.

3.3.1.4. Modeling Input Formats

We considered two different formats for representing a patient's longitudinal trajectory as input for modeling. The first method was a Sequential string of events. In this format each patient's events follow the exact chronology in which they appear within the EHR. As an analogy, in this format a patient's trajectory is like a sentence and the goal of the model is to predict the final word the sentence (always a CDAI score in this case of either controlled or uncontrolled). The potential advantages to this format are based on its flexibility: A patient's trajectory is presented to the model in the exact order that it occurred in the hospital, each patient can have an arbitrary number of types of variables, time between patient events can be modeled with fidelity, differences in the numbers, types, and order of patient events can be easily represented for each individual patient. The potential disadvantage of this format is also related to its flexibility: different patients will have different numbers of events and therefore potentially drastic differences in the length of their trajectories which necessitates a more complex model capable of handling the longest and most complicated sequence. From a back-propagation perspective, longer sequences fed into an RNN increase the likelihood of encountering a vanishing gradient thus hindering the models ability to learn. Within this format the order of variables cannot be anticipated for each patient. Therefore, we prepended the raw values for each variable with a string containing the name of the variable. For example a raw CDAI score of 10 was converted into 'cdai10'. Each unique string was then mapped to a unique index. To address the loss of 'nearness' introduced by converting continuous variables into uncorrelated strings, we added an embedding layer to all Sequential architectures. Ideally, this would not only allow the network to learn that cdai10 was more similar to cdai11 than cdai50, but also that low CDAI scores were more similar to high ESR or CRP values that appear in close longitudinal proximity for patients.

Example Sequential format:

- Conceptual: (event1_type_value, dT1, event2_type_value, dT2,eventn_type_value)
- Using Real Variables: (ESR11, CRP22, ESR24, methotrexate, CDAI21, PROM50)

The second input format that we explored was to force our data to conform to sliding time windows of a fixed interval as is done with traditional Time Series forecasting. In this format, a window interval is decided upon (for example, 3 months) and variables are decided (for example CDAI, Steroids, CRP). In each window, a single value is entered for each variable. This format is almost universally employed for fields in which variable recording can be guaranteed to be consistent at specific time intervals, such as the stock market or EKG measurements. Outpatient care is by nature inconsistent in the frequency of a patients visits to their care provider and the number of variables that are measured for the patient at each visit. This inconsistency is the result of a combination in patient adherence, the individual and cyclic nature of chronic disease and its severity, and provider preference the types and frequency of measuring patient variables and changing treatment strategies. When the sampling frequency and variable measurement is inconsistent, such as for patients with RA (chronic disease), strategies must be employed to deal with the cases of either having no history of a variable within a given window or having more than one value present for a variable within a single window. Missing values were imputed with zeros since the field of deep learning has learned empirically that neural networks learn to ignore zero-values assuming that they do not legitimately occur as real input values, which is true for our data. Additionally, there is no sensibly accurate method for imputing the variables that we have selected as they can change drastically over time within a person and have only modest correlation to each other at best. Attempts at non-zero imputation were likely to induce spurious correlations, injecting noise into the signal, especially when considering our the relatively small size of our training cohort. Since our goal was to predict the most recent event, if multiple values occurred for a given variable within a single time window, we selected the most recent value. If a patient's clinical history was too short to fill all windows for a particular experiment, all values for that patient's window were replaced with zeros.

3.3.1.5. Longitudinal Modeling using Deep Learning

Since patients and their outcomes change over time, and deep RNNs have previously demonstrated superior performance to traditional machine learning methods for chronological EHR data[3], we focused on deep learning models but experimented with many different possible ways to represent time dependencies.

Unlike Random Forests, Support Vector Machines, or Linear Regression, there is no discrete Deep Learning algorithm. Deep Learning is a term used to describe machine learning models based on neural networks of multiple layers and algorithms for their optimization such as stochastic gradient decent and its derivatives. Each layer is composed of a varying number of nodes. The manner in which the nodes operate and are connected to nodes in other layers determines the how the layer transforms the input it receives into the output that it generates. Deep Learning can be viewed as a hierarchical transformation of the input data, each layer acting as a distinct function changing the data in a different way, into the representation of the original input data that makes the predictive task as straightforward as possible. An architecture is an arrangement of layers placed together and representing either the modeler's theory or an experimental finding about which functions will generate the best representation of the input for a given problem.

3.3.1.6. Overview of Relevant Deep Learning Layer Types

We considered 5 potential layer types to represent patient longitudinal trajectories: Dense, Time-Distributed, Convolutional, and Recurrent (LSTM, GRU). Each of these layers have distinct methods for generating representations of their input data. Dense layers have no representation of chronology or proximity. Therefore, neither the order of patient events nor which events happened close in time for a patient can be represented. In this way, models composed entirely of dense layers, known as Multilayer Perceptrons, are conceptually similar to Random Forests or kernelized regression. Time-Distributed layers learn a single dense mapping function that is applied to every timestep of the input data. In our case, this can be thought of as a generating one representation for each encounter or window. Convolutional layers can represent proximity but not chronology. Therefore, the representations that they generate can account for which events happened close in time for a patient, but not the global order of events for a patient's trajectory. Models composed of Convolutional blocks, known as Convolutional Neural Networks, offer a distinct advantage in that they use a local pooling of proximity values to reduce the size of the input space, thus reducing the complexity of the model and in-theory increasing its ability to generalize. Despite lacking an explicit ability to directly represent the order of sequences, CNNs have shown to work very well as language models[4], including extracting information from clinical text[5]. LSTMs and GRUs are different variations of Recurrent Neural Networks (RNN), which are the family of architectures that explicitly model chronological sequences of events. LSTMs and GRUs have slightly different mechanisms for learning sequential representations which can lead to differences in performance on different data sets. In general, LSTMs are more robust but due to that they are slower to train.

3.3.1.7. Model Training

The UCSF cohort was divided into three sub-cohorts for model building and testing: training, validation, and testing. To ensure that the sub-cohorts were representative of the overall population, we calculated the proportion of patients in each CDAI outcome category (60% were Controlled, 40% were Uncontrolled). We then performed a Stratified Random Split, keeping 20% (n = 116) of the patients aside for testing (these patients' data were never trained on, the data was used only to test the final model) and using 80% for model training and development. We then performed an additional stratified random split on the patients assigned, allocating 80% for direct model training (n = 369) and the remaining 20% (n=93) for model validation. This validation cohort was used to assess the generalizability during the model selection process (by varying model architectures and hyperparameters). All continuous variables (CDAI, ESR, CRP) were then linearly scaled to range between zero and one with min/max scaling using the minimum and maximum values for each of these variables found in the training cohort. Thus, the training data were used for model optimization and the validation data were used to guard against overfitting during model selection and hyperparameter tuning. Once a final model was produced, we combined the validation cohort with the training cohort to train the model on both cohorts before the final single evaluation on the test cohort.

The ZSFG patient cohort was less than half the size of the UCSF cohort and we chose not to involve it in model selection. The ZSFG cohort was split in two, the test cohort was matched to the size of the UCSF test cohort as closely as possible (n=117) so that model performance could be evaluated across equally sized patient populations. A training cohort, comprised of the remaining patients (n = 125), was created from the remaining patients. Membership in the cohorts was assigned through a random stratified split as described above.

The goal of any deep learning architecture is to learn a representation of the original input data that maximizes the success rate for the predictive task. In this case, the input data is each individual patients' clinical RA trajectory and the task is to predict what each patients' disease activity state will be at their next visit. The final representation that the architecture generates is captured by a vector which represents the patients chronology, which we have called a Patient Trajectory Vector (PTV). During training, the deep learning architecture is used to generate a PTV. The PTV is fed into a logistic classifier which makes a binary prediction (controlled, uncontrolled) of the patient's disease state at the time of their next visit. The difference between the sigmoidal prediction of the outcome and the patient's actual outcome is the error. The error is then back-propagated into the PTV and then into the architecture itself. The parameters for the architecture are then gently updated in directions that would have led to a better PTV representation for that patient resulting in a sigmoidal output closer to the ground truth (zero for Controlled, one for Uncontrolled). These updated transformations are then applied to next training patient sample and evaluated. In practice, models are updated based on the results of batches of samples instead of single samples because batch-training has been shown to produce models that converge faster[6].

For models that included both patient variables that changed over time, such as lab values and CDAIs, and static variables that did not change over time, such as demographics, the variables were separated according to whether or not they were time-dependent. These separate inputs for each patient were fed into two independent deep networks; a recurrent network for the time-dependent variables and a purely dense network for static variables. The two network outputs were concatenated to form a final joint representation and passed to the logistic classifier. Back propagation flowed through both networks allowing joint learning of static and time-dependent representations.

There are many different strategies that can be applied for model optimization. The most common methods for optimization include: experienced intuition, grid searches, random searches, or some form of Sequential Model-based Global Optimization (SMBO) techniques. Since, to our knowledge, no model architectures for multivariable time series deep learning to predict future health outcomes for a chronic disease have been published, there was no data to guide intuition for selecting the optimal variables. While grid searches are popular because they are easy to conduct, Bergstra and Bengio[7] have shown that it is more efficient to randomly search through values while employing a method to intelligently narrow the search space than it is to loop over a fixed sets of hyperparameter values in a grid. SMBOs are algorithms that begin with a random search over the hyperparameter space, and then use the results of the models built with that search to fit one or more surrogate functions that describe the relationship between a set of possible hyperparameters and model generalization. The algorithm then begins optimizing the surrogate function with the goal of identifying points in the hyperparameter space that will lead to improved model performance on data unseen by the model during training.

Our goal was not only to identify the best performing single architecture and hyperparameters, but also to uncover trends in performance associated with different ways of representing patient trajectories and the types of patient variables that were most essential for accurate predictions of future patient outcomes. Therefore, we set up separate SMBO experiments for each combination of architecture family: dense, time-distributed, convolutional, and recurrent architecture as well as architectures combining different layer types. Once the best combination of patient value types and model architecture was identified, additional experiments were performed to determine the impact of the length of patient history to include as input on model performance. Overfitting, the substantial divergence of model log-loss performance between data used for training and unseen data, was rigorously monitored by comparing the model's performance on the training cohort to that of the development cohort. Training for any model was stopped as soon as overfitting was detected. Models were ranked by their generalization or hold-out accuracy on the development cohort, their performance on the training data was never recorded. Model training, optimization, and selection were performed using the TensorFlow[2] computational engine wrapped with Keras[8] as a front end on Amazon Web Services Elastic Cloud (EC2) P2XLarge Linux GPU servers. Additional python libraries were used for data preprocessing and model evaluation including Pandas[9], Matplotlib[10], scikit-learn[11], and Numpy[12].

3.3.1.8. Transfer Learning and Fine Tuning

Transfer Learning is the deep learning practice of taking a model that has been fully trained on one data set and updating the model's parameters by retraining the final dense layer on data from a new data set while keeping the rest of the model parameters frozen. This is generally considered most appropriate when the two datasets are highly similar. For our work, that meant training the model on the UCSF data, then updating the weights for the PTV using the training cohort from ZSFG while keeping the Time-Distributed Dense and GRU layers frozen.

Fine Tuning is a general case of Transfer Learning where layers other than the final dense layer of a model, including potentially all layers, are updated by training on a new data set. This approach is generally most applicable when the two data sets are part of the same general domain but are otherwise very different from each other. We experimented successively unfreezing one additional layer from the top down.

3.3.2. Supplemental Discussion

Deep Learning is notorious for requiring training sample sizes far above the number of EHR records for patients with most chronic diseases (even if patients at many large hospitals were aggregated). To overcome this, during our model building process we leveraged physician knowledge and experience to select a small number of raw variables with known clinical importance. This reduced the number of variables the model needed to sift through to learn from, and presumably reduced the number of patient samples necessary to properly train models, by many orders of magnitude. We found that beginning with a small number of clinically important variables, even if the consequences of their complex time-dependent interactions are not perfectly understood, has the added benefit of increasing model interpretability.

The deep neural architecture that performed best was constructed in a way in which a human might approach the problem. The time-distributed layer essentially creates a summary of each time-window for the patient. The recurrent units then look for longitudinal patterns in the chronological summaries from each patient. The dense PTV then generates a single representation of the patient's overall trajectory. It is this complete trajectory representation that the logistic classifier uses to forecast the patient's future CDAI category. The fact that this model also contained the smallest number of trainable parameters and utilized a high degree of regularization also makes sense intuitively, especially considering the relatively small size of the training cohort and the large differences between the two testing cohorts.

3.3.3. Supplementary Tables

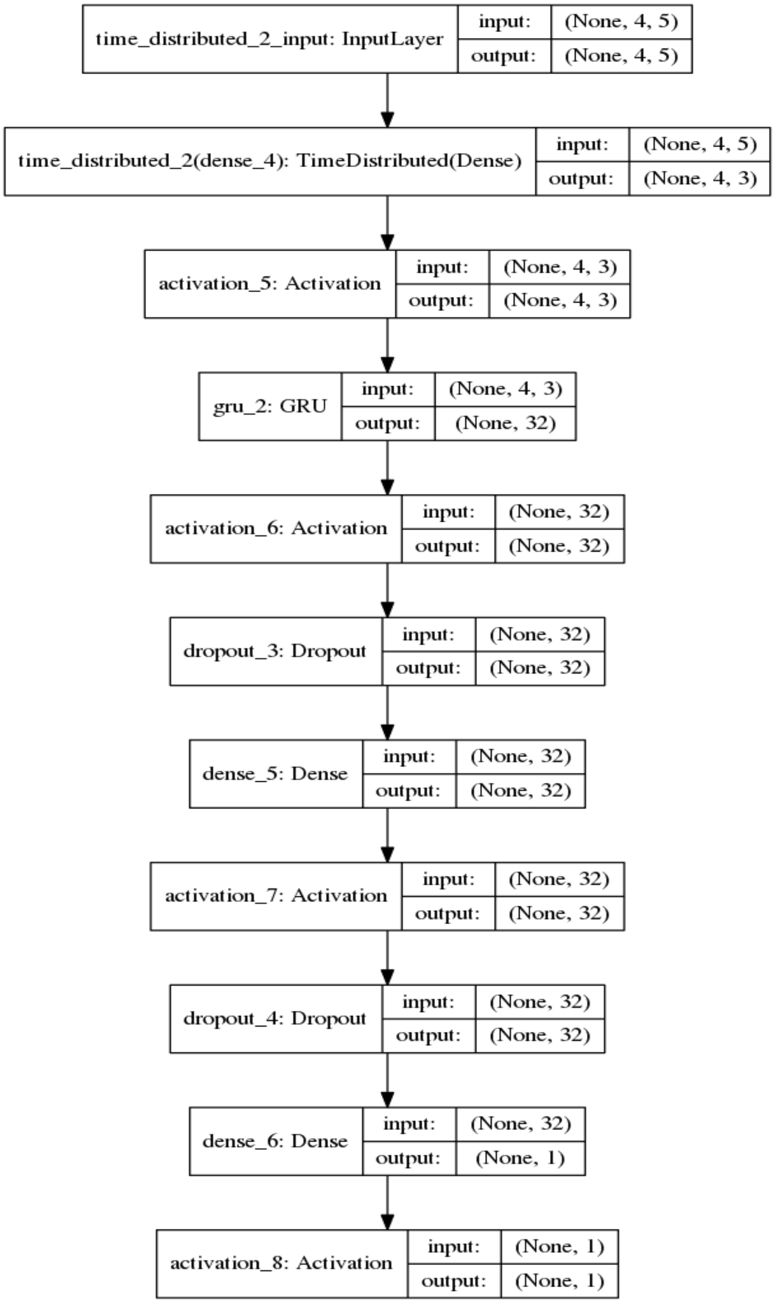
Supplemental Table 3.1: Contrastive Comparison of Machine Learning Methods

ML Method	Models Time	Interactions	Unequal Length Inputs	Can handle missing data	Can handle high dimensions	Number of training samples needed	Potential to overfit
Cox	Yes	No	No	No	No	Low	Low
Random Forest	No	Yes	No	Yes	No	Low	Low
LASSO	No	No	No	No	Yes	Medium	Medium
RNN	Yes	Yes	Yes	Yes	Yes	High	High

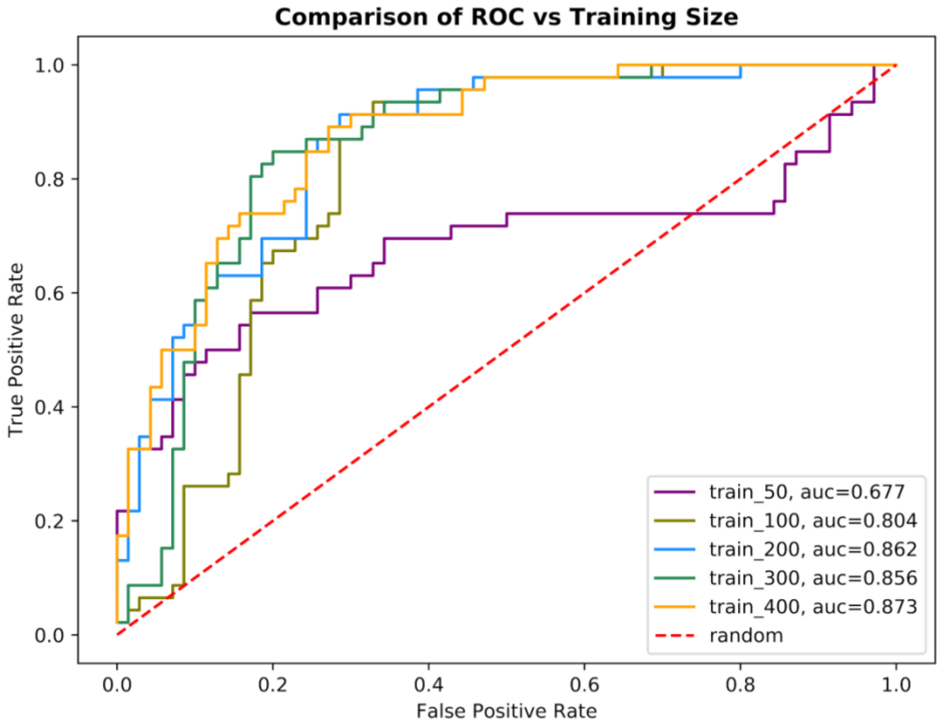
Supplemental Table 3.2: List of medications identified for RA treatment and considered DMARDs for the purposes of this work

Drug Name	Class
Methotrexate	Small Molecule
Sulfasalazine	Small Molecule
Hydroxychloroquine	Small Molecule
Leflunomide	Small Molecule
Azathioprine	Small Molecule
Auranofin	Small Molecule
Chloroquine	Small Molecule
Cyclophosphamide	Small Molecule
Cyclosporine	Small Molecule
Gold	Small Molecule
Minocycline	Small Molecule
Mycophenolate	Small Molecule
Penicillamine	Small Molecule
Myocrisin	Small Molecule
Abatacept	Biologic
Adalimumab	Biologic
Anakinra	Biologic
Certolizumab	Biologic
Etanercept	Biologic
Golimumab	Biologic
Infliximab	Biologic
Rituximab	Biologic
Tocilizumab	Biologic
Inflectra	Bio-Similar
Remsima	Bio-Similar
Benepali	Bio-Similar
Maball	Bio-Similar
Tofacitinib	JAKs Inhibitor

3.3.4. Supplementary Figures



Supplementary Figure 3.1: Architecture of Best Performing Longitudinal Deep Learning Model



Supplementary Figure 3.2: Sensitivity Analysis Comparing Forecasting Performance versus Training Size

Forecasting performance increases non-linearly with the number of samples available for training. There is a sharp increase in performance between 50 and 100 samples. The net size of performance gains becomes smaller as the sample size is increased. It is important to note that these experiments are conducted post-hyperparameter-optimization. Therefore, they reflect the numbers necessary to train the optimal model but do not reflect the numbers necessary to identify the optimal model.

References

1. Buttorff C, Ruder T, Bauman M. *Multiple chronic conditions in the United States.* RAND Santa Monica, CA; 2017.

2. Centers for Medicare and Medicaid Services (CMS). National health expenditures 2014 highlights. 2016.

3. Choi E, Schuetz A, Stewart WF, Sun J. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association.* 2016;24(2):361-370.

4. Kim Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:14085882.* 2014:1-6.

5. Gehrmann S, Dernoncourt F, Li Y, et al. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PLoS One.* 2018;13(2):e0192360.

6. Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent. 2010; Heidelberg.

7. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *J Mach Learn Res.* 2012;13:281-305.

8. Chollet F. Keras: Deep learning library for theano and tensorflow. *URL: https://keras io/k.* 2015;7(8):T1.

9. McKinney W. Data structures for statistical computing in python. Paper presented at: 9th Python in Science Conference 2010.

10. Hunter JD. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering.* 2007;9(3):90-95.

11. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine Learning in Python. *J Mach Learn Res.* 2011;12:2825-2830.

12. Oliphant TE. *Guide to NumPy, 2nd edn. CreateSpace Independent Publishing Platform.* Santa Monica; 2015.

Chapter 4

Predicting the Future

4.1. Permissions

Portions of this chapter was originally published in JAMA Network Open (JAMA Netw Open. 2019;2(3):e190606) and is available under a Creative Commons license.

4.2. Forecasting Individual Rheumatoid Arthritis Patient Outcomes Using Deep Learning on EHR Data

4.2.1. Abstract

Importance: Knowing the future state of a patient would enable a physician to customize current therapeutic options to head off disease worsening, but predicting that future state requires sophisticated modeling and information. If artificial intelligence models were capable of forecasting future patient outcomes they could be used to aid clinicians and patients to prognosticate outcomes or simulate potential outcomes under different treatment scenarios.

Objective: To engineer an artificial intelligence system to prognosticate the state of disease activity of patients with Rheumatoid Arthritis (RA) at their next clinical visit, and to quantify its ability.

Design: A retrospective multi-cohort observational study ranging from Januray 2012 to February 2018.

Setting: Rheumatology clinics at two distinct health systems with different EHR platforms; a university hospital (UH), and a public safety net hospital (SNH). The UH and SNH had significantly different patient populations and treatment patterns.

Participants: 578 patients at the university hospital and 242 patients at the safety net hospital met the inclusion criteria for RA, a complex systemic inflammatory disease with a variable course that may be difficult to predict.

Exposure: Structured data was extracted from the Electronic Health Record (EHR) including exposures (medications) along with patient demographics, labs, and prior measures of disease activity. We developed a longitudinal deep learning method to predict disease activity for RA patients at their next rheumatology clinic visit and evaluated inter-hospital generalization and model interoperability strategies.

Main Outcome(s) and Measure(s): Model performance was quantified using the Area Under the Receiver Operating Characteristic Curve (auROC). Disease activity in RA can be measured using a composite index score.

Results: 578 patients at the UH were included, with a mean age of 57, 83% were female. At the SNH, there were 242 patients included with a mean of 60, 81% were female. Patients at the UH were seen more frequently (median time between visits 100 days vs 180 days at the SNH) and more frequently prescribed higher-class medications (biologics) (63% vs 29%). At the UH, the model reached an auROC of 0.912 (95% CI: [0.862, 0.960]) on a test cohort of 116 patients. The UH-trained model had an auROC of 0.741 (95% CI: [0.649, 0.827]) in the SNH test cohort (n=117), despite marked differences in the patient populations. In both settings, a baseline prediction utilizing each patients' most recent disease activity score had statistically random performance.

Conclusions: Building accurate models to forecast patient outcomes using EHR data in a complex disease is possible. Our findings suggest that these models can be shared across hospitals with diverse patient populations.

4.2.2. Introduction

Rheumatoid arthritis (RA) is a complex systemic inflammatory disease characterized by joint pain and swelling that affects approximately one in one hundred people world-wide[1]. A chronic autoimmune disease, it is associated with significant morbidity and high costs of care. Disease progression varies greatly between people, and while numerous treatment options exist, individual responses to treatment vary widely[2]. While advances in therapeutics and clinical disease management have greatly reduced the proportion of treated patients living with uncontrolled disease activity, remission and durable response are less common. Data from the American College of Rheumatology's RISE registry indicates that 42% of patients nationwide had moderate or high disease activity at their most recent visit[3]. These data suggest that additional tools to facilitate and personalize disease management are needed.

Given the volume of data available in EHRs, the number of possible patient treatment and outcome trajectories resulting from heterogeneous patient comorbidities, medications and other factors far out-number what a human, even an experienced physician, can fully utilize. Many machine learning methods have been applied to clinical data such as Cox Regression[4], Random Forests[5], and LASSO[6]. However, these are often not well-suited to forecast the future from EHR data, given unequal numbers of data points between patients, large amounts of missing data, and high variable dimensions with time-dependent interactions (Supplementary Table S1). Deep Learning, a sub-discipline of Artificial Intelligence, has redefined computer vision[7] and demonstrated multiple successes in clinical applications[8] involving image data for melanoma[9], retinopathy[10], metastatic breast cancer[11], and other biomedical[12] and healthcare[13,14] domains. Deep Learning is being applied to a rapidly increasing number of EHR-related datasets (A survey of recent advances in deep learning techniques for electronic health record (EHR) Analysis.) and like the application of technology to any new field there are numerous opportunities and challenges (Opportunities and obstacles for deep learning in biology and medicine. Journal of The Royal Society Interface). A subfamily of deep learning called Recurrent Neural Networks (RNNs) have become state of the art in longitudinal predictions[15], solving complex problems in sequence modeling fields such as language translation[16] and self-driving cars[17]. Longitudinal deep learning models have previously been applied to EHR data[18] (Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review) classifying cardiovascular arrhythmias[19] as well as predicting inpatient mortality and emergency department re-admissions[20]. To our knowledge, there have not been attempts to forecast RA disease activity for future visits using any deep or machine learning approach.

Previous deep learning studies had tens of thousands of samples available for model training, far exceeding the number of samples that would be available for most outpatient conditions. To date, no study has investigated the generalizability of deep learning models using a smaller sample sizes typical of most complex chronic conditions cared for in the ambulatory setting.

In the current work, we aimed to utilize structured data from the EHR to build a model that would most accurately predict future RA disease activity. If successful, the ability to forecast disease activity could be clinically utilized to inform the aggressiveness of treatment on an individualized basis at each clinical visit. Models developed for predicting RA disease activity will be informative for other health conditions with quantifiable outcomes in the outpatient setting.

4.2.3. Methods

This study was approved by the UCSF Committee on Human Research (protocol number 15-18282).

4.2 3.1. Data Sources

Data for this study were extracted from the EHRs of two different hospitals; a university rheumatology clinic, the University of California San Francisco (UH) and a safety net rheumatology clinic, Zuckerberg San Francisco General Hospital (SNH). UH uses an EPIC EHR system that contains records on approximately one million total patients and dates back to January 2012. The UH data for this study were accessed on July 1, 2017. SNH uses separate EHR vendors for inpatients and outpatients; eClinicalWorks is used for outpatients and the EHR contains records on 65,000 unique individuals and dates back to January 2013. SNH data for the study were collected on February 27, 2018. A detailed description of the methods of EHR access can be found in the Supplementary Methods.

4.2.3.2. Definition of RA cohort

Patients had to have (Figure 4.1): two RA-related ICD-9 diagnostic codes (any of 714.0, 714.1, or 714.2) spaced a minimum of 30 days apart by a Rheumatologist and been prescribed at least one Disease Modifying Antirheumatic Drug (DMARD). These criteria have shown high specificity in a recent RA cohort study (https://www.ncbi.nlm.nih.gov/pubmed/22623324). To further increase specificity, we required that each patient have a minimum of two Clinical Disease Activity Index (CDAI) scores, which are only assigned by Rheumatologists for RA patients at both clinics in this study. Additionally, we required patients to have one RA diagnostic laboratory value (either C-reactive protein [CRP] or erythrocyte sedimentation rate [ESR]). Together, we believe this verifies that included patients were being treated for RA at the clinic for a minimum of four months. Final cohort sizes that met inclusion criteria were 578 patients at UH and 242 patients at SNH.

4.2.3.3. RA Disease Outcome Metric

The ACR endorses six different disease activity measures. The CDAI, a composite index of patient and physician assessments along with scoring of tender and swollen joints, is the most frequently used activity measure in the RISE registry and is the primary score used at both UH and SNH. CDAI is recorded as a raw score (0-72) but subsequently binned into four categories: Remission (< or =2.8), Low (2.9-10), Moderate (10-22), or High (>22) disease activity[21,22 ,23]. These four categories can then be further aggregated into a binary disease activity state, Controlled (Remission or Low activity, CDAI < 10) or Uncontrolled (Moderate or High activity, CDAI > 10).

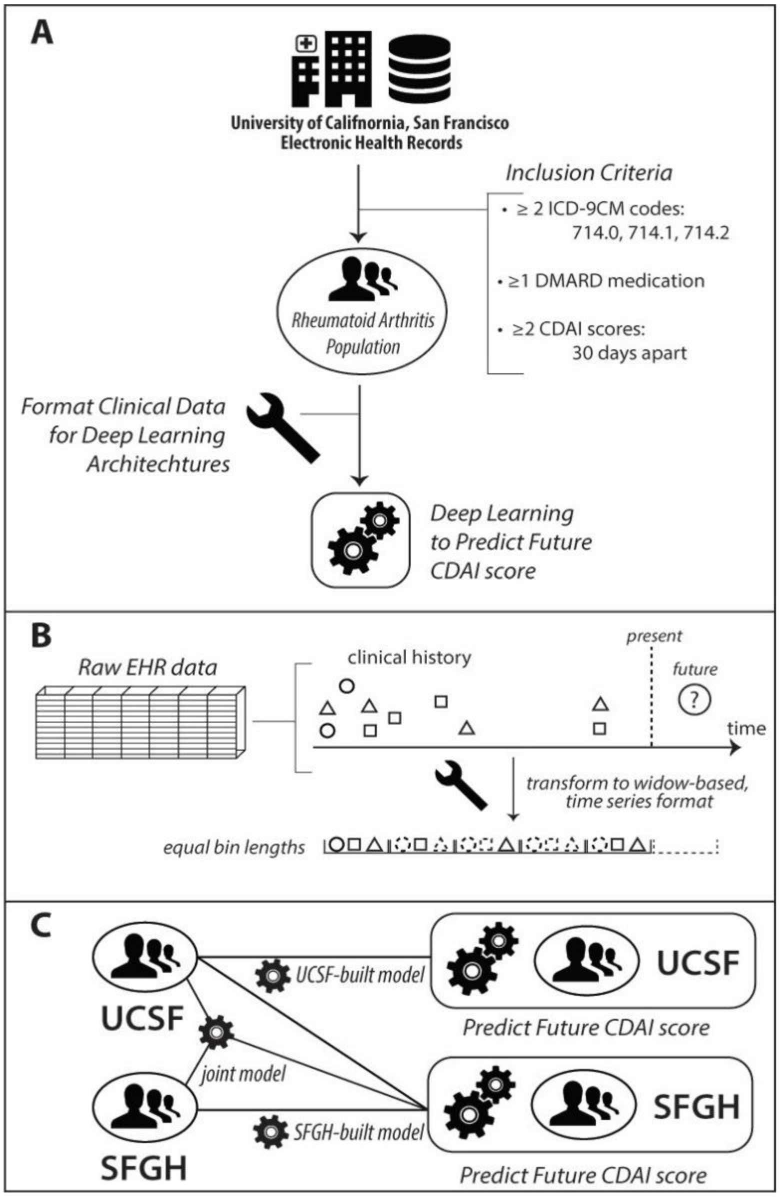

A
University of Califnornia, San Francisco
Electronic Health Records
Inclusion Criteria
• ≥ 2 ICD-9CM codes: 714.0, 714.1, 714.2
• ≥1 DMARD medication
• ≥2 CDAI scores: 30 days apart
Rheumatoid Arthritis Population
Format Clinical Data for Deep Learning Architechtures
Deep Learning to Predict Future CDAI score
B
Raw EHR data
clinical history
present
future
time
transform to widow-based, time series format
equal bin lengths
C
UCSF
SFGH
UCSF-built model
joint model
SFGH-built model
UCSF
Predict Future CDAI score
SFGH
Predict Future CDAI score

Figure 4.1: Inclusion criteria and study design for predicting RA clinical outcomes using deep learning methods. A) Workflow and design of the current study. B) Clinical data manipulation of relevant variables for deep learning. Shapes refer to clinical variables that are contained with bins of equal lengths (i.e., windows). Shapes with dashed lines represent missing data that are set to 0. C) Replication cohort experimental design

Figure 4.1: Inclusion Criteria and Study Design

4.2.3.4. Variables Utilized in Model

Given the relatively small number of patients available and the complexity of time series models, we chose to only include variables with known clinical significance in the models. These variables are known to be associated with disease activity, however no study that we are aware of has shown them to be predictive of future disease activity. We included the following: prior CDAI, ESR and CRP, DMARDs (Supplementary Table S2), oral and injected glucocorticoids, autoantibodies (presence of rheumatoid factor [RF] and/or anti-cyclic citrullinated peptides [anti-CCPs]), and demographics (age, sex, race/ethnicity). We chose to include only the first occurrence of each medication given the lack of reliable medication stop dates in the EHR. Considering each variable at each of four different time windows results in a reasonably large time-dependent total variable space of 165 total variables (29 possible DMARDs, eight possible Steroids, CDAI, ESR, and CRP at each time window in addition to the five static variables: demographics plus anti-CCPs and RF).

4.2.3.5. Modeling

Data were sorted chronologically by patient. The patient's demographics and history of clinical and laboratory variables were used to predict their most recent disease activity (Figure 4.1B). Extensive experimentation was performed to determine the optimal methods to format the chronological data as input and construct and train the most generalizable deep learning model for outpatient forecasting within this dataset. Complete information pertaining to model input, building, and selection are provided in the Supplemental Methods. A Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD) checklist is included in the Supplement. The code to build and train the model can be found on github (https://github.com/beaunorgeot/deep_clinical_forecasting).

4.2.3.6. Comparative Baselines

As a first baseline, we built a classifier that uses a Bayesian prior on the likelihood of each outcome category (Outcome Posterior Classifier). For example, if the ratio of Controlled to Uncontrolled outcomes is 60:40 respectively, as in the case of the UH cohort, the model would assign a forecasting prediction of Controlled, 60 percent of the time. As a second baseline (Change Posterior Classifier), we built a classifier with two elements of prior knowledge. First was each individual patient's previously recorded outcome. Second was the likelihood of changing outcome classes from one encounter to the next. We then built a model that considers each patient's previous outcome class as well as the likelihood within the cohort to switch classes in order to forecast the patient's future outcome class. For example, if the probability of switching outcome classes is 30 percent, the model would look at the previous outcome for each patient, and to forecast the patients future state, it would change the class of 30 percent of the patients while maintaining the class of the remaining 70 percent of the patients.

4.2.3.7. Evaluation Criteria

We chose the AUROC as our primary evaluation metric. In addition to AUROC, we performed sensitivity analyses in order to better evaluate the top model's potential clinical utility. We assessed how often the model was confident and wrong, assuming a threshold at probability of Uncontrolled is equal to 0.50, and compared model performance between groups of patients whose CDAI score at the predicted visit was either Remission, Low, Moderate, or High Disease

Activity. We explored how forecasting models may be shared across institutions that may not be able to directly share patient data. We evaluated the impact of the number of training samples on the models' performance to estimate the patient cohort size at which a hospital should decide to use a model trained at a different hospital, instead of building one of their own.

4.2.3.8. Model Explanation and Interpretability

We calculated Permutation Importance Score (PIS)[24] to measure the contribution of each independent variable, including time, to the overall model performance measured by AUROC. We generated a graphic, which we have called a Confusion Plot, by collecting the final dense representation learned by the model for each patient and plotting them using T-SNE, colored by outcome category, to assess the coherence of the representations learned by the model.

4.2.3.9. Statistical Analysis

AUROC Confidence Intervals: AUROC confidence intervals (CI) were generated on UH validation cohort for model selection and for UH and SNH test cohorts for final performance assessment using the Delong Method (CITE ME). Models with an auROC CI spanning 0.5 are not statistically different from random performance.

PIS Confidence Intervals: Variables at each time point whose PIS CI spanned the baseline AUROC were considered insignificant.

4.2.3.10. Performance in a Distinctly Different Cohort

We assessed three different methods of employing the model on patients from the second health system (SNH) (Figure 4.1C). First, we trained a model from scratch on the SNH training cohort using the top-performing architecture selected via Bayesian optimization at UH. Second, we tested the UH model directly on the SNH test cohort. Third, we utilized model transfer-learning and fine-tuning to update the fully trained UH model using the SNH training cohort. An explanation of theory of transfer-learning and fine-tuning as well as the methods that we applied can be found in the Supplementary Methods.

4.2.4. Results

4.2.4.1. Clinical Cohort Comparison

The UH and SNH clinics contained substantially different patient populations based on a number of factors (Table 4.1). Comparatively, the UH population was larger, predominantly White and non-Hispanic, and seen by rheumatologists with nearly double the frequency compared to the population at SNH. UH patients were more than twice as likely as SNH patients to be prescribed higher-class medications (biologics) and were also on a broader spectrum of treatments.

Table 4.1: Characteristics of Individuals with Rheumatoid Arthritis in the Two Health Systems Studied

Population Characteristic	**University Clinic N = 578**	**Safety Net Clinic N = 242**
Age in years, Mean ± SD	57 (15)	60 (15)
Female, n (%)	477 (83)	195 (81)
Race/Ethnicity, n (%)		
White	296 (51)	30 (12)
African American	33 (6)	19 (8)
Hispanic	97 (17)	89 (37)
Asian	101 (17)	70 (30)
Other	51 (9)	34 (13)
EHR System	Epic	eClinicalWorks
Median Number of CDAI Scores per Patient	6	4
Median Time Between CDAI	100 days	180 days
DMARD, n (%)		
Conventional Synthetic	534 (94)	191 (79)
Biologic	364 (63)	70 (29)
Tofacitinib	29 (5)	0 (0)

N: Number; SD: Standard deviation; EHR: Electronic Health Record, DMARD: disease modifying antirheumatic drug; CDAI: clinical disease activity index. DMARD numbers reflect patients prescribed a DMARD at the clinic prior to their index date. Supplementary Table S2 provides a breakdown of medications considered for each DMARD category.

4.2.4.2. Primary cohort (University Clinic) results

The best performing model was small, highly regularized, and consisted of a time-distributed layer, followed by recurrent GRU layers and a final dense layer (Supplementary Figure S1). Fixed time intervals of 120 days, random sampling during training, equal penalization of errors for both classes, use of a combination of clinical, medication, and laboratory variables, and one year of each patient’s history prior to their index date provided the best results. The best deep learning model (Figure 4.2) demonstrated excellent forecasting performance (AUROC= 0.912, 95% CI [0.862, 0.960]) on the University Clinic test cohort (n=116). Both baselines demonstrated near random performance: Outcome Posterior Classifier: AUROC = 0.535, 95% CI [0.442, 0.630]; Change Posterior Classifier: AUROC = 0.554, 95% CI [0.460, 0.636]).

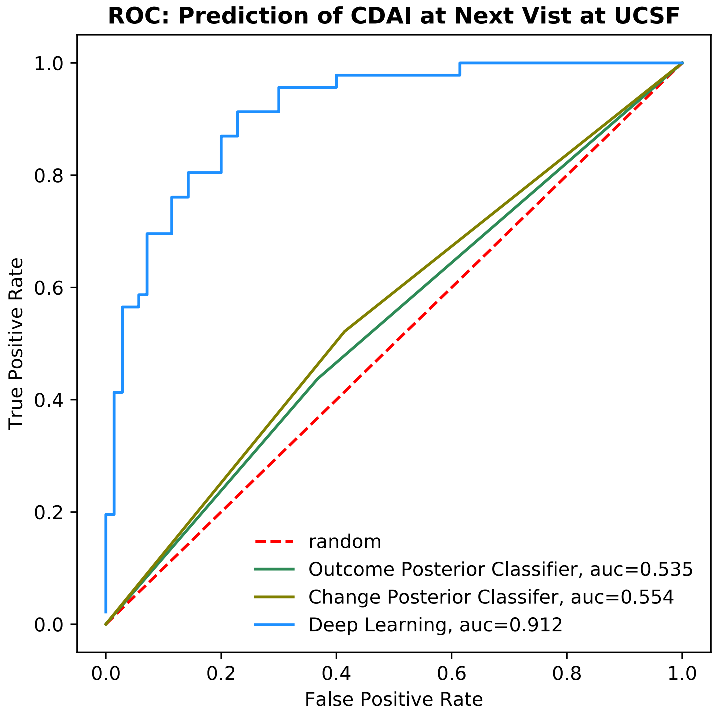


Figure 4.2: Forecasting Performance on Test Cohort at UH. The distribution of outcomes from the training cohort at UH was 60 percent ‘Controlled’ and 40 percent ‘Uncontrolled’ according to the Clinical Disease Activity Index (CDAI). This prior was used to train the Outcome Posterior Classifier at UH (Green Line, AUROC=0.535). The likelihood of switching outcomes between visits within the training cohort was 25 percent. This prior was used to train the Change Posterior Classifier at UH (Olive Line, AUROC=0.554). Deep Learning produced the best results (Blue Line, AUROC=0.912).

Figure 4.2: Forecasting Performance on Test Cohort at UH

4.2.4.3. Sensitivity and Model Explanation

A sensitivity analysis comparing forecasting performance to the number of samples available for training revealed a non-linear increase in performance with linear increases in sample size (Supplementary Figure S2). CDAI was important for forecasting performance in each time window (Combined PIS=40) followed by Time itself (PIS=11). ESR and CRP variables contributed small but significant predictive power to the two most recent time windows (Combined PIS = 2,3 respectively). Steroids, as a class, at the current time window had a PIS of 4, with Prednisone alone having a PIS of 2, but were not significant in other windows. Multiple

DMARDs were significant but with PIS less than two. The model was confident (probability greater than 0.8 or less than 0.2) and incorrect only two times out of the 116 test samples, or 1.5% of the time. These errors occurred for patients whose future visit CDAI score fell on the threshold between the outcome classes (CDAI =10) +/- 2. Performance was equal for patients whose future CDAI was clinically determined as either Remission, Low Activity, or High Activity. Predictive performance was lowest for patients whose future disease activity was Moderate; most of the incorrectly classified patients in this group had CDAI scores near the classification threshold (CDAI in the range of 10-14). The Confusion Plot (Figure 4.3) appears as a nearly one-dimensional manifold (a curve). Instead of dichotomous clusters for each outcome category, the model learned a continuous representation of the patients. Distinct decision boundaries based on the Confusion Plot can be seen.

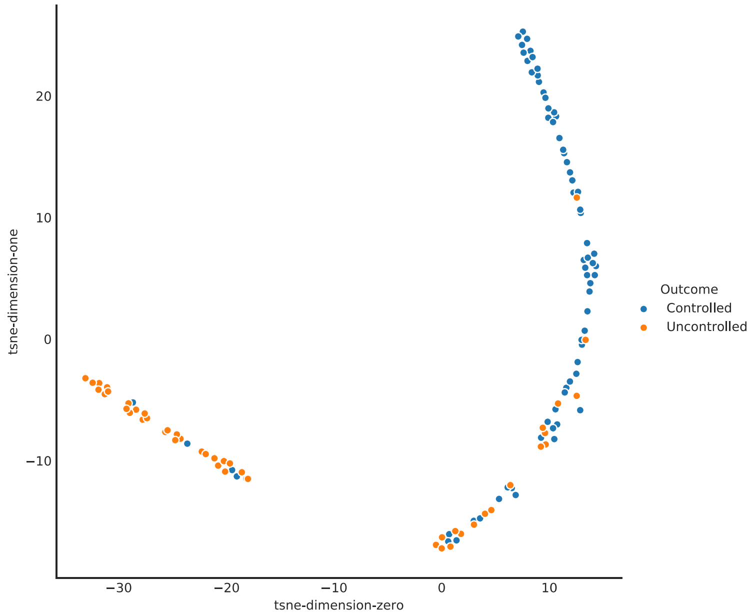


Figure 4.3: Confusion Plot consisting of the final embedding of the model, the Learned Patient Trajectory Vectors, visualized using t-SNE, colored by the ground truth of the patients outcome at their next visit. Yellow for uncontrolled . The model places observations onto a one-dimension manifold with Controlled and Uncontrolled outcomes clustering along different ends of the manifold.

Figure 4.3: Confusion Plot, Learned Patient Trajectory Vectors

4.2.4.4. Safety Net Cohort Results

When the top-performing model architecture was trained from scratch on the Safety Net Clinic training cohort (Figure 4.4; n=125) it produced reasonable results (AUROC=0.623, 95% CI [0.522, 0.724]) on the Safety Net Clinic test cohort (n=117). Employing a model that was trained on all the UH patients (n=578) directly on the Safety Net Clinic test cohort dramatically increased forecasting performance (AUROC=0.741, 95% CI [0.649, 0.827]). Utilizing transfer-learning and fine-tuning to update the UH-trained model using the Safety Net Clinic training cohort did not provide any additional improvements in performance (AUROC = 0.739). Both baselines demonstrated random performance: Outcome Posterior: AUROC = 0.507, 95% CI [0.391 - 0.615]; Change Posterior Classifier: AUROC = 0.544, 95% CI [0.460 - 0.622]).

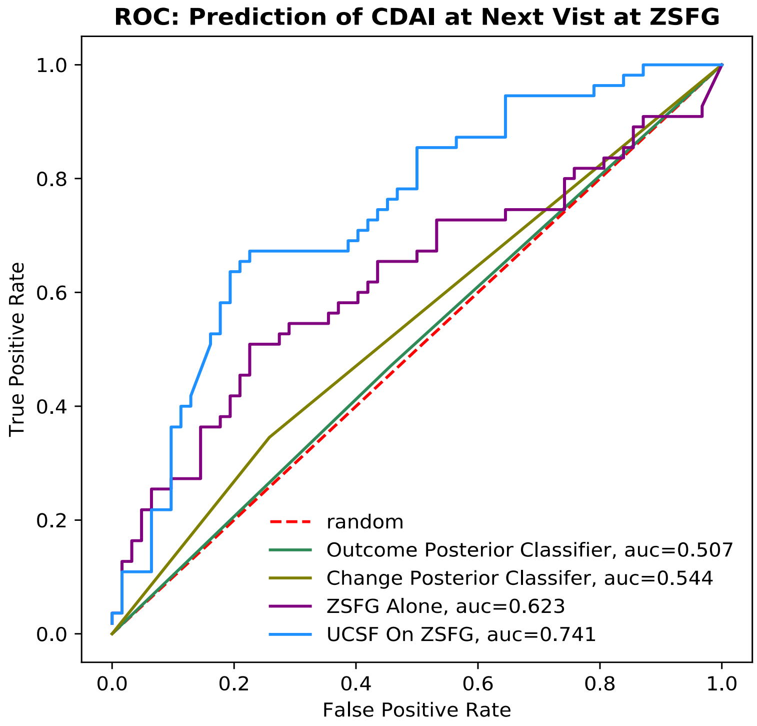


Figure 4.4: Safety Net Cohort Results. The distribution of outcomes from the training cohort at ZSFG was 50 percent 'Controlled' and 50 percent 'Uncontrolled'. This prior was used to train the Outcome Posterior Classifier at ZSFG (Green Line, AUROC=0.507). The likelihood of switching outcomes between visits within the training cohort was 25 percent. This prior was used to train the Change Posterior Classifier at ZSFG (Olive Line, AUROC=0.544). Training the deep learning model exclusively on the ZSFG train cohort produced an AUROC that was substantially better than random (Purple Line, AUROC = 0.623). Training the deep learning model on the larger UH patient cohort produced the best overall results test cohort (Blue Line, AUROC =0.741)

Figure 4.4: Forecasting Performance on Test Cohort at SNH

4.2.5. Discussion

4.2.5.1. Main Findings

In this study, we used deep learning to forecast future RA disease activity scores across two health systems and compared those results to prediction models that only used a patient's most recent CDAI. Contrary to our expectations, a patient's most recent CDAI alone was actually a very poor predictor of their index CDAI, as evidenced by the statistically random results of both Baselines. The history of disease activity, lab values, and medications all together were required to create the strongest predictor of the disease activity at the next visit. Just over 20 variables were found to be significantly important for predictive accuracy, a relatively small number, however these variables have time-dependent interactions which adds considerable complexity. For example, the best deep learning model substantially outperformed the Multilayer Perceptron (which acted as a surrogate for Logistic Regression) (Supplemental Results Table 1) demonstrating the utility of more complex DL models for this task.

Our results show that deep learning models can be trained on cohorts of only a few hundred patients to accurately forecast RA patient outcomes using EHR data. We also found that our model performed well when applied to a second health system with a distinct sociodemographic population and separate EHR system. Given the many differences in the demographics and social determinants between the patients in these centers, the ability of the model to function significantly above random is highly promising. By considering no more than the most recent year of each patient's history but allowing patients to have as little as four months of history, the model could have utility for patients at all stages of their care. While the amount of data that a rheumatologist must synthesize in one visit in order to make decisions is large and growing, the results presented here indicate that use of artificial intelligence models to assist with this in the near future is promising.

4.2.5.2. Prior Work

Early successes in the application of deep learning to clinical forecasting[19,20 ,25] demonstrated that longitudinal deep learning models outperformed traditional machine learning approaches and that reasonable predictive performance was possible. However, these studies were limited in their clinical utility by the inclusion of patients without clinical risk indicators for the outcomes being predicted, the sheer numbers of patients used for training, and a lack of evaluation of model performance across hospitals with diverse patient populations. This work addresses these open questions by focusing on a clinically relevant patient population and outcome at both a large university hospital and an associated safety-net clinic. The model trained on the larger UH population produced the best results on the SNH population, demonstrating the power of larger training sizes and the interoperability of models between hospitals with diverse patient populations. While it is perhaps discouraging that utilization of transfer-learning and fine-tuning methods to update the fully trained UH model using the SNH training cohort did not provide any additional improvements in performance, we suspect that this is due to the fact that the SNH training cohort was probably too small and thus suffered from over-fitting.

4.2.5.3. Limitations

With data from two distinct hospital systems and just over 800 total patients, inferences about large scale generalizations cannot be made. Accordingly, this work is limited to being a promising proof-of-concept. There are numerous inherent biases in medicine, perhaps most notably are those relating to sicker patients generally having a great number of data points. We sought to address this bias at multiple levels. Most notably, by giving all patients the same number of time windows and setting that to be equal to the smallest median number of visits in either cohort (Table 4.1) and then by dropping all but the most recent values within each window for a patient. Additionally, including fewer variables in the model lessens the risk of spurious associations between variables. However, the potential for theses biases runs deep. For example, physicians may choose to order or not order labs for a given patient at a given time point based on factors that are not modeled here, including physician preference. Similarly, some physician-patient combinations may be more or less likely to switch a patient's treatment strategy. Intimately tied to these challenges is the decision about what to do when the value of a patient's variable is missing from a time window. Our choice to replace missing continuous variables with a value never observed in our data set (zero) is not perfect, as the replaced value more closely resembles healthy patients than sick ones, but it does seem to be the replacement option that is least likely to reinforce this bias. Statistical imputation or forward filling are likely to introduce or reinforce bias for a health condition that varies so much between individuals and within a person over time and where most patients' disease activity is uncontrolled. While we strove to reduce biases in our data and modeling, we cannot fully eliminate them. Using the treating physician as variable to model, while not possible with this current study, could potentially reduce bias further in the future. To the best of our knowledge, there are no clinical methods for explicitly forecasting individual patient disease activity states at future visits nor has any methodology for this ever been employed in a clinic. While this underscores the need for the work introduced in this study, it leaves us without any clinical baseline to compare machine learning results to.

Finally, the performance of the UH-trained model on the SNH test cohort (AUROC=0.741) is too low to be of immediate clinical utility, however the performance is evidence that the model learned something robust and transferable. Given the notable differences in the two clinical populations, a method of exploring whether the differences in model performance between the populations was due exclusively to the differences in the treatment populations would have added clarity to the study. However, while the patient populations are different in many ways that we can measure (Table 4.1), we know that they are also different in many ways that we cannot reliably measure (other socioeconomic factors, environment, social structure, insurance coverage, and more) making an actual patient matching algorithm impossible given the relatively small sizes of both populations. For complex chronic diseases like RA, patient populations that number in the hundreds are unlikely to capture enough clinical or social variation to adequately represent the complete disease spectrum. Our sensitivity analysis revealed that increasing the training set size lead to non-linear increases model performance. Thus, adding EHR data from other institutions seems likely to result in additional gains in predictive power as well as insights into the subtler factors responsible for the model's performance.

4.2.5.4. Summary and Implications

The future decision support that we envision will involve aggregating data from multiple institutions, training the model on all of that data, and then deploying the model in small clinics as well as large hospital systems giving everyone access to the most robust models trained on largest and most diverse patient populations possible. Using such a forecasting model will help clinicians and patients understand predicted disease trajectories. This in turn, will help inform the aggressiveness of treatment. We find that there are many clinical situations where there is equipoise about whether and how to augment therapy for RA. Patients may have been stable for some time, but come to the current visit with a CDAI score just over the threshold of moderate disease activity. Alternately, they may have been in moderate disease activity over several visits and have been experiencing adverse effects related to their current DMARD regimen. In situations like these, where waiting until the next visit to consider any medication changes seems like a reasonable option, having a prediction from the algorithm that indicates that the CDAI score at the next visit will likely be worse may push a provider and patient to action. These situations "at the margin" are the ones that are most likely to benefit from the algorithm. Given the algorithm’s already strong performance at identifying patients that will have controlled disease activity at their next visit, probability thresholds could be analyzed to specifically improve outcomes for these patients “at the margin”. As a patient’s health status and other variables change, the model will adapt its predictions, allowing both patients and clinicians to use this information to inform treatment changes dynamically. As we move toward personalized medicine, such models can be used to simulate trajectories given different treatment scenarios. The addition of molecular, genomic, and other types of data to EHR data to generate treatment response trajectories would allow a more personalized medicine approach to RA care.

With large national registries, such as the American College of Rheumatology’s RISE registry, now available for rheumatic and other diseases, we see a rich future in the application of deep learning to longitudinal patient care. Model performance is nearing the point where they are good enough to warrant launching a prospective clinical trial to evaluate their usefulness in aiding clinicians and patients to prognosticate RA outcomes or simulate outcome trajectories under different treatment scenarios.

4.2.6. Conclusion

In the current study, we built an accurate longitudinal deep learning model to forecast patient outcomes in two distinctly different rheumatoid arthritis populations that numbered in the hundreds, much smaller than what once believed to be necessary for DL. These models can be shared across hospitals with different EHR systems and diverse patient populations. In the future, models built from large pooled patient populations are likely to be the most accurate, giving everyone access to the most robust models trained on largest and most diverse patient populations possible. The methods used to develop models for predicting RA disease activity will be informative for other health conditions with quantifiable outcomes.

4.2.7. Additional Support

Use of UCSF de-identified data was made possible by Dana Ludwig MD and the UCSF Information Technology Services Academic Research Systems team and the Clinical Data Research Consultations service. We thank Nadav Rapport PhD for providing code related to the R library MetaMap to map drug names to more standardized names. We thank Thomas Peterson PhD for providing suggestions for the Discussion. Research reported in this publication was supported by funding from the UCSF Bakar Computational Health Sciences Institute and the National Center for Advancing Translational Sciences of the National Institutes of Health under award number UL1 TR001872. Additional support was provided by the Amazon Cloud Research Fellowship (BN), The Silicon Valley Community Foundation (BN), the ARCs Fellowship Foundation (BN), the Russell/Engleman Medical Research Center for Arthritis (JY, GS), AHRQ R01 HS024412 (JY,GS), and the NIH grants P30AR070155 (MS, JY), NLM K01LM012381 (MS), K23AR063770 (GS), U24CA195858 (TG), and R01GM079719 (AB). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. Role of the Funder/Sponsor: The funding institutions played no part in the design or conduct of the study; collection, management, analysis, or interpretation of the data; preparation, review, or approval of the manuscript; or decision to submit the manuscript for publication.

References

1. Spector TD. Rheumatoid arthritis. *Rheumatic diseases clinics of North America.* 1990;16(3):513-537.
2. Singh JA, Saag KG, Bridges SL, Jr., et al. 2015 American College of Rheumatology Guideline for the Treatment of Rheumatoid Arthritis. *Arthritis Rheumatol.* 2016;68(1):1-26.
3. Yazdany J, Bansback N, Clowse M, et al. Rheumatology Informatics System for Effectiveness: A National Informatics-Enabled Registry for Quality Improvement. *Arthritis Care Res (Hoboken).* 2016;68(12):1866-1873.
4. Harrell FE, Jr., Lee KL, Pollock BG. Regression models in clinical studies: determining relationships between predictors and response. *J Natl Cancer Inst.* 1988;80(15):1198-1202.
5. Strobl C, Boulesteix AL, Zeileis A, Hothorn T. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics.* 2007;8:25.
6. Odgers DJ, Tellis N, Hall H, Dumontier M. Using LASSO Regression to Predict Rheumatoid Arthritis Treatment Efficacy. *AMIA Joint Summits on Translational Science proceedings AMIA Joint Summits on Translational Science.* 2016;2016:176-183.
7. Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision.* 2015;115(3):211-252.
8. Greenspan H, Ginneken Bv, Summers RM. Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique. *IEEE Transactions on Medical Imaging.* 2016;35(5):1153-1159.

9. Esteva A, Kuprel B, Novoa RA, et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature.* 2017;542(7639):115-118.

10. Gulshan V, Peng L, Coram M, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus PhotographsAccuracy of a Deep Learning Algorithm for Detection of Diabetic RetinopathyAccuracy of a Deep Learning Algorithm for Detection of Diabetic Retinopathy. *JAMA.* 2016;316(22):2402-2410.

11. Wang D, Khosla A, Gargeya R, Irshad H, Beck A. Deep Learning for Identifying Metastatic Breast Cancer. *ArXiv e-prints.* 2016.

12. Ching T, Himmelstein DS, Beaulieu-Jones BK, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of the Royal Society, Interface.* 2018;15(141).

13. Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics.* 2018;19(6):1236-1246.

14. Miotto R, Li L, Kidd BA, Dudley JT. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific Reports.* 2016;6:26094.

15. Karpathy A, Johnson J, Li FF. *Visualizing and Understanding Recurrent Networks.* 2015.

16. Hassan H, Aue A, Chen C, et al. *Achieving Human Parity on Automatic Chinese to English News Translation.* 2018.

17. Huval B, Wang T, Tandon S, et al. *An Empirical Evaluation of Deep Learning on Highway Driving.* 2015.

18. Che Z, Purushotham S, Cho K, Sontag D, Liu Y. *Recurrent Neural Networks for Multivariate Time Series with Missing Values.* Vol 82016.

19. Schwab P, Scebba G, Zhang J, Delai M, Karlen W. *Beat by Beat: Classifying Cardiac Arrhythmias with Recurrent Neural Networks.* 2017.

20. Rajkomar A, Oren E, Chen K, et al. Scalable and accurate deep learning for electronic health records. *NPJ Digital Medicine.* 2018;1(1):18.

21. Ritchie MD, Denny JC, Crawford DC, et al. Robust replication of genotype-phenotype associations across multiple diseases in an electronic medical record. *American journal of human genetics.* 2010;86(4):560-572.

22. Smolen JS, Landewe R, Breedveld FC, et al. EULAR recommendations for the management of rheumatoid arthritis with synthetic and biological disease-modifying antirheumatic drugs: 2013 update. *Annals of the rheumatic diseases.* 2014;73(3):492-509.

23. Smolen JS, Breedveld FC, Burmester GR, et al. Treating rheumatoid arthritis to target: 2014 update of the recommendations of an international task force. *Annals of the rheumatic diseases.* 2016;75(1):3.

24. Breiman L. Random Forests. *Machine Learning.* 2001;45(1):5-32.

25. Choi E, Schuetz A, Stewart WF, Sun J. Using recurrent neural network models for early detection of heart failure onset. *J Am Med Inform Assoc.* 2017;24(2):361-370.

Chapter 5

MI_CLAIM

5.1. Minimum Information about Clinical Artificial Intelligence Modeling

5.1.1. Abstract

Artificial Intelligence (AI) models have become widely used tools for the generation of clinical insights, prognostics, diagnostics, and classification. Although many significant results have been derived from clinical AI studies, one limitation has been the lack of standards for presenting and exchanging the results from such models. Here we present a proposal, the Minimum information for Clinical Artificial Intelligence (MI_CLAIM), that describes the minimum information required to ensure that the performance of clinical AI models can be easily interpreted and that results derived from model analysis can be independently verified in similar settings. The ultimate goal of this work is to establish a standard for designing, recording and reporting AI-based clinical informatics studies, which will in turn facilitate transparency and the establishment of trust and ultimately enable the utilization of such models in the clinical setting.

With respect to MI_CLAIM, we concentrate on defining the content and structure of the necessary information rather than the technical format for capturing it.

5.1.2. General Principles of the Claim Design

As a starting point, we propose that for the results from clinical AI models to have the most value, they should satisfy the following requirements: (i) the recorded information about each study should be sufficient to interpret the clinical utility of the results and should be detailed enough to enable comparisons to similar studies and permit replication in similar settings and (ii) the code to build, train, and evaluate an identical model (including examples of expected input data formats) should be openly provided to enable external validation and utilization of successful models (See Table 5.1). The Figure 5.1 further shows the interconnection of these parts/components of clinical AI study.

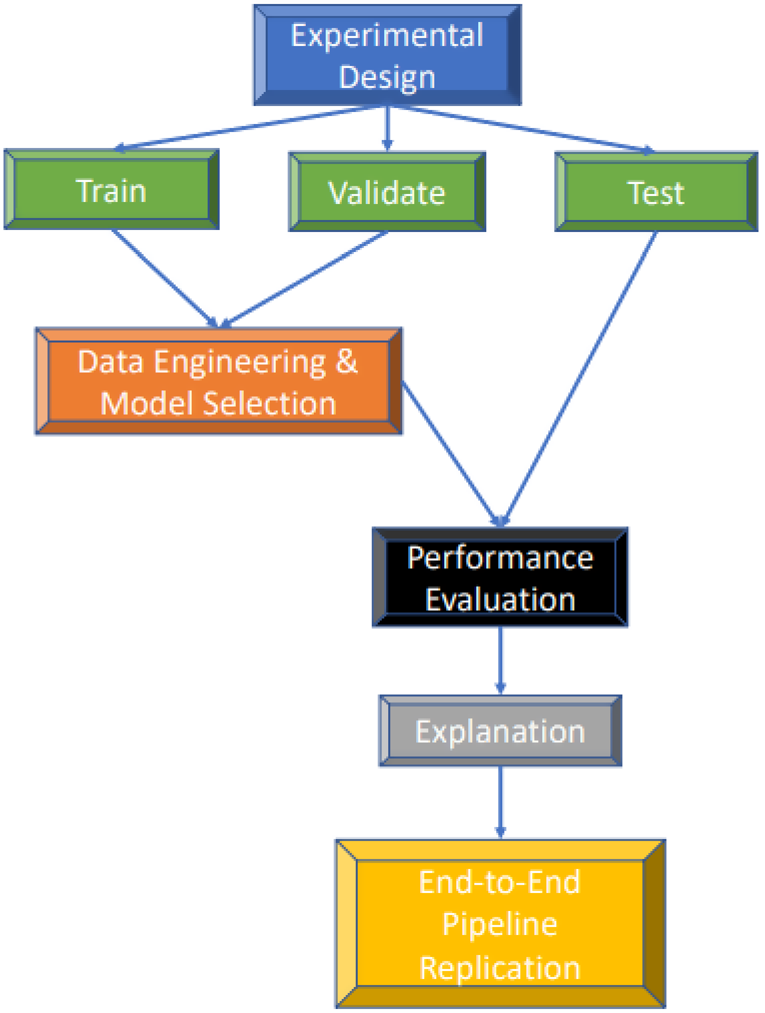


Figure 5.1: The (Six) Parts of MI_CLAIM - A Schematic Representation of the 6 Components of a Clinical AI Study

The first requirement recognizes that overfitting is a primary concern when dealing with flexible models and clinical data, while transparency about the distribution of clinical variables and demographics present in the model building and testing cohorts is essential to properly assess fairness/bias for all groups and the overall clinical utility.

The second requirement addresses the truth that sharing clinical data often is neither possible due to institutional patient privacy policies, nor would it be advisable to share even at institutions without such safeguards in place. Clinical data is much more sensitive than other data, such as microarray data, and should only be handled by people with proper training (which is currently decided at the institution level). In any case, the validation of the exact results is generally of less interest than whether or not the results validate in a new cohort of patients, which researchers at different institutions can do with their own data using security features determined by their institution. Therefore in the space of clinical AI models, the limiting factor for validation is not the raw data itself but replication of the exact model building pipeline (including any feature engineering or transformations). The code for complete pipelines, beginning with a few examples of the raw input data (in it's proper format, but populated with random numbers) and ending w/ performance evaluation should be provided as well-documented scripts or notebooks, including exact environment requirements, such that a new researcher can run the pipeline end to end, without any modifications necessary to the code. This provides the new researcher with everything necessary to rapidly validate the results in their own cohorts and will additionally facilitate the transfer of pipelines across clinical use-cases.

Table 5.1: The (Six) Parts of MI_CLAIM - A Schematic Representation of the 6 Components of a Clinical AI Study

Part/Component of Clinical AI Study	Description
Part 1: Experimental Design	This section describes the study as a whole. Normally, this can be broken down into four subsections: The clinical setting, performance measures, population composition, and baselines. (a) The clinical problem and the workflow in which a successful model would be employed, and in what ways, if any, the experimental design in which the model will be trained and tested differ from that (including the acquisition of data). (b) The Performance Measurements that will be used to evaluate the results and how do those measurements would translate to successes and failures in the clinical setting. (c) The composition or makeup of the population available for training and testing and how representative that sample is of a real-world populations for the clinical question at hand. Additionally, whether performance among certain sub-groups of the population is important or if aggregate performance statistics across the population is all that matters. (d) Are there any current solutions employed in the clinic that can act as baselines?
Part 2: Establishment of Out of sample validation	This section is tightly coupled to Section 3 and builds upon the Experimental Design by detailing the steps that will be taken to prevent information leakage, overfitting, generalization out of sample, and therefore a meaningful interpretation of the study results. Of paramount importance is the splitting of all available samples into two groups at the very beginning of the study, one for development and an independent test cohort. Truly appreciating the importance of this simple task requires an intuitive understanding of three fundamental statistical concepts: populations, samples, and parameters (aka: measurements, features, variables). To briefly discuss these concepts, let's imagine that we're talking about type 2 diabetes. In truth the Population is the total of all patients anywhere w/T2D. This is the population that we'd really like to learn about. But since we don't have access to all of the them, we have to use all the people/patients that we do have access to as a surrogate for the true population. But it is crucial that we always keep in mind that our 'population' is not the true Population. All of our best practices are set up to maximize the inferences that we can draw about the true population using our surrogate population. Samples are the groups of individual people that we draw from our surrogate population. The most common sample types that we talk about in machine learning are training, validation, and testing cohorts. Cross-validated splits are also examples of samples drawn from our (surrogate) population. In any given sample there is an exact correlation between independent variables and the dependent variable(s). But that correlation tells you absolutely nothing about the TRUE-ASSOCIATION between variables and outcomes in the TRUE POPULATION. Parameters (also known as Features or variables) are things that you have measured about each of the people in your sample or population. For example: height, medications prescribed, BMI, or eye color. If you build models or conduct statistical tests on the entirety of your surrogate population (all the data you have), you will have turned your 'population' into a sample and YOU WILL LOSE ANY AND ALL ABILITY TO MAKE ANY CLAIM AT ALL about how your model performs in the population that you care about or the predictive importance or association of your variables and the outcome. You instantly move from predictive to descriptive statistics. Members of the test cohort should reflect the population and distribution of the clinical outcomes of interest. We recommend stratified sampling where possible, and reporting a comparison of statistics describing the distribution of variables and outcomes within training and testing populations otherwise. The development cohort may be used in any manner that facilitates data engineering and model selection. The two most common

Part/Component of Clinical AI Study	Description
	approaches are either cross-validation, which is typically used for smaller datasets, or sub-dividing the development cohort into training and internal validation set for large sample sizes. Under no circumstances can cross-validation be used as a replacement for an independent test cohort. Validation at a separate clinic or hospital system is necessary to make any claims about generalization.
Part 3: Data Engineering and Model Selection	With an independent test set established, the development cohort can now safely be used to estimate the best (a) format of data and (b) type of model to solve the clinical problem. For all studies this section should begin with data providence, clearly specifying where the data in its most raw form came from and how it was formatted. It should then (a) describe any transformations that were done to the data prior it being fed into the model as input. For traditional machine learning studies, these transformations will typically be feature engineering, for deep learning models this frequently involves the normalization of continuous variables and one-hot, or integer, encoding of categorical variables. Next, the section should describe the type of models that were evaluated and how a top performing combination of model and data formatting will be selected. Typically these two elements of optimal data format and model are highly interdependent and therefore the process to arrive at the best combination is often iterative. An example statement might look like: “5-fold cross-validation on the development cohort was used to evaluate the results of a grid search comparing number of input features, number of variables to consider at each split, number of splits, and number of trees for random forest models. No other model types were considered. The top performing approach was selected on the basis of median AUC.” This section should also describe the process preparing the baseline methods for use, if any were available.
Part 4: Out of Sample Evaluation	With the optimal model selected and comparative baselines tuned from Section 3 it is time to evaluate them once, and only once, against the test cohort. This section will include a typical results table with the performance of the baselines and models tested along with appropriate statistics for significance. If any important sub-groups of patients were identified in Section 1, performance of the baseline and model in each of those subgroups should also be provided in an identically formatted table.
Part 5: Model Explanation/Interpretation	Having some intuition of how complex models are behaving is relevant to most clinical problems and typically serves one, or more of three purposes. First, it may provide a sanity check that the model reached its accuracy by focus on relevant inputs and not unanticipated artifacts of the data. Second, it can uncover bias which model users should be aware of. This bias could relate to fairness or anticipated points of failure. Third, there are many potential tasks that clinical AI models might be applied to that no human is definitively capable of performing well. In these cases, it may be useful to harness what the model has learned to generate testable hypotheses to move those fields forward. We agree with Spiegelhalter that useful explanations should contain at least two elements; Global explanations consisting of what the model learned overall about the relationship between the independent and dependent variables, and local explanations, consisting of why the predictions for specific cases were arrived at. SHAPs and MAgEC provide two alternative approaches for both the global and local examination and interpretation of most models.

Part/Component of Clinical AI Study	Description
	If no explanation methods are available important insights into model behavior can still be gleaned by a more detailed examination of its performance. For classification models a description of the top-5 cases where the model was most confident and correct, most confident and incorrect, and least confident is a good starting point. The same philosophy can be applied to regression models by examining the cases where the model had the largest error above the true answer, the largest error below the true answer, and the number of times it predicted the median value.
Part 6: Reproducible Pipeline	The code for complete pipelines, beginning with a few examples of the raw input data (in its proper format, but populated with random numbers) and ending w/ performance evaluation should be provided as well-documented scripts or notebooks, including exact environment requirements, such that a new researcher can run the pipeline end to end, without any modifications necessary to the code. This provides the new researcher with everything necessary to rapidly validate the results in their own cohorts and will additionally facilitate the transfer of pipelines across clinical use-cases. The goal here is not for a new researcher to replicate the results but the exact process by which the results were generated. This enables the new researcher to determine whether the results validate in their own clinical settings and facilitates the transfer of pipelines from one clinical task to another rapidly speeding up prototyping and helping the entire field to develop best practices.

5.1.3. Discussion

My goal is to develop a standard that can serve both clinical scientists and data scientists. To that end, I hope that this description will stimulate discussion of the proposed MI_CLIAM standards and I encourage the clinical community, as well as the AI community, to provide me with their views on how this standard can be improved. For this purpose an e-mail discussion group has been set up with the release of this manuscript as an independent document outside of my dissertation.

Chapter 6

MAgEC

6.1. Understanding and Leveraging Complex Models: Framework for understanding, explaining, and comparing clinical AI models

6.1.1. Introduction

In medicine, and indeed in many other fields, the information needed to understand and intelligently act within the overall landscape is comprised of 3 overlapping hierarchies of understanding: Cases, or what variables/factors about a specific individual contribute to or drive their outcome; Features, or the average effect of each variable across many cases; and Populations, patterns in cases that allows them to be logically grouped together. Therefore, any attempt to examine and understand a model in these types of settings must address what the model has learned in regards to each of these hierarchical elements of information. However, historically methods for examining models have generally addressed only one or at most two elements in the hierarchy which has lead to an incomplete understanding of the models which has lead to an inability to adequately assess true model performance including examining it for bias and anticipating it's performance in future uses; as well as an ability to completely leverage what models have to learned to generate new knowledge in imperfectly understood fields such as medicine.

Further, many methods for understanding models has been model-specific which has lead to two deep challenges;:first it has made model builders make decisions for trading off the model types that they would like to employ to best solve the current problem with the types of methods that they would like to use examine what the model has learned and potentially explain it to other stakeholders; second, it has made it difficult/impossible to objectively compare what multiple different types of models trained to do the same task have learned. Since there has not been a direct way of comparing what different types of models have learned about the same data and task, this has lead to an often unaddressed and insidious oversight in the field which is confusing what a specific model has learned about the association of input and response variables with some underlying truth about the nature of the actual association between the inputs and the responses.

There are two broad reasons that understanding what a model has learned is important. In some areas, the task for which a model is trained for is well understood (such as radiology/pathology/lending) in which case models should be examined in order to verify that they are indeed focusing on plausibly causal variables and not artifacts and that they are behaving fairly and ethically. In other areas (such as forecasting outcomes or selecting optimal treatment strategies in complex disease) it may be that no human is definitively capable of performing the task for which the model is being trained, in which case it is critical to harness what the model has learned to generate testable hypotheses to move those fields forward (and the model examination tasks discussed prior become more of a sanity check in these less understood tasks).

Here we present a unifying framework for understanding and explaining clinical AI models that address the above desiderata and additionally provide a method to implement it. We present these through two relevant examples; a well understood task using the public Pima Indians Diabetes Dataset, and a very new task of using longitudinal model to forecast future outcomes for patients with Rheumatoid Arthritis. Source code, Jupyter Notebooks, and data (for one of the tasks) are made available.

6.1.2. Methods

6.1.2.1. Goal

To develop a simple and intuitive method for explaining models of arbitrary complexity to an audience that are not experts with the inner workings of gradient-based models. The method must and be able to generate local and global explanations in a unified manner.

6.1.2.2. MAgEC

Conceptually, what our method does is to linearize a model around a given feature and time point (if using a longitudinal model). This is intuitive. Complex functions often behave linearly in the very near vicinity of a given point. This can be thought of as a local partial derivative.

Each patient can be represented with a two dimensional table, where each row is a variable and each column is a time point. Each cell in the table is the value for each variable at each time point.

The first step in the method is to calculate the predicted outcome for each case(patient) using the original/observed data. The predicted outcome is either a probability for classification tasks or a real number for regression tasks. The next step is to iteratively alter the value for each cell in each patient's table, while maintaining the original data values for all other cells, run each patient with the updated cell value through the model, and calculate how that single alteration affects the predicted outcome. For example, in a binary classification task, a particular case/patient ‘A’ may have a predicted probability of having outcome == 1 as .68 using the original data. After perturbing a single variable,X, and re-running the patient-A through the model, we might observe that the patient-A’s probability of outcome==1 has changed from .68 to .72. To compare the effect of variable X on the outcome, for patient-A, we take the difference of the logits between the observed data and the perturbed data. The result then (the difference of logits: new_logit - baseline_logit) can be interpreted identically to the coefficient in a linear or logistic regression model (the constant/linear effect of X on the outcome). Indeed, we show that in the case of a regression model, DECs exactly recover the variable coefficients when applied to the mean of the sum of the individual cases for each variable/timepoint.

The nature of the perturbation within a cell is determined by the original data type of the variable, continuous or categorical. Continuous variables are perturbed by increasing their value by a marginal amount, such as 1e-3, which represents the logit for that variable as the limit goes to zero. This can be described as the change in the predicted outcome with a nearly zero increase in the given variable. The amount of the change is technically a hyperparameter of the method, which we refer to as epsilon by convention, however our initial experiments have shown the method results to be extremely robust to changes in epsilon (see supplement) and we currently leave it as a fixed parameter. Categorical variables are assumed to be one-hot encoded. The state of categorical variable is altered by switching the encoding from 0 to 1 if the original state was a 0, and vice-versa if the original state was a 1. Categorical variables with more than two levels are treated cohesively. For example, each level is altered iteratively, one at a time, within the category to ensure that invalid combinations of variables do not occur.

We then apply the process for each cell in each case/patient’s two-dimensional representation table. Therefore, this method takes in a two-dimensional table for each case where the cell values are the feature values and it outputs a two-dimensional table for each case where the cell values are the differences between the logit generated by the original feature and the logit generated by the perturbed data.

6.1.2.3. Cases

To understand what the model has learned about how changes in each variable will affect the outcome for individual cases (case heatmaps) this is the end of the technical process all that remains is interpretation of the results. For a given case, if a given feature was present in their original input data, and the logit for that feature is positive and the case's actual outcome was positive, then model attributes a portion of the case's improvement to that feature. If a given feature was not present in the original input data for a case, then the logit represents the model's prediction of the change in the case's outcome if that feature was added. ((Probably belongs in Discussion: Care should be taken to ensure a realistic interpretation of these results in each situation. In particular, in medicine it may be important to bin your model's variables into categories of potentially actionable/testable (such as treatment choices) and non-actionable (such as diagnoses, vitals, demographics). While this method may provide an implicit treatment recommendation, for example in the form of a predicted response to each possible treatment option, these recommendations should be considered for research purposes only and not for actual clinical care. The recommendation comes from the explanation, which is developed from what the model has learned. Not only is it likely that a given model is imperfect from a prediction perspective, and indeed all explanations should be interpreted in the context of a model's overall ability to perform the task for which it was trained, but model's are also subject to the same sampling bias that physicians are. Controlled, real-world, trials will be necessary to validate any insights or promising approaches discovered through model explanation/examination.))

6.1.2.4. Features

To understand what the model has learned about the ‘big picture’ of how a particular variable influences the outcome, on average, a coefficient for each feature (at each timepoint) can be generated by calculating the mean of the logits for all of the individual cases for that feature/timepoint. We call these Directional Effect Coefficients (DECs). As with logistic regression, DECs are a linearization of the outcome to the input feature and (keep the following here, or put it with Cases??),they provide the constant association between the feature and the outcome, and can be interpreted as the model’s (average) sensitivity to the input feature for each patient.

6.1.2.5. Population

Populations are defined on the basis of similarity or distance. For example, similarity could mean living in a similar region or it could mean having traits in common. To identify populations or subgroups, one must first define what measure of similarity they will use for comparison. For me, an appealing metric of similarity in the clinical setting is the association between independent variables and dependent variables, or the effect of an input variable on the patients outcome. Applying this metric we can say that two patients are similar with regards Variablei if, while accounting for all other variables, a similar change in Variablei for both patients results in a similar change in their clinical outcome. For example, if the outcome of interested in is hA1C at their next visit, two patients may be considered similar with respect to

BMI if, while accounting for all other variables such as medications and demographics, changing each person's BMI by a set amount results in a nearly identical change in their hA1C at the next visit. In addition to providing a convenient measure of similarity between patients, this approach could also serve as a method for treatment recommendation. If we can identify patients that have similar responses to a particular drug, while accounting for all of their other variables, we could use that to select an optimal treatment for them. Using MAgEC, this similarity score can be generated by simply plotting coefficients against patients, ordered by coefficient magnitude, to identify those patients who are most similar.

6.1.2.6. Datasets

(1) The Boston Housing Dataset was used to generate a linear regression model to which MAgEC was applied to. The coefficients from the model itself were compared to the MAgEC coefficients to assess the fidelity of MAgEC coefficients in a purely linear model.

(2) The Pima are a group of Native Americans living in Arizona. A genetic predisposition allowed this group to survive normally to a diet poor of carbohydrates for years. In the recent years, because of a sudden shift from traditional agricultural crops to processed foods, together with a decline in physical activity, made them develop the highest prevalence of type 2 diabetes. For this reason they have been subject of many studies.

The dataset includes data from 768 women with 8 medical diagnostic predictor variables and one target variable, Outcome

Variables:

- Number of times pregnant
- Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Diastolic blood pressure (mm Hg)
- Triceps skin fold thickness (mm)
- 2-Hour serum insulin (mu U/ml)
- Body mass index (weight in kg/(height in m)^2)
- Diabetes pedigree function (see the paper)
- Age (years)
- The last column of the dataset indicates if the person was diagnosed with diabetes within 5 years (1) or not (0)

Source: The diabetes data containing information about PIMA Indian females, near Phoenix, Arizona has been under continuous study since 1965 due to the high incidence rate of Diabetes in PIMA females. The dataset was originally published by the National Institute of Diabetes and Digestive and Kidney Diseases, consisting of diagnostic measurements pertaining to females of age greater than 20.(Smith et al, 1988)

(3) Data and Model from Assessment of a Deep Learning Model Based on Electronic Health Record Data to Forecast Clinical Outcomes in Patients With Rheumatoid Arthritis (Norgeot et al, 2019)

6.1.3. Results

MAgEC coefficients perfectly match regression coefficients on linear models (Table 6.1). All three models that were developed to fit the Pima Indians dataset (logistic regression, random forest, MLP) achieved comparable results on the test set (~79%, See Table 6.2) which is consistent with published results. Interestingly, when compared on the basis of coefficients, either regression or MAgEC, the models learned different magnitudes of effect for the variables and sometimes different directions of effect (Figure 6.1: MLP MagECs vs Logistic Regression Coefficients).

Table 6.1: Boston Housing Data: Comparison of MAgEC to Regression Coefficients in Linear Model - Direction of Effect Coefficients

Our Method		**Actual Linear Coefficients from Linear Regression Model**	
	0		Coefficients
features		features	
NOX	-17.795759	NOX	-17.795759
DIS	-1.475759	DIS	-1.475759
PTRATIO	-0.953464	PTRATIO	-0.953464
LSTAT	-0.525467	LSTAT	-0.525467
CRIM	-0.107171	CRIM	-0.107171
TAX	-0.012329	TAX	-0.012329
AGE	0.000751	AGE	0.000751
B	0.009393	B	0.009393
INDUS	0.020860	INDUS	0.020860
ZN	0.046395	ZN	0.046395
RAD	0.305655	RAD	0.305655
CHAS	2.688561	CHAS	2.688561
RM	3.804752	RM	3.804752

Note: Boston Housing Data

Table 6.2: PIMA Results - PIMA Model Performance

Model	Accuracy	AUC
Logistic Regression	78.6	
Random Forest	79.2	
MLP	79.2	

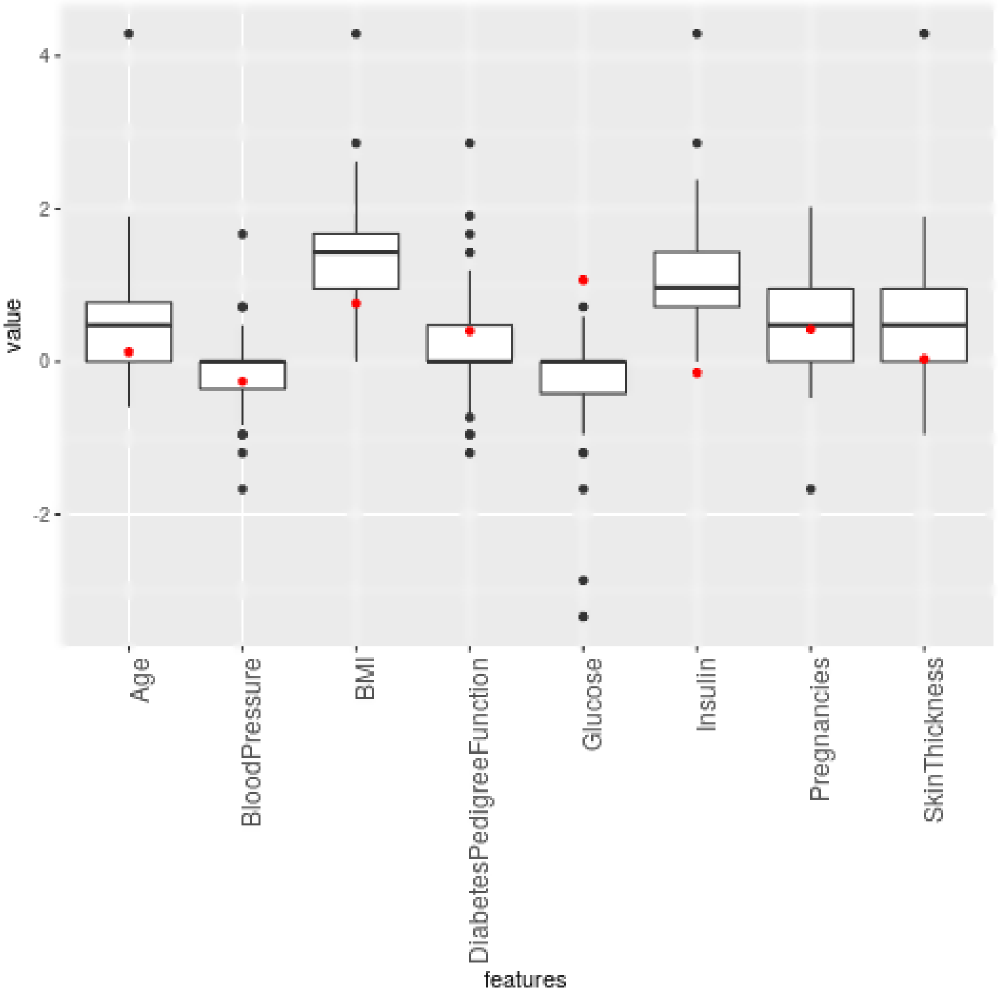


Figure 6.1: MLP MagECs vs Logistic Regression Coefficients - Comparison of MLP MagEC Distribution and Logistic Regression coefficients directly from the model.

Box and Whisker plot (and all black dots) show the distribution of MAgEC coefficients for each case within the test set. Red dots represent the logistic coefficients learned by the logistic regression model.

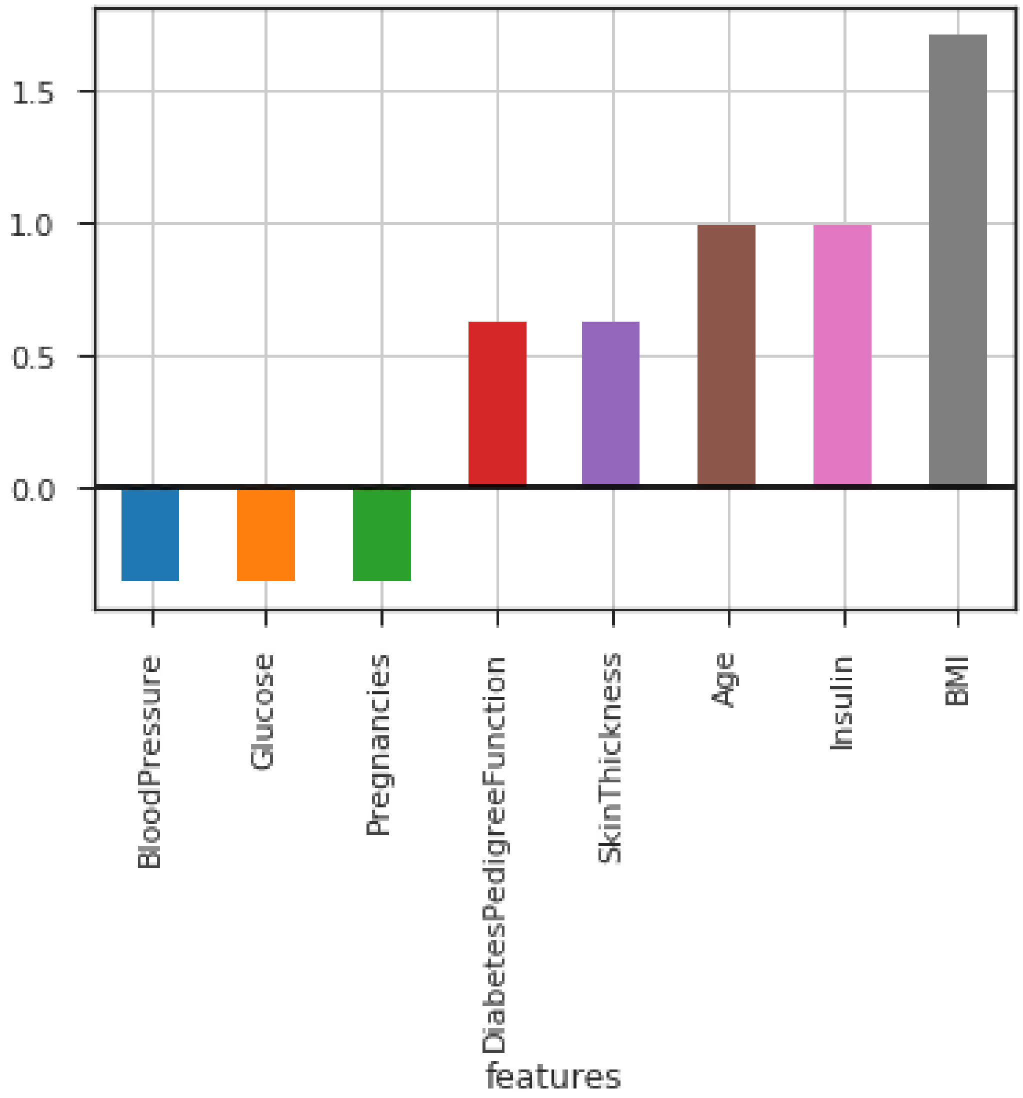


Figure 6.2: PIMA MagECs from MLP for a single patient

Similar patients: changes in bloodpressure expected to have similar changes in the probability of developing diabetes within 5 years, given all other variables

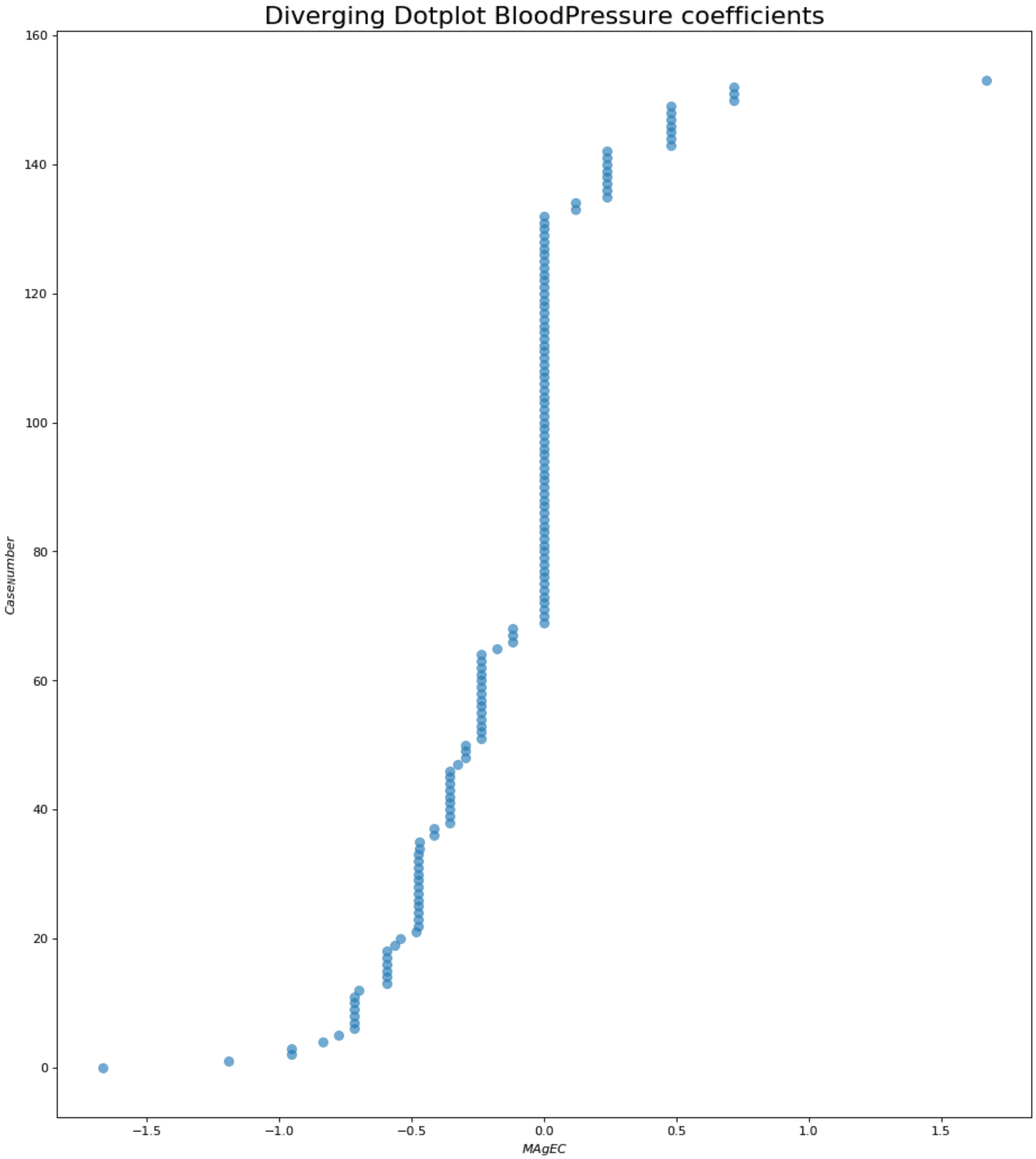


Figure 6.3: MagEC Clustering of PIMA patients on BloodPressure

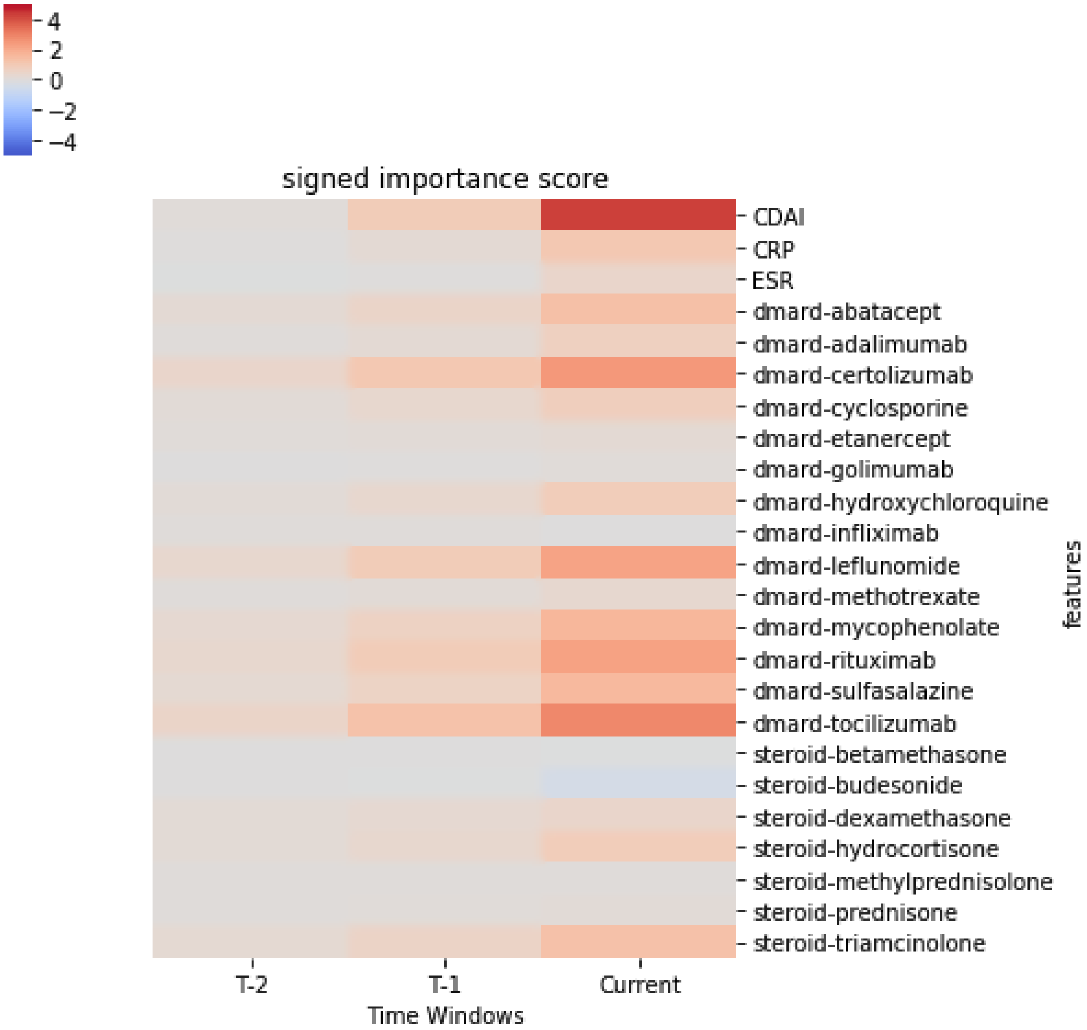


Figure 6.4: RA Results - Feature-Level RA MagEC

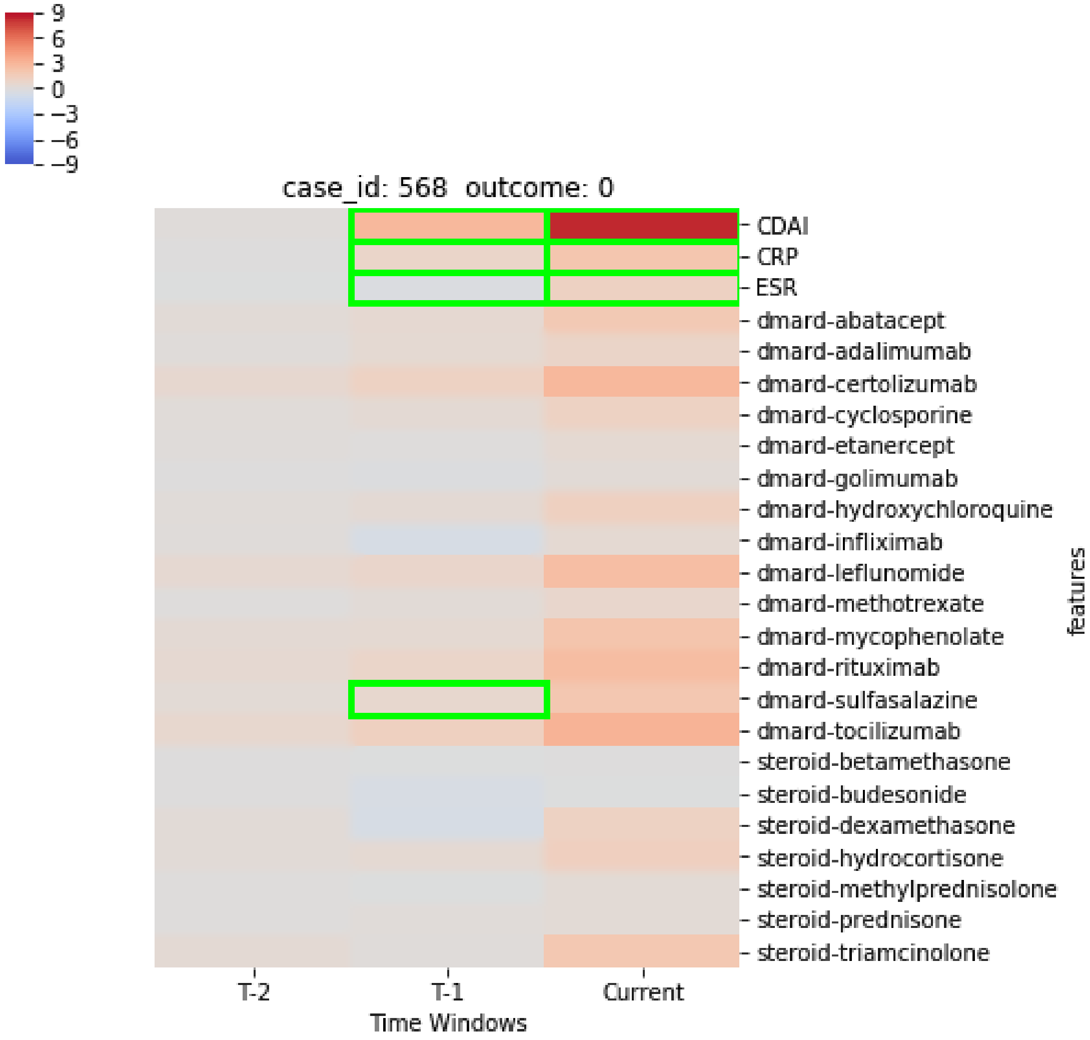


Figure 6.5: Individual RA Case MagEC

** Table showing case number (randomized), case outcome, MagEC, and variables that were actually present for that case.

6.1.4. Discussion

6.1.4.1. Summary

MAgEC provides an intuitive explanation of arbitrarily complex models for anyone that is familiar with regression coefficients. It provides explanations on both the global and local levels and also adds in a population level (or similarity between cases) explanation. While it is perhaps most useful in the context of deep learning, MAgEC can also be used to extend utility of

Random Forests and SVMs and allowing what many different model types have been learned to be compared directly. When MAgECs are visualized as a heatmap for temporal models they make it possible to understand complex temporal trends from a single image.

6.1.4.2. Comparison to Prior Work

SHAPs and LIME are existing methods in this space. LIME functions by generating surrogate data for a particular patient and then fitting a linear model to the surrogate data to generate an explanation. As such, it is currently limited to generating local explanations only. LIME is known to be highly sensitive to its methodological hyperparameters. I hve found it to generate non-sensical results on the data for this study, assigning direction and magnitudes of effect that did not align with classification labels. This was verified independently by a collaborator.

SHAPs functions by comparing gradient changes between cases and some background, essentially yielding how the current case differs from the average case with respect to each variable. I found this method well implemented and the results to be logically consistent. While it does not currently support global explanations, it could be directly extended to do so. SHAPs, while flexible is not truly model agnostic, it only supports models that learn via gradient descent. As such, it is not possible to compare deep learning models to random forests or regression models, two models that are highly utilized in clinical research, to determine if what they have learned is similar or divergent. A greater challenge of the SHAPs method is the explanation method itself, explaining complex models using gradients is simply not intuitive for most non-ML experts. Thus it may not facilitate trust and transparency to non-expert audiences. We consider MagEC an alternative, not an improvement over SHAPs, useful for a different set of practitioners and potentially a different set of problems.

Both SHAPs and LIME have demonstrated utility in images data and NLP, in addition to tabular data. At the moment MAgEC only supports tabular data.

6.1.4.3. Limitations

Currently there is no mathematically rigorous way to set epsilon, the unit of change, for continuous variables. Different values of epsilon have been shown to generate slight changes in the MAgECs. This is intuitive and its relevance is unclear. Choosing a value between 1e-3 and 1e-6 produces reasonably stable results in practice.

MAgEC has been designed initially for clinical models, which have traditionally used exclusively tabular data. At the moment MAgEC has no application extensions for images or NLP. Extensions for NLP models are fairly straightforward and are planned. Image extensions are not currently on the roadmap, I feel like SHAPs does an excellent job in that space already and there is no need for an alternative.

6.1.4.4. Future Directions

Predicted ‘best treatments’ can be assigned by generating MAgECs for a patient, extracting only the coefficients for potential treatment options and ranking them in ascending order. The highest rank is presumably the best option. This assertion could be strengthened for each patient by identifying prior patients who actually got that drug and had similar coefficients for that drug and checking their outcomes.

An approach to setting epsilon, either as one exact number or calibrating it for each model, should be explored.

Currently, MAgECs provide direction and magnitude of effect for single variables while accounting for all other variables. Ideally, coefficients or something equivalent to directly describe interactions (like a DEC for age*preg*glucose that is LEARNED not feature engineered) would be desirable. In theory, this could currently be done within the current MAgEC framework by perturbing combinations of features simultaneously, but the computational cost for most models would be prohibitive.

6.1.5. Conclusion

MAgEC provides an intuitive explanation of arbitrarily complex models for anyone that is familiar with regression coefficients. It provides explanations on both the global and local levels and also adds in a population level (or similarity between cases) explanation. While it is perhaps most useful in the context of deep learning, MAgEC can also be used to extend utility of Random Forests and SVMs and allowing what many different model types have been learned to be compared directly. When MAgECs are visualized as a heatmap for temporal models they make it possible to understand complex temporal trends from a single image.

References

1. Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261--265). IEEE Computer Society Press.
2. Norgeot B, Glicksberg BS, Trupin L, et al. Assessment of a Deep Learning Model Based on Electronic Health Record Data to Forecast Clinical Outcomes in Patients With Rheumatoid Arthritis. JAMA Netw Open. 2019;2(3):e190606. doi:10.1001/jamanetworkopen.2019.0606

Chapter 7

DeepMANN

7.1. Mann

A Deep Learning Pipeline to predict phenotype from microarray data

7.2.1. Background/Introduction

The ability to predict phenotype from genotype is extraordinarily valuable. Diagnostically, it could be used to determine which treatment regime is best suited to an individual. Prognostically, it could be used to begin interventions long before a phenotype was visible, thus potentially improving or even preventing disease states. Pharmaceutically, it could be used to select the drug or combination of drugs most likely to maximize effect and minimize adverse reactions.

Microarrays provide a fast and inexpensive way to sample the genome of an individual by checking to see how many copies of the most commmon nucleotide that individual posses at each of ~500k positions. Since humans are generally assumed to be exclusively diploid, theses arrays record 1 or 3 possible values at each position: 0 (the nucleotide at neither chromosome matches the reference), 1 (a nucleotide on 1 chromsome matches but the other does not), 2 (both match).

Genome Wide Association Studies (GWAS) seek to identify genetic differences between individuals that possess (cases) or lack (controls) a phenotype of interest by collecting microarray data on large numbers of individuals and then looking at each position on the microarray and comparing the prevelence of the number of zero's one's and two's between the cases and controls. The phenotye of interest is most often the presence of a particular disease or medical condition, or its absence.

The initial goal of GWAS was to increase biological understanding and discover targets for potential treatments by identifying a small number of powerful driver mutations (Single Nucleotide Polymorphisms or SNPs) that controlled a given phenotype. This was based on the assumption that most phenotypes possessed a small number of genetic determinants whose state determined the phenotype. The search was for a small number of drivers, each with an enormous effect size.

This approach necessitates that individual mutations be both relatively common in the population (so that they can be discovered among cohorts of thousands) and capable of dramatically altering the phenotype.

Studies over the last two decades have increasingly shown that the genetic contribution to human phenotypes is influenced by a large number of polymorphisms, most of which have a relatively small effect size. Given this knowledge, new approaches were developed that modeled phenotype as the additive effect of an increasing larger number of polymorphisms. Theses approaches search for a large number of drivers, each with a small effect. Some methods, such as Fast-LMM can even look at the entire set of microarray data to approximate the effect of each of the ~500k SNPs on an array.

The short coming of these approaches is the fundamental assumption that mutations have a linear relationship with a phenotype. Biology is fundamentally complex: each stage in central dogma inolves multiple pathways, and a single gene (coded by thousands of nucleotides) within one pathway is often involved in multiple other pathways. Thus, while there are a small number of exceptions where a single mutation is so disruptive that it's effect is synonymous with a change in phenotype, the rule is that there is an extraordinarily complex and non-linear relationship between DNA (nucleotides) and phenotype.

For nearly all common or complex diseases, it is not possible to predict phenotype from genotype at a clinically useful level. However, microarray data is extremely plentiful. A methodolgy capable of overcoming the shortcomings of GWAS and additive effect assumptions by capably modeling the complex non-linear relationship between genotype and phenotype could have significant clinical impact.

Deep Neural Networks (DNNs) have come to prominence in many different fields over the last 5 years because they are remarkably capable of identifying complex non-linear patterns. This capability comes from applying a series linear mappings followed by non-linear transformations which allows heirarchical learning of features. In the space of genetics this could allow the equivalent of using nucleotide mutations as raw inputs, combining those mutations into pathways, then learning pathway interactions, and finally determining which set(s) of pathways and pathway interactions are associated with healthy or disease states. (Obviously important to note that a DNN isn't going to re-create a biological pathway, it's going to create a function that models something equivalent to a pathway.)

DNNs are not magical, they learn slowly and require a large number of samples to effectively train. Empiracally, this often necessitates tens of thousands of samples per class. Furthermore the complexity of a DNN is directly limited by the relative number of input features to the number of samples presented for learning. This means that number of input features must be kept reasonably small and can only grow as the number of sample grows.

DNNs may be well suited for GWAS/microarray data since studies already exist that contain the tens of thousands of samples nessary to train a network capable of modeling complex non-linear relationships. Conducting further microarray studies is inexpensive and DNNs can be updated with new knowledge one sample at a time, there's no need to retrain the entire network when new data becomes available.

Scope

Complex diseases are by definition complex. Not only with complex genetic components, but also with complex environmental components, and even complex gene-environment components. Currently, deepMANN considers only the genetic contribution to phenotype and therefore it's predictive power is thresholded at a maximal upper bound set by each disease's heredity. Thus, while we hope the approach will be equally successful at uncovering the genetic patterns underlying diverse diseases, the ability to predict phenotype given genotype should be expected to vary considerably from disease to disease. Additionally, the true genetic contribution the overwhelming majority of human phenotypes is unknown. While this makes it difficult to quantify the method's absolute performance, it is fairly simple to determine its relative performance by comparing its predictive power to the current best practices.

Here we assess the predictive power of DNNs using a small number of SNPs and highly complex non-linear functions and compare it to Fast-LMM which uses the additive effects of all SNPs on the microarray.

7.2.2. Methods (Reader's Digest Version)

Microarray data was downloaded from the Welcome Trust Inflammatory Bowel Disease (IBD) cohorts comprised of 14925 individuals of which there were approximately 1800 cases and 13125 controls. The data came divided into a BED file containing the polymorphism data and a text file containing the phenotype information for each inidividual; labeling them as either a case or control.

The order of the individuals was shuffled so that cases and controls were randomly interspersed. The complete dataset was then split the into a training cohort comprised of 80% of the individuals (12000 members) and a testing cohort comprised of the remaining 20% of individuals (2925 members).

Due to the large class imbalance, the primary metric that was used to assess model performance was area under the receiver operating curve (auc_roc).

7.2.3. Data Preparation & Model Fitting

FAst-LMM

The FAast-LMM model was fit on the training cohort using all 360,000 SNPs present within the BED file. The model was first tested on the training cohort to ensure that the fit had occurred correctly and then tested on the testing cohort to determine the predictive power of the model out of sample.

Deep Neural Networks

Since the DNNs would be unable to effectively learn using all 360k SNPs, initial feature selection was done by performing GWAS on the training cohort using linear regression from the

FAst-LMM package. The top-N SNPs (where N was either 500 or 2000) were selected and the genotypes for each SNP were collected on all individuals. The SNPS were then arranged in order by chromosome number and chromosome position. Different models were built using different types of DNNs including Convolutional Networks and Multilayer Perceptrons each both 500 and 2000 SNPs.

The models were first tested on the training cohort to ensure that the fits had occurred correctly and then tested on the testing cohort to determine the predictive power of the model out of sample.

DNN Architecture Logic

Three different DNN architectural structures were developed representing two conceptually different approaches and a middle ground between them.

1. Pure Convolutional Network
2. Convolution followed by Neural Network
3. Multilayer Perceptron

Convolutional networks (convNet) have defined the success of Neural Networks in recent years. They comprise the architecture that has resulted in ultra-human performance on image recognition tasks and are also used in state of the art speech recognition and streaming vision such as that used by self-driving vehicles. Conceptually, convNets operate by searching for localized patterns within the input. An eye is an eye, regardless of whether it appears in the top, bottom, or middle of a picture. At the next level in the heirarchy, two eyes next to each other with a nose in the middle can make up a face. ConvNets were used as a deepMANN architecture for two reasons; as a potential filter for linkage disequilibrium (LD) and to capture any localized patterns in mutation. ConvNets have recently been showing success when employed on DNA sequence data, however since microarrays do not sample the genome at equa-distance positions it is unlikely that deep convNets will be the optimal solution as convNets discard all but the most powerful local signals.

Multilayer Perceptrons (MLPs) consist of multiple fully connected layers of neurons. They care nothing about locality and instead search for global patterns using the entirty of the data from the previous layer in the current layer. In general, MLPs are capable of modeling much more complex functions than convNets can, but this comes at the expense of requiring much more computational resources and raw input in order to effectively learn. The additional computation necessary is primarily memory: MLPs have considerably more absolute parameters, additionally no parameters within an MLP are redundant as is the case for convNets, and finally MLPs do not discard any information during forward propogation while convNets generally discard the majority of their data. The fact that MLPs have more absolute parameters to learn is what necessistates the greater volume of raw input.

A middle ground between the convNet and the MLP was reached by effectively attaching a deep MLP to a relatively shallow convNet with goal of using convolution primarily to filter LD and then using the MLP to model the heirachrical global complex non-linearity by which genotype becomes phenotype.

Optimization Choices

Binary cross-entropy was selected as the loss function to optimize. The class imbalance, which at only 8:1 is much smaller than that observed in most biological settings, initially prevented the DNNs from being to learn anything at all. The solution was to modify the loss function so that it more heavily penalized mistakes made on the under represented class by multiplying the loss for that class by the ratio of the class imbalance. This proved sufficient to allow the maximum network performance. Penalizing the loss at a ratio greater than the class imbalance did not result in additional gains in the prediction of the under represented class (even when the penalty was 10x the class ratio, or ~80x normal).

ADAM (Adaptive Moment Estimation) was chosen as the learning alogrithm; it is one of the current best performers in the industry and literature (as of Aug 2016) and does especially well if the input data is sparse (such as when using one-hot encoding for the convolutional networks). ADAM does per parameter optimization. Nodes that receive large gradients will see their effective learning rate reduced. Nodes that receive small or infrequent updates will see their effective learning rate increased. Adam accomplishes this by storing a running average of gradients (1st and 2nd momentum). So the downside is that it requires more memory. The enormous benefit is that it converges very quickly which not only speeds up training but also reduces the liklihood of being trapped in a local minima.Additionally, ADAM requires no user adjustment of learning rate or momentum to achieve optimal performance.

Architecture Development Process

Initial network structure was decided by considering the size of the raw input and attempting to adapt and combine the best practices from the fields of regulartory genomics and computer vision. Initially the training process was manually monitored and architecture adjustments were made to identify the number, type, and order of layers that resulted in reasonable performance. This was followed by grid searches to optimize the network's hyperparameters such as number samples to use prior to gradient descent update, number of training cycles, strength and type of regularization, the number and size of filters and pooling sizes for the convolutional layers, and number of neurons in each layer.

7.2.4. Results

GWAS Results

This is included only for reference purposes. It's the results of the simple regression and shouldn't be treated too seriously.

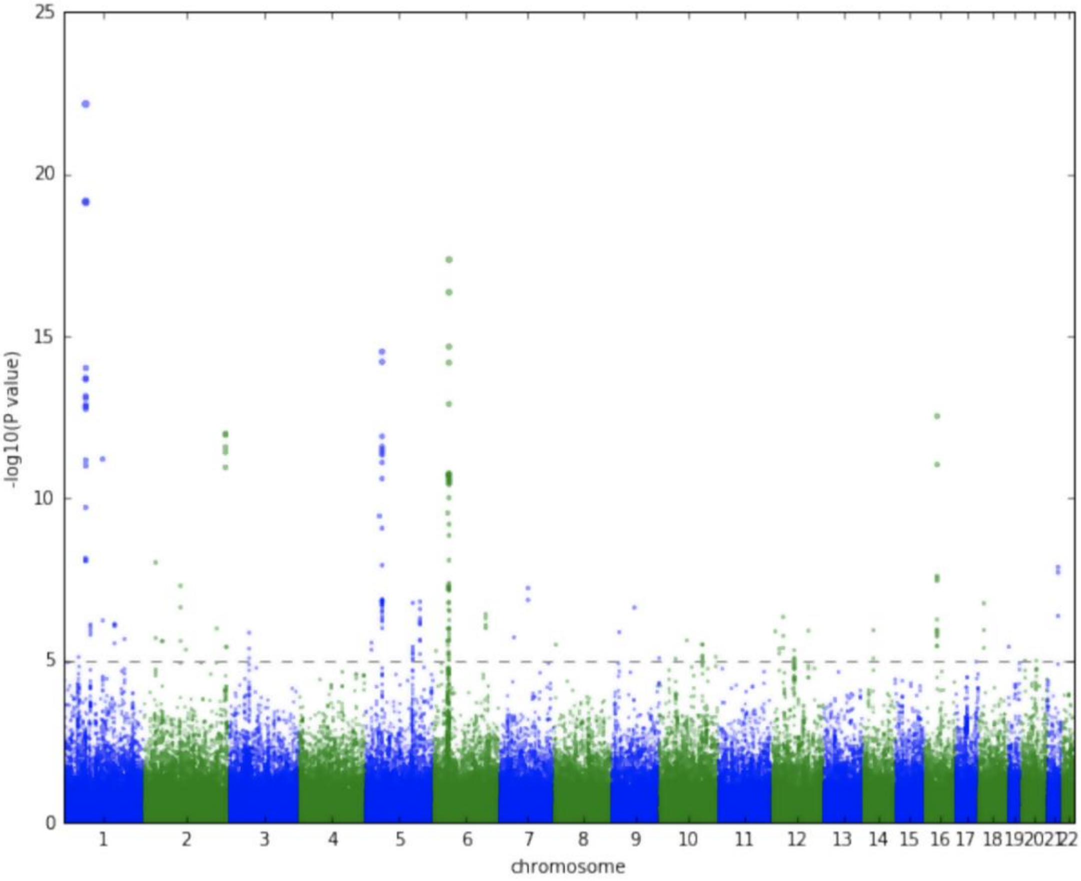


Figure 7.1: GWAS Results

FAst-LMM

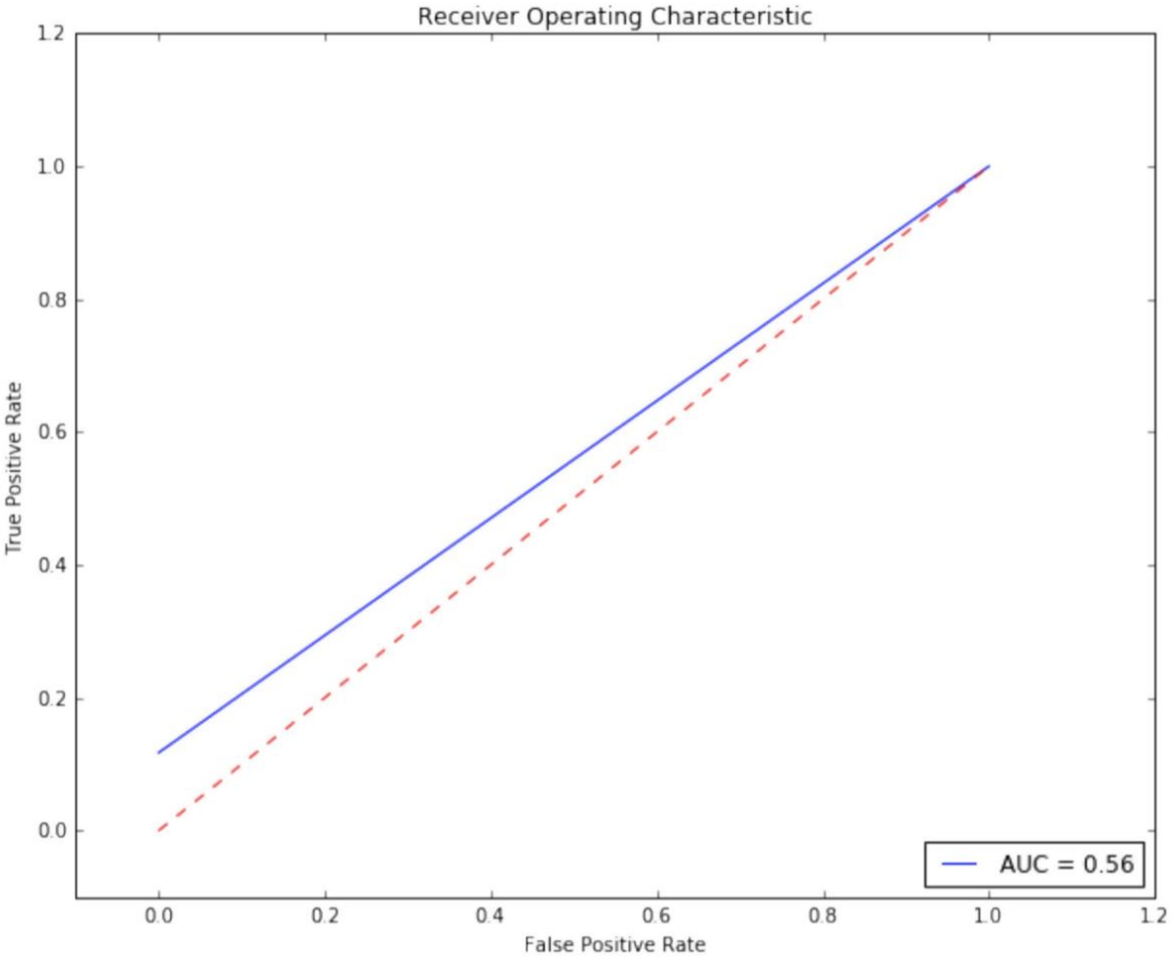


Figure 7.2: FAst-LMM In Sample Performance

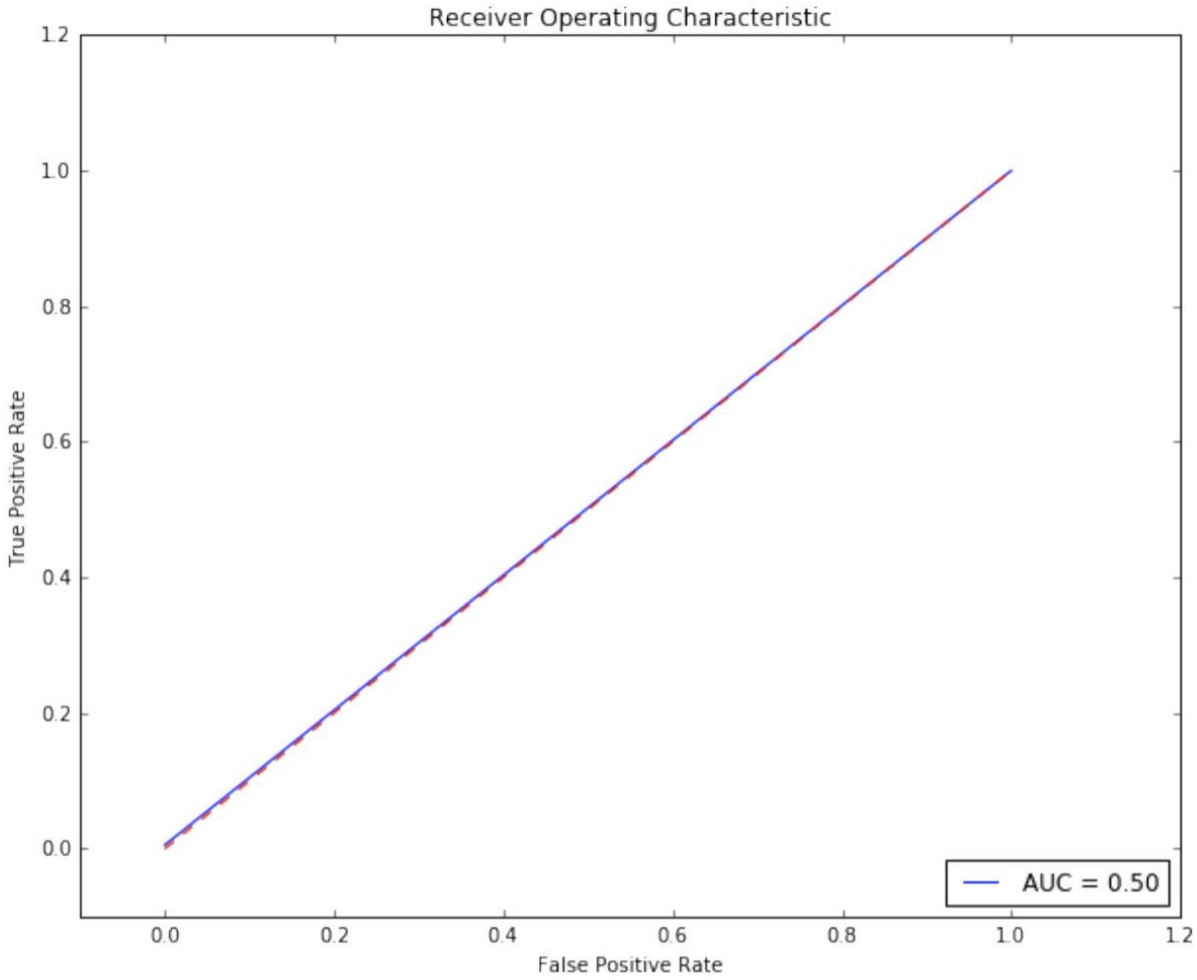


Figure 7.3: FAst-LMM Out of Sample Performance

DNNs

There were multiple different architectures, each with very different parameters that all converged to the same ROC scores. These are representative examples that reached the peak roc.

MLP

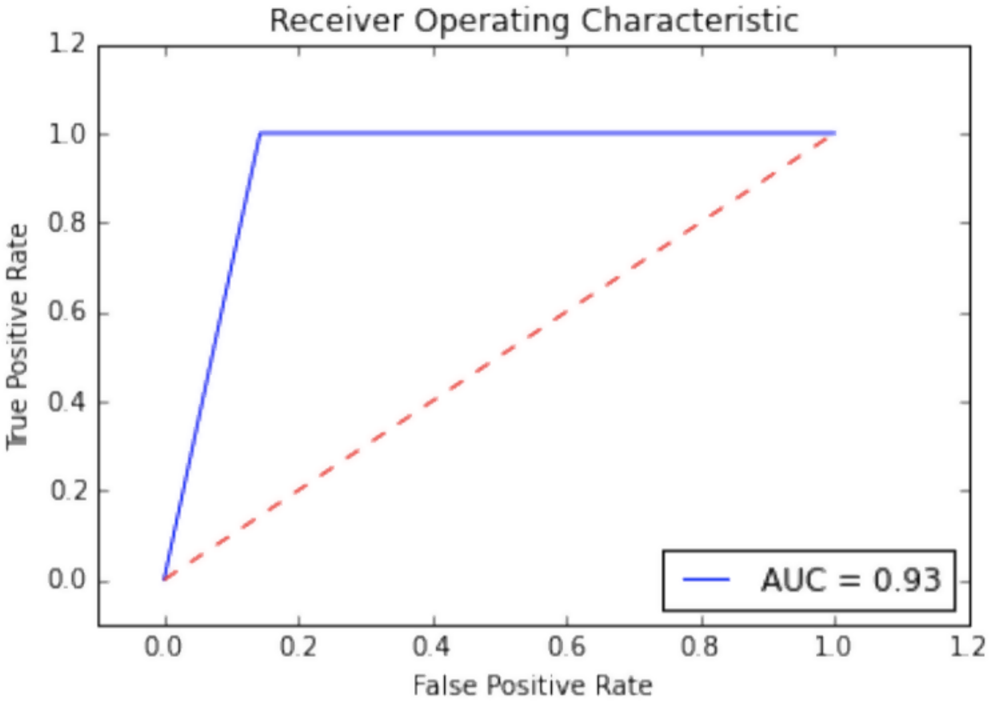


Figure 7.4: MLP2000 In sample ROC

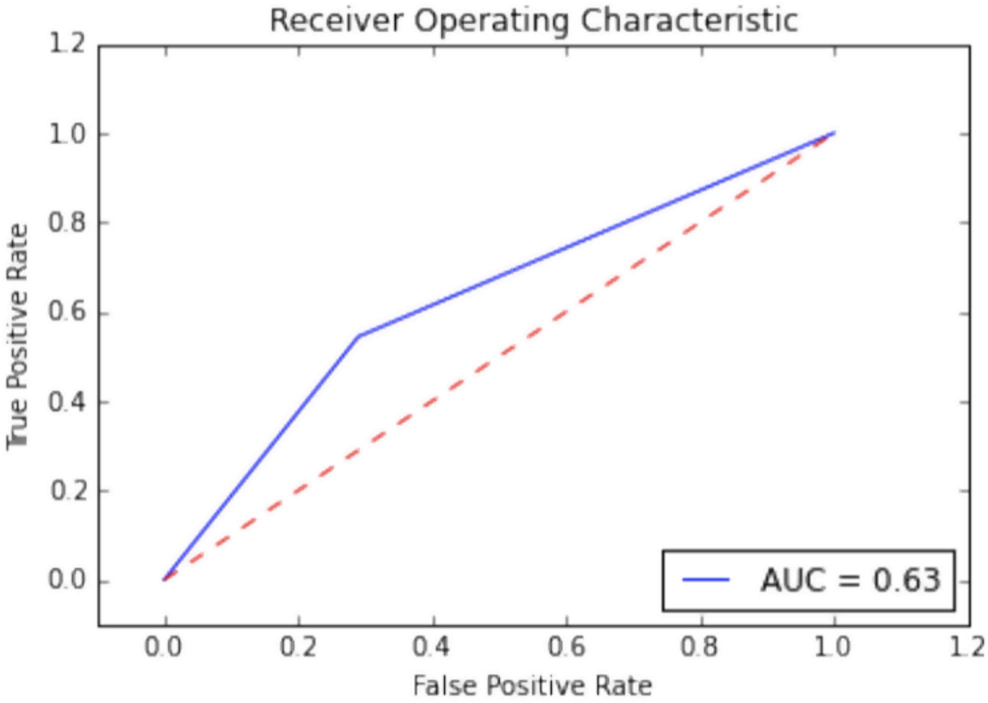


Figure 7.5: MLP2000 Out of sample ROC

Parameters:
* initialization = random uniform distribution
* dropout_rate = 0.5
* l2_regularization amount = 0.4
* input_layer_neurons = 2000
* neuronsfc1 = 300
* neuronsfc2 = 1500
* neuronsfc3 = 1500

ConvNet + DNN

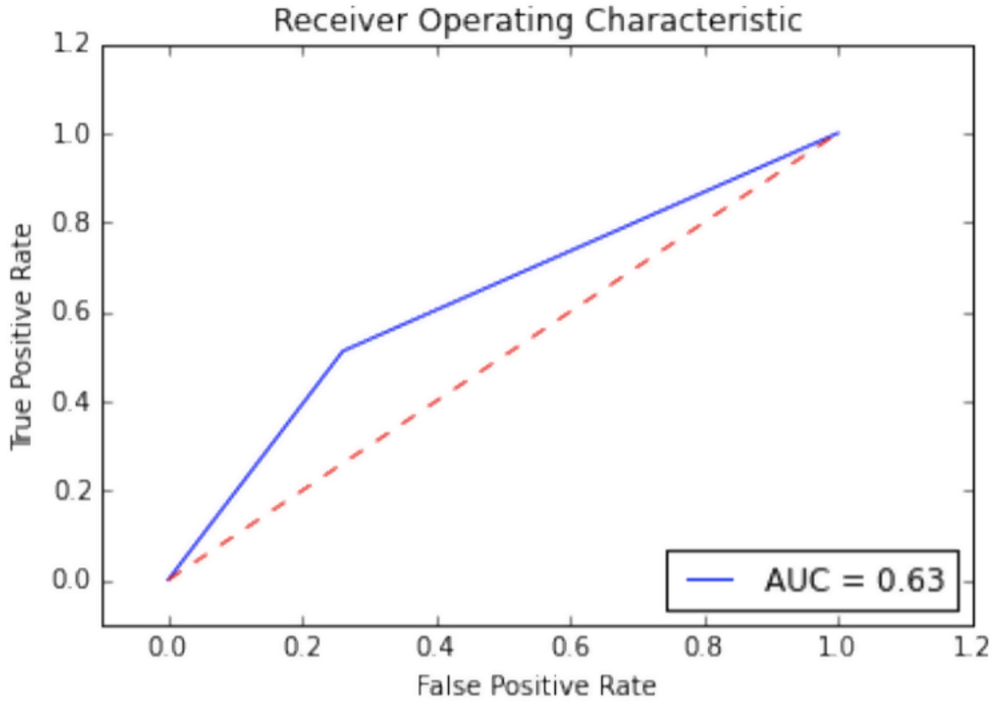


Figure 7.6: conv_500 out of sample

Parameters:
* dropout_rate = 0.5
* neurons = 100 in each dense layer
* zero padding used preseve dimensions
* 16 filters, each 3x3
* Pooling Size = 1x2 (max of 2 nearest)

Quick Summary:

The linear model had zero recall out of sample. Multiple architectures of DNNs outperformed had an roc_auc of .63 (Overall accuracy was between 69:71%).

7.2.5. Discussion

The DNNs reached a clear limit at .63 roc_auc, although the conv+NN converged to it using only 500 SNPs as input, while the MLP required 2000 to reach it.

This limit is likely to have been caused by either of 2 factors (or a combination):

1. *Not enough samples to learn additional complexity*: Class imbalance aside, with only ~1600 examples in the positive class to learn from the network is severely hampered. Reasonably complex DNNs empirically require a minimum of around 10,000 examples per class. Having a greater number of samples, even while maintaining the class imbalance, could allow exploitation of a more complex architecture which could provide greater predictive power.
2. *Genetic signal reached*: It is possible that SNP data is only capable of .63 roc_auc for IBD.

7.2.6. Future Directions

1. Use deepMANN on more interesting phenotypes with larger sample sizes
2. The ultimate medical goal is not to predict phenotype from genotype, but to create the most accurate prediction possible using all of the available data. DNNs have proven to be well suited to build strong multi-modal models. The current approach sets a foundation that be easily expanded upon.

Chapter 8

Philter

8.1. Philter

8.1.1. Abstract

There is a great and growing need to ascertain what exactly is the state of a patient, in terms of disease progression, actual care practices, pathology, adverse events, and much more, beyond the paucity of data available in structured medical record data. Ascertaining these harder-to-reach data elements is now critical for the accurate phenotyping of complex traits, detection of adverse outcomes, efficacy of off-label drug use, and longitudinal patient surveillance. Clinical notes often contain the most detailed and relevant digital information about individual patients, the nuances of their diseases, the treatment strategies selected by physicians, and the resulting outcomes. However, notes remain largely unused for research because they contain individually identifying data, or Protected Health Information (PHI). Previous clinical note de-identification approaches have been rigid and still too inaccurate to see any substantial real-world use, primarily because they have been trained with too small medical text corpora. To build a new de-identification tool, we created the largest manually annotated clinical note corpus for PHI and develop a customizable open-source de-identification software called Philter ("Protected Health Information filter"). Here we describe the design and evaluation of Philter, and show how it offers substantial real-world improvements over prior methods.

Keywords: Natural Language Processing (NLP); Electronic Health Records (EHR); De-identification; Clinical Notes; Protected Health Information (PHI);

8.1.2. Introduction

Structured EHR fields, primarily comprised of elements such as high-level demographics and billing codes (ICD), are currently the most utilized in determining the state of a patient, in terms of clinical care details or disease state. Many of these fields are often used in clinical research, and are now starting to be used to determine human phenotypes[1] for genome-wide association studies, and can be used to facilitate automated improvements of healthcare decision making[2]. However in many cases this information is not detailed enough to provide appropriate insights. Additionally, procedural, diagnostic, and medication billing coding are often incomplete, inconsistent, subjective, and inaccurate (often due to the needs of billing prioritizing over the needs of science), and this could even lead to false insights[3,4]. Clinical notes often contain the richest and most relevant information available about disease phenotypes, treatments, and outcomes as well as the clinical decision-making process. This written medical narrative frequently captures patient experience and event ordering timelines. To date, there have been many studies that have successfully used data from clinical notes for discoveries, including detection of drug adverse outcomes[5], identification of off-label drug use[6], surveillance of disease states[7], and identification of clinical concept relatedness[8].

With nearly the entire United States healthcare system now adopting electronic health records (EHRs), but with most of the actual clinical details captured in these free-text notes, transforming information contained within clinician notes into a computable resource is essential for medical research and improving patient care. However, clinical notes contain legally Protected Health Information (PHI), which prevent their use in most research applications.

Removal of PHI from clinical notes is a challenging task because the potential number of words that could be PHI are limitless. There are many different methods for recording and formatting patient note data across the health system landscape, and each health system serves a distinct patient population resulting in differences in the distribution of types of PHI across health systems[9] and the probability that a given word is PHI or a medical term (e.g.: ‘MA’).

The current state-of-the-art in de-identification systems still have real-world weaknesses because there are only a small number of corpora openly available for algorithm development and testing[10-14]. Priorities around de-identification software performance in recent years have been driven largely by de-identification competitions, most notably the Integrating Biology and the Bedside (i2b2) competitions in 2006 and 2014, which have emphasized a balanced approach of information retention and patient privacy, instead of national guidelines (https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html) which focus exclusively on privacy. It is clear that real-world performance is generally still below the threshold of compliance regulations for removing PHI, resulting in a lack of broader use of these tools to de-identify notes for research[9,15 ,16]. Every piece of PHI not identified and removed represents a potential violation of patient privacy and also a potentially expensive lawsuit. Even at 95% recall (i.e., percent of PHI removed), the amount PHI still remaining across millions of clinical notes would be staggering.

With an incredibly diverse patient population being treated at the University of California, San Francisco (UCSF), yielding over 70 million clinical notes collected within our Electronic Health Records (EHR), we required an efficient, accurate, and secure method for removing PHI from notes in order to make these data usable by researchers while minimizing the risk of PHI exposure. We developed a privacy-centric approach to removing PHI from free-text clinical notes using both rule-based and statistical NLP approaches. The algorithm utilizes an overlapping pipeline of methods that are state-of-the-art in each application including: pattern matching, statistical modeling, blacklists, and whitelists. We built this software tool as a self-contained system that could be deployed on any major computing platform and can operate without an internet connection, allowing it to be run in secure environments.

We have called this algorithm Philter (Protected Health Information filter). In this work, we describe the engineering of Philter and its evaluation against other systems. As we have discovered most existing tools in this field do not have actual open source availability, we have released Philter as open source code, and envision tens of thousands of health systems finding it useful.

8.1.3. Online Methods

8.1.3.1. Corpora: UCSF Corpus

To create the UCSF corpus of clinical notes, 4,500 notes were randomly selected from over 70 million notes from all departments at UCSF by assigning a hash identity to each note ID, randomly permuting the order of the hashed ID, then randomly selecting 4,500 hashed note IDs. Words were then manually annotated for PHI-categories by one of our three trained annotators. The annotators used Multi-document Annotation Environment (MAE)[17]. The MAE tool was configured with PHI elements following the HIPAA Safe Harbor guidelines with a couple of additional categories to identify provider information (Supplemental Table 8.1). 4,500 notes were annotated twice, with a second annotator reviewing and correcting the mark-up of the first annotator and Inter-Rater Reliability was calculated. When in doubt, annotators chose the more conservative option, for example marking an unclear name as belonging to a patient vs a physician. We generated a distribution of the randomly sampled notes and found more than one hundred note categories, note types, departments of origin, and provider specialties. We randomly assigned 2,500 notes to use for the development of a new de-identification algorithm (see Supplemental Table 8.2 for a distribution of the departments represented) and 2,000 notes to test algorithm performance (Supplemental Table 8.3).

The UCSF Committee on Human Research approved our study protocol [study # 16-20784].

8.1.3.2. I2b2 corpus

The i2b2 2014 de-identification challenge test corpus consists of 514 notes and was downloaded on July 18, 2017[10,11]. However, annotations of words as either safe or PHI within this corpus do not exactly follow the HIPAA guidelines for Safe Harbor, specifically in regards to locations and dates[18]. We therefore changed the annotations for words from the following categories: years in isolation, seasons (e.g. winter, spring), days of the week, patient/doctor initials in isolation, country names and ages under 90 from PHI to safe. The i2b2 2014 corpus replaced real PHI with surrogates. In a few instances, the surrogate values are for patient identification numbers were unrealistic, being four digits or less. These were removed.

8.1.3.3. Evaluating De-identification Performance

If PHI is allowed through a de-identification system, that yields a recall error, in that the PHI was not found. If safe words are obfuscated, that yields a precision error, in that extra text was unnecessarily removed. Since preventing exposure of PHI is our highest priority, we wanted to devise a system that minimized recall errors, even at the expense of greater precision errors.

Each PHI word that evades detection increases the risk of patient re-identification. Therefore, we evaluate performance at the word-level. In this analysis, we count as True

Positives (TP) those PHI words that were correctly labeled as PHI while the False Positives (FP) are non-PHI words that were incorrectly labeled as PHI. Likewise, True Negatives (TN) are non-PHI words correctly labeled as non-PHI while False Negatives (FN) are PHI words incorrectly labeled as non-PHI.

Since we chose to optimize our method to maximally maintain patient privacy, we chose recall as our primary measure of performance (Equation 1), which represents the portion of PHI words that were identified correctly:

$$Recall = {}^{TP}\!/_{TP + FN} \qquad (1)$$

However, de-identified clinical notes only have value if they retain as much non-PHI information as possible. Thus, we also measure precision (Equation 2), which represents the portion of filtered words that were non-PHI:

$$Precision = {}^{TP}\!/_{TP + FP} \qquad (2)$$

To account for precision, we selected the F2 score (Equation 3) as our secondary performance measure, which is a weighted average of recall and precision that values recall twice as much as precision:

$$F2 = {}^{5 * Precision * Recall}\!/_{(4 * Precision) + Recall} \qquad (3)$$

8.1.4. Algorithm

8.1.4.1. Algorithm Concept and Overview

The categories of PHI, the values of PHI, and the context surrounding PHI within a note can change drastically between types of notes, between departments within a health system, and between different health systems. In contrast to this, we believe that words that are not PHI have considerably less variability. Therefore, we started with an approach of identifying words that are not likely to be PHI. Approaches to identify words that are likely to be PHI were then incorporated into the algorithm for additional security and precision.

8.1.4.2. Algorithm Control, Customization, and Output

To optimize ease of use and modularity, while ensuring that the complete algorithm performs as expected, the pipeline is controlled by a simple text configuration file in the JSON format. We store the position of each character in memory so that tokens identified as PHI may be replaced with an obfuscated token of exactly the same length (e.g.,: 'John Smith' becomes '**** *****'). Therefore, the original structure of the note is perfectly preserved, with the exception that asterisks in the original note are replaced with spaces. The priority with which a token is marked as PHI or safe is dictated by the order of processes in the configuration file and is entirely customizable. We built an evaluation script that automatically compares de-identified notes to annotated gold-standards at the character level to quantify global and PHI category-specific performance.

8.1.4.3. Algorithm Pipeline

At the beginning of the pipeline, a custom script tokenizes individual words within each note by separating them on whitespace and symbols (i.e., -, /, #, &, periods, etc). Next, short phrases that have a high probability of not being PHI are identified using pattern matching with a custom library of 133 "safe" regular expressions. Then, a custom library of 171 regular expressions is used to identify predictable PHI entities such as salutations, emails, phone numbers, dates of birth, social security numbers, and postal codes. In both cases, the regular expressions search for specific words, phrases, and/or numbers and utilize the immediate context surrounding each word to identify matches. For example, if a number appears adjacent to the word 'age' or 'years old', that number is interpreted as an age and is PHI if it is greater than or equal to ninety, as per HIPAA guidelines for Safe Harbor methods. On the other hand, a number referring to dosage (e.g., 50 mg) is not interpreted as PHI.

At this stage, the Python NLTK module is used to tag each word with a part of speech (POS) to address the challenge of dealing with words that could be either safe or PHI, using statistical modeling to determine the structure of each sentence and document. For example, the word 'White' in the context of 'White fluid found at...' is an adjective and therefore safe, while 'Patient John White presents with...' is a proper noun and is PHI.

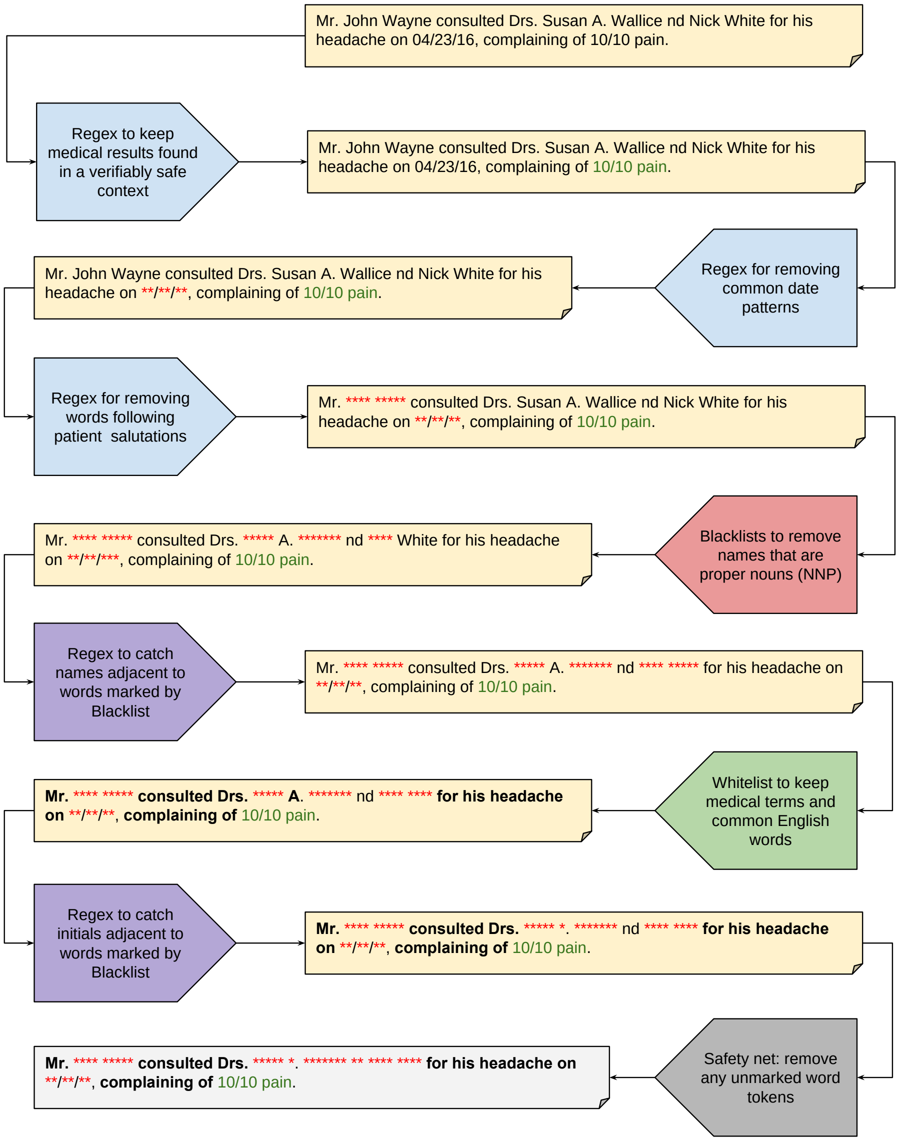
We assembled a blacklist of names using last names occurring 100 or more times in the 2010 U.S. census, and first names occurring five or more times for each year of birth between 1879-2017 from the U.S. Social Security website. To minimize occurrences of names that are also common words (i.e. new, walks, knee, home, child, etc.) in the blacklist, we removed a total of 855 words from the blacklist that were the greatest contributors to precision errors during training (complete documentation of blacklist creation is available on the public github repository). All names added to the final blacklist were tokenized on whitespace and symbols, and converted to lowercase. The blacklist was separated into a first names blacklist and a last names blacklist, and the two lists were incorporated into the full pipeline in succession. During the blacklist stage of PHI-searching, if a token is in at least one of the blacklists and is labeled as a proper noun by NLTK (e.g.POS tag = NNP), it is marked as PHI.

Next, an additional name removal step is implemented using a combination of regular expression and blacklist matching. We created a custom library of 4 regular expressions that search for common last name patterns in clinical notes (e.g. Jane Doe or Doe, Jane), and potential names are marked as PHI if an adjacent token was previously marked as PHI by a blacklist.

At this point, the pipeline employs a safety mechanism to catch PHI that occurs in unexpected formats, such as previously unseen names, words with incorrect POS tags, or misspellings. This is accomplished by identifying previously unlabeled (label = PHI/Safe) tokens that are most likely not PHI. This is accomplished using a custom whitelist of ~195,000 tokens comprised of medical terms and codes extracted from common medical word banks and ontologies (e.g., UMLS, SNOMED, MeSH, etc.), common medical abbreviations, the 20,000 most common English words and an additional list of common English verbs with varied tenses. All Social Security and 2010 Census names were removed from the whitelist, and some common English and medical words were then added back to the whitelist to maintain acceptable precision measurements (Complete whitelist documentation can be found on the github repository). All tokens that have not already been categorized as PHI or Safe by an earlier portion of the pipeline, with the exception of tokens with numeric POS tags, are passed through the whitelist.

A final active filtering process is used to identify patient and provider initials. We created a single regular expression that searches for initials patterns in clinical notes (e.g., Doe, J. or Jane S. Doe), and these regex matches are marked as PHI if one or more adjacent tokens were previously marked as PHI by a blacklist.

At the conclusion of the pipeline a token can have one of three possible labels: marked for exclusion, marked for inclusion, or unmarked. To maximize patient privacy, only words marked for inclusion are retained (Figure 8.1).



Mr. John Wayne consulted Drs. Susan A. Wallice nd Nick White for his headache on 04/23/16, complaining of 10/10 pain.

Regex to keep medical results found in a verifiably safe context

Mr. John Wayne consulted Drs. Susan A. Wallice nd Nick White for his headache on 04/23/16, complaining of 10/10 pain.

Regex for removing common date patterns

Mr. John Wayne consulted Drs. Susan A. Wallice nd Nick White for his headache on **/**/**, complaining of 10/10 pain.

Regex for removing words following patient salutations

Mr. **** ***** consulted Drs. Susan A. Wallice nd Nick White for his headache on **/**/**, complaining of 10/10 pain.

Mr. **** ***** consulted Drs. ***** A. ******* nd **** White for his headache on **/**/***, complaining of 10/10 pain.

Mr. **** ***** consulted Drs. ***** A. ******* nd **** ***** for his headache on **/**/**, complaining of 10/10 pain.

Mr. **** ***** consulted Drs. ***** A. ******* nd **** **** for his headache on **/**/**, complaining of 10/10 pain.

Regex to catch initials adjacent to words marked by Blacklist

Mr. **** ***** consulted Drs. ***** *. ******* nd **** **** for his headache on **/**/**, complaining of 10/10 pain.

Mr. **** ***** consulted Drs. ***** *. ******* ** **** **** for his headache on **/**/**, complaining of 10/10 pain.

Figure 8.1: Philter Algorithm pipeline

8.1.4.4. Optimization

2500 notes in the UCSF development corpus were used to develop the optimal Philter algorithm. Each portion of the pipeline, as well as the overall ordering of the pipeline, was modified to obtain the greatest overall performance metrics. Examples include changes to regular expression patterns, the tokens present in the White and Black lists, and the POS tags used to match against the lists. Optimization was done iteratively, developing against 500 notes at a time from the development set, testing against the next 500 notes in the development set, then repeating, growing the size of the development set by the previous 500 notes each time.

8.1.4.5. Comparators

Ferrandez et al.[9], performed a head-to-head comparison of multiple de-identification systems on multiple corpora, which revealed that the PhysioNet de-identification tool[11], had the best out-of-the-box performance. To identify PHI, the PhysioNet algorithm uses a combination of regular expressions and three types of lookup dictionaries (known names of patients and hospital staff, generic names of people and locations, and common words along with UMLS terms considered by their team unlikely to be PHI).

We selected the PhysioNet de-identification tool as the strongest comparator that met our criteria and downloaded the source code from PhysioNet 's[14] website (https://www.PhysioNet .org/physiotools/deid/) on February 12, 2017.

The National Library of Medicine's Scrubber tool, first published in 2013[19] takes the approach of maximizing recall and valuing real-world generalization over public challenge competition results. It has been continually revised and improved since its initial creation and investigators have even launched a trial[20] with updates as recent as 2018. The tool makes use of other public tools, including Apache's cTAKES[21] and UIMA projects[22], to compare the likelihood of words being PHI based on their relative frequency of appearance in public documents such as medical journals and LOINC codes to private physician notes under the reasonable assumption that words that appear in public documents are unlikely to be PHI. We selected the NLM Scrubber tool as our second comparator and downloaded the most recent version (v.18.0928) from the NLM website (https://scrubber.nlm.nih.gov/files/). Unfortunately, NLM Scrubber software does not maintain the original character alignment of scrubbed notes and comes with no method to automatically evaluate its performance against annotated notes. We had to design an evaluation script for this software and have made the script available to the community on our GitHub repository.

8.1.4.6. Framework for secure de-identification and evaluation

Figure 8.2 outlines the environment we designed to build and run Philter on clinical notes while ensuring security of the original notes and providing a framework for reporting PHI that was not filtered by the algorithm.

To ensure security, clinical notes were kept on a server with an encrypted drive protected behind an institutional firewall and through access-controlled VPN at all times from initial software development through institutional release. Access to the server was only permitted via password-protected SSH protocol from points inside the VPN, and only from devices which themselves had encrypted stores or hard drives. The raw clinical notes were loaded onto the server through a Clarity-level text document extraction from UCSF's Epic EHR system.

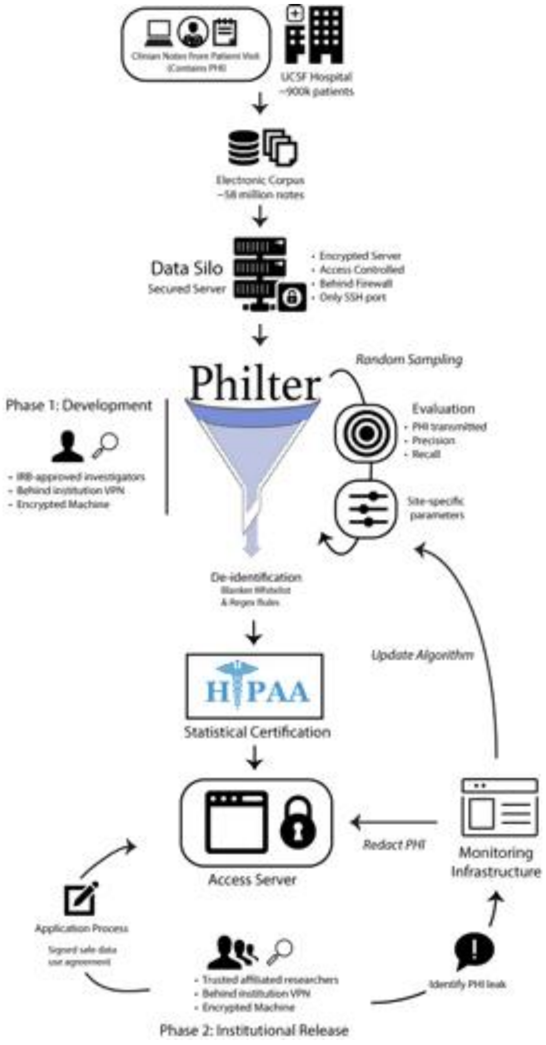


Figure 8.2: De-identification Ecosystem

8.1.4.7. Measuring compute time

We calculated the run time of our pipeline using batches of 500 notes on a 32 core Linux machine with 16GB of RAM using the native Linux Time function, ‘time’, to estimate the feasibility of running Philter at a large scale. We conducted two experiments. First, a single batch of 500 notes, with a total size of 2.2Mb, was run as single process and timed. Second, 20 batches of the 500 notes were run simultaneously as multiple processes and timed.

8.1.4.8. Sensitivity Analysis

In addition to Recall and F2 performance, we were interested in we were interested in the distribution of PHI across each category of PHI along with the number of TPs and FPs resulting from the best de-identification tool.

8.1.4.9. Open source code

The Philter package is written in machine-portable Python. The package can be installed via PIP, the Python package installer, and the source code along with detailed design descriptions as well as installation and use instructions can be obtained through the public repository open-sourced, under an MIT License (https://github.com/beaunorgeot/philter-ucsf-beta).

8.1.5. Results

The Inter-Rater Reliability, for PHI vs Safe tokens, between first and second pass annotators in the UCSF corpus was greater than 99.99%, with the second annotator identifying an average of 39 additional PHI tokens and converting an average of 21 tokens from PHI to Safe per 500 notes.

We compared overall recall and precision and per-PHI-category Recall across the three algorithms (Physionet, Scrubber, and Philter) on two corpora; the 2,000 note UCSF test corpus mentioned above and the publicly available 514 note 2014 i2b2 test corpus.

Primary and Secondary result metrics on both corpora are displayed in Table 8.1, with precision listed as a reference. On the UCSF test corpus: Physionet had a recall of 85.10% and an F2 of 86.15%, Scrubber had a recall of 95.30% and an F2 of 91.59%, and Philter had a recall of 99.46% and an F2 of 94.36%. On the 2014 i2b2 test corpus: Physionet had a recall of 69.84% and an F2 of 73.05%, Scrubber had a recall of 87.80% and an F2 of 85.22%, and Philter had a recall of 99.92% and an F2 of 94.77%.

Table 8.1: Performance Comparison of Tools and Corpora

	UCSF			I2B2		
	P	R	F_2	P	R	F_2
PHIlter	78.28	99.46	94.36	78.58	99.92	94.77
Physionet	90.62	85.10	86.15	89.49	69.84	73.05
Scrubber	79.24	95.30	91.59	76.26	87.80	85.22

Note: P = Precision, R=Recall.

Philter also outperformed both of the other algorithms for each category of PHI on both corpora, in addition to having the highest overall recall (See Table 8.2).

Table 8.2: Remaining PHI Analysis by Tool, UCSF Test Corpus

PHI Category	**Instances of PHI remaining (PHIlter)**	**Instances of PHI remaining (Physionet)**	**Instances of PHI remaining (Scrubber)**
Age >= 90	0	0	0
Patient_Vehicle_or_Device_Id	0	18	0
Patient_Account_Number	0	35	4
Patient_Medical_Record_Id	0	445	0
Patient_Social_Security_Number	0	0	6
Patient_Phone_Fax	0	0	1
Patient_Initials	2	120	132
Patient_Name_or_Family_Member_Name	6	211	93
Patient_Address	7	25	16
Patient_Unique_ID	20	442	34
Email	0	1	1
URL_IP	4	20	153
Date	7	257	269
Provider_Certificate_or_License	0	276	99
Provider_Name	12	546	90
Provider_Initials	12	236	217
Provider_Address_or_Location	43	1597	210
Provider_Phone_Fax	45	49	43

Note: PHI counts for PHIlter, Physionet and Scrubber performance on the UCSF corpus. Instances of PHI represent single tokens within the span of multiple or single-token items of PHI. Patient-only PHI is highlighted in blue, provider-only PHI is highlighted in yellow, and patient/provider PHI is highlighted in green.

Table 8.3: Remaining PHI Analysis by Tool, I2B2 Corpus

PHI Category	**Instances of PHI remaining (PHIlter)**	**Instances of PHI remaining (Physionet)**	**Instances of PHI remaining (Scrubber)**
AGE	0	1	0
DEVICE	0	6	0
MEDICALRECORD	0	524	18
PATIENT	2	154	92
DATE	0	4590	1587
FAX	0	2	0
PHONE	0	31	67
ZIP	0	3	1
USERNAME	1	92	92
STREET	2	27	21
LOCATION-OTHER	2	9	12
IDNUM	2	297	206
CITY	2	14	52
DOCTOR	5	197	186

PHI counts for PHIlter, Physionet and Scrubber performance on the I2B2 corpus

8.1.5.1. Sensitivity Analysis: Distribution of PHI and Philter Recall by Category

The raw count of PHI varied noticeably between the two corpora, but Philter's recall consistently generalized across the categories for each corpus (Supplemental Tables 4 and 5).

Results of additional sensitivity analyses regarding the precision errors caused by each element of the algorithm pipeline (Supplemental Tables 6) and the impact of partial PHI removal (Supplemental Tables 7 and 8) and can be found in the Supplement.

8.1.5.2. Philter Compute Time

The amount of real (wall-clock) time necessary to run 500 notes as a single process was 323 seconds. The amount of real time necessary to simultaneously process 20 batches of 500 notes, 10,000 notes total, was 401 seconds.

8.1.6. Discussion

8.1.6.1. Principal Results

In this study we developed an algorithm, Philter, that utilizes an overlapping pipeline of multiple state-of-the-art methods and compared it to the two strongest real-world competitors on the basis of recall. Philter demonstrated the highest overall recall on both corpora, had the highest recall in each category of PHI on both corpora, and generalized well between the corpora. Philter's recall on the 2014 i2b2 test corpus is the highest reported in the literature. A key design decision was the use of rules to separate PHI from Safe words while using a statistical method to improve precision. The overall size of the UCSF corpus at 4,500 manually annotated notes is the largest in the world that we are aware of. Likewise, the UCSF test corpus, at 2,000 notes, is the largest corpus to be tested and reported in the literature.

8.1.6.2. Limitations

Despite Philter's strong performance, with recall values equal to or greater than 99.5%, recall still was not perfect. The portions of PHI that were not identified were edge cases around existing patterns. For example, there were six total tokens that were missed for patient names in the UCSF test corpus. These tokens actually came from one single patient, whose name was six tokens long. The 6 token name appeared twice in one note, and each time Philter successfully removed three of the names, likely making the actual patient's name difficult or impossible to re-identify. The solution to this and similar problems are almost trivially easy to fix but they underscore the need to test de-identification systems on very large and diverse corpora to continually discover and refine edge cases.

The statistical portion of the pipeline was the most problematic from a precision perspective. The POS tagger frequently confused capitalized words, either at the beginning of sentences or all-capital words within sentences, as proper nouns. We found a very high overlap between common English words and medical terms (See, Whitelist) with names taken from the Census and Social Security. Precisely 16,095 names were found to be either medical terms or common English words. Therefore, an incorrect POS tag of NNP frequently resulted in a False Positive.

The decision not to include institution-specific information, such as a map between patient names and note identification numbers, could be considered a limitation. At the time of development, we chose not to include such information for numerous reasons. First, our lists of patient names are messy (it was not uncommon for drug names to appear as patient names in our databases). Second, even after rigorous initial cleaning, our patient name lists only detected 80% of name PHI within the corpus. This is in part due to the fact that patient family member names frequently appear within notes and in part due to misspellings of names. Third, relying on the use of inside data would not produce an algorithm that was generalizable out of the box. We believe that patient name-to-note maps could make a small but valuable addition to the pipeline and we envision placing it prior to the Names Blacklist steps. However, at the time of this writing, despite extensive development, we still are not ready to incorporate them. If we find that doing so improves performance in the future, we will provide the steps necessary to reproduce our process at other institutions on our github README.

8.1.6.3. Comparison with Prior Work

With more EHR systems being deployed across the world, there is still an incredible need for text processing tools, and de-identification is a key utility that can enable many readers and programmers to access those notes in a safer manner. While challenges and competitions have been run for nearly 10 years, there is still a pragmatic need for safe, efficient, open-source de-identification tools.

The field has been dominated by two separate approaches to designing de-identification algorithms. The first uses a rule-based system to detect PHI, while the second approach uses statistics to assign probabilities of PHI to words. Rule-based systems primarily use regular expressions and/or blacklists of words to tag PHI. Statistical methods employ machine learning, traditionally Conditional Random Fields and increasingly Recurrent Neural Networks, to learn patterns based on words and their context. Rule-based systems typically have better recall, while statistical methods typically have better precision. Rule-based systems are inherently predictable allowing their success and failures to be anticipated. Statistical systems are much faster to build; however, they are often difficult to interpret and performance on new data is more unpredictable. For example, the organizers of the 2006 i2b2 challenge discovered that the best performing algorithm in the competition, which utilized a statistical approach, suffered serious failures when de-identifying notes that came from the same hospital but were not drawn from the competition corpus[23].

The sparsity of available notes for de-identification system development and testing has provided a tremendous challenge to developing robust de-identification approaches because the nature of PHI contained within a note may differ significantly depending on the hospital or department they were generated from. Ferrandez et al.[9] demonstrated this by showing different proportions of categories of PHI distribution between the VHA, i2b2, and the Swedish Stockholm corpora. For example, Provider Names comprised only nine percent of the overall PHI in the VA corpus, but were nineteen percent of the PHI in i2b2, while there were no occurrences of Provider Names in the Stockholm corpus. Conversely, Patient Names make up only four and five percent of the VA and i2b2 PHI, respectively, but over 20 percent of the Stockholm corpus. ID Numbers were barely present in the VA corpus, totaling less than half of one percent of the PHI, but were responsible for more than twenty four percent of the PHI in the i2b2 corpus.

Between the systems selected as comparators for this study, the Physionet tool is the oldest and most 'proven'; it has great precision but does not effectively remove PHI. Scrubber is a newer software and the designers traded precision to get much improved recall. Unfortunately, neither of these approaches can be easily modified. Since PHI varies widely from corpus to corpus and the needs of those performing de-identification are diverse, the lack of customizability of these tools presents real-world usability challenges.

The NLM Scrubber software assumes that words appearing frequently in public documents are unlikely to be PHI, and although this assumption appears reasonable, it is not justifiable given our own findings. As mentioned above, we found over 16,000 names in the census and Social Security data that were either common English words or medical terms. This may explain the 20X difference between Scrubber and Philter in the number of patient name tokens that remained after filtering.

In addition to outperforming the comparators selected for this study, Philter sets new state-of-the-art recall results on the 2014 i2b2 corpus. The challenge winner, the Nottingham system, had a recall of 96.29 (Table 6: micro-averaged, token-wise, HIPAA category)[12]. Philter also demonstrates higher recall than the results reported for the more modern deep learning based de-identification systems (Dernoncourt et al.[24] i2b2 recall 97.38; Lui et al.[25] recall 93.8). Interestingly, the only publically available de-identification system used in the aforementioned competition, MITRE's MIST tool[26], faired quite poorly (HIPAA token recall of .805) even when supplemented with the well regarded Stanford NER tagger and pre-trained on an additional private corpus from Kaiser.

It is fair to note that the i2b2 Challenge systems and the deep learning systems mentioned in this manuscript attempted to maximize F1 rather than recall. While we believe that this is a flawed approach within the de-identification community (considering recall is the primary concern from a patient privacy standpoint), we acknowledge that tuning these systems to maximize PHI removal could potentially improve their recall performance.

As mentioned above, the POS tagger portion of the pipeline was the most problematic element from a precision perspective. Despite having lower recall and being subject to several statistical system challenges, such as lack of transparency and great risk of poor generalization to new corpora, we are excited by the very high precision of the deep learning approaches previously referenced[24,25]. We can imagine replacing the current NLTK POS tagger in the Philter pipeline with a deep learning version of the same.

8.1.7. Conclusions and Future Directions

In summary, Philter providers state-of-the-art de-identification performance while retaining the majority of relevant medical information. We envision that PHI removal can be further optimized using a crowd-sourcing approach with lots of exposure to many hospitals and notes. For this reason, we have made Philter open-source and highly customizable. We believe the system is capable of 100% recall with enough exposure and community involvement. The simple to use software will accept any text file as input, is fully modular to allow the community to improve the algorithm or adapt it to each users' specific needs, easy to evaluate, and executable in a secure environment. The software comes pre-configured, as the pipeline described in this manuscript, to produce the de-identification results that most closely follow HIPAA Safe Harbor guidelines.

8.1.8. Acknowledgements

The authors wish to thank Ozlem Uzuner and Amber Stubbs of the University of Albany for detailed information on the i2b2 dataset and early manuscript revisions. Research reported in this publication was supported by funding from the UCSF Bakar Computational Health Sciences Institute and the National Center for Advancing Translational Sciences of the National Institutes of Health under award number UL1 TR001872. Additional support was provided by the Amazon Cloud Research Fellowship (BN), The Silicon Valley Community Foundation (BN), the Russell/Engleman Medical Research Center for Arthritis (JY, GS), AHRQ R01 HS024412 (JY, GS), and the NIH grants P30AR070155 (MS & JY), NLM K01LM012381 (MS), K23AR063770 (GS), U24CA195858 (TG), and R01GM079719 (AB). The content is solely the responsibility of

the authors and does not necessarily represent the official views of the National Institutes of Health.

8.2. Supplement

8.2.1. Supplemental Background

8.2.1.1. Motivation

Initially, our plan was to identify a pre-existing de-identification system that we could use for this task. The number of open-source publicly available de-identification software systems is very small. We began the search for such a system by examining the HIPAA-defined, token-based, recall results of the i2b2 2014 de-identification challenge[12]. Unfortunately, the top performing entry, Nottingham system[27], was specifically fine-tuned for both the i2b2 dataset as well as the i2b2 evaluation script (using a post-processing script to modify tokens to maximize scoring), potentially limiting its generalization and resulting in over-optimistic assessment of performance. Additionally, the Nottingham system is not publicly available for use. Interestingly, the only publicly available de-identification algorithm that was used in the competition, MITRE's MIST tool[26], faired quite poorly (HIPAA token recall of .805) even when supplemented with the well regarded Stanford NER tagger and pre-trained on an additional private corpus from Kaiser.

A wider literature review of post-i2b2 challenge identified a couple of potentially promising candidates that used Deep Recurrent Neural Networks and reported results on the i2b2 2014 corpus for comparison[24,25]. However, the Lui et al.[25] system is not publicly available in any form, and while the Dernoncourt et al.[24] team have made available a Named Entity Recognition tagger based on their work, the de-identification system reported in their paper is not available.

8.2.1.2. Existing De-Identification Corpora

There are a very small number of public corpora that have been labeled for PHI and are available to develop or test de-identification algorithms. The Informatics for Integrating Biology and the Bedside (i2b2) program, released a corpus of 889 discharge summaries as part of a challenge in 2006 to evaluate state-of-the-art systems for automatically targeting and removing PHI[14]. In 2008, PhysioNet released a corpus of 2,434 nursing notes that they used to build a software de-identification tool[10,11]. In 2014, i2b2 released another corpus as part of a new challenge consisting of 1,304 longitudinal clinical narratives derived from 295 hand-selected diabetic patients at risk for coronary artery disease [12,13].

8.2.2. Supplemental Methods: UCSF Corpora

Supplemental Table 8.1: PHI Categories

PHI Categories
Age >= 90
Patient_Vehicle_or_Device_Id
Patient_Account_Number
Patient_Medical_Record_Id
Patient_Social_Security_Number
Patient_Initials
Patient_Name_or_Family_Member_Name
Patient_Address
Patient_Unique_ID
Email
URL_IP
Date
Phone_Fax
Provider_Certificate_or_License
Provider_Name
Provider_Initials
Provider_Address_or_Location

Note: Supplemental Table 8.1 Distribution of 2500 training notes Across Departments

Supplemental Table 8.2: Distribution of 2500 training notes Across Departments

Department_Specialty	**Count**
Gastroenterology	233
Obstetrics	225
Radiology	181
General Internal Medicine	177
Pulmonology	161
Pulmonary Function and Bronchoscopy	133
Ophthalmology	128
Obstetrics and Gynecology	121
Emergency Medicine	117
Family Medicine	103
Dermatology	82
Cardiology	75
Reproductive Endocrinology and Infertility	60
Kidney Transplantation	54
Endocrinology and Metabolism	51
Urologic Oncology	50
Hepatology	48
Primary Care	46
General Pediatrics	43
Neurology	40
Orthopedic Surgery	39
Liver Transplant	38
Neurosurgery	38
Anesthesiology	35
Pediatric Gastroenterology	35
Otolaryngology, Head and Neck Surgery	33
Radiology MR	30
Rheumatology	27
Radiology CT	26
Hematology and Oncology	25
Urology	25
Lung Transplant	20
Breast Care - Cancer Center	19
Pediatric Nephrology	19
Psychiatry	19
Allergy and Immunology	15
Interventional Radiology	15
Pediatric Cardiology	15
Geriatric Medicine	13
Lab	13
Nephrology	13
Pediatric Endocrinology	13
Pediatric Neurology	13

Department_Specialty	Count
Gastrointestinal Oncology	12
Physical Therapy	12
Dysplasia	11
HIV Program	10
Infusion and Transfusion	10
Pediatric Oncology	10
Pediatric Rheumatology	10
Gynecologic Oncology	9
Prenatal Diagnosis	9
Pain Medicine	8
Radiation Oncology	8
Anticoagulation	6
Heart Transplant	6
Nuclear Medicine	6
Pathology	6
Adolescent Medicine	5
Employee Health Services	5
Pediatric Hematology	5
Pediatric Otolaryngology, Head and Neck Surgery	5
Thoracic Oncology	5
General Surgery	4
Genetics - Cancer Center	4
Investigational Therapy	4
Optometry	4
Pediatric Pulmonology	4
Plastic Surgery	4
Executive Health	3
Home Health Services	3
Orthotics	3
Pediatric Immunology	3
Pediatric Urology	3
Sleep Medicine	3
Audiology	2
Colorectal Surgery	2
Endocrine Surgery	2
Orthopedic Surgical Oncology	2
Pediatric Anesthesiology	2
Pediatric Orthopedic Surgery	2
Pediatric Physical Medicine and Rehabilitation	2
Pediatric Surgery	2
Respiratory Therapy	2
STOR Immunizations Converted	2
Surgical Oncology	2
Thoracic Surgery	2

Department_Specialty	**Count**
Vascular Lab	2
Cardiothoracic Surgery	1
Clinical Research	1
Craniofacial Anomalies	1
Diabetes Services	1
Hospice and Palliative Medicine	1
Hospital Medicine	1
Infectious Diseases	1
Interpreting Services	1
Melanoma	1
Pediatric Bone Marrow Transplant	1
Pediatric Infectious Disease	1
Pediatric Infusion and Transfusion	1
Pediatric Occupational Therapy	1
Pediatric Pulmonary Function	1
Social Services	1
Support Service - Cancer Center	1
Vascular Surgery	1

Supplemental Table 8.3: Distribution of Testing Notes Across Departments

Department_Specialty	**Count**
Obstetrics	95
Radiology	73
Pulmonology	71
General Internal Medicine	70
Gastroenterology	69
Ophthalmology	66
Pulmonary Function and Bronchoscopy	64
Emergency Medicine	60
Endocrinology and Metabolism	51
Obstetrics and Gynecology	51
Family Medicine	50
Kidney Transplantation	38
Cardiology	34
Dermatology	30
Hepatology	27
Primary Care	26
Reproductive Endocrinology and Infertility	26
General Pediatrics	22
Liver Transplant	20
Neurosurgery	19
Pediatric Gastroenterology	19
Urologic Oncology	18
Hematology and Oncology	17
Neurology	17
Orthopedic Surgery	17
Radiology CT	15
Otolaryngology, Head and Neck Surgery	13
Radiology MR	13
Urology	13
Rheumatology	12
Anesthesiology	11
Gastrointestinal Oncology	10
Interventional Radiology	10
Breast Care - Cancer Center	9
Lung Transplant	9
Nephrology	8
Pediatric Endocrinology	8
Geriatric Medicine	7
Lab	5
Pediatric Nephrology	5
Anticoagulation	4
Dysplasia	4
Executive Health	4

Department_Specialty	**Count**
Pediatric Cardiology	4
Pediatric Rheumatology	4
Psychiatry	4
Radiation Oncology	4
General Surgery	3
Interpreting Services	3
Investigational Therapy	3
Neuro-Interventional Radiology	3
Pathology	3
Pediatric Neurology	3
Respiratory Therapy	3
Thoracic Oncology	3
Adolescent Medicine	2
Allergy and Immunology	2
Employee Health Services	2
Gynecologic Oncology	2
Heart Transplant	2
HIV Program	2
Infusion and Transfusion	2
Orthopedic Surgical Oncology	2
Pediatric Pulmonology	2
Prenatal Diagnosis	2
Surgical Oncology	2
Audiology	1
Endocrine Surgery	1
Endocrinology	1
Hospital Medicine	1
Integrative Medicine	1
Melanoma	1
Nuclear Medicine	1
Optometry	1
Pain Medicine	1
Pediatric Diabetes	1
Pediatric Hematology	1
Pediatric Oncology	1
Pediatric Orthopedic Surgery	1
Physical Therapy	1
Plastic Surgery	1
Sleep Medicine	1
Social Services	1
Symptom Management	1

8.2.3. Sensitivity Analysis

In addition to Recall, F2 performance, and our primary sensitivity analysis, we were interested in two additional sensitivity analysis. First, we were interested in determining the impact of partial de-identification successes, specifically, were there instances where only a portion of the PHI was removed that made the changed remaining associated tokens from PHI to safe. An example would be obscuring part of a date (eg: 1/1/2018 → */*/2018) or most of a name (eg: John A Smith → **** A *****). Second, while not emphasizing Precision as a de-identification metric, we wanted to catalog which elements of the Philter pipeline were the greatest contributors to precision errors to better anticipate which types of non-PHI words were most likely to be erroneously removed.

8.2.4. Supplemental Results

Supplemental Sensitivity Analysis One: What PHI Actually Remains after de-identification. Even when de-identification failed to completely remove an entire PHI entity, approximately 20% of the time it removed enough of the entity to make it no longer recognizable as PHI

Supplemental Sensitivity Analysis Two: Precision Errors

The portions of the pipeline that search for names were the most significant contributors to precision errors.

Supplemental Table 8.4: Recognizable PHI Analysis (PHIlter, UCSF Test Corpus)

PHI Category	Recognizable PHI
Age >= 90	0
Patient_Vehicle_or_Device_Id	0
Patient_Account_Number	0
Patient_Medical_Record_Id	0
Patient_Social_Security_Number	0
Patient_Phone_Fax	0
Patient_Initials	0
Patient_Name_or_Family_Member_Name	6
Patient_Address	4
Patient_Unique_ID	11
Email	0
URL_IP	0
Date	6
Provider_Certificate_or_License	0
Provider_Name	11
Provider_Initials	6
Provider_Address_or_Location	40
Provider_Phone_Fax	45

Supplemental Table 8.4. Recognizable PHI counts for PHIlter performance on the UCSF corpus. We defined “recognizable PHI” as any annotated identifier that was not PHI according to HIPAA after surrounding PHI was removed. There were 158 total FNs for Philter on the UCSF corpus initially, with 129 recognizable as PHI by human analysis after de-identification. Refer to the “Not Recognizable PHI” column in Supplemental Table 8.3 for detailed information on criteria used for determining recognizable PHI.

Supplemental Table 8.5: Recognizable PHI Analysis (PHIlter, I2B2 Corpus

PHI Category	Recognizable PHI
AGE	0
DEVICE	0
MEDICALRECORD	0
PATIENT	2
DATE	0
FAX	0
PHONE	0
ZIP	0
USERNAME	0
STREET	2
LOCATION-OTHER	2
IDNUM	0
CITY	2
DOCTOR	4

Supplemental Table 8.5. Recognizable PHI counts for PHIlter performance on the i2b2 test corpus. There were 16 total FNs for Philter on the UCSF corpus initially, with 12 recognizable as PHI by human analysis after de-identification.

Supplemental Table 8.6: False Positive Count by PHIlter Configuration File Element on the UCSF corpus

Filter	False Positive Count
Last Names Blacklist (lastnames_minus_fps.json)	1830
Whitelist	1725
First Names Blacklist (firstnames_minus_fps.json)	1236
'filters/regex_context/names_regex_context3.txt'	649
'filters/regex_context/initials.txt'	508
'filters/regex/dates/mm_yy_transformed.txt'	366
'filters/regex/addresses/hospital2.txt'	356
'filters/regex/dates/mm_dd_transformed.txt'	301
'filters/regex_context/names_regex_context2.txt'	252
'filters/regex/addresses/in_city_transformed.txt'	242
'filters/regex/ucsf_regex/ucsf_neighborhoods.txt'	226
'filters/regex/contact/xxx_xxx_xxxx.txt'	191
'filters/regex/salutations/post_salutations_2chars.txt'	172
'filters/regex/dates/dd_mm_transformed.txt'	161
'filters/regex/dates/month_name_transformed.txt'	108
'filters/regex/dates/mm_dd_yy_transformed.txt'	102
'filters/regex/salutations/pre_salutations_2chars.txt'	101

Supplemental Table 8.6. Each row name corresponds directly a file process within the pipeline and its relative location on the software filepath. False positive (FP) counts for PHIlter configuration file elements with FP counts >=100. Because multiple filters matched some FPs, FP counts do not reflect total number of FPs generated by PHIlter, but rather the total number of times each filter matched any FP.

Supplemental Table 8.7: UCSF corpus TP/FN Counts

PHI Category	TPs	FNs	Recall
Age >= 90	11	0	100.00%
Patient_Vehicle_or_Device_Id	550	0	100.00%
Patient_Account_Number	35	0	100.00%
Patient_Medical_Record_Id	471	0	100.00%
Patient_Social_Security_Number	30	0	100.00%
Patient_Initials	721	2	99.72%
Patient_Name_or_Family_Member_Name	1579	6	99.62%
Patient_Address	3996	7	99.83%
Patient_Unique_ID	652	20	97.02%
Email	120	0	100.00%
URL_IP	468	4	99.15%
Date	13396	7	99.95%
Phone_Fax	1469	45	97.03%
Provider_Certificate_or_License	369	0	100.00%
Provider_Name	5045	12	99.76%
Provider_Initials	721	12	98.36%
Provider_Address_or_Location	3998	43	98.94%

Supplemental Table 8.7. TP/FN counts and recall per PHI category for PHIlter performance on the UCSF test corpus. The following annotated PHI categories were not considered PHI for performance evaluation purposes, and not included in performance analysis:.

Supplemental Table 8.8: Overall Recall Per PHI Category (PHIlter, I2B2 Test Corpus)

PHI Category	TPs	FNs	Recall
AGE	7	0	100.00%
DEVICE	12	0	100.00%
MEDICALRECORD	721	0	100.00%
PATIENT	1445	2	99.86%
DATE	11880	0	100.00%
FAX	6	0	100.00%
PHONE	407	0	100.00%
ZIP	143	0	100.00%
USERNAME	91	1	98.91%
STREET	414	2	99.52%
LOCATION-OTHER	12	2	85.71%
IDNUM	377	2	99.47%
CITY	338	2	99.41%
DOCTOR	3231	5	99.85%

References

1. Kirby JC, Speltz P, Rasmussen LV, et al. PheKB: a catalog and workflow for creating electronic phenotype algorithms for transportability. *J Am Med Inform Assoc.* 2016;23(6):1046-1052.

2. Norgeot B, Glicksberg BS, Butte AJ. A call for deep-learning healthcare. *Nature medicine.* 2019;25(1):14-15.

3. Makary MA, Daniel M. Medical error-the third leading cause of death in the US. *BMJ (Clinical research ed).* 2016;353:i2139.

4. O'Malley KJ, Cook KF, Price MD, Wildes KR, Hurdle JF, Ashton CM. Measuring diagnoses: ICD code accuracy. *Health Serv Res.* 2005;40(5 Pt 2):1620-1639.

5. Iqbal E, Mallah R, Rhodes D, et al. ADEPt, a semantically-enriched pipeline for extracting adverse drug events from free-text electronic health records. *PLoS One.* 2017;12(11):e0187121.

6. Jung K, LePendu P, Chen WS, et al. Automated detection of off-label drug use. *PLoS One.* 2014;9(2):e89324.

7. Afzal N, Sohn S, Scott CG, Liu H, Kullo IJ, Arruda-Olson AM. Surveillance of Peripheral Arterial Disease Cases Using Natural Language Processing of Clinical Notes. *AMIA Joint Summits on Translational Science proceedings AMIA Joint Summits on Translational Science.* 2017;2017:28-36.

8. Finlayson SG, LePendu P, Shah NH. Building the graph of medicine from millions of clinical narratives. *Scientific data.* 2014;1:140032.

9. Ferrández Ó, South B, Shen S, Jeffrey Friedlin F, Samore M, Meystre S. *Evaluating current automatic de-identification methods with Veteran's health administration clinical documents.* Vol 122012.

10. Goldberger AL, Amaral LA, Glass L, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation.* 2000;101(23):E215-220.

11. Neamatullah I, Douglass MM, Lehman L-wH, et al. Automated de-identification of free-text medical records. *BMC Med Inform Decis Mak.* 2008;8:32-32.

12. Stubbs A, Kotfila C, Uzuner O. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1. *Journal of biomedical informatics.* 2015;58 Suppl:S11-19.

13. Stubbs A, Uzuner O. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of biomedical informatics.* 2015;58 Suppl:S20-29.

14. Uzuner O, Luo Y, Szolovits P. Evaluating the state-of-the-art in automatic de-identification. *J Am Med Inform Assoc.* 2007;14(5):550-563.

15. Deleger L, Molnar K, Savova G, et al. Large-scale evaluation of automated clinical note de-identification and its impact on information extraction. *J Am Med Inform Assoc.* 2013;20(1):84-94.

16. Meystre SM, Friedlin FJ, South BR, Shen S, Samore MH. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology.* 2010;10:70.

17. Rim K. Mae2: Portable annotation tool for general natural language use. Paper presented at: 12th Joint ACL-ISO Workshop on Interoperable Semantic Annotation2016.

18. Deleger L, Lingren T, Ni Y, et al. Preparing an annotated gold standard corpus to share with extramural investigators for de-identification research. *Journal of biomedical informatics.* 2014;50:173-183.

19. McMurry AJ, Fitch B, Savova G, Kohane IS, Reis BY. Improved de-identification of physician notes through integrative modeling of both public and private medical text. *BMC Med Inform Decis Mak.* 2013;13:112.

20. WebCite. Glossary of Data Element Definitions. 2019; http://www.webcitation.org/74K94OPQf.

21. Savova GK, Masanz JJ, Ogren PV, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J Am Med Inform Assoc.* 2010;17(5):507-513.

22. Ferrucci D, Lally A. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering.* 2004;10((3-4)):327-348.

23. Moore RC, Bilmes J, Chu-Carroll J, Sanderson M. Proceedings of the Human Language Technology Conference of the NAACL, Main Conference. 2006.

24. Dernoncourt F, Lee JY, Uzuner O, Szolovits P. De-identification of patient notes with recurrent neural networks. *J Am Med Inform Assoc.* 2017;24(3):596-606.

25. Liu Z, Tang B, Wang X, Chen Q. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics.* 2017;75s:S34-s42.

26. Aberdeen J, Bayer S, Yeniterzi R, et al. The MITRE Identification Scrubber Toolkit: design, training, and assessment. *International journal of medical informatics.* 2010;79(12):849-859.

27. Yang H, Garibaldi JM. Automatic detection of protected health information from clinic narratives. *Journal of biomedical informatics.* 2015;58 Suppl:S30-38.

Chapter 9

Deep Cumulative Dosage Determination

9.1. Abstract

Glucocorticoids, one of the most common classes of steroids, are prescribed with a great variety of cumulative dosages, often including complex tapers, due to their wide utility across a large number of health conditions. Cumulative steroid use is associated with several serious health conditions and is also an important surrogate outcome for auto-immune treatments. Currently, determining cumulative dosage is done through manual chart review which is both time consuming and error-prone. Here, I detail the process by which deep learning can be used in the Natural Language Processing space to automatically determine the total cumulative dose of a steroid that a patient has received directly from the EHR 'Sig' field.

9.2. Introduction

Glucocorticoids are one of the most commonly prescribed classes of medications in the United States. They are essentially immune-suppressors, most common administered as pills, and are primarily indicated for inflammatory conditions ranging from Asthma, COPD, and allergies to Rheumatoid Arthritis, Tendinitis, and Multiple Sclerosis. Glucocorticoids are generally considered safe for short term use but long term use is associated with potentially serious risks including Osteoporosis, cardiovascular complications, and diabetes. For many of the health conditions in which they are used, these steroids serve as secondary treatments, taken during acute attacks when front line treatments fail. For example, the frontline treatment for Rheumatoid Arthritis are a group of drugs collectively known as Disease Modifying

Antirheumatic Drugs (DMARDs). DMARDs are designed to specifically treat RA, however if a patients disease suddenly ‘Flares up’, glucocorticoids may be additionally prescribed to suppress the entire immune system. In this way, glucocorticoid usage can also be a surrogate for efficacy of frontline treatments (such as DMARDs).

Deep Learning has become state-of-art for many Natural Language Processing (NLP) tasks, including sentiment analysis and document classification which both closely mirror the task of assigning cumulative dosage, a discrete number, from a string of text.

9.3. Methods

9.3.1. Data

Two fields were extracted for patients who had received Prednisone, the most common glucocorticoid, prescriptions: “Sig” and “Dosage”

Sig fields within the EHR contain the physicians short hand written instruction to the patient about what dosage of the drug to take, when, and for how long, including information on potentially tapering the dosage taken over time.

The Dosage field contains the volume of prednisone contained within each pill (not the cumulative dosage).

The data itself varies widely in terms of sig length, complexity, and quality

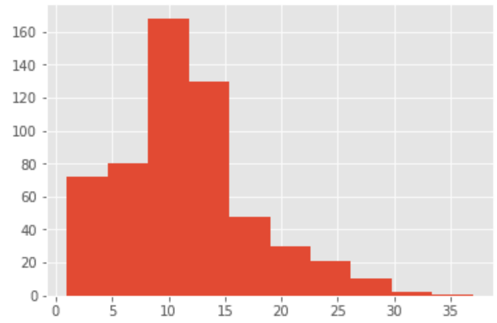


Figure 9.1: Sig Word Count

Figure 9.1: Length of sig, in words, on the x-axis. Count of the number of sigs that contained that word-count on the y-axis.

9.3.2. Data Processing

The clinical researchers have determined that exact cumulative dosage is less relevant binning the raw cumulative dosage into categories which describe levels of use. These categories were determined to be: ‘low’ if the cumulative dose was less than 5mg, “moderate” if the cumulative dosage was between 5-10mg, “high” if the cumulative dosage was between 10-20mg, and “very high” if the cumulative dosage was greater than 20mg.

9.3.3. Gold Standard

To assign gold standard labels, 845 charts were manually reviewed by trained clinical researchers and assigned one of the four labels. 25% of these, or 212 total samples were held aside for testing, the remaining samples were used to conduct three experiments

9.3.4. Base Model

I was initially most interested in the type of data, as opposed to model architecture, that would best solve this type of problem. Therefore, for prototyping purposes I chose a simple single architecture that seemed to perform well initially on this task and multiple related tasks.

Table 9.1: Pred-sigs Base Architecture

Layer (type)	**Output Shape**	**Param #**
embedding_1 (Embedding)	(None, 20, 64)	18368
conv1d_1 (Conv1D)	(None, 18, 32)	6176
max_pooling1d_1 (MaxPooling1)	(None, 9, 32)	0
dropout_1 (Dropout)	(None, 9, 32)	0
lstm_1 (LSTM)	(None, 32)	8320
dropout_2 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 16)	528
dropout_3 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 4)	68

Total params: 33,460
Trainable params: 33,460
Non-trainable params: 0

9.3.5. Experiments

1. Sig field alone into the base architecture
2. Language model built using all labeled training sigs, then transferred to the classification task
3. Dual input model using sig field and dosage

For all experiments, sig length was treated as a hyperparameter.

9.4. Results

Table 9.2: Initial Experimental Accuracy

Experiment	Test Set Accuracy
Sig Field Alone	70%
Lang Model + Sig Field	68% **
Sig + Dosage	77%

Table 9.3: Classification Report of Sig+Dosage Experiment

	precision	recall	f1-score	support
0	0.76	0.93	0.84	86
1	0.81	0.61	0.70	62
2	0.62	0.60	0.61	35
3	0.88	0.79	0.84	29
avg/total	0.77	0.76	0.76	212

Table 9.4: Sensitivity Analysis: Incorrect Predictions for Class "High" (2) Sig+Dosage

		Sig	Dosage	Pred_class/True_Class	
34	mg total by mouth daily take 20mg a day for one month then decrease by 5mg every 2 weeks until you reach 10mg a day	10.0	690	1	2
52	take 5 mg by mouth 4 four times daily	1.0	706	0	2
56	generic for deltasone take 2 tablets by mouth daily	10.0	715	0	2
57	mg total by mouth daily take 2 tablet 20 mg total by mouth for 3 days then decrease to 1 tablet 10 mg total daily	10.0	649	1	2
73	tablet 5 mg total by mouth daily 30mg d x 1 week 20mg d x 1 week 15mg d x 1 week then 10mg d	5.0	642	1	2

		Sig	Dosage	Pred_class/True_Class	
106	take 1 tablet 2 5 mg total by mouth daily for a total of 12 5mg	2.5	641	0	2
111	mg x 5 days take 3 tabs 30mg x 5 days take 2 tabs 20mg x 5 days then take 1 tab 10 mg thereafter	10.0	696	0	2
133	take 2 tablet by mouth every day	10.0	675	0	2
157	take 1 5 tablets daily	10.0	598	1	2
158	day for 1 week then reduce to 40 mg for 4 days then reduce to 20 mg for 4 days then reduce to 10 mg	10.0	704	1	2
161	take 1 tablet 1 mg total by mouth daily please take 16 mg total per day	1.0	708	0	2
171	2 16 2 22 then 55 mg 2 23 2 29 then 50 mg 3 1 3 7 then 45 mg 3 8 3 14	10.0	707	1	2
177	take 5 mg by mouth daily taking 15 20mg daily	5.0	606	3	2
188	30mg d x 1 week 20mg d x 1 week 15mg d x 1 week then 10mg d	10.0	686	1	2

Columns from left to right: ‘signature’, ‘dosage’ , ‘sample_index’, ‘predicted class label’, ‘true class label’.

9.5. Discussion and Future Directions

This is a difficult problem. Even as person with a background in clinical research, it is often impossible to tell what the dosage is, just looking at the sigs themselves. Examples of this can be seen in the Sensitivity Analysis table above. Often times context that is known to a clinical chart reviewer but is not present in the sig or dosage information itself is necessary to determine the cumulative dosage. Additionally, the writing of the Sigs sometimes appears to be a language puzzle of sorts, for example “take 1 tablet 1 mg total by mouth daily please take 16 mg total per day”. Was the answer 1mg total by mouth daily or 16mg?

In fact, not all of the errors were caused by the machine, throughout the development process we consistently found examples that the machine had labeled correctly but had been mislabeled by the expert human annotators. Additionally, there are pure data errors. We have encountered multiple sigs which clearly contain two different sigs that have been concatenated, probably as a result of copy/pasting by the physician or a bug within the Epic EHR system.

Despite all of these challenges, and the relatively short amount of time that has been dedicated to the project so far, the initial results are quite promising. Result of the top-performing approach are already hovering in the range of results offered by a commercial software company to UCSF at 6-figure licensing cost.

The target accuracy in order to reach research utility is approximately 85%. To that end, next steps are fairly straight forward. First, Python scripts to identify and rectify double sigs will be implemented. I am also in the process of designing some ‘helper’ functions to crawl through the Sigs, in an attempt to provide the context known to experts but not present in the data itself. This helper context will be passed in as a separate input to the network. Finally, I will re-do Experiment Two, using all 50k glucocorticoid sigs instead of just the 800 that were part of the initial dataset.

The task is difficult and unlikely to ever reach above 90% accuracy due to the inherent noise, errors, and ambiguity in the data itself. However, tasks such as this highlight the power of current deep networks in the NLP space. In the space of a few hours of development time I was able to achieve current state-of-the-art results that had been developed by teams of people over a substantial period of time using older techniques.

Chapter 10

Medical Research Topic Labeling

10.1. Introduction

The NIH and other agencies are funding high-throughput genomics (‘omics) experiments that deposit digital samples of data into the public domain at breakneck speeds. This high-quality data measures the ‘omics of diseases, drugs, cell lines, model organisms, etc. across the complete gamut of experimental factors and conditions. The importance of these digital samples of data is further illustrated in linked peer-reviewed publications that demonstrate its scientific value. However, meta-data for digital samples is recorded as free text without biocuration necessary for in-depth downstream scientific inquiry. Deep learning is revolutionary machine intelligence paradigm that allows for an algorithm to program itself thereby removing the need to explicitly specify rules or logic. Whereas physicians / scientists once needed to first understand a problem to program computers to solve it, deep learning algorithms optimally tune themselves to solve problems. Given enough example data to train on, deep learning machine intelligence outperform humans on a variety of tasks. Today, deep learning is state-of-the-art performance for image classification, and, most importantly for this proposal, for natural language processing.

This proposal is about engineering Crowd Assisted Deep Learning(CrADLe) machine intelligence to rapidly scale the digital curation of public digital samples. We will first use our NIHBD2K-funded Search Tag Analyze Resource for Gene Expression Omnibus (STARGEO.org)to crowd-source human annotation of open digital samples. We will then develop and train deep learning algorithms for STARGEO digital curation based on learning the associated free text meta-data each digital sample. Given the ongoing deluge of biomedical data in the public domain, CrADLe may perhaps be the only way to scale the digital curation towards a precision medicine ideal. Finally, we will demonstrate the biological utility to leverage CrADLe for digital curation with two large-scale and independent molecular datasets in: 1) The Cancer Genome Atlas(TCGA), and 2) The Accelerating Medicines Partnership-Alzheimer's Disease (AMP-AD). We posit that CrADLe digital curation of open samples will augment these two distinct disease projects with a host big data to fuel the discovery of potential biomarker and gene targets. Therefore, successful funding and completion of this work may greatly reduce the burden of disease on patients by enhancing the efficiency and effectiveness of digital curation for biomedical big data.

10.2. Methods

10.2.1. Goal

We sought to develop two independent deep learning models with large-scale human curation of GEO, and we would combine that intelligence to facilitate most accurate automated digital curation.

10.2.2. Data

Currently, STARGEO catalogues 1,122,750 digital samples drawn from 31,379 experiments that can be curated. Therefore, to curate all the experiments with human curation at about 3 minutes (2.17 minutes for primary curation + 48 seconds or secondary curation) an experiment means about 20 validated experiments can be produced an hour between a primary and secondary curator. This proposal to develop deep learning models of human curation is justified because it will take about 65 days to curate all of STARGEO and validate it among two curators, but only 17 days if accurate machine digital curation can make an accurate primary annotation, given an on-demand Upwork crowd can provide rapid biocuration validation. Given increased training data generally improves DL classification models. our CrADLe approach to annotate STARGEO with human and machine intelligence is not only feasible, but likely a cheaper, faster, more accurate and most importantly a much more scalable than other annotation efforts.

Given large enough datasets to train on, DL has proved superior for learning patterns in medical imaging for prognosis of Alzheimer’s disease and mild cognitive impairment, organ segmentations and detection, ultrasound interpretation, etc. For computer vision applications, convolutional neural network (CNN) models have skyrocketed to state-of-the-art performances as open architectures such as Google’s Inception v3, Inception v4, and the Google/Microsoft hybrid Inception-Resnet accuracyrates greater than 97% that outperform humans for image recognition. Most recently, Google developed and validated a best-in-class DL model of diabetic retinopathy in retinal fundus photographs had 90.3% sensitivity and 98.5% specificity for detecting referable disease. Using 128 175 retinal images, which were graded 3 to 7 times for diabetic retinopathy, diabetic macular edema, and image gradability by a crowd of 54 US licensed ophthalmologists, google created a best-in-class labeled imageset to achieve state-of-the-art performance in diagnosing diabetic retinopathy. Most applicable to this project to deep learn patterns of free text, recursive Neural Networks (RNNs) for sentiment analysis of free text pushes the state of the art in single sentence positive/negative classification from 80% up to 85.4%. RNNs are the only model that can accurately capture the effects of negation and its scope for both positive and negative phrases. Given that state-of-the-art sentiment analysis of free text is not precise with any method, combining state of the art deep learning of NLP with a large-scale on demand crowd validation as we propose here to ensure precision, is both justified and warranted to effect intelligent machine curation of digital samples.

10.2.3. Models

For preliminary analysis, we built two different deep learning frameworks, one GeneDL model based on a CNN of gene expression of 26 genes, and the other Words DL model based on free text Recurrent Long Short Term Memory (LSTM) framework state-of-the-art sentiment analysis of free text. Our training set was based on STARGEO crowd-annotated samples of 1903 breast cancer vs 1045 breast tissue controls across 41 experiments run on 26 different platforms.

10.3. Results

The original LTSM achieved 0.96 area under the ROC performance, by using word embeddings as the initial layer in the prediction model to explicitly learn representations in relation to the specific prediction task thereby utilizing every sample in the training corpus to it's maximum potential and further improving accuracy of predictions. Furthermore, a modification of the LSTM model that included CNN features performed best with complete perormance of out of sample AUC. In stark contrast to the LTSM architectures for words, Gene model achieved only AUC of 0.81under the ROC performance. This is because of a paucity of the input feature space of genes given the diversity of platforms and gene configurations measured in the public data as only 26 genes were measures across ALL digital samples across every platform. Nonetheless, we selected a batch normalized, fully connected, feed-forward framework with dropout for predicting labels from gene expression data to take complete advantage of potential effect of gene-gene interactions while forcing the network to learn patterns instead of memorizing them. Rectified Linear Units used for activation to prevent neural saturation and a vanishing gradient. During our initial research the top performing model reached 11 layers arranged into triple repeated blocks of full connectivity, normalization, activation, then dropout.

10.4. Conclusions and Future Directions

Deep learning approaches to medical data curation are extremely promising, already equaling human performance but at a fraction of the time and cost. Increasing AUC from current measures of approximately 0.95 up to >0.999 could potentially be achieved by ensembling predictors.

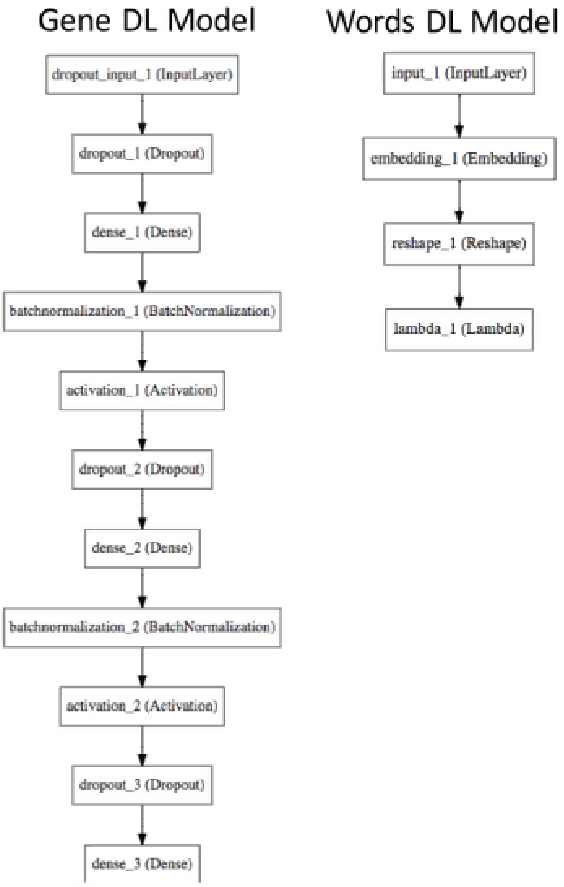


Figure 10.1: Simple DL models to compare DL techniques based on STARGEO crowed-annotated digital samples of 1903 breast cancer vs 1045 breast tissue controls across 41 experiments run on 26 different platforms. Gene DL curation model (left) is trained from data on 26 genes that comprises 11 layers utilizing dropout and maxnorm for regularization, batchnormalization, Rectified Linear Units (ReLU) for activation. The simpler Words DL curation model uses recurrence of word embeddings to classify samples based on free text of arbitrary lengths.

Figure 10.1: Topic Labeling Architectures

References

1. Garay JP, Gray JW. Omics and therapy–a basis for precision medicine. *Molecular oncology*. 2012;6(2):128-139.

2. Chen R, Snyder M. Promise of personalized omics to precision medicine. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine.* 2013;5(1):73-82.

3. Highnam G, Mittelman D. Personal genomes and precision medicine. *BioMed Central.* 2012;13:324.

4. Khoury MJ, Gwinn ML, Glasgow RE, Kramer BS. A population approach to precision medicine. *American journal of preventive medicine.* 2012;42(6):639-645.

5. Mirnezami R, Nicholson J, Darzi A. Preparing for precision medicine. *New England Journal of Medicine.* 2012;366(6):489-491.

Chapter 11

Conclusion

11.1. The Past: A Summary

Where my work has diverged from that of other founders in this field, it has been primarily on the types of problems that interest us. Many have made significant contributions to the automation of tasks that humans already have some competency in, such as radiological and pathological diagnostics, and differential diagnosis. Here they have sought to make processes that already work, work even better, reducing error rates and the burden of repetitive tasks. My own work has been driven by the motivation to create methods that might enable doctors and healthcare systems to systematically perform tasks that are not possible today. I find myself most excited by tasks primarily related to predicting, and directing, the course of future.

Looking back at the checklist of requirements from Chapter 1 that I argued were necessary to establish a new field of healthcare that utilizes data on previous actions and outcomes to enable smarter ongoing choices; the groundwork that has been laid in this short time is actually quite impressive. Approaches for modeling individual patients have been identified and validated on the clinically meaningful task of forecasting disease activity for individuals with Rheumatoid Arthritis; a task for which no current clinical standard previously existed. Furthermore, I have shown that these models can be trained in one hospital system and still function effectively on dramatically patients in a different hospital system, providing evidence that the models were able to learn something robust and transferable about individual disease trajectories. I have provided simple and intuitive methods to dissect what any model has learned about the relationship between clinical input variables and outcomes, as well as explanations for model predictions on a case by case basis. I have shown how the method can be extended to identify patients that are similar on the basis of how input variables, such as treatments, will affect their outcomes, providing a method for data-driven individualized treatment selection. Philter, will enable the large-scale automated learning from clinical notes, which have previously been inaccessible to researchers due to the sensitivity of the information within them, unlocking the insights of the physicians that generate them. Perhaps most importantly, I have established a standard for designing, recording and reporting AI-based clinical informatics studies, which will in turn facilitate transparency and the establishment of trust and ultimately enable the utilization of such models in the clinical setting.

11.2. The Future: A Roadmap

The field, of course, is still in its infancy. The exciting proof-of-concepts that I have described above are exactly that: proofs-of-concept. The application of modern AI to health data is evolving so rapidly that my own research has made my manuscripts somewhat outdated before they were even published. This can result in a temptation to constantly strive for improvement of methods in the research setting at the expense of translating and testing reasonable approaches in real-world clinical settings. From my perspective, the next step is obvious. We must demonstrate whether physicians armed with insights gleaned from AI trained on relevant patients can provide better care for their own patients than physicians acting alone. We need not expect dramatic improvements initially, incremental improvements, like interest in a bank account, when compounded daily eventually yields remarkable changes.

My primary hypothesis, one that is shared by every statistician, informatician, and data scientist in the world, is that the more completely we sample from the true population of individuals to build our models, the more accurate those models will become. Said more simply, humans have a lot of individual variation in every aspect, health and disease are no exception. A single physician, no matter how experienced or gifted is only able to see a small portion of that variation in their career, therefore the model that their mind builds of the disease and response to treatment is inherently quite incomplete. AI models built using the intersections of thousands of doctors and millions of patients are likely to capture much more of the true variation and will therefore be more robust and reliable. It follows then that the best model would be one trained on all of the patients in the world. While philosophically correct, this line of thinking has recently lead to demands among researchers for the open sharing of patient data across all institutions and strong criticisms of the policies currently which prevent it. Many of those fighting for the public sharing of health data are informaticians who are quite divorced from the intimate nature of this data. Many people would rather share their financial information than the health information of themselves and their families; and no one is arguing for the open sharing of financial data. In short; of course we must facilitate the safe sharing of data but first we must earn trust through patient engagement over the use of their data, and rock-solid protocols for how data can be accessed, for what reasons, and by whom.

I’d like to part with a final thought about forests and trees. It is so easy to get swept up in the excitement about algorithms crunching big data to determine optimal treatments at an individual level (trees) that it’s tragically easy to forget that disease is much easier to prevent than to cure (forest). According the CDC, 90% of American healthcare spending is on chronic diseases, 70% of which are completely preventable with healthy lifestyle practices. In fact, the prevention of nearly all health conditions currently effecting the first-world can summarized so simply that it is almost laughable: don’t smoke, eat a balanced diet of natural foods, exercise vigorously in safe age-appropriate manner for an hour a day 5 days a week, sleep 7-9 hours a night, and don’t sweat the small stuff. The reason that these simple practices are not followed by all are complex. The poor do not have access to natural foods or education about the true risks of the ‘typical American lifestyle’. Curing lung cancer, or developing the perfect diabetes drug, is just sexier than convincing people not to smoke or to eat right and exercise more; attention and investment dollars are allocated accordingly. Finally, humans, like nearly all other species, evolved in environments of scarcity; we’re naturally gluttonous when possible. Those interested in applying AI for the maximal possible impact on the future health of the country would be remiss to ignore the forest for the trees. In addition to curing cancer and finding the optimal drug, AI must also be used to develop methods to engage people of all backgrounds about their lifestyles, empower them to make healthy changes, and provide better access to healthier nutrition.

Library Release

Publishing Agreement

It is the policy of the University to encourage the distribution of all theses, dissertations, and manuscripts. Copies of all UCSF theses, dissertations, and manuscripts will be routed to the library via the Graduate Division. The library will make all theses, dissertations, and manuscripts accessible to the public and will preserve these to the best of their abilities, in perpetuity.

I hereby grant permission to the Graduate Division of the University of California, San Francisco to release copies of my thesis, dissertation, or manuscript to the Campus Library to provide access and preservation, in whole or in part, in perpetuity.

Author Signature _ ___ Date _____5/7/2019