Interpreting Black-Box Models
using Hierarchical and Temporal Feature Abstractions

by

Akshay Sood

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

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2021

Date of final oral examination: 08/12/2021

The dissertation is approved by the following members of the Final Oral Committee:
Mark Craven, Professor, Biostatistics and Medical Informatics
Aws Albarghouthi, Associate Professor, Computer Sciences
David Page, Professor, Biostatistics and Bioinformatics
Michael Gleicher, Professor, Computer Sciences
Vikas Singh, Professor, Biostatistics and Medical Informatics

In memory of my mother, Dr. Rita Sood (*1956 - 2021*)

Acknowledgments

I am grateful to many people for their guidance, encouragement and support during the course of this PhD.

First, I would like to thank my advisor, Mark Craven. Mark's integrity, professionalism, and dedication to his work and his students are among his many excellent qualities as a researcher and an educator. Mark has always been willing to commit time to listen to my ideas and concerns, to provide guidance, and to correct my mistakes with patience and encouragement. I have learned and benefited from his clarity of thought and expression, his attention to detail, and his emphasis on clear and precise writing. Finally, I could not have asked for a more understanding mentor during the years of my mother's illness.

I thank the members of my thesis committee. I have had many enjoyable interactions with Vikas Singh during my PhD, and I am deeply grateful for his support during my mother's illness. I have had several useful discussions with David Page and his students about machine learning projects using electronic health records. I appreciate his encouragement of and interest in my work. I would also like to thank Michael Gleicher and Aws Albarghouthi, with whom I have had insightful discussions about interpretability in machine learning, and whose incisive comments have helped me think more clearly about my work.

I would like to thank my collaborators. Larry Hanrahan and Theresa Guilbert provided data and clinical guidance on the asthma exacerbations project. I have had enjoyable collaborations with Beth Burnside and Matthew Churpek. I appreciated the time working with and mentoring Madeline Abbott. I am especially grateful to Alex Cobian for guiding me during the early years of my PhD, Kyubin Lee

for his companionship during his final year, and Yuriy Sverchkov for our many collaborations and discussions during my time at UW-Madison. I have enjoyed regular meetings with them and other members of the Craven Group over the years: Nathan Bollig, Gary Pack, Sid Kiblawi, Saswati De, among others.

I am grateful for the support of the Departments of Computer Sciences and Biostatistics and Medical Informatics. I thank Angela Thorp, Shelley Maxted, and Beth Bierman for their tireless efforts on behalf of students. I would like to acknowledge the financial support of the Center for Predictive Computational Phenotyping, the UW Institute for Clinical and Translational Research, and NIH grants U54 AI117924 and UL1 TR000427.

I thank various colleagues and friends who helped make grad school an enjoyable experience for me: Swapnil Haria and Sukriti Singh for many adventures around Madison, Akhil Guliani for board games and food comas, Arpit Agarwal for sailing fails, Jia-shen Boon for many collaborations and good times, Yash Govind, Neha Godwal, Ritika Oswal, Amrita Roy Chowdhury, Samartha Patel, Jinman Zhao, Satyam Dhar, Matt Gawlik, Ellen Greta, Ronak Mehta, Sathya Ravi, Vamsi Ithapu and many others.

Finally, I thank my family for their boundless support and love. My partner Ginny, who has been my dearest companion through the best and worst of times; my brother Varun, who has always been an inspiration; my father Arun, whose stoicism belies the depth of his caring; and my mother Rita, who was the best mother and role model I could have asked for. She passed away before she could see me graduate, but I know that she would have been proud.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The emergence of increasingly complex models and large, rich feature representations in machine learning promises new opportunities by enabling more accurate modeling of phenomena of interest, but also presents substantial challenges. The very complexity that often contributes to the accuracy of such models makes them 'black-boxes' that are hard for humans to interpret. In many application domains, such as healthcare, interpretability is a key requirement for the deployment and acceptance of such models.

In this dissertation, we advance the state of the art for interpretability in machine learning by proposing novel approaches that leverage abstractions over the features to better interpret black-box models. We develop these approaches through their application to black-box models that we have trained to address specific tasks of interest.

In particular, we consider the task of modeling asthma exacerbations, a prevalent acute respiratory condition, using electronic health records (EHRs). We develop an algorithm to phenotype asthma exacerbations from EHRs. Using the phenotyped exacerbations, we explore a variety of representations and modeling approaches for the task of predicting future exacerbations, and develop an approach to identifying subpopulations of asthma patients sharing distinct temporal and seasonal exacerbation patterns.

We develop methods that use hierarchical and temporal feature abstractions to interpret black-box models while meeting key interpretability desiderata. We use a model-agnostic, permutation-based approach that characterizes models across the distribution of instances. By leveraging feature hierarchies, we interpret models at multiple resolutions in terms of their important features, feature groups, and interactions. For models over temporal or sequential representations, we develop an approach that identifies the importance of salient features with respect to their temporal ordering as well as localized windows of influence. Our approach is statistically grounded using a hypothesis testing and false discovery rate control methodology. We apply these methods to learned models in challenging biomedical

domains, including a model that we have trained to predict asthma exacerbations.

Additionally, we provide software packages that include efficient, distributed implementations of our methods, and tools to readily visualize the explanations generated by them.

1 INTRODUCTION

1.1 The Need for Model Interpretability

Advances in machine learning, supported by advances in computing, reflect a trend towards increasing model complexity. This has led to a proliferation of models that learn rich representations over large, complex parameter spaces, most readily evident in the advent of deep learning. Such models have increasingly been applied in domains with a high degree of social impact, such as healthcare, but this very complexity makes them black-boxes whose decision-making is hard to explain, a critical deficit in many such domains. There are several principal reasons why it might be important to interpret or explain the decision-making of black-box models:

- **Trust**: Trust in the accuracy of a model's predictions and its underlying rationales is a key requirement for its deployment and acceptance in many domains, for end users as well as other stakeholders.

- **Legal and ethical imperatives**: A 'right to an explanation' to be given to individuals affected by algorithmic decisions may be ethically and legally required, such as by laws like the Equal Credit Opportunity Act (ECOA) or the General Data Protection Regulation (GDPR).

- **Scientific discovery**: Interpreting black-box models may enable new insights into a problem domain by uncovering previously unrecognized salient features and associations that the models have learned, and generating hypotheses that may be tested to establish causal relationships.

- **Model development**: Explanations may aid in improving the predictive performance of models by detecting and avoiding overfitting, diagnosing and removing bias and discrimination in models' decisions, gaining insight into differences among input representations, and identifying weaknesses in changing or adversarial environments.

Complex models may be hard for humans to interpret for several reasons: (i) they may capture relationships between the features and the outputs that are highly non-monotonic and/or non-linear, (ii) their outputs may depend on decisions made by large numbers of computational units, as in the case of large decision trees, neural networks, or ensemble methods, and (iii) they may be trained over complex data consisting of large, structured feature spaces, such as in visual, genetic or clinical domains.

Methods that interpret models seek to unravel this complexity in various ways in order to make it easier for humans to understand models' decisions. In the context of supervised learning, these approaches may be broadly classified as (i) methods that design models to be *transparent*, i.e., inherently more interpretable than complex models by some measure, and (ii) methods that generate *post-hoc* explanations, i.e., explanations of the behavior of learned black-box models.

1.2 Methods for Interpreting Black-box Models

The increasing complexity and social impact of algorithms have led to a concomitant rise in post-hoc methods to interpret black-box models. There are two chief ways in which a model may be considered a black-box: (i) from a mechanistic perspective, where the internal workings of the model may be partially or completely hidden from the user, so that the user may only have access to the model's output for a given input, and (ii) from the perspective of human understanding, where, despite complete information about the model, such as its architecture, parameters and computations, the model's complexity may make it hard to interpret. *Model-specific* explanation methods rely on knowledge of the internal workings of the model and may only be used to interpret the latter kind of black-box models. *Model-agnostic* methods make few assumptions about the model and may be used to interpret both kinds of black-box models.

Figures 1.1 and 1.2 show two other broad characterizations of explanation methods. Explanations are commonly generated using one of two methodologies: (i) *model translation*, where an *explanatory model* is trained to approximate the predic-

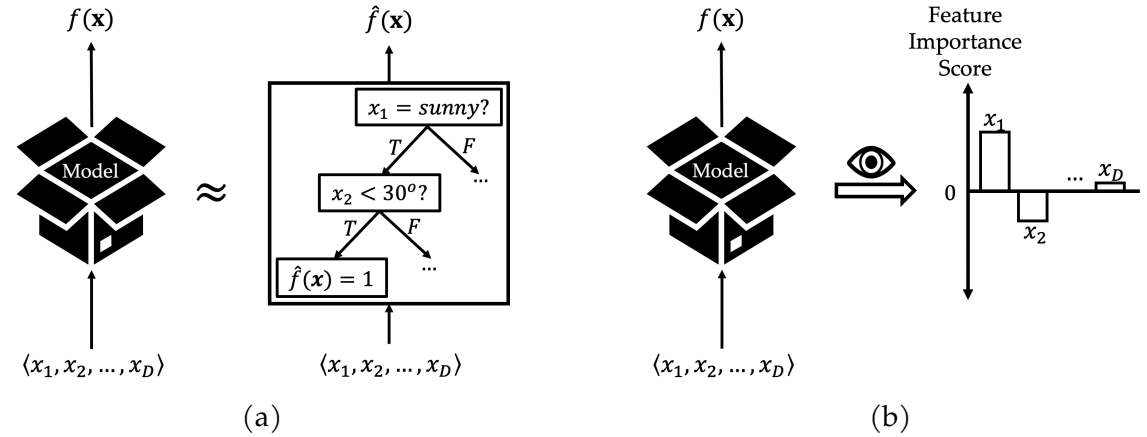


Figure 1.1: Illustration of methodologies used to explain a black-box model $f(\mathbf{x})$, showing (a) model translation, where the black-box is approximated by an explanatory model represented by an interpretable decision tree, and (b) model inspection, where the black-box is examined directly in order to identify importance scores for the input features.

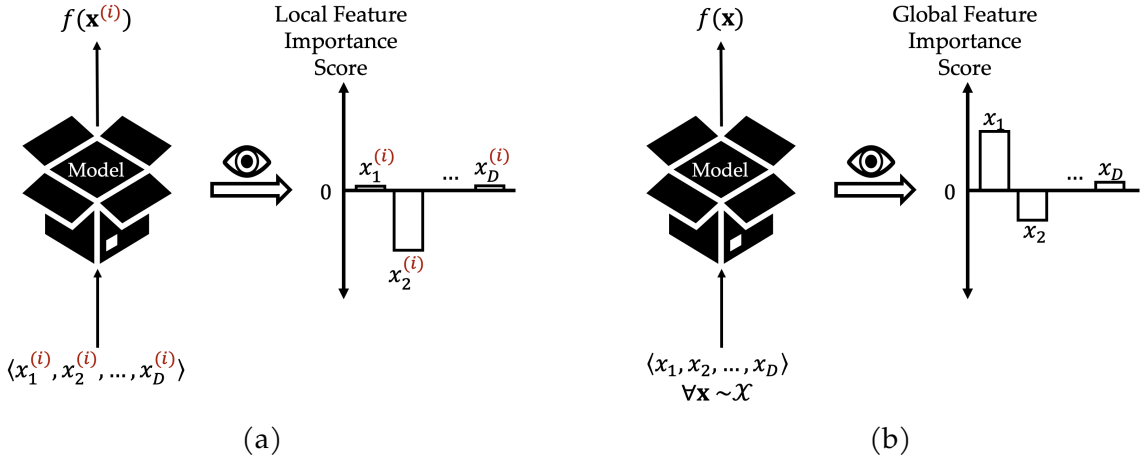


Figure 1.2: Illustration of locality of explanations for a black-box model $f(\mathbf{x})$, showing (a) local explanations, identifying feature importance scores for the model's prediction for a given instance i , and (b) global explanations, identifying feature importance scores characterizing the model across the distribution of instances $\mathcal{X}$.

tions of the black-box model, while being easier to interpret (Figure 1.1a), and (ii) *model inspection*, where the black-box model is examined directly in order to identify its properties, such as feature importance scores that quantify the influence of each feature on the model's behavior (Figure 1.1b). The *locality* of an explanation describes whether it (i) explains the model's predictions for specific instances using *local explanations* (Figure 1.2a), or (ii) characterizes the model over the entire distribution of instances using *global explanations* (Figure 1.2b).

Explanations or interpretations yielded by different methods vary widely in their forms, use cases, limitations, and computational costs. Common forms of explanations include (i) visualizations of model behavior, such as saliency maps, (ii) decision sets, rules, or trees that translate the model's decision-making into small, easily comprehensible descriptions, and (iii) importance scores summarizing the effect of different features on the model's predictions.

The degree to which an explanation is comprehensible and aids model interpretability may be influenced by a number of factors, including the nature of the black-box model as well as the form of the explanation. For instance, for linear models, importance scores can precisely capture the magnitude and direction of the causal influence of each feature on the model's predictions, while no such relationship necessarily holds for non-linear models. Explanations having lower *syntactic complexity* by virtue of smaller, more concise descriptions are also more *simulatable*, making it easier for humans to mentally represent the model and simulate its behavior. In the presence of large, complex feature spaces, explanations may be more comprehensible when expressed at the right level of granularity using a description language leveraging feature groups and other abstractions.

While there is general consensus on the importance of interpretability in machine learning, the subject of what constitutes a 'good' explanation and how it may be measured continues to be actively discussed (Barocas et al. 2020; Doshi-Velez and Kim 2017; Miller 2018). It is widely recognized that approaches must be cognizant of how the model is used and interpreted, and that they should include the needs of the stakeholders in this process (Gleicher 2016; Kumar et al. 2020; Preece et al. 2018). Although interpretability eludes clear definitions and solutions outside of

specific settings, it is nevertheless possible to identify key desiderata that are widely applicable across explanations and explanation methods:

- **Comprehensibility**: The extent to which the explanation aids human understanding. Comprehensibility is the main goal of model interpretability, yet it is hard to formally define and objectively measure.
- **Fidelity**: The extent to which the explanation accurately captures the reasoning of the black-box model. In the case of model translation, this may be measured based on how closely the predictions of the explanatory model match those of the black-box model.
- **Accuracy**: In the case of model translation, the performance of the explanatory model on test data. Together with fidelity, it reflects the ability of the explanatory model to serve as a substitute for the black-box model.
- **Stability**: The stability of a local explanation, capturing the intent that small perturbations of the input should have small effects on the explanation. Lack of stability may make the explanation vulnerable to adversarial manipulations and erode trust.
- **Model Agnosticism**: The ability of the explanation method to function with little or no knowledge of the internal workings of the black-box model, allowing more general and retrospective applicability and avoiding obsolescence.

1.3 Our Approach

In this work, we attempt to address shortcomings in existing work on model interpretability while satisfying key interpretability desiderata. We propose a model-agnostic method that leverages feature hierarchies to provide global explanations of learned models in terms of their important feature groups in addition to important *base features*, i.e., features that are input to the model, facilitating the explanation of models at multiple resolutions. We also propose Temporal Importance Model Explanation (TIME), a model-agnostic, global explanation method that advances

the state of the art in the explanation of models over temporal or sequential representations.

Whereas most work on model interpretability has focused on local explanations, we focus on global explanations because they are important for clinical and many scientific domains. In clinical domains, it is important to provide an overall description of what a model does before it is deployed, not just be able to explain individual predictions after deployment. Moreover, global explanations offer the possibility of identifying previously unrecognized risk or protective factors, and important windows of exposure for a given condition. While local explanations may be used to justify specific decisions, global explanations are often advantageous for model diagnostics, feature engineering, bias detection, trust, and discovery (Doshi-Velez and Kim 2017; Ibrahim et al. 2019).

Our approach interprets black-box models using explanation vocabularies based on hierarchical and temporal abstractions over the features. Leveraging these abstractions can provide more comprehensible descriptions of the model than using base features (in case of a tabular representation) or timesteps (in case of a temporal representation) by providing more concise explanations as well as by interpreting models at multiple resolutions. In particular, they can enable better explanations of models over large feature spaces, where base features may be numerous and may have low individual significance, making it harder to detect and to interpret them. Feature abstractions can also produce more faithful explanations of black-box models by grouping together correlated features and more accurately assessing their importance. Finally, they can significantly increase the computational efficiency of computing explanations by pruning the space of features and feature groups to be examined.

We consider a feature to be important if the model's performance degrades on average when the feature is perturbed via permutation. We assess feature importance by (i) examining the model loss, rather than the model output, in order to capture how the perturbation affects the accuracy of the model's predictions, and (ii) by performing hypothesis testing to test the statistical significance of this effect. We provide a direct, statistically grounded approach towards global explanations,

as opposed to generating global explanations by heuristically aggregating local explanations, an approach that is commonly used by existing methods. Thus, our approach also avoids issues resulting from a lack of stability of local explanations. We use a novel application of hierarchical false discovery rate (FDR) control to perform multiple test correction for statistical tests of base features, feature groups, and temporal properties of the features.

Permutations serve a twofold purpose in our approach: (i) to compute importance scores for features, and (ii) (in case of TIME) to test the significance of features using permutation tests, a widely-used, non-parametric statistical significance test. While several methods have employed permutation-based feature importance scores, and some methods have used hypothesis testing based on permutation tests to examine feature importance, combining the two approaches is a novel aspect of our work. Moreover, the generality of permutations allows our approach to be model-agnostic.

Our interest in interpreting black-box models is motivated in part by applications of machine learning in biomedical domains. In particular, we are interested in risk prediction models using electronic health record (EHR) data. We focus on modeling asthma exacerbations, a prevalent acute respiratory condition, using EHRs. We develop an approach to phenotyping asthma exacerbations from EHRs and explore a variety of feature representations and models for the task of predicting future exacerbations. In order to examine differences between modeling approaches and to identify potential risk factors for exacerbations that the models may have learned, we develop an approach to interpret complex models that is well-suited to large, structured feature spaces, such as those characterized by EHR data. We use our approach to examine a long short-term memory (LSTM) model used to predict asthma exacerbations, as well as a random forest model used to identify viral genotype-to-phenotype associations. We build on this work to develop TIME, an approach to interpret black-box models over temporal or sequential representations, and use it examine an LSTM model used to predict in-hospital mortality from intensive care unit (ICU) data.

1.4 Thesis Statement and Contributions

The central thesis of this work is that by leveraging an explanation vocabulary comprising hierarchical and temporal abstractions over the features in conjunction with a permutation-based approach for feature importance and statistical testing, we can interpret learned black-box models while meeting key interpretability desiderata.

The major contributions of this dissertation are the following:

1. **Modeling asthma exacerbations from electronic health records**:

   We address the task of modeling asthma exacerbations using EHRs. The motivation for this analysis is to improve patient care by anticipating exacerbations, and to identify potentially unrecognized risk factors for exacerbations that may be indicated in EHR variables. We develop an algorithm for phenotyping asthma excarbations from EHRs and use this to identify exacerbations in our patient cohort. Using the phenotyped exacerbations, we consider the task of predicting exacerbations from a patient's clinical history as represented in their EHR. For this task, we perform a comparison over a variety of feature representations, including fixed-length as well as distributed representations, and a variety of modeling approaches, including logistic regression, random forests, and long short-term memory networks. We are able to learn models that predict exacerbations with a moderately high degree of accuracy. We also consider the task of inferring temporal exacerbation phenotypes from EHRs using a mixture of semi-Markov models. We show that our approach is able to identify subpopulations of asthma patients sharing distinct temporal and seasonal patterns in their exacerbation susceptibility.

2. **Understanding learned models by identifying important features at the right resolution**:

   We propose a model-agnostic global explanation method that leverages feature hierarchies to interpret learned black-box models in terms of their im-

portant features. Our approach (i) tests feature groups, in addition to base features, and tries to determine the level of resolution at which important features can be determined, (ii) uses hypothesis testing to rigorously assess the effect of each feature on the model's loss, (iii) employs a hierarchical approach to control the false discovery rate when testing feature groups and base features for importance, and (iv) uses hypothesis testing to identify important interactions among features and feature groups. We evaluate our approach using synthetic data where the ground-truth importance of features and feature groups is known, as well as by analyzing complex models in two challenging biomedical applications: a random forest model trained to learn viral genotype-phenotype associations, and an LSTM model for predicting asthma exacerbations.

3. **Feature importance explanations for temporal black-box models**:

   Existing methods to explain black-box models are often specific to architectures and data where the features do not have a time-varying component. We propose TIME, a method to explain models that are inherently temporal in nature. Our approach (i) uses a model-agnostic permutation-based approach to analyze global feature importance, (ii) identifies the importance of salient features with respect to their temporal ordering as well as localized windows of influence, and (iii) uses hypothesis testing to provide statistical rigor. We evaluate our approach using synthetic data where the ground-truth importance of features and their temporal properties are known, as well as by analyzing an LSTM model trained to predict in-hospital mortality from ICU data. We perform comparisons against a number of baseline explanation methods. We show that our approach significantly outperforms other methods on synthetic data and performs competitively with the best methods on real data, while generating more interpretable explanations.

1.5 Thesis Organization

This thesis is organized as follows. Chapter 2 reviews existing work related to model interpretability and predictive modeling using electronic health records. Chapter 3 describes our work on modeling asthma exacerbations from EHRs, including the use of models that may be treated as black-boxes. Then, we shift our discussion to approaches for interpreting black-box models. Chapter 4 presents a method to interpret black-box models at multiple resolutions using feature hierarchies. Chapter 5 presents TIME, a method used to interpret models over temporal or sequential representations. Finally, Chapter 6 summarizes our contributions and discusses directions for future research.

2 RELATED WORK

In this chapter, we discuss work related to different aspects of this dissertation. In Section 2.1, we review the literature on interpretability in machine learning. We present an overview of the problem and examine it from two perspectives: (i) designing models that inherently easier to interpret, and (ii) designing methods that interpret learned black-box models. We then discuss our approach to interpreting models and how it relates to existing work. In Section 2.2, we review the literature on predictive modeling using electronic health records, focusing on two tasks in particular: (i) representation learning for EHRs, and (ii) outcome prediction.

2.1 Interpretability in Machine Learning

The subject of explanation has a rich history in philosophy, psychology and cognitive science, and its study in the context of artificial intelligence (AI) can be traced back to expert systems (Miller 2018; Simon 1992). Research on explanation in machine learning has burgeoned in recent years, driven by the development and adoption of increasingly accurate and complex models. Various terms have been used to address the subject in the literature, including *interpretability* (Lipton 2016), *explainability* (Fong and Vedaldi 2017), *comprehensibility* (Gleicher 2016), *understanding* (Koh and Liang 2017), *intelligibility* (Weld and Bansal 2018), and *explainable artificial intelligence* (Gunning 2017). In this work, we interchangeably use the terms 'interpretability' and 'explainability', as well as the terms 'explanation' and 'interpretation'. In general, we use 'interpretability' when referring to the subject of study and 'explanation' when referring to its methods and outputs.

A clear formulation of what it means to be interpretable is not generally agreed upon, and most explanation methods focus on a narrow set of well-defined problems. Some authors attempt to overview interpretability in machine learning and organize its different facets. Lipton (2016) provides a taxonomy of interpretability desiderata and methods for supervised learning models, and argues for the need to

improve the problem formulation for interpretability. Gleicher (2016) expands the scope for considering interpretability beyond the modeling process, and examines the need for interpretability, the stage at which it might be important, and the stakeholders, as well as how explanations might be constructed and evaluated. Doshi-Velez and Kim (2017) posit that the need for explanations arises due to an incompleteness in the formalization of modeling tasks. They propose a taxonomy of approaches to evaluate interpretability and hypothesize a data-driven approach to discover 'factors' of interpretability. Miller (2018) reviews findings on human explanation from the social sciences and argues for the need to incorporate these ideas into approaches to interpret models. Selbst and Barocas (2018) examine the problem of interpretability in machine learning from the perspective of human intuition and its shortcomings from a sociolegal standpoint. Barredo Arrieta et al. (2020) present an overview of the field and discuss opportunities and challenges for integrating it into a more general concept of 'responsible AI'. Other overviews and surveys of the field have been conducted by Adadi and Berrada (2018), Guidotti et al. (2018), Mohseni et al. (2018), Montavon et al. (2018), and Mueller et al. (2019).

We organize the literature on model interpretability by extending taxonomies proposed by other authors (Barredo Arrieta et al. 2020; Gleicher 2016; Guidotti et al. 2018; Lipton 2016). First, we consider models that are transparent, i.e., inherently more interpretable than other, black-box models. We then discuss methods to generate post-hoc explanations of learned black-box models. We identify distinct attributes that characterize these methods and explore each attribute in greater detail. We conclude the section by comparing our work to existing approaches to interpret models.

2.1.1 Transparent Models

In the context of model interpretability, *transparency* represents an inherent property of the model that allows for comprehensible descriptions of its decision-making processes. Models may be considered transparent in various ways (Lipton 2016):

- *Simulatability*: The degree to which humans are able to mentally represent the

entire model at once, including its input data, parameters and computations, in a 'reasonable' amount of time. By this measure, models comprising smaller representations and fewer computations may be considered more transparent than other models. For instance, sparse linear models may be considered more simulatable than dense ones, and decision trees that are shallow or have few nodes may be considered more simulatable than those that are deep or have many nodes.

- *Decomposability*: The property that individual parts of the model, including its inputs, parameters, and computations, are easily interpretable. For example, decision trees with node splits that are individually interpretable, as well as generalized additive models (GAMs) (Lou et al. 2012) may be considered decomposable.

- *Algorithmic transparency*: Transparency associated with the model's learning algorithm. For instance, optimization over convex loss surfaces associated with linear model training may be considered more transparent than optimization over highly non-convex loss surfaces associated with neural network training.

Conversely, *complexity* or *opacity* signifies a lack of transparency about the model's decision-making processes. Model complexity may result from several causes, including (i) non-monotonic or non-linear relationships between the features and the outputs, (ii) decision-making based on a large numbers of computational units, and (iii) data complexity resulting from the use of large, structured feature spaces.

Complex models are often considered black-boxes due to the difficulty in readily interpreting them, despite access to their mechanisms and learned parameters. Additionally, models may be treated as black-boxes when their internal workings are withheld in order to maintain trade secrets or competitive advantages (Burrell 2016).

Figure 2.1 characterizes learned models as transparent or black-box models based on the degree to which they are inherent interpretable (by some measure).

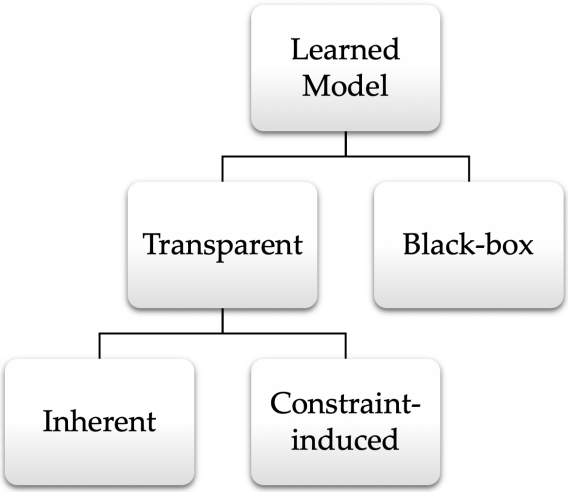


Figure 2.1: Characterization of learned models as transparent or black-box, depending on the degree to which they are inherently interpretable. Transparent models may be further categorized into models that are inherently transparent and models that are generally regarded as complex but are rendered more transparent by means of specific constraints.

Among the models that may be treated as transparent, we examine two kinds of models: (i) those that are generally considered inherently transparent, and (ii) those that are nominally complex, but have specific constraints imposed on them so as to render them interpretable in some way.

Inherently transparent models. Simpler models such as association rules, decision trees, and linear models are often associated with higher transparency and lower predictive performance compared to more complex models such as neural networks and random forests (Caruana et al. 2015). Huysmans et al. (2011) perform an empirical comparison of decision table, tree and rule-based models and conclude that decision tables are the most comprehensible to inexperienced users in their setting. Freitas (2014) reviews issues with the interpretability of different kinds of inherently transparent models and examines the drawbacks of using representation size alone to evaluate their interpretability.

Constraint-induced transparency. The imposition of constraints in the training process, such as by means of regularization, may be used to induce transparency in models that may otherwise be regarded as complex. Freitas (2014) proposes the use of semantically-derived monotonocity constraints to improve the comprehensibility of classification models. Ho et al. (2014) use sparsity and non-negativity constraints to extract interpretable phenotypes from EHRs using tensor factorization. Choi et al. (2016b) use non-negativity constraints to learn more interpretable embeddings of medical concepts. Alvarez-Melis and Jaakkola (2018b) use a regularization scheme to train locally linear neural networks, thereby generating more interpretable local explanations. Other examples of such approaches include Faruqui et al. (2015), Lee et al. (2019a), Lei et al. (2016), Plumb et al. (2020), Ross et al. (2017), Wu et al. (2017), and Yang et al. (2020).

2.1.2 Black-box Models

Transparent models have traditionally been deployed in domains where interpretability is deemed critical, such as healthcare (Rajkomar et al. 2018), but this often comes at the cost of predictive performance as compared to more complex models. The increasing adoption of high-performing complex models in such domains necessitates the development of post-hoc explanation methods that can interpret learned models.

2.1.2.1 Methods for Post-hoc Explanation of Black-box Models

Figure 2.2 identifies distinct attributes characterizing post-hoc explanation methods, including (i) methodology, i.e., whether the explanation is derived by training a transparent explanatory model to approximate the black-box model, or by inspecting the black-box directly, (ii) locality, i.e., whether the method explains the prediction for a specific instance, or characterizes the model over the entire distribution of instances, (iii) model specificity, i.e., whether the method addresses a specific model architecture, or if it is model-agnostic, (iv) form of explanation, such as visualization or feature importance, and (v) other attributes, namely, the expla-

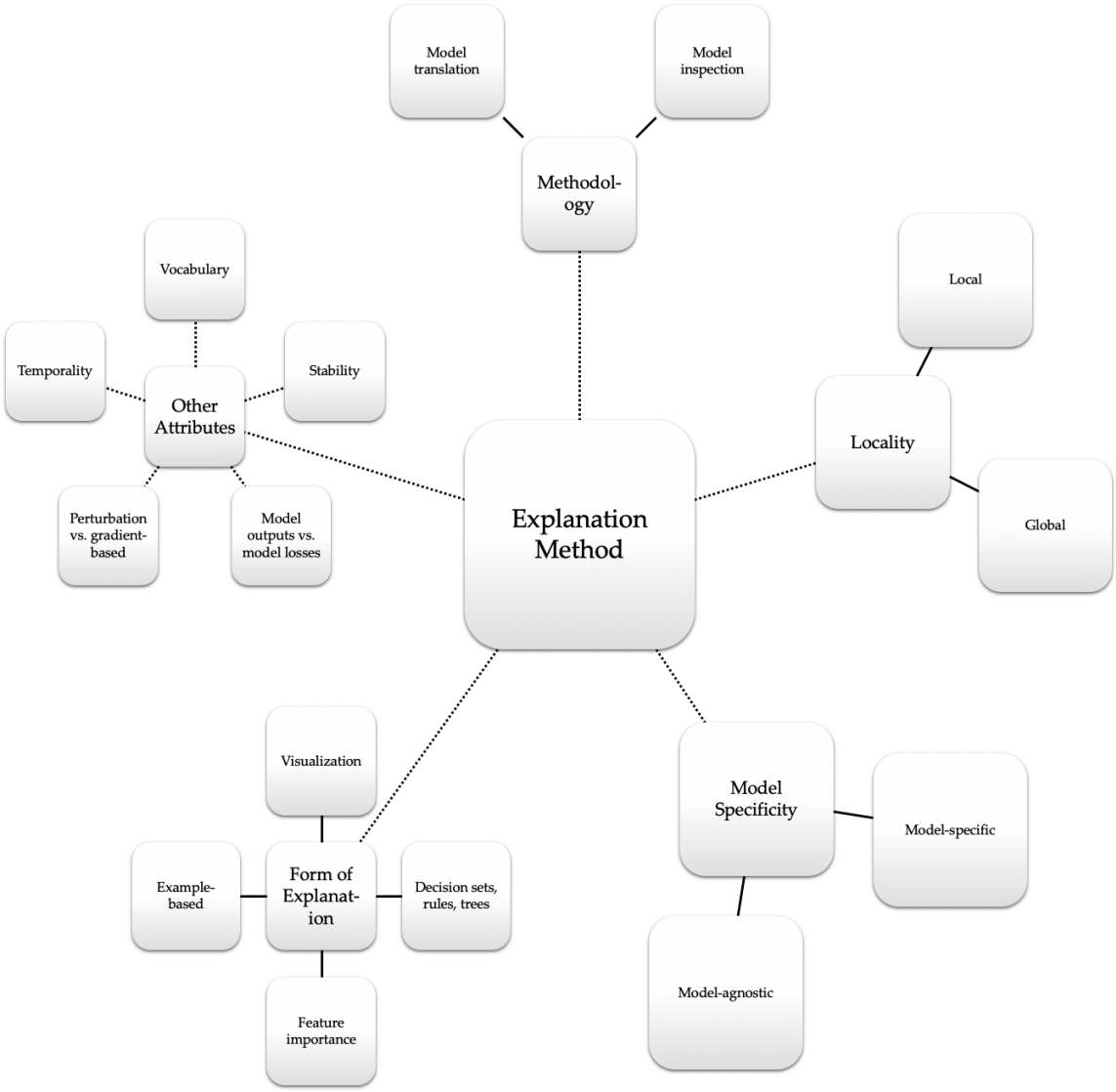


Figure 2.2: Characterization of post-hoc explanation methods for black-box models based on distinct attributes, including (i) methodology, (ii) locality, (iii) model specificity, (iv) form of explanation, and (v) other attributes. Solid lines indicate *is-a* relationships and dotted lines indicate *aspect-of* relationships.

nation vocabulary, stability, whether the model targets model outputs or losses, whether the method is perturbation or gradient-based, and whether the method addresses models with tabular or temporal representations.

Methodology. Explanation methods may be broadly classified based on whether they perform model translation, i.e, generate explanations by first training an explanatory model that approximates a black-box model, or model inspection, i.e., examine the behavior of a black-box model directly.

Model translation: Model translation trains an explanatory model that is designed to be transparent in order to approximate the predictions of a black-box model, and then uses the explanatory model to generate explanations for the black-box model. It is also referred to as *model reprojection* (Gleicher 2016) and is related to *knowledge distillation* (Hinton et al. 2015) and *mimic learning* (Ba and Caruana 2014). Many explanation methods rely on a model translation approach, including local (Lundberg and Lee 2017; Ribeiro et al. 2016), global (Craven and Shavlik 1996; Faruqui et al. 2015; Hara and Hayashi 2016), model-specific (Faruqui et al. 2015; Hara and Hayashi 2016) and model-agnostic (Craven and Shavlik 1996; Lundberg and Lee 2017; Ribeiro et al. 2016; Ribeiro et al. 2018) methods.

Model inspection: Model inspection refers to examining the behavior of a black-box model directly without first training an explanatory model to approximate the black-box model. This may be done in a variety of ways, such as by means of visualization or feature importance scores (Section 2.1.2.1). Methods that perform model translation may also perform model inspection on the explanatory model rather than the black-box model when the explanatory model is also complex, e.g., Schwab and Karlen (2019).

Locality. Explanation methods may be classified as *local* or *global* depending on whether they explain a black-box model's predictions on individual instances or its behavior across the entire distribution of instances, respectively.

Local explanations: Also referred to as *prediction interpretability* (Alvarez-Melis and Jaakkola 2017), *outcome explanations* (Guidotti et al. 2018) or *instance explanations* (Mohseni et al. 2018), local explanations seek to explain the model's predic-

tions for specific instances or local regions in the neighborhood of specific instances (Alvarez-Melis and Jaakkola 2017; Fong and Vedaldi 2017; Koh and Liang 2017; Lei et al. 2016; Lundberg and Lee 2017; Ribeiro et al. 2016; Ribeiro et al. 2018).

Global explanations: Also referred to as *model interpretability* (Alvarez-Melis and Jaakkola 2017) or *model explanations* (Guidotti et al. 2018), global explanations provide an overall description of a learned model and characterize its predictions over the entire distribution of instances (Bau et al. 2017; Craven and Shavlik 1996; Hara and Hayashi 2016; Henelius et al. 2014; Karpathy et al. 2015). Some methods are designed for local explanations but may also be used to generate global explanations by aggregating local ones (Lundberg and Lee 2017; Ribeiro et al. 2016; Ribeiro et al. 2018), while other methods are designed specifically for global explanations (Breiman 2001; Craven and Shavlik 1996; Ibrahim et al. 2019).

Model specificity. Some explanation methods are specific to certain model architectures, while others are model-agnostic and may be used with many types of models.

Model-specific: Many explanation methods focus on specific models or model classes, such as random forests (Breiman 2001; Hara and Hayashi 2016; Louppe et al. 2013), word vectors (Faruqui et al. 2015), convolutional neural networks (Mahendran and Vedaldi 2014; Simonyan et al. 2013; Zhang et al. 2018) or recurrent neural networks (Karpathy et al. 2015; Lei et al. 2016).

Model-agnostic: Some methods make few assumptions about the learned models, treating them as black-boxes (Alvarez-Melis and Jaakkola 2017; Craven and Shavlik 1996; Datta et al. 2016; Fong and Vedaldi 2017; Henelius et al. 2014; Lundberg and Lee 2017; Ribeiro et al. 2016). Model-agnostic methods do not usually require access to the internal workings of model and thus may be used to explain a wide variety of models.

Form of Explanation. Different forms of explanations may be generated by interpreting black-box models, depending on the explained model and the explanation method. We discuss some common forms here.

Visualization: Visualization techniques developed to interpret black-box models include neural interpretation diagrams (Olden and Jackson 2002), partial dependence plots (Friedman 2001; Friedman and Popescu 2008), t-SNE (Maaten and Hinton 2008), and saliency maps (Shrikumar et al. 2017; Simonyan et al. 2013; Sundararajan et al. 2017). Visualization as a means of interpreting specific model types is also common (Bau et al. 2018; Karpathy et al. 2015).

Decision sets, rules, and trees: Explanation methods for black-box models, in particular model translation methods, commonly generate explanations in the form of decision sets (Carter et al. 2018; Lakkaraju et al. 2019), rules (Craven and Shavlik 1994; Pastor and Baralis 2019; Ribeiro et al. 2018), and trees (Breiman and Shang 1996; Craven and Shavlik 1996; Frosst and Hinton 2017), due to their ability to transparency.

Feature importance: Many explanation methods return scores or rankings for important features, both for local explanations (Lundberg and Lee 2017; Ribeiro et al. 2016; Shrikumar et al. 2017; Štrumbelj and Kononenko 2014) and for global explanations (Breiman 2001; Datta et al. 2016; Fisher et al. 2019; Gregorutti et al. 2017; Zeiler and Fergus 2014).

Example-based: Some methods use salient examples as explanations, analogous to case-based reasoning in humans. These may be used for both local (Caruana et al. 1999; Jeyakumar et al. 2020) and global (Ribeiro et al. 2016) explanations.

Other attributes. Other attributes used to characterize an explanation method include the explanation vocabulary, the stability of the method, whether it targets model outputs or model losses, whether it uses perturbations or gradients to examine the model, and whether it is designed to interpret models using tabular or temporal representations.

Vocabulary: Some methods attempt to expand the explanation vocabulary beyond the base features in order to generate more comprehensible explanations. Different terms have been used to describe the components of such an expanded vocabulary, including *interpretable components* (Ribeiro et al. 2016), *cognitive chunks* (Doshi-Velez and Kim 2017), and *interpretable atoms* (Alvarez-Melis and Jaakkola 2018b).

Bau et al. (2017) measure the interpretability of internal representations by mapping hidden-variable responses to known human-labeled concepts. Fong and Vedaldi (2017) explain the predictions of image classification models in terms of most relevant image perturbations. Kim et al. (2018) explain black-box model predictions in terms of a vector of *concept activations*, where a subset of human-annotated examples embody each concept. Alvarez-Melis and Jaakkola (2018b) develop neural network models trained to predict and explain jointly, with explanations in terms of *interpretable basis concepts* learned by autoencoding the original features, while regularizing via sparsity and similarity to concept prototypes. Zhou et al. (2018) decompose neural activations of an input image into pretrained semantically interpretable components.

Stability: The stability of a local explanation method refers to the degree to which the explanation changes, given small perturbations to the instance being explained. Lack of stability may make the explanation vulnerable to adversarial manipulations, and while some explanation methods are designed to be stable (Alvarez-Melis and Jaakkola 2018b), many widely used explanation methods are not (Alvarez-Melis and Jaakkola 2018a; Dombrowski et al. 2019; Ghorbani et al. 2017).

Model outputs vs. model losses: Explanation methods typically target either the output predictions or the corresponding loss values of the black-box model. Model outputs may be used to identify all features that the model is sensitive to, including features without any predictive value, i.e., features subject to overfitting. Examples of such methods include feature occlusion (Zeiler and Fergus 2014), LIME (Ribeiro et al. 2016) and SHAP (Lundberg and Lee 2017). Model losses incorporate target labels in addition to model outputs, making it possible for explanations to identify features that are both relevant to the model and to the task being modeled, which may be potentially useful for feature selection and gaining insight into the target domain. Examples include methods developed by Breiman (2001), Covert et al. (2020b), Gregorutti et al. (2017), and Schwab and Karlen (2019).

Perturbation vs. gradient-based: Many explanation methods can be categorized as either perturbation-based, i.e., methods that perturb the value of a feature or group of features (Lundberg and Lee 2017; Ribeiro et al. 2016; Zeiler and Fergus

2014), or gradient-based, i.e., methods that use gradient information available for differentiable models (Simonyan et al. 2013; Sundararajan et al. 2017). Gradient-based methods may also be viewed as performing infinitesimal perturbations as they calculate gradients (Covert et al. 2020a). We discuss perturbation-based methods in greater detail in Section 2.1.2.2.

Temporality: Most existing explanation methods are designed for tabular, as opposed to temporal, representations. Ismail et al. (2020) demonstrate the inaccuracy of commonly used model-agnostic and gradient-based methods when used to explain temporal models. Some approaches have focused on interpreting recurrent neural networks (Ismail et al. 2019; Karpathy et al. 2015; Suresh et al. 2017) and attention-based models (Choi et al. 2016c; Zhang et al. 2019), while others have explored constraint-based transparency for temporal models using tree regularization (Wu et al. 2017) and game-theoretic characterizations (Lee et al. 2018). However, these approaches require specific model architectures, limiting their applicability. Recent work has begun to address model-agnostic explanation for temporal models. Tonekaboni et al. (2020) propose FIT, a method to assign importance scores for sequence-sequence models, and Bento et al. (2020) propose TimeSHAP, an extension of SHAP (Lundberg and Lee 2017) to temporal models, but these approaches focus on local rather than global explanations.

2.1.2.2 Perturbation-based methods

Perturbation-based methods generate explanations in terms of important features by perturbing features using operations such as occlusion, noise addition, or substitution. Such perturbations can be interpreted as a form of feature 'removal' that unifies many explanation methods (Covert et al. 2020a). We discuss some important categories of perturbation-based methods, including those that are closely related to our work.

Reference-value-based methods. Reference-value-based methods perturb features by replacing their values with reference values. These may be zero val-

ues (Schwab and Karlen 2019; Zeiler and Fergus 2014), default values (Ribeiro et al. 2016), or values sampled from a uniform distribution (Suresh et al. 2017).

Shapley-value-based methods. Shapley-value-based methods use *Shapley values*, a solution concept in cooperative game theory that satisfies a number of desirable properties, to attain feature attributions for black-box models. These may also be viewed as perturbation-based methods, where the perturbations vary according to the method variant (Covert et al. 2020a), such as marginalizing with conditional (Covert et al. 2020b; Lundberg and Lee 2017) or marginalizing with uniform (Štrumbelj and Kononenko 2014) distributions. Shapley-value-based explanations are axiomatically justified, but are computationally intractable to compute exactly and may be difficult for users to interpret (Kumar et al. 2020).

Permutation-based methods. Permutation-based methods rely on permutations of features to ascertain feature importance. Breiman (2001) first proposed using permutations to identify important features in random forests, and many variants of feature importance using permutations have since been studied (Altmann et al. 2010; Fisher et al. 2019; Gregorutti et al. 2015; Ojala and Garriga 2010; Strobl et al. 2008). The simplicity and generality of permutations makes them attractive as a tool for model-agnostic explanation.

Hypothesis Testing: Hypothesis testing may be used in conjunction with permutations to test the statistical significance of feature importance. In particular, several approaches (Burns et al. 2020; Golland et al. 2005; Ojala and Garriga 2010) rely on permutation tests (Good 2013), a type of widely used non-parametric statistical test, to identify feature importance using hypothesis testing.

2.1.3 Comparison to Our Work

In Chapter 4, we present a method to explain learned models in terms of their important base features, feature groups, and interactions by leveraging feature hierarchies. In Chapter 5, we propose TIME, a method to explain models having

temporal or sequential representations in terms of their important features and temporal properties.

Our methodology is based on model inspection rather than model translation. We use a model-agnostic, perturbation-based approach to generate global, feature importance-based explanations of learned models. Our approach identifies important features that have predictive value by targeting model losses. We use hypothesis testing and hierarchical FDR control to test the statistical significance of important features. We expand the explanation vocabulary using hierarchical and temporal feature abstractions in order to generate explanations that are more comprehensible, faithful, efficient to compute, and well-suited for models over large feature spaces. Our approach avoids issues of stability that might affect local explanations when they are aggregated to generate global explanations.

In Chapter 4, we examine perturbations using both reference-value-based methods and permutation-based methods in order to explain tabular models using feature hierarchies. We use the Wilcoxon signed-rank test for hypothesis testing. In Chapter 5, we build on this approach to generate explanations of temporal models. We restrict ourselves to perturbations based on permutations, but develop permutation-based feature importance scores further and use a hypothesis testing methodology based on permutation tests in order to make fewer assumptions about the learned models. While permutation-based feature importance scores and hypothesis testing to examine feature importance have separately been explored by existing works, combining the two approaches is a novel aspect of our work.

2.2 Predictive Modeling using Electronic Health Records

Electronic health record adoption in the US has increased dramatically in the last decade (Adler-Milstein et al. 2015; Charles et al. 2013), accompanied by a rise in research aimed at using large-scale EHR data for a variety of health analytic tasks (Goldstein et al. 2017; Hripcsak and Albers 2013; Shickel et al. 2017; Xiao et al. 2018).

EHR data do not directly represent patients' health, but rather their interactions with the healthcare system, and are rich but challenging data to work with, due to issues of data heterogeneity, completeness, accuracy, complexity and bias (Hripcsak and Albers 2013; Shickel et al. 2017). Early research on risk prediction models using EHRs has generally focused on traditional generalized linear models, using only a small and selective subset of available features, and data from a single center (Goldstein et al. 2017). Simpler and more transparent models are most commonly deployed in clinical practice (Rajkomar et al. 2018), in part due to the importance of explainable decision-making in the healthcare domain (Ghassemi et al. 2018).

Recent work in EHRs has seen a proliferation of deep learning approaches, following their success in advancing the state-of-the-art in a number of machine learning tasks (LeCun et al. 2015). Shickel et al. (2017) and Xiao et al. (2018) provide surveys of works applying deep learning methods across a broad range of EHR-related tasks. We focus on two tasks in particular: (i) representation learning and (ii) outcome prediction.

Representation learning. EHRs are typically populated with time-series of encounters that include coded data such as diagnoses, procedures and medications, in addition to free text in clinical notes. Traditional fixed-length vector representations for these codes are highly sparse and high-dimensional, making them challenging to work with (Miotto et al. 2016). Recent works have employed unsupervised deep learning approaches to learn latent vector embeddings, for both medical concepts and patients, that capture such relationships and support downstream analytic tasks. These include modified Restricted Boltzmann Machines (Tran et al. 2015), skip-gram models (Choi et al. 2015; Choi et al. 2016b; Choi et al. 2016d; Choi et al. 2016e; De Vine et al. 2014) and stacked autoencoders (Miotto et al. 2016; Suk and Shen 2013).

Outcome prediction. An important class of applications using EHRs is predicting various outcomes of interest, such as in-hospital mortality, discharge diagnoses, and disease onset within a certain time interval in the future. Several approaches have

been used to build predictive models of future events using longitudinal EHR data. The irregularly sampled nature of EHRs and variability in patient record density makes EHRs challenging to model (Xiao et al. 2018). Some of the earlier approaches use even-sized temporal windows over densely sampled ICU data for multilabel classification, using multilayer perceptrons in Che et al. (2015) and long short-term memory (LSTM) models in Lipton et al. (2015). Later methods deal directly with irregularly sampled data. Recurrent neural networks have frequently been used, following their success in modeling sequential predictions tasks in natural language processing (Sutskever et al. 2014). These include LSTMs (Esteban et al. 2016; Lipton et al. 2016; Pham et al. 2016; Rajkomar et al. 2018), and gated recurrent units (GRUs) (Che et al. 2016; Che et al. 2017; Choi et al. 2016a; Choi et al. 2015). Other approaches include neural attention models (Choi et al. 2016c; Rajkomar et al. 2018) and convolutional neural networks (Cheng et al. 2016; Nguyen et al. 2016; Razavian and Sontag 2015). While most of these approaches work with the structured data in the EHR (such as coded diagnoses, medications, procedures), some methods have also been used to process unstructured data such as clinical notes (Jagannatha and Yu 2016; Rajkomar et al. 2018).

Several approaches employ skip-gram embedding methods to learn event vector representations before passing them to a learning model (Choi et al. 2015; Choi et al. 2016b; Choi et al. 2016d; Pham et al. 2016). DeepCare (Pham et al. 2016) derives separate vectors for embedding diagnosis codes and interventions (procedure and medication codes combined). One of the models we train to predict asthma exacerbations uses Med2Vec (Choi et al. 2016b) to obtain separate vector representations for coded diagnoses, problem diagnoses and interventions, and concatenates these together along with event vectors for other temporal variables, before passing them to an LSTM.

Although deep learning methods have been used to generate increasingly accurate predictions, they are often considered opaque in their inner workings and hard to interpret (Lipton 2016). This lack of interpretability is particularly problematic in the clinical domain (Lipton et al. 2015; Miotto et al. 2016; Suk and Shen 2013), and several authors have proposed approaches to address these concerns.

Following the categories outlined by Shickel et al. (2017), these approaches include: (i) examining the types of inputs that maximize activations in the model's hidden units (Che et al. 2015; Cheng et al. 2016; Choi et al. 2016b; Nguyen et al. 2017), (ii) constrained representation learning (Choi et al. 2016b; Tran et al. 2015), (iii) t-SNE for visualization (Nguyen et al. 2017; Tran et al. 2015), (iv) model translation (Che et al. 2017), and (v) attribution via neural attention models (Choi et al. 2016c; Rajkomar et al. 2018).

3 MODELING ASTHMA EXACERBATIONS FROM ELECTRONIC HEALTH RECORDS

Asthma is a prevalent chronic respiratory condition, and acute exacerbations represent a significant fraction of the economic and health-related costs associated with asthma. In this chapter, we describe a novel study that is focused on modeling asthma exacerbations from data contained in patients' electronic health records (EHRs). This work makes the following contributions: (i) we develop an algorithm for phenotyping asthma exacerbations from EHRs, (ii) we determine that models learned via supervised learning approaches can predict asthma exacerbations in the near future (AUROC ≈ 0.77), and (iii) we develop an approach, based on mixtures of semi-Markov models, that is able to identify subpopulations of asthma patients sharing distinct temporal and seasonal patterns in their exacerbation susceptibility.

This work was performed in collaboration with Alexander Cobian, Madeline Abbott, Yuriy Sverchkov, Lawrence Hanrahan, Theresa Guilbert, and Mark Craven, and was published in the proceedings of the AMIA Joint Summits on Translational Science (Cobian et al. 2020).

3.1 Introduction

Asthma is a chronic condition that affects about 300 million people worldwide (To et al. 2012) including about 8% of the U.S. population (Winer et al. 2012). Asthma exacerbations, which frequently require acute care, can be life-threatening events and account for a significant fraction of the asthma disease burden (Dougherty and Fahy 2009). Well-characterized triggers of exacerbations in asthmatic patients include respiratory viruses, allergens, environmental pollutants, occupational exposures, and medications such as aspirin and other non-steroidal anti-inflammatory drugs (Wark and Gibson 2006). Additionally, having had a prior exacerbation is a significant risk factor for recurrent exacerbations (Dougherty and Fahy 2009).

In this study, we address two questions that are pertinent to understanding and managing exacerbations. First, we consider to what extent exacerbations can be predicted given a patient's clinical history as represented in their electronic health record. Prior studies on predicting asthma exacerbations have employed small sets of manually selected features, and have been devised and evaluated using smaller patient populations (in the context of clinical trials in some cases) (Bateman et al. 2015; Hoch et al. 2017; Loymans et al. 2018). In contrast, we are interested in determining how effectively exacerbation risk models can be learned from EHR data in a setting in which we are agnostic about which features are useful predictors. To address this question, we first devise a phenotyping algorithm for exacerbations and apply it to electronic health records for a cohort of 28,101 asthma patients. We then use supervised learning methods to train models to predict exacerbations in advance, given prior entries in a patient's EHR. The motivation for this analysis is to (i) improve patient care by anticipating exacerbations, and (ii) identify potentially unrecognized risk factors that may be indicated in EHR features.

The second question we address is to consider whether distinct temporal exacerbation phenotypes can be elicited from EHR data. The facts that patients have varying exacerbation triggers and that some patients are more prone to exacerbations indicate that there are diverse asthma phenotypes. We address the task of identifying temporal/seasonal phenotypes by clustering patients according to the temporal patterns of their exacerbations. The motivation for deriving such temporal/seasonal phenotypes is severalfold: (i) to characterize seasonal exacerbation frequency at a local scale, (ii) to be able to better detect associations between environmental factors and exacerbations by analyzing subpopulations that have similar temporal/seasonal exacerbation profiles, and (iii) to improve our exacerbation risk-assessment models by conditioning on a patient's temporal/seasonal exacerbation phenotype.

3.2 Cohort

The patient data used in this study is sourced from a clinical data warehouse for the University of Wisconsin Health system. The data we use consists of electronic health records for 28,101 asthma patients. The information we extract from the EHRs comprises demographic features and time-stamped events. The demographic features include age, sex, race, and ethnicity (Hispanic or non-Hispanic). The time-stamped events include problem list diagnoses and other diagnoses (both encoded using ICD-9), procedures (with associated CPT-4 codes), medications (with each drug represented in a three-tiered hierarchy), primary complaints and departments associated with clinical encounters, readings of six vital signs (systolic and diastolic blood pressure, temperature, pulse, respiration, and oxygen saturation, all encoded in terms of being high, low or normal), and asthma control test (ACT) scores, encoded in terms of being well-controlled (≥ 20), somewhat controlled ($16 \leq$ score ≤ 19), or poorly controlled (≤ 15).

Patients were selected for inclusion in our study if they had one or more ICD-9 codes of 493.xx (asthma) anywhere in their problem diagnosis list, or two or more such codes anywhere among other coded diagnoses. EHR data for all of these patients was available between January 1, 2007 and December 31, 2011. This study was reviewed and approved by the University of Wisconsin Health Sciences IRB as protocol M-2009-1273.

3.3 Methods

In this section, we describe approaches to three tasks that we have addressed: (i) phenotyping asthma exacerbations from EHRs, which is a necessary precursor for the other two tasks, (ii) predicting a near-term asthma exacerbation given a patient's clinical history as represented in the EHR, and (iii) identifying subpopulations of asthma patients who have similar temporal/seasonal patterns in their exacerbations.

3.3.1 Phenotyping Asthma Exacerbations

For the purpose of clinical studies, an exacerbation is typically defined in terms of an urgent visit to a health care provider for asthma symptoms coupled with a need for treatment with oral corticosteroids. Based on these criteria and accepted operational definitions (Bousquet 2000; Busse et al. 2012; Reddel et al. 2009), we implemented a phenotyping algorithm for recognizing exacerbations from events recorded in an EHR.

Our approach phenotypes an exacerbation when three components are observed in close temporal proximity: (i) a qualifying clinical encounter, (ii) a co-occurring respiratory diagnosis, and (iii) a prescription for, or administration of, oral corticosteroids. We define an exacerbation as beginning if a patient's EHR includes one of several types of clinical encounters, co-occurring with a respiratory diagnosis code recorded on the same date, and followed within seven days by a prescription of oral corticosteroids. Any further prescriptions of oral corticosteroids within five days of the last prescription are considered extensions of the same exacerbation. The full interval of the exacerbation begins with the co-occurring encounter and respiratory diagnosis and ends five days after the last oral corticosteroid prescription. This approach is illustrated in Figure 3.1.

A qualifying clinical encounter is detected by meeting one of the following conditions: (i) an encounter with type Hospital Encounter, Office Visit, Telephone, Orders Only, or of the generic type Appointment when additionally associated with a visit type of Office Visit, (ii) a recorded encounter associated with an inpatient or urgent care department, or (iii) a charge associated with a CPT code in the range 99221-99223 (initial hospital care), 99231-99233 (subsequent hospital care), 99251-99255 (inpatient consultation), or 99281-99285 (emergency department visit). The associated respiratory diagnosis codes that can indicate the start of an exacerbation are ICD-9 493.x (asthma), 46[0-6].x (acute respiratory infections), 48[0-6].x (pneumonia), 490.x (bronchitis nos), 491.x (chronic bronchitis), 519.x (other diseases of respiratory system), or 786.x (symptoms involving respiratory system and other chest symptoms).

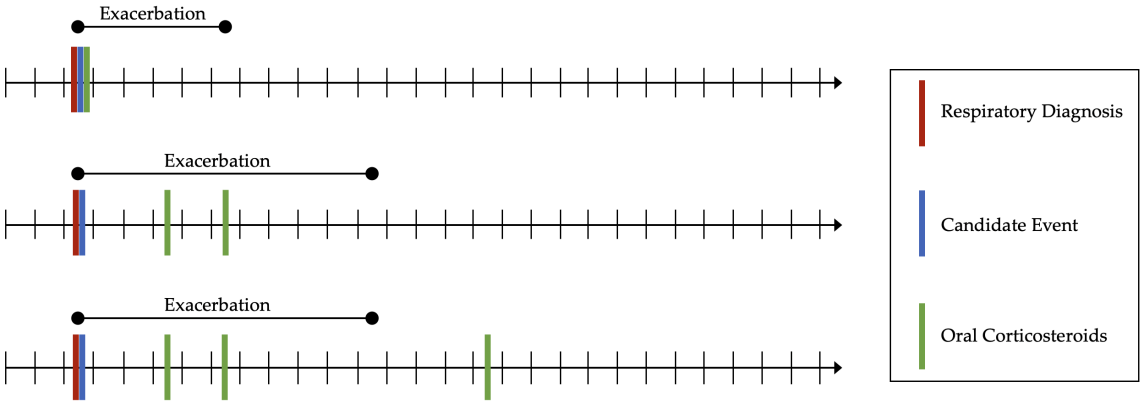


Figure 3.1: An illustration of the exacerbation phenotyping task. The figure shows three example patient timelines and the resulting exacerbation event that is recognized in each. Short, vertical black lines on the timeline represent days. Vertical red, blue and green lines represent events recorded in the EHR. The duration of a phenotyped exacerbation is represented by the extent of the corresponding horizontal black line over the timeline.

3.3.2 Predicting Asthma Exacerbations

Given our phenotyping algorithm to identify exacerbations in electronic health records, we can use supervised learning methods to train models that try to predict these exacerbations in advance. There are various ways in which this task can be framed. Here we approach the problem as a classification task: given a patient's history up to a given decision date, we want our model to predict whether the patient will experience an exacerbation within the next 90 days or not. The event window of 90 days was selected because follow-up visits for asthma tend to be 3-6 months based on guideline recommendations. In this section, we describe three approaches we use to learn classification models for this task.

We investigate a number of feature representations for this task. One approach is to represent static and time-stamped event features together using a fixed-length vector representation, comprising a summary of the event features concatenated with the static features. Alternatively, we represent event features by formulating a sequence of vectors for each patient, with each vector representing the events at a

given time-stamp. This sequence, together with a fixed-length vector representing the static features, forms a representation that preserves the patient's temporal history instead of summarizing it.

The first learning method we apply is logistic regression with L_1 and L_2 regularization (Hoerl and Kennard 1970; Tibshirani 1996). Here we use a fixed-length vector representation comprising binary features to represent the occurrence of each event feature in each of two different temporal windows: (i) over the last six months, and (ii) over the entire observation period, prior to the decision date. We perform internal cross-validation to tune the strength of the regularization.

A second learning method we apply to this task is random forests (Breiman 2001). We test two different fixed-length vector representations here: with event features summarized based on (i) their occurrence in different temporal windows (as for logistic regression), and (ii) recency. In the latter case, for each event type (e.g. for each possible diagnosis), we include a numeric feature whose value represents the number of days since the last occurrence of the event. For example, a single feature in this representation indicates how long it has been since the patient has had an ICD-9 code in the 020.xx range. For the case in which a patient has not had the event recorded within the period covered by our data set, we set the feature value to ∞. Note that for random forests, the scale of each feature is not important, and thus values of ∞ are not problematic since it is the relative ordering of feature values that matters. We tune the maximum tree depth as well as the number of sampled features per split using internal cross-validation.

A third learning method we apply is a Long Short-Term Memory (LSTM) neural network (Hochreiter and Schmidhuber 1997). In contrast to the logistic regression and the random forest, where the features summarize the patient's temporal history, the LSTM network is able to directly process the sequence of events in the history. However, some event types, namely problem diagnoses, other diagnoses, and interventions (procedures and medications), comprise large vocabularies (our cohort includes observations of 4,398 problem diagnoses, 6,533 other diagnoses, and 8,745 interventions) of which only a small subset is recorded at each encounter. Instead of working directly with the resulting sparse, high-dimensional vectors, we first

map these event types to an embedded space, resulting in dense, lower-dimensional vectors that are then used to form the event sequence for the LSTM. To learn the weights for the embedding layer, we use Med2Vec (Choi et al. 2016b), a method that obtains distributed representations of medical concepts, while capturing the context represented by the ordering of EHR visits as well as the co-occurence of codes within an EHR visit. Separate embeddings of size 200 are generated for problem diagnoses, other diagnoses, and interventions. These are then concatenated, along with the other event features, to produce the event representation at each time-stamp in the record. The ordered sequence of events forms the input sequence for the LSTM. We use an LSTM cell state of size 100 and a sigmoid output layer. The static demographic features feed directly into the output layer. For the loss function, we use binary cross-entropy with L_2 regularization. Figure 3.2 shows the LSTM network architecture.

3.3.3 Identifying Subpopulations of Asthma Patients

To address the second task of identifying subpopulations of patients who share common temporal/seasonal patterns in their exacerbations, we develop a clustering approach based on a mixture of semi-Markov models. The motivation for this approach is to identify groups of patients who have commonality in the (i) durations of their exacerbations, (ii) durations of periods in which their asthma is controlled, and (iii) seasonal dependence of their exacerbations.

As illustrated in Figure 3.3a, the data that is input to this approach consists of a state sequence for each patient along with a duration for each state. We can think of each patient as transitioning between two states, exacerbated and controlled, or perhaps remaining in the controlled state throughout the observation period. Since we cannot detect an exacerbation that starts before the observation period, we assume that all patients are in the controlled state at the beginning of their sequence. Moreover, because these sequences are both left- and right-censored (i.e., we observe the state sequence only for the period from January 1, 2007 to December 31, 2011), we divide the general controlled state into three separate states: first-

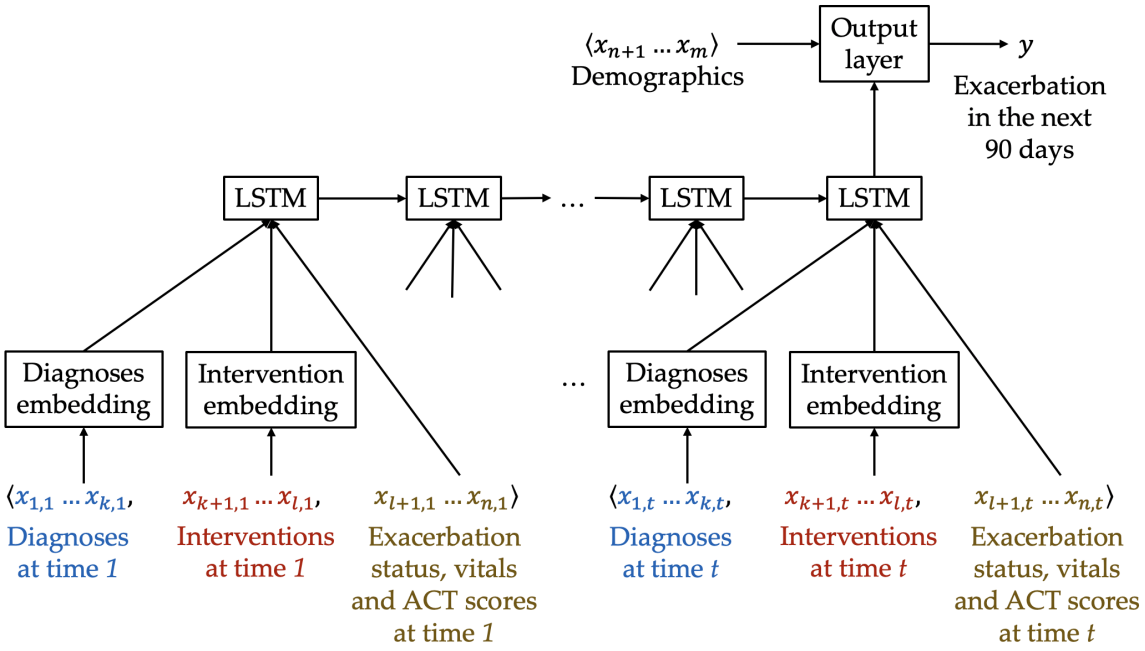


Figure 3.2: The LSTM network for predicting exacerbations. Time-stamped event features $x_1 \ldots x_n$ are represented by formulating a sequence of vectors, with each vector representing the events at a given time-stamp $x_{1,t} \ldots x_{n,t}$. Diagnoses and interventions are embedded into dense, lower-dimensional vectors. The static demographic features $x_{n+1} \ldots x_m$ feed directly into the output layer of the network.

controlled, internal-controlled, and last-controlled. Since we assume that all patients are in a controlled state at the beginning of a sequence, all sequences begin with a first-controlled state. An internal-controlled state represents the period between two exacerbated states during which a patient's asthma is controlled. Only patients who are recorded as having at least two exacerbated states during the study period have internal-controlled states. Finally, patients who have had at least one exacerbated state will also have a last-controlled state. With this partitioning of the controlled states, we can separately estimate the durations of sojourns in the internal-controlled state, thereby avoiding the bias that would be imposed in estimating controlled state durations by also including the censored durations of the first-controlled and last-controlled states.

Each state has an associated duration (with days as the units), and thus we can

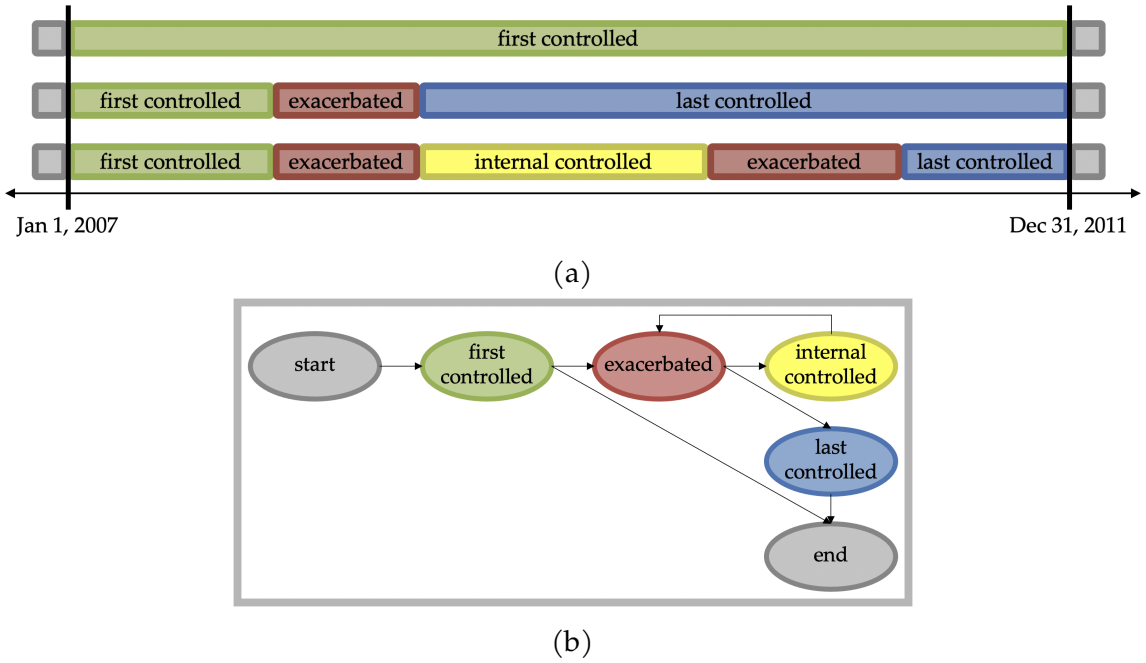


Figure 3.3: Modeling exacerbation state sequences using a semi-Markov model. (**a**) Example state sequences for three patients. (**b**) A semi-Markov model for characterizing asthma exacerbations. Nodes represent states and edges represent allowable transitions. Aside from the silent start and end states, each state has a duration distribution.

represent patient p's exacerbation history as follows:

$$\mathbf{s}^{(p)} \equiv \langle s_1^{(p)}, \ldots, s_{L_p}^{(p)} \rangle \qquad \mathbf{d}^{(p)} \equiv \langle d_1^{(p)}, \ldots, d_{L_p}^{(p)} \rangle$$

where $\mathbf{s}^{(p)}$ represents the state sequence for patient p, $\mathbf{d}^{(p)}$ represents the corresponding duration sequence, $s_i^{(p)}$ represents the ith state in patient p's history, $d_i^{(p)}$ represents the duration of this ith state, and L_p represents the length of the history in terms of the number of state visits. A semi-Markov model (Rabiner 1989)

represents the probability of a patient's history as:

$$P\left(\mathbf{s}^{(p)}, \mathbf{d}^{(p)}\right) = P\left(s_1^{(p)} \mid \mathsf{start}\right) \cdot P\left(d_1^{(p)} \mid s_1^{(p)}\right) \cdot \prod_{i=2}^{L_p} \left[P\left(s_i^{(p)} \mid s_{i-1}^{(p)}\right) \cdot P\left(d_i^{(p)} \mid s_i^{(p)}\right)\right] \cdot P\left(\mathsf{end} \mid s_{L_p}^{(p)}\right)$$

where each $P\left(s_i^{(p)} \mid s_{i-1}^{(p)}\right)$ term represents a state-transition probability, and each $P\left(d_i^{(p)} \mid s_i^{(p)}\right)$ term represents the probability of staying in state $s_i^{(p)}$ for the duration $d_i^{(p)}$. Because we assume that all sequences begin in the first-controlled state, $P\left(\text{first-controlled} \mid \text{start}\right) = 1$. Likewise, $P\left(\text{end} \mid \text{last-controlled}\right) = 1$. Figure 3.3b depicts the states and transitions for such a model.

In order to capture the effect of seasonal determinants of exacerbations, we can extend the above model to use inhomogenous duration distributions for the controlled states. Specifically, our approach uses distinct duration distributions for the controlled states conditioned on the month in which the patient entered the controlled state. In this way, the timing of a patient's transition to the exacerbated state can depend on the time of year. To implement this inhomogeneity, we extend the represention of patient p's exacerbation history to indicate the month $m_i^{(p)}$ in which each state $s_i^{(p)}$ is entered:

$$\mathbf{s}^{(p)} \equiv \langle s_1^{(p)}, \ldots, s_{L_p}^{(p)} \rangle \qquad \mathbf{d}^{(p)} \equiv \langle d_1^{(p)}, \ldots, d_{L_p}^{(p)} \rangle \qquad \mathbf{m}^{(p)} \equiv \langle m_1^{(p)}, \ldots, m_{L_p}^{(p)} \rangle.$$

We then condition on the month sequence when determining the probability of

the states and durations:

$$P\left(\mathbf{s}^{(p)}, \mathbf{d}^{(p)} \mid \mathbf{m}^{(p)}\right) = P\left(s_1^{(p)} \mid \mathsf{start}\right) \cdot P\left(d_1^{(p)} \mid s_1^{(p)}\right) \cdot \prod_{i=2}^{L_p} \left[P\left(s_i^{(p)} \mid s_{i-1}^{(p)}\right) \cdot P\left(d_i^{(p)} \mid s_i^{(p)}, m_i^{(p)}\right)\right] \cdot P\left(\mathsf{end} \mid s_{L_p}^{(p)}\right).$$

Note that, in this formulation, the duration of the sojourn in the first state does not depend on the month since all the sequences begin on the same date. Additionally, for $P\left(d_i^{(p)} \mid \mathsf{last\text{-}controlled}, m_i^{(p)}\right)$ and $P\left(d_i^{(p)} \mid \mathsf{exacerbated}, m_i^{(p)}\right)$ in our models, there is no dependence on $m_i^{(p)}$. We make this choice for the last-controlled state because the durations in this state are censored and hence not informative. Although it would be reasonable to have the duration distribution for the exacerbated state depend on the month, we posit that there is not a strong dependence here and choose not to incorporate it into our representation.

To represent duration distributions, we use histograms at the time granularity of days. The duration for all states is capped at 1,826 days (five years) which is the length of the patient histories. All controlled states have a minimum duration of one day and the exacerbated state has a minimum duration of five days (since our exacerbation phenotyping procedure specifies this as the minimum duration). To contend with the sparsity of our data when estimating durations, we use Gaussian kernel density estimation (with bandwidth = 0.3) followed by discretization to days to smooth the histograms.

In order to cluster patients into distinct subpopulations, we construct a mixure of semi-Markov models as shown in Figure 3.4. Each component of the mixture incorporates the set of states shown in Figure 3.3b, aside from there being a common start state. Thus, for example, instead of having one first-controlled state, there is one per component. The transition probabilities from the start state represent prior probabilities of mixture components (i.e., mixture weights). By allowing the parameters in the component semi-Markov models to vary from one component to

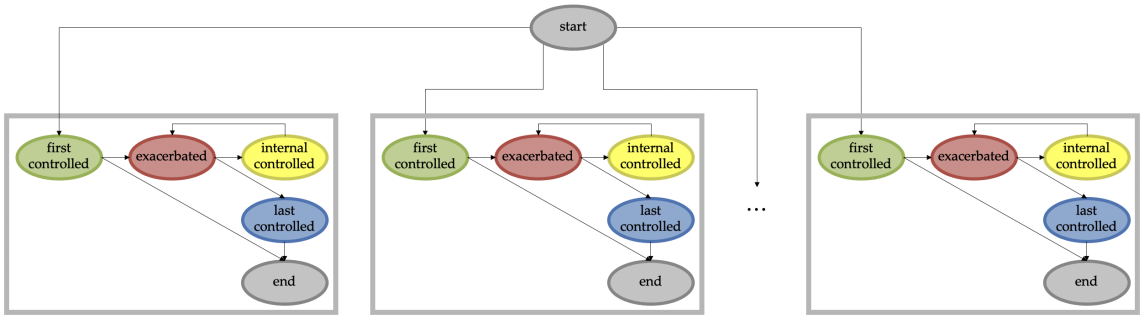


Figure 3.4: A mixture of semi-Markov models for characterizing asthma exacerbations. The mixture components are shown enclosed in gray boxes.

another, we can learn state transitions and duration distributions that characterize different subpopulations.

We learn the parameters for the mixture of semi-Markov models using an Expectation Maximization approach. To initialize the duration parameters for each state, we randomly select from the training set 10 events corresponding to the given state (and month when applicable) and use these events to estimate the associated duration distribution. The transitions going out of each state are initialized to a uniform distribution. To mitigate the effect of local optima in the EM procedure, the parameter estimation process is restarted 10 times, each time re-initializing the model with a different randomly selected subset of events from the training data. For a given number of components k, we then select the model that maximizes the likelihood of the training-set data.

3.4 Results

In this section, we describe the results from phenotyping asthma exacerbations from EHRs, predicting near-term exacerbations, and identifying subpopulations of asthma patients who have similar exacerbation patterns.

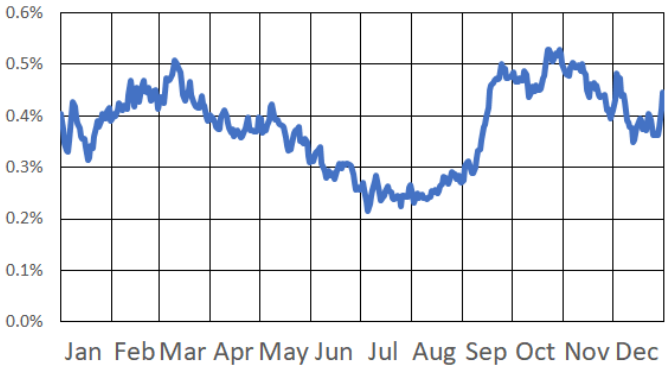


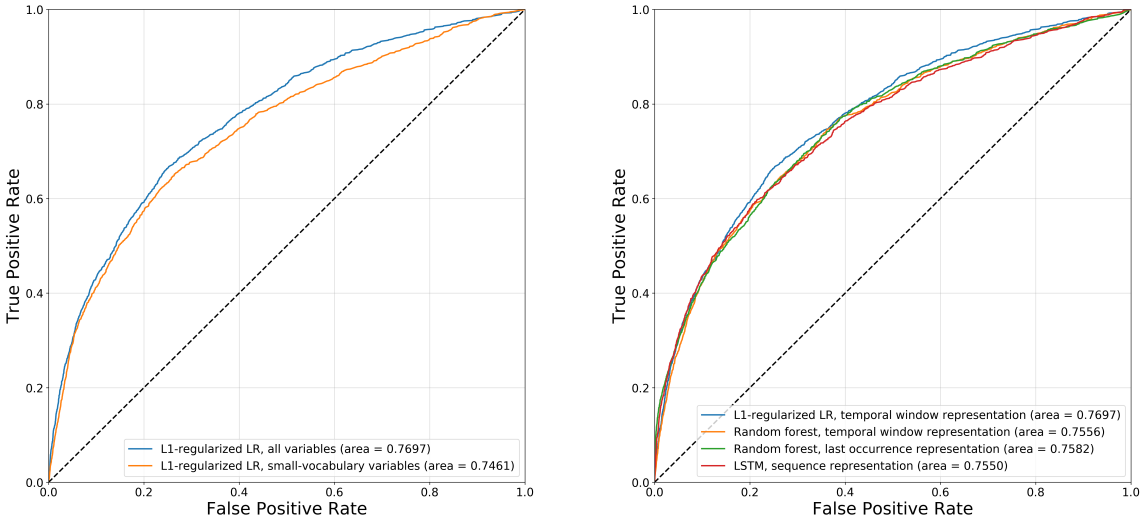
Figure 3.5: Plot of exacerbation frequency by time of year in our cohort. The y-axis represents the percentage of patients in our cohort who are in the midst of an exacerbation event on a given day of the year.

3.4.1 Phenotyping Asthma Exacerbations

We apply our asthma exacerbation phenotyping algorithm to the electronic health records for 28,101 asthma patients. The algorithm identifies a total of 14,447 exacerbations in these records. Figure 3.5 shows how the frequency of exacerbations varies by time of year in our patient cohort. Several notable features are present in this plot, including a spring peak corresponding to pollen-triggered exacerbations, an early fall peak corresponding to the increase in respiratory virus illness as children return to school, and a smaller peak centered on the holiday travel season.

3.4.2 Predicting Asthma Exacerbations

We evaluate our supervised learning approach for predicting asthma exacerbations using 10-fold cross-validation. In the present study, we consider one decision date per patient. For a patient in the training set, we train on data in the patient's EHR that precedes the decision date and determine the class label for the patient according to whether they experienced an exacerbation within 90 days after the decision date or not. For a patient in the test set, the learned model is given data in the patient's EHR that precedes the decision date and then predicts whether the



(a) (b)

Figure 3.6: ROC curves for asthma exacerbation prediction, comparing **(a)** L_1-regularized logistic regression models, with and without the inclusion of large-vocabulary EHR categories (diagnoses and interventions), and **(b)** the best-performing logistic regression model (L_1 regularization and a temporal window-based representation), random forests (temporal window-based and last occurrence-based representations), and LSTM (sequence-based representation).

patient will have an exacerbation within the next 90 days.

To ensure that our models are seasonally independent, we choose decision dates such that they are uniformly distributed throughout the days of the year, and we have at least 90 days on record after the decision date for each patient. Moreover, we left-censor the patient histories as needed to ensure that we have observation periods of the same length for every patient.

Figure 3.6a shows the receiver operating characteristic (ROC) curves for logistic regression models learned using L_1 regularization over a fixed-length representation based on the occurrence of event features in two temporal windows: (i) spanning the last six months, and (ii) spanning the entire observation period prior to the decision date. We show results with and without the inclusion of the large-vocabulary features in the EHR, namely the problem diagnoses, other diagnoses

and interventions (medications and procedures). In this way, we can evaluate the predictive value gained from the inclusion of these richer but more complex EHR features for the purpose of predicting asthma exacerbations. We also train and evaluate L_2-regularized logistic regression models but omit their results as they yield lower area under the ROC curve (AUROC) values than the L_1-regularized models.

The results shown in Figure 3.6a suggest that, given an asthma patient's past electronic health record, we are able to predict whether they will have an exacerbation in the near future with some degree of accuracy. The inclusion of large-vocabulary features yields a small but significant boost in AUROC, indicating the value of these richer but more complex features in predicting exacerbations.

Figure 3.6b shows ROC curves comparing multiple classifiers and representations used to predict asthma exacerbations, namely: (i) the best-performing logistic regression model, using L_1 regularization and a temporal window-based representation, (ii) random forest models using temporal-window and last occurrence-based representations, and (iii) the LSTM model using a sequence-based representation. Notably, logistic regression outperforms the more complex models given the same representation (random forest) as well as other representations (random forests, LSTM).

In order to gain insight into which EHR features are most valuable in predicting asthma exacerbations, we analyze the best-performing L_1 logistic regression model by ranking its coefficients in decreasing order of magnitude, and list the top-25 associated features in Table 3.1. These results suggest that while exacerbations in the last six months are the single greatest predictor for exacerbations in the near future, a diverse set of features are useful as predictors. Perhaps surprisingly, the majority of the most important features correspond to events observed at any point in the patient's past observation period, as opposed to more recent events observed in the last six months. While small-vocabulary features such as previous exacerbations, ACT scores, vitals and demographics provide significant predictive value (as indicated in Figure 3.6a), the large-vocabulary features (diagnoses and interventions) dominate the list of most important predictors upon their inclusion.

Table 3.1: Top-25 features of the best-performing L_1-regularized logistic regression model, ranked in decreasing order of coefficient magnitude.

Coefficient	Window	Category	Feature
0.90	6 Months	Exacerbations	Asthma exacerbation
0.48	Ever	Prescription meds	Corticosteroids
0.41	Ever	Diagnoses	V58.65: Long-term (current) use of steroids
0.35	Ever	Exacerbations	Asthma exacerbation
0.29	Ever	Procedures	Periodic preventive medication, infant
-0.27	Ever	Problem diagnoses	493.90: Unspecified asthma
-0.22	Ever	Diagnoses	493.81: Exercise-induced bronchospasm
-0.22	Ever	Problem diagnoses	493.00: Extrinsic asthma, unspecified
0.21	Ever	Procedures	Hospital discharge day management <30 min
-0.21	Ever	Diagnoses	493.90: Unspecified asthma
-0.18	N/A	Demographics	Race: White
0.18	Ever	Administered meds	Anticholinergics
-0.17	Ever	Procedures	Office outpatient visit <5 min
0.17	6 Months	Procedures	Office outpatient visit <15 min
0.17	Ever	Procedures	Breathing capacity test
-0.16	Ever	Diagnoses	V03.89: Other specified vaccination
0.15	6 Months	Charges	IV infusion therapy/prophylaxis
0.15	6 Months	Diagnoses	493.90: Unspecified asthma
-0.14	Ever	Procedures	Urinalysis
0.13	Ever	Prescription meds	Penicillin Combinations
-0.13	Ever	Procedures	Office outpatient visit <15 min
0.12	Ever	Procedures	Residual lung capacity
0.12	Ever	Diagnoses	493.92: Unspecified asthma with acute exbn
0.12	Ever	Charges	HB-visit units 46+ minutes room usage
0.12	N/A	Demographics	Age: 55-60 Years

Notably, some asthma diagnosis codes are negatively associated with future exacerbations. A possible explanation is that these codes tend to be associated with less acute cases of asthma.

3.4.3 Identifying Subpopulations of Asthma Patients

For the second task considered, we evaluate our approach to clustering patients using a mixture of semi-Markov models. We partition the patients such that 80% of them are in a training set, and the remaining 20% are in a test set. For each specified

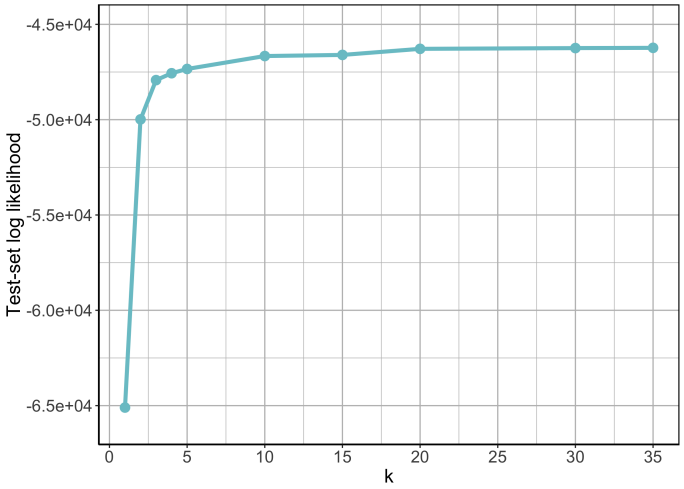


Figure 3.7: Log-likelihood of the test set data as the number of components, k, is varied.

number of mixture components, k, we learn a model using data from patients in the training set. We then evaluate the model by determining the likelihood of the test-set patients under that model. To mitigate the effect of local optima in the EM procedure, we use 10 multiple random restarts for a given value of k and then select the model that maximizes the likelihood of the training-set data.

Figure 3.7 shows the resulting test-set log-likelihoods for values of k ranging from 1 (a single semi-Markov model) to 35. We can draw several conclusions from these results. First, the models with multiple components explain the test-set data better than the individual semi-Markov model. Second, the log-likelihood keeps increasing as we add components to model until about 20, and it then levels off. Even with 35 components, however, we do not see evidence of overfitting.

To gain insight from the models, we inspect the learned parameters. Figure 3.8 shows selected duration distributions from a learned mixture of semi-Markov models when the number of components $k = 5$. Each row represents a component. The first column shows the duration distribution for the exacerbated state, and subsequent columns show duration distributions for the internal-controlled state conditioned on entering the state in the months of January, April, July or October.

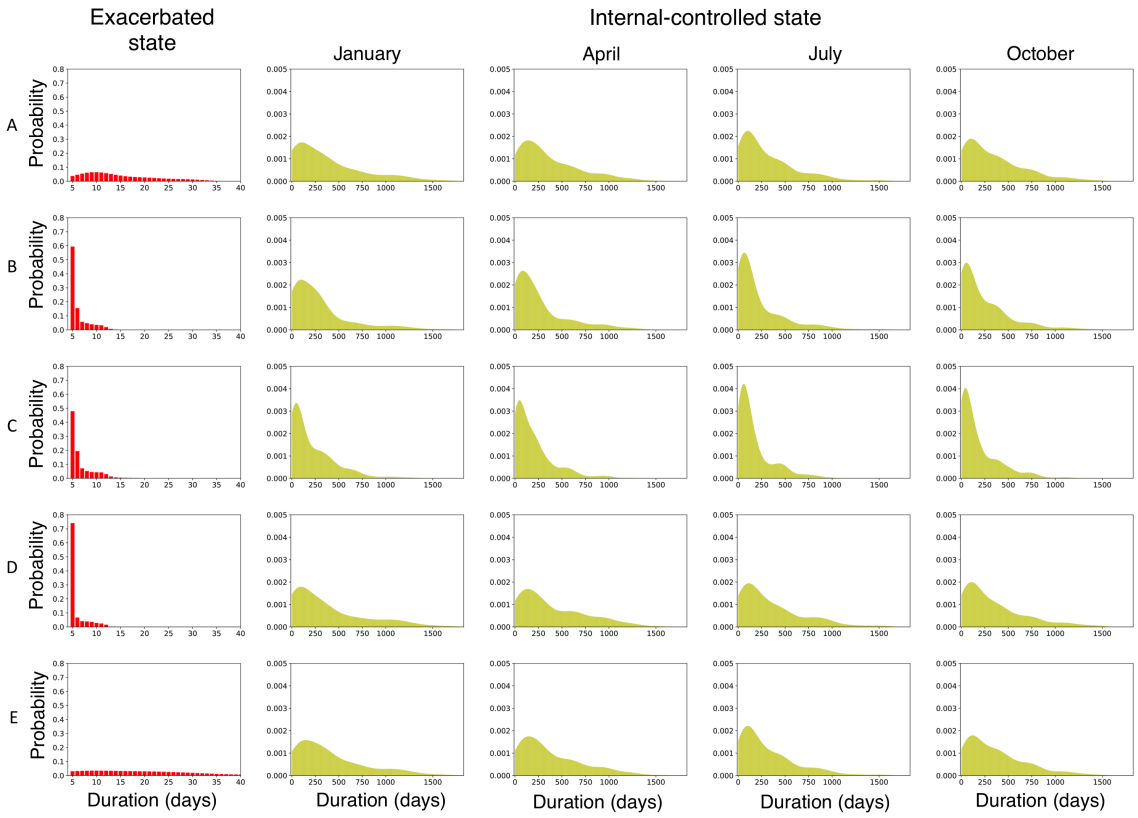


Figure 3.8: Selected duration distributions from the mixture of semi-Markov models. Each row shows a selection of the learned state duration distributions for a mixture component in a 5-component model. The first column shows the duration distribution for the exacerbated state. Subsequent columns show duration distributions for the internal-controlled state, conditioned on entering the state in January, April, July or October.

Recall that the internal-controlled state has a separate duration distribution for each month of the year; we show the distributions for only four months due to space limitations. Table 3.2 shows the transition probabilites for each component in the mixture.

The distributions shown in Figure 3.8 and Table 3.2 illustrate several notable differences among the components. Component A mostly represents patients who

Table 3.2: Transition probabilities for the 5-component mixture of semi-Markov models.

Component	Transition Probabilities	
	first-controlled to exacerbated	exacerbated to internal-controlled
A	0.7509	0.7317
B	0.9965	0.2150
C	0.9981	0.5894
D	0.9985	0.5237
E	0.0003	0.4996

struggle to keep their asthma under control. Within this component, the duration distribution for the exacerbated state has a long tail, and the probability of transitioning to the internal-controlled state is relatively high indicating that many patients in this component have experienced multiple exacerbations during the observation period. However, this component also seems to represent the patients who did not experience *any* exacerbations during the observation period. This is indicated by the relatively low transition probability (0.7509) from the first-controlled state to the exacerbation state. The patients who do not take this transition remain in the first-controlled state for the entirety of the observation period. Component B represents patients who have infrequent exacerbations. This is indicated by the relatively low probabability of transitioning from the exacerbated state to internal-controlled state, meaning that most of these patients had only one exacerbation during the observation period. Components C and D are similar to one another except that patients in the former generally have somewhat more prolonged exacerbations and shorter sojourns in the internal-controlled state. Component E represents patients who rarely, if ever, experience exacerbations. The probability of transitioning to the exacerbated state is near zero and the duration distributions are very close to their initialized values.

The internal-controlled duration distributions show heterogeneity across the months, generally being more peaked in the proximity of fall. However, with the 5-component model, we do not see components with pronounced specificity

for seasonal exacerbation patterns (e.g., we do not see a component that obviously corresponds to fall exacerbators). We see such clusters in some of the models with more components.

These results demonstrate that our mixture of semi-Markov models approach is able to identify subpopulations of patients who exhibit meaningful differences in the temporal patterns of their exacerbations.

3.5 Discussion

We have presented approaches and empirical results that address two key tasks in modeling asthma exacerbations from electronic health records. First, we considered to what extent exacerbations can be predicted given a patient's clinical history as represented in their electronic health record. Our results indicate that learned models are able to predict exacerbations with a moderately high degree of accuracy (AUROC ≈ 0.77) when given such information. The ability to predict asthma exacerbations is important to identify the patients that require more aggressive treatment plans and closer medical followup to improve patient outcomes. Second, we considered whether distinct temporal exacerbation phenotypes can be elicited from EHR data. Our approach to this task, which is based on a mixture of semi-Markov models, is able to identify subpopulations of asthma patients sharing distinct temporal and seasonal patterns in their exacerbation susceptibility.

4 UNDERSTANDING LEARNED MODELS BY IDENTIFYING IMPORTANT FEATURES AT THE RIGHT RESOLUTION

In many application domains, it is important to characterize how complex learned models make their decisions across the distribution of instances. One way to do this is to identify the features and interactions among them that contribute to a model's predictive accuracy. In this chapter, we present a model-agnostic approach to this task that makes the following specific contributions. Our approach (i) tests feature groups, in addition to base features, and tries to determine the level of resolution at which important features can be determined, (ii) uses hypothesis testing to rigorously assess the effect of each feature on the model's loss, (iii) employs a hierarchical approach to control the false discovery rate when testing feature groups and individual base features for importance, and (iv) uses hypothesis testing to identify important interactions among features and feature groups. We evaluate our approach by analyzing random forest and LSTM neural network models learned in two challenging biomedical applications.

This work was performed in collaboration with Kyubin Lee and Mark Craven and was published in the proceedings of the 33rd AAAI Conference on Artificial Intelligence (Lee et al. 2019b). The code for running the algorithm and generating experimental results is available at `https://github.com/Craven-Biostat-Lab/mihifepe` and as Python package 'mihifepe' on PyPI.

4.1 Introduction

In many application domains, it is important to be able to inspect, probe, and understand models learned by machine learning systems. It may be the case that the machine learning approaches that provide the best predictive performance in a

given domain learn models that are highly challenging to inspect and understand. For this reason, a number of approaches have been developed for gaining insight into complex learned models such as random forests and deep neural networks.

Methods for gaining comprehensible descriptions of learned models may broadly be categorized based on (i) explanation locality, (ii) model specificity and (iii) explanation targets, among other characterizations (as discussed in Section 2.1.2.1). Existing methods largely focus on explanations in terms of base features, i.e., features that are input to the models. However, in many domains, particularly with large feature spaces, base features may not represent the right level of resolution at which to interpret models. The multiplicity of base features may lead to explanations that are less concise and hence less interpretable. Moreover, the effects of individual base features on the model may be small or statistically insignificant.

Some methods attempt to expand the expanation vocabulary beyond the base features. This includes the use of latent representations (Bau et al. 2017), derived representations (Alvarez-Melis and Jaakkola 2018b; Kim et al. 2018; Zhou et al. 2018), and image regions (Fong and Vedaldi 2017; Ribeiro et al. 2016). However, these methods largely focus on explanations that are local, that are specific to certain model architectures and/or to the visual domain, and that target the model's sensitivity to all features (via model outputs) rather than to features relevant to the modeling task (via model losses).

In this chapter, we present a model-agnostic approach that explains models globally and at multiple resolutions by leveraging feature hierarchies to expand the explanation vocabulary. Our approach explains models in terms of important features, groups of features, and interactions among them. The prior research that is most closely related to ours includes methods that aim to provide model interpretability by identifying important features through perturbations of input (Breiman 2001; Li et al. 2016; Zeiler and Fergus 2014).

The specific contributions of our approach are the following. First, it is well-suited to tasks with large, structured feature spaces, where the base features might not provide the best level of resolution for characterizing what is important to the learned models. Our approach tests feature groups, in addition to base features, and

tries to determine the level of resolution at which we can determine the important features. Second, we go beyond just ranking features according to their importance, and instead use hypothesis testing to assess the effect of each feature on the model's loss. Given the potentially large number of hypothesis tests that must be performed, we use a hierarchical approach to control the false discovery rate when testing feature groups and base features for importance. Third, we propose a method based on hypothesis testing to identify important interactions among base features and feature groups.

We evaluate our approach by analyzing random forest and LSTM neural network models learned in two application domains: identifying viral genotype-to-disease-phenotype associations, and predicting asthma exacerbations from electronic health records. Additionally, we validate our approach using synthetic data sets where we know which features and feature groups are relevant.

4.2 Methods

In this section, we describe the key elements of our model-agnostic approach for characterizing learned models.

4.2.1 Identifying Important Features via Perturbation

Algorithm 1 outlines a general approach to identifying important features in a learned model. It measures how the output of the model, or its loss, varies when individual features in a given set of instances are perturbed in some way. Breiman (2001) proposes an approach based on this idea as a way to characterize learned random forest models. In Breiman's method, the perturbation is performed by permuting the values of the given feature across a set of instances. However, the approach can be generalized to other perturbations, such as replacing their values by zero values (Li et al. 2016; Zeiler and Fergus 2014), other constant values (Fong and Vedaldi 2017; Ribeiro et al. 2016), or values sampled from a noise distribution (Fong and Vedaldi 2017; Suresh et al. 2017).

Algorithm 1: General approach to identifying important features via perturbation

input : learned model f, feature set $\mathcal{F}$, test set $\mathcal{X} = \{(\mathbf{x}^{(1)}, y^{(1)}) \dots (\mathbf{x}^{(M)}, y^{(M)})\}$

output: set $\{(j, v_j) \,|\, j \in \mathcal{F}\}$ summarizing the effect v_j on loss $\mathcal{L}$ when perturbing each feature j

foreach feature $j \in \mathcal{F}$ **do**
  foreach instance $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{X}$ **do**
    foreach perturbation $p = 1, \dots P$ **do**
      let $\tilde{\mathbf{x}}_j^{(i,p)}$ represent $\mathbf{x}^{(i)}$ with feature j perturbed in some way
      calculate perturbed model output $f\left(\tilde{\mathbf{x}}_j^{(i,p)}\right)$ for the p^{th} perturbation
    compare loss $\mathcal{L}\left[y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right]$ to $\frac{1}{P}\sum_{p=1}^{P} \mathcal{L}\left[y^{(i)}, f\left(\tilde{\mathbf{x}}_j^{(i,p)}\right)\right]$
  calculate summary statistic v_j characterizing the effect of perturbing feature j on $\mathcal{L}$

A key extension of this idea in our approach is that it uses hypothesis testing to determine whether a given feature has a generally consistent effect on the model's loss across the distribution of instances. We do this using held-aside test instances so that our importance assessment measures whether a feature truly impacts a model's predictive accuracy. In the results presented here, we use the Wilcoxon matched-pairs signed-rank test to assess the null hypothesis that the median difference between pairs:

$$\mathcal{L}\left[y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right] - \frac{1}{P}\sum_{p=1}^{P} \mathcal{L}\left[y^{(i)}, f\left(\tilde{\mathbf{x}}_j^{(i,p)}\right)\right] \tag{4.1}$$

is zero. Here $\tilde{\mathbf{x}}_j^{(i,p)}$ is defined as $\mathbf{x}^{(i)}$ with feature j perturbed on the p^{th} permutation. For perturbations that do not involve randomness, such as erasure, $P = 1$ and $\tilde{\mathbf{x}}_j^{(i,1)}$ denotes the single perturbation that can be done to feature j.

We use the Wilcoxon test instead of a paired t-test due to significant non-

normality in the changes to loss introduced by feature perturbations. Here, we use the one-tailed version of the test, corresponding to the null hypothesis that the median difference is *greater* than zero, in order to focus on features that provide predictive value to the model. Alternatively, we could use a two-tailed test to also detect features whose perturbation *decreases* loss, thereby indicating overfitting.

4.2.2 Considering Feature Groups

The approach described in Algorithm 1 is typically applied to the set of features that are used as input to the model, which we refer to as *base features*. We argue that, in many domains, characterizing the importance of base features may not be the right level of resolution for gaining a thorough understanding of a learned model. In some domains, there may be a large number of features that are important to the model, and it may be difficult to discern which high-level factors are most important for the model's predictions unless groupings of related features are considered. For example, models that perform risk assessment from electronic health records often have thousands of base features representing distinct diagnoses. Our understanding of such a model is likely to be aided by analyzing the importance of groups of related diagnoses, or even the entire set of diagnoses, in addition to very specific ones. Moreover, it might be the case that few, if any, individual base features show a statistically significant change to the model's loss when perturbed, or the effect sizes of these changes to the loss are small. In such cases, we can potentially detect statistical significance and larger effect sizes by considering groups of related features.

In contrast to assessing feature importance only at the level of base features, our approach also assesses the importance of *feature groups*. We assume that we are given a hierarchy in which internal nodes represent groups of features, and leaf nodes represent base features. We can then apply Algorithm 1 to both base features and feature groups in order to determine which are important.

In some application domains, such as risk assessment from EHRs, there are standard ontologies which can be used to define the hierarchy of feature groups.

For example, the International Classification of Diseases (ICD) and the Clinical Classifications Software (CCS) both define hierarchies of semantically related groups of diagnoses and procedures. In a risk-assessment application, the base features might represent the occurrence of specific recorded diagnoses in a given patient's EHR, such as reflux esophagitis (ICD-9 code 530.11) or acute esophagitis (ICD-9 530.12). We could test the importance of such features by erasing all occurrences of the given diagnosis from patients' records and measuring the resulting loss. Moreover, we might test the importance of the feature groups esophagitis (ICD-9 530.1), which has five children diagnoses including the two listed above, or diseases of the esophagus (ICD-9 530), which has 28 descendant diagnoses. To test a feature group, we could erase all recorded diagnoses that are encompassed by the group.

In other application domains, the feature groups might be derived from data. For example, in our viral genotype-to-phenotype task, we calculate feature groups using a hierarchical clustering method. Our base features are *haplotype blocks*, which are variable-sized regions of the genome that have been inherited as a unit from one of two parental virus strains. Our feature groups consist of sets of neighboring haplotype blocks (i.e., larger regions of the viral genome).

In a natural language domain, we might define feature groups on the basis of syntactic or semantic categories. For example, if the base features are binary indicators of the presence of specific words and dependency-parse paths, we might define feature groups for syntactic categories such as relative clauses, and test the importance of such a feature with a perturbation that simulates the "erasure" of all relative clauses from instances, thereby setting to zero word and dependency-path features that were derived these clauses.

In an image classification domain, the base features might correspond to pixels and we might define feature groups to represent superpixels or objects as feature groups. Perturbations could involve replacing a region with a constant value, injecting noise, or blurring (Fong and Vedaldi 2017).

In contrast to approaches for hierarchical feature selection (Wan and Freitas 2018), the hierarchies used by our approach do not necessarily represent *is-a* or *generalization-specialization* relationships. Each internal node needs only to group

features that are related in some meaningful way (e.g., neighboring regions of a genome). Moreover, our approach is not focused on feature selection per se, but instead on characterizing which feature groups are important in a given learned model.

4.2.3 Controlling the False Discovery Rate

Given a hierarchy over the features, we can compute the effect of perturbing each base feature and each feature group using Algorithm 1 across a given set of instances. We treat each node in the hierarchy as representing the null hypothesis that perturbing the corresponding feature group does not have a significant effect on the loss function, in the sense that the median of the differences computed using Formula (4.1) is zero. A hypothesis is rejected if this median difference is statistically significantly different from zero, and a hypothesis is tested only if its parent hypothesis has been rejected.

However, there is a notable multiple-comparisons problem due to the potentially large number of hypotheses tested. For instance, there are 8,740 hypotheses to be tested (counting both base features and feature groups) in the asthma exacerbation prediction task that we address. Moreover, when adjusting for multiple comparisons, we need to take into account the hierarchical organization of the hypotheses being tested. We address this issue by using the hierarchical false discovery rate (FDR) control methodology developed by Yekutieli (2008), as described in Algorithm 2.

This algorithm uses a recursive procedure to consider a hierarchical set of hypotheses, which in our case consist of feature groups to be tested. If the null hypothesis is rejected for a given node in the hierarchy (i.e., we determine that a feature group is important), then the children of that node are tested using the Benjamini-Hochberg procedure (Benjamini and Hochberg 1995) to control false discoveries. Otherwise, the descendants of the given node are not tested. The algorithm returns a subtree representing the set of feature groups and base features for which the null hypothesis was rejected.

Algorithm 2: Using hierarchical FDR control to identify important features

input : Tree $\mathbb{T}$ of hypotheses to be tested along with their associated p-values, significance level q

output: A subtree $\mathbb{S}$ of $\mathbb{T}$ corresponding to hypotheses rejected while controlling FDR at significance level q

function `HierarchicalFDR` (*node*)**:**
- // node has already been rejected
- *rejectedSet* = { *node* }
- **if** *node* is not leaf **then**
  - let $P_{(1)} \leq \ldots \leq P_{(k)}$ be the set of ordered p-values of *node*.children
  - // Apply Benjamini-Hochberg procedure to children
  - let $r = \max\{i : P_{(i)} \leq \frac{i \times q}{k}\}$
  - **if** $r > 0$ **then**
    - *rejectedChildren* = set of r hypotheses corresponding to $P_{(1)} \leq \ldots \leq P_{(r)}$
  - **foreach** *child* $\in$ *rejectedChildren* **do**
    - *rejectedSet* = *rejectedSet* $\cup$ `HierarchicalFDR` (*child*)
- return *rejectedSet*

begin
- **if** $\mathbb{T}$.*root.pvalue* $> q$ **then**
  - $\mathbb{S}$ = empty tree
- **else**
  - $\mathbb{S}$ = `HierarchicalFDR` ($\mathbb{T}$.*root*)

Using this algorithm, we can identify the set of feature groups and base features that have a significant effect on a model's loss, while controlling the rate of false discoveries in this set. Of particular interest is the set of *outer* nodes: those nodes for which we reject the null hypotheses (i.e., determine that they are important) and that have no children for which we reject the null hypotheses. These nodes represent the highest level of resolution at which we can determine the importance of features and feature groups.

The key assumptions made by this approach, which are reasonable in our context, are that (i) if a given feature significantly affects the loss when perturbed, a group of features containing this feature will also significantly affect the loss when perturbed, (ii) the p-values for siblings are independently distributed, and (iii) p-values for true null hypotheses are uniformly distributed in [0,1].

4.2.4 Identifying Important Interactions

In addition to identifying individual base features and feature groups that are important, we would also like to identify interactions among them that a given model has determined as important. Here we consider cases in which the model outputs a scalar value. For this analysis, we do not treat a given model completely as a black box, but instead assume that we know the transfer function that produces the model's outputs. Let $g(\mathbf{x}^{(i)})$ denote the function that maps $\mathbf{x}^{(i)}$ to the value that is input to the transfer function $h(\cdot)$, and $f\left(\mathbf{x}^{(i)}\right) = h\left(g(\mathbf{x}^{(i)})\right)$ indicate the output of the model. For example, $h(\cdot)$ might be a logistic activation function in a neural network for a binary classification task, in which case $g(\cdot)$ would represent the part of the network that maps from $\mathbf{x}^{(i)}$ to the net input of the logistic function. Or in a random forest trained for a regression task, $h(\cdot)$ would represent the identity function, and $g(\cdot)$ would represent the average of the values predicted by the individual trees in the forest.

Our notion of an interaction among features is based on the concept of additivity. We define an interaction between feature j and feature k to mean that changes in

$g(\cdot)$ when we perturb both features are non-additive (for some instances):

$$\left[g\left(\tilde{\mathbf{x}}_j^{(i)}\right) - g\left(\mathbf{x}^{(i)}\right)\right] + \left[g\left(\tilde{\mathbf{x}}_k^{(i)}\right) - g\left(\mathbf{x}^{(i)}\right)\right] \not\approx \left[g\left(\tilde{\mathbf{x}}_{j\wedge k}^{(i)}\right) - g\left(\mathbf{x}^{(i)}\right)\right] \tag{4.2}$$

where $\tilde{\mathbf{x}}_{j\wedge k}^{(i)}$ denotes instance $\mathbf{x}^{(i)}$ with feature j and feature k perturbed jointly.

To identify interactions that are important, we use hypothesis testing to assess whether a candidate interaction exhibits non-additivity. We can do this by considering the median difference between pairs formed by the two sides of the inequality above. In the results presented here, we use the Wilcoxon matched-pairs signed-rank test to assess the null hypothesis that the median difference between the pairs is zero. This approach to testing interactions can be applied to base features, feature groups, and mixtures thereof.

Alternatively, we can consider whether a candidate interaction exhibits non-additivity which has a generally consistent effect on the model's loss across the distribution of instances. We can do this by assessing the difference between pairs:

$$\mathcal{L}\left[y^{(i)}, h\left(g\big(\tilde{\mathbf{x}}_{j\wedge k}^{(i)}\big)\right)\right] - \mathcal{L}\left[y^{(i)}, h\left(g\big(\mathbf{x}^{(i)}\big) + \Delta g\big(\tilde{\mathbf{x}}_j^{(i)}\big) + \Delta g\big(\tilde{\mathbf{x}}_k^{(i)}\big)\right)\right] \tag{4.3}$$

where $\Delta g\big(\tilde{\mathbf{x}}_j^{(i)}\big)$ is defined as $\left[g\big(\tilde{\mathbf{x}}_j^{(i)}\big) - g\big(\mathbf{x}^{(i)}\big)\right]$ (i.e., the change in $g(\mathbf{x}^{(i)})$ that results from perturbing feature j). However, the null distribution may not be as straightforward to work with in this case because, depending on the loss function, the difference in variances of the inner terms on each side may lead to the loss terms having different means.

4.3 Results

In this section, we evaluate our approach by (i) assessing its ability to detect important features and interactions while controlling FDR on synthetic data sets, and (ii) applying it in two biomedical domains in which it is essential to understand learned models.

4.3.1 Evaluation on Synthetic Data Sets

To verify that our approach is able to identify important features and interactions while controlling the false discovery rate, we first evaluate it using data sets for which we know the relevant features. In this setting, we formulate a ground-truth function of the form:

$$y^{(i)} = \sum_{j \in \mathcal{R}_L} \alpha_j x_j^{(i)} + \sum_{\substack{(j,k) \in \mathcal{R}_I \\ j \neq k}} \alpha_{jk} x_j^{(i)} x_k^{(i)} \tag{4.4}$$

where $\mathcal{R}_L$ and $\mathcal{R}_I$ represent the subset of relevant linear and interaction terms respectively, and α_j and α_{jk} are corresponding coefficients that determine how feature j and interaction (j, k) contribute to the output. Note that a feature is considered important if belongs to $\mathcal{R}_L$, or is a component of an interaction that belongs to $\mathcal{R}_I$, or both. We represent a "learned" model using a function that approximates the ground-truth function:

$$f\left(\mathbf{x}^{(i)}\right) = \sum_{j \in \mathcal{R}_L} \alpha_j x_j^{(i)} + \sum_{\substack{(j,k) \in \mathcal{R}_I \\ j \neq k}} \alpha_{jk} x_j^{(i)} x_k^{(i)} + \gamma^{(i)} \tag{4.5}$$

where $\gamma^{(i)} \sim N(0, \sigma^2)$ represents the deviation of the model's output from the ground-truth function for some instance i in the feature space. This formulation is intended to simulate the situation in which a learned model provides a fairly accurate representation of the underlying target function, but incorporates irrelevant features and other deviations which have a small impact on the model's outputs.

We generate synthetic data sets by drawing feature vectors from a given distribution, and then using Equation 4.5 to determine $f(\mathbf{x}^{(i)})$ for each $\mathbf{x}^{(i)}$, and similarly for each perturbation of $\mathbf{x}^{(i)}$. Here we present results in which our instance spaces have 500 binary features, and each underlying ground truth function has 50 important features and 50 important interactions selected from among these, with coefficients $\alpha_j \sim U(0,1) \quad \forall j \in \mathcal{R}_L$ and $\alpha_{jk} \sim U(0,1) \quad \forall (j,k) \in \mathcal{R}_I$. The feature vectors are constructed by sampling each feature from an independent Bernoulli distribution.

Table 4.1: Average power and FDR for features and interactions on synthetic data sets as the number of instances M in the test set is increased.

M	Features		Interactions	
	Power	FDR	Power	FDR
32	0.722	0.019	0.046	0.132
64	0.800	0.024	0.370	0.014
128	0.850	0.026	0.543	0.030
256	0.895	0.029	0.682	0.035
512	0.919	0.036	0.777	0.040
1024	0.936	0.035	0.840	0.039
2048	0.948	0.029	0.877	0.048
4096	0.960	0.029	0.913	0.045
8192	0.967	0.033	0.935	0.046
16384	0.975	0.032	0.949	0.039

We define feature groups by creating a balanced binary hierarchy with features randomly assigned to leaf nodes and feature groups represented by internal nodes. A feature group is considered important if it contains at least one important feature in its subtree. We perform perturbations by erasure (i.e., set the feature to zero in all instances) and use Equation 4.1 to perform hypothesis testing, followed by the hierarchical FDR procedure (Algorithm 2) with $q = 0.05$.

To analyze interactions, we use the base features identified as important in the preceding analysis to construct a set of potential interactions to test. This allows us to prune the large search space of all possible interactions, albeit at the cost of decreased power. We then use Equation 4.2 to perform hypothesis testing of these interactions, and use the Benjamini-Hochberg procedure to control FDR among this set.

Table 4.1 shows the results of applying our method as the number of instances in the "test set" is varied. The results in the table represent averages over 100 randomly generated models and datasets. For each test-set size, we report both the average power of the method (i.e., the fraction of relevant features and interactions that are identified as important) and the average false discovery rate (i.e., the

Table 4.2: Average power and FDR for features and interactions on synthetic data sets as the noise coefficient σ is increased.

σ	Features		Interactions	
	Power	FDR	Power	FDR
0.00	0.999	0.000	0.991	0.000
0.01	0.983	0.034	0.966	0.000
0.02	0.982	0.034	0.966	0.048
0.04	0.980	0.034	0.964	0.047
0.08	0.974	0.034	0.958	0.048
0.16	0.964	0.034	0.920	0.049
0.32	0.938	0.034	0.866	0.048
0.64	0.887	0.033	0.766	0.049
0.128	0.770	0.033	0.564	0.050

fraction of putatively important features and interactions that are irrelevant). The middle columns show average power and FDR for determining important features and feature groups, and the rightmost columns show average power and FDR for determining important interactions. Table 4.2 shows the effect of varying the coefficient σ for sampling the noise values $\gamma^{(i)}$ for each learned model (Equation 4.5). Here, the number of instances is fixed at 10,000. The results in Tables 4.1 and 4.2 indicate, not surprisingly, that the average power of our method to detect relevant features and interactions increases with larger test sets, and decreases with larger values of σ. Importantly, for all conditions, the average FDR is controlled at the 0.05 level.

The analyses of both features and interactions show similar trends. However, the average power for discovering important interactions trails the average power for discovering important features for any given test set size/noise level. This is because we only test an interaction if its constituent features have already been identified as important during the preceding feature analysis.

4.3.2 Real Application Domains and Models

The first real domain we consider is focused on identifying the genetic components of Herpes simplex virus type 1 (HSV-1) that are responsible for various dimensions of eye disease. Here we analyze random forest (RF) models that have learned mappings from variations in viral genotypes to three different eye disease phenotypes (Kolb et al. 2016). Each instance corresponds to a genetically distinct strain of the virus, and there are 65 recombinant strains generated by mixed infection of two parental strains. We represent each genotype as a vector of 547 features, where each feature corresponds to a *haplotype block* which is variable-sized regions of the genome that has been inherited as a unit from one of the two parental virus strains. The value of each binary feature indicates from which parental strain the haplotype block was inherited. The phenotypes (blepharitis, stromal keratitis, and neovascularization) for each instance are numeric scores indicating the disease severity resulting from infection in mice by a given strain. We choose to analyze learned RF regression models since they show statistically significant predictive accuracy for all three phenotypes and they demonstrate better cross-validated predictive accuracy than penalized linear regression models (Lasso and Ridge) for two of the three phenotypes, as well as other models that we train for this task. The cross-validated R^2 values for the blepharitis, stromal keratitis, and neovascularization models are 0.45, 0.56, and 0.48, respectively. Each learned RF model comprises 1,000 trees.

The second application domain we address is predicting asthma exacerbations from electronic health records. The cohort, task and model trained for the task are extensively described in Chapter 3, but pertinent details are reproduced here for completeness.

The data set consists of information derived from EHRs for a cohort of 28,101 asthma patients from the University of Wisconsin Health System over a five-year period. The information extracted from the EHRs includes demographic features and time-stamped events corresponding to encounters with the healthcare system. These events include problem-list and other coded diagnoses, procedures, medications, vitals, asthma control scores, and prior exacerbations. We also include

features representing the time since the last event, represented at multiple scales.

We train an LSTM model (Hochreiter and Schmidhuber 1997) to predict whether a patient would experience an exacerbation within the next 90 days or not, given their clinical history as represented in the EHR. We select a cell state of size 100 and a sigmoid output layer. The coded diagnoses, problem diagnoses, and interventions (procedures and medications) all comprise large vocabularies (6,533 for coded diagnoses, 4,398 for problem diagnoses, and 8,745 for interventions) of which only a small subset is recorded at each encounter. Therefore, we first map event vectors for each of these sets to an embedded space using Med2Vec (Choi et al. 2016b), resulting in shorter, dense fixed-length vectors. Separate embeddings of size 200 are generated for each of these sets, which are then concatenated, along with the other temporal features, to produce the event representation at each timestamp in the record. The ordered sequence of events forms the input sequence for the LSTM. The static demographic features are provided as input to the output sigmoid layer. We use 10-fold cross-validation to assess the predictive accuracy of the LSTM and obtain an area under the ROC curve (AUROC) of 0.757.

4.3.3 Feature Groups and Perturbations

For the HSV-1 application, our feature hierarchy represents neighboring regions of the viral genome. We compute the hierarchy using a constrained hierarchical clustering method applied to the base features, which represent haplotype blocks. This clustering method uses Hamming distance to compare columns (features) in our data matrix, and a complete linkage function, such that every pair of features in a given cluster is within a specified bit difference of each other. The agglomerative clustering operator groups features that are correlated (i.e., exhibit similar inheritance patterns) across the viral strains. Since we want our hierarchy to group *neighboring* haplotype blocks that are correlated, we constrain the clustering method such that hierarchy adheres to the linear ordering of the haplotype blocks with the HSV-1 genome. Thus, the merging step during clustering can be applied only to features or feature groups that are adjacent to each other in the genome. The

resulting hierarchy consists of 547 leaf nodes (base features) and 546 internal nodes (feature groups).

The perturbations we use to interrogate models in this domain are based on permutations. For a given feature or feature group, we randomly permute and reassign the values for the feature or feature group in the data matrix. When doing such permutations for feature groups, the values in the group for each instance are treated as a unit, being permuted and reassigned together. We do this perturbation 500 times for each feature or feature group when assessing its importance.

We consider two hierarchies over features for the asthma exacerbation prediction task. We construct a top-level hierarchy representing broad categories of EHR-elicited features (diagnoses, demographics, etc.). The second hierarchy we use is the standard ICD-9 hierarchy of diagnoses. In this application, we use erasure perturbations which involve zeroing out features or feature groups of interest. For event-based features, the erasure operation we use removes all occurrences of the feature from a patient's history. For features that are encoded in an embedded representation, the erasure operation is applied to the patient's history and then the embedding of the associated events is recomputed while retaining the same embedding model.

4.3.4 Identifying Important Features

In this section, we examine which features and feature groups we identify as being important to the learned models in both application domains, while controlling the false discovery rate at the 0.05 level (i.e., $q = 0.05$). Table 4.3 summarizes the results of our feature importance analysis of learned models for four tasks across both domains. The first row in the table indicates the number of base features and feature groups that are assessed for each model. The second row indicates the number of base features and feature groups that have an unadjusted p-value < 0.05 after performing hypothesis testing as described in Section 4.2. The third row shows the number of features that we ascertain to be important after performing hierarchical FDR control. The last two rows indicate, among the nodes surviving FDR control,

Table 4.3: Summary of hypothesis testing results for feature importance analysis in both application domains.

Nodes	HSV-1 Genotype-phenotype Association			Asthma Exacerbation
	Blepharitis	Stromal Keratitis	Neovascularization	ICD-9
Total nodes (base features + feature groups)	1,093	1,093	1,093	8,740
Nodes with unadjusted $p < 0.05$	242	148	111	3,480
Nodes rejected at q level < 0.05	107	110	80	3,179
Outer nodes	40	36	24	2,120
Feature groups among outer nodes	6	3	3	159

the total number of outer nodes and the number of outer nodes that correspond to feature groups. Recall that outer nodes refer to nodes at the highest resolution at which we can detect important features, i.e., nodes that survive FDR control but have no children that do.

Figure 4.1 provides a visual depiction of these results for the blepharitis phenotype model. Among the 1,093 base features and feature groups that are tested, we determine that 107 are important when controlling the FDR at $q = 0.05$. Moreover, the set of 40 outer nodes represents the highest level of resolution at which we can say that a viral genomic region is important to the phenotype. In the case of the blepharitis phenotype, six outer nodes are feature groups, representing genomic regions that are associated with the phenotype but for which we cannot localize precisely which base features are important. Figure 4.2 shows features identified as important for all three disease phenotypes, mapped to the genomic coordinates of the virus. Using our feature importance analysis of learned RF models, we are able to significantly narrow down the genetic determinants of disease from a large number of candidate regions. Several of these regions validate what is previously known about HSV-1 pathogenicity, and others indicate novel disease determinants (Kolb et al. 2016). Moreover, the results suggest a high degree of underlying causality among the three disease phenotypes, given the fact that there is substantial overlap among the regions identified as important.

Figure 4.3a shows the results of the feature importance analysis of the asthma exacerbation prediction model for feature groups at the highest level of the feature hierarchy. These results suggest that the most informative feature groups are

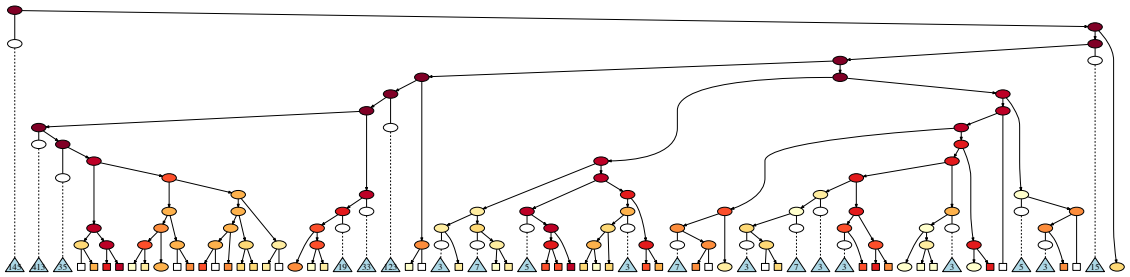


Figure 4.1: Feature importance analysis of the random forest model for blepharitis. Ovals represent feature groups, squares depict base features, and triangles depict subtrees of the hierarchy that were not tested by the FDR procedure. Color intensity indicates the magnitude of the associated p-value. White nodes are those that were tested but did not survive the FDR procedure.

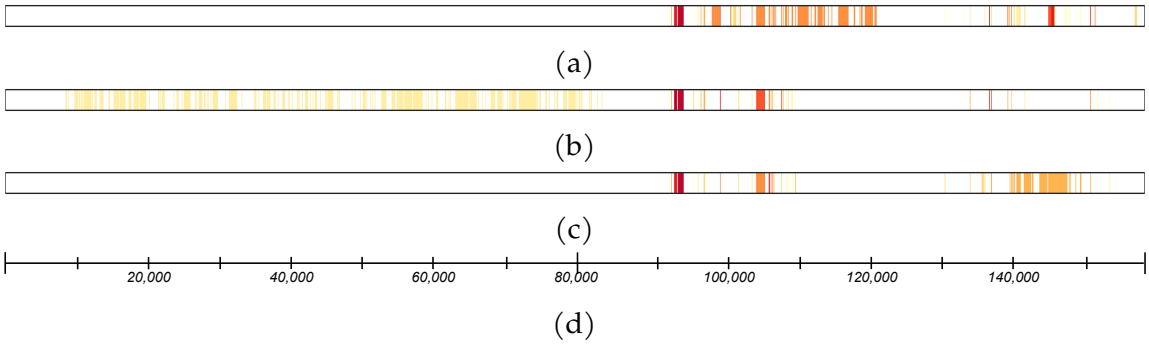


Figure 4.2: Important features mapped to the HSV-1 genome coordinates for all three disease phenotypes: (a) blepharitis, (b) stromal keratitis, (c) neovascularization. Color intensity indicates the magnitude of the associated importance p-value.

coded diagnoses (DIAGNOSES), intervals between events (TIMESTAMPS), and interventions (which includes medications and procedures). We note that even when all the features are erased (ROOT), the model still performs better than random, with AUROC = 0.537. This is likely due to the fact that the number of encounters in a patient's history is associated with the exacerbation risk. Even when we erase all other information, we leave the number of events in a patient's history intact. Figure 4.3b depicts a subset of features identified as important after performing hierarchical FDR analysis on the diagnosis feature group. These results

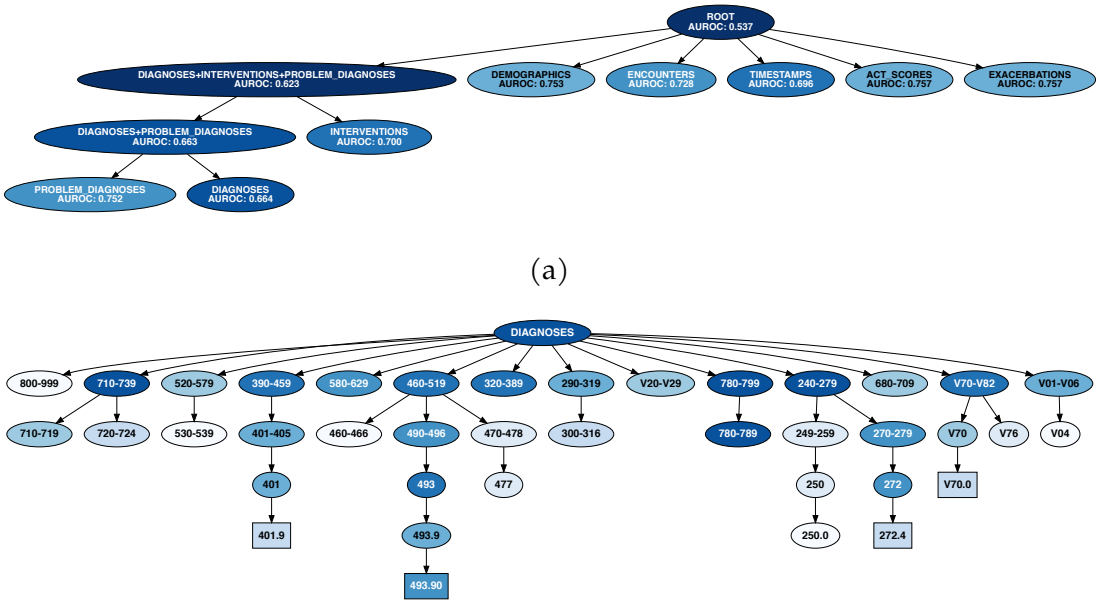


Figure 4.3: Feature importance analysis of the LSTM model for predicting asthma exacerbations. Darker shades correspond to larger effect sizes, i.e., lower model AUROCs when the feature groups are perturbed. (a) Subtree showing important feature groups at the highest level of the feature hierarchy. (b) Subtree showing important features and feature groups comprising the ICD-9 hierarchy of diagnoses. Note that the root node in panel (b) corresponds to the DIAGNOSES node in panel (a).

are also summarized in Table 4.3. A large number of hypotheses are rejected at FDR control level $q = 0.05$, indicating that many features and feature groups have some predictive signal for this task. Figure 4.3b shows a subtree of nodes surviving FDR control while having the largest effect sizes. The features identified as important include those with known connections to asthma, such as the respiratory diseases subtree (460-519) terminating at asthma (493.90), and the mental disorders subtree (290-319) (Scott et al. 2007). We also identify as important some features whose relationships to to asthma are not as clear, such as the metabolic diseases feature group (240-279).

4.3.5 Identifying Important Interactions

We apply our approach to identifying important interactions to the RF models trained to capture HSV-1 genotype-phenotype associations. We evaluate two sets of candidate interactions. First, we assess interactions across all pairs of outer nodes that are identified as important using feature importance analysis. We identify 780, 630, and 276 candidate pairwise interactions for blepharitis, stromal keratitis, and neovascularization, respectively. After performing hypothesis testing, we use the Benjamini-Hochberg procedure to control the false discovery rate at the 0.1 level. Only one candidate interaction across the three phenotype models survives FDR control, namely, an interaction between two base features for the stromal keratitis model, out of which one feature has the largest effect size among the outer nodes. Second, we consider interactions between a set of nodes located at an intermediate level in the feature hierarchy that survives FDR control during feature importance analysis. We are able to detect several significant interactions for the stromal keratitis phenotype. Among 435 candidate interactions tested, three interactions are identified as significant.

4.4 Discussion

We have presented a model-agnostic approach to understanding learned models by identifying important features at various levels of resolution. The key contributions of our approach are that it employs hypothesis testing, along with hierarchical feature groupings and a hierarchical-FDR control method, in order to rigorously assess which features and groups of features have a significant effect on a model's loss. We have also presented an approach for testing important feature interactions.

We demonstrated and evaluated our approach in the context of two biomedical domains. In both domains, our method lent insight into complex learned models by determining important features and feature groups. Additionally, we identified important interactions in one of our HSV-1 models. The analysis of the asthma exacerbation prediction model showed the differential impact of EHR categories

on the predicted outcome. We also examined which diagnoses, as defined by the ICD-9 hierarchy, are important in determining the model's predictions. Our analysis highlighted several known and some unknown (but potentially important) diagnoses associated with the asthma exacerbations, as determined by the model. Finally, our approach identified important features across a range of resolutions in both domains, from large feature groups down to base features, facilitating concise yet accurate descriptions of the model and aiding the goal of model interpretability.

5 FEATURE IMPORTANCE EXPLANATIONS FOR TEMPORAL BLACK-BOX MODELS

In this chapter, we propose TIME, a method to explain models that are inherently temporal in nature. Our approach (i) uses a model-agnostic permutation-based approach to analyze global feature importance, (ii) identifies the importance of salient features with respect to their temporal ordering as well as localized windows of influence, and (iii) uses hypothesis testing to provide statistical rigor.

This work was performed in collaboration with Mark Craven and appears at the 36th AAAI Conference on Artificial Intelligence, with a preprint available at `http://arxiv.org/abs/2102.11934`. The code for running the algorithm and generating the experimental results is available at `https://github.com/Craven-Biostat-Lab/anamod` and as Python package 'anamod' on PyPI.

5.1 Introduction

Existing research on model interpretability has largely focused on explaining models trained over tabular data, where each feature takes a single value per instance, instead of explaining temporal models, where the instances consist of sequences or time series. Ismail et al. (2020) demonstrate the unreliability and inaccuracy of commonly used model-agnostic and gradient-based methods when used to explain temporal models. Some approaches have focused on interpreting recurrent neural networks (Ismail et al. 2019; Karpathy et al. 2015; Suresh et al. 2017) and attention-based models (Choi et al. 2016c; Zhang et al. 2019), while others have explored methods to encourage temporal models during training to be more interpretable using tree regularization (Wu et al. 2017) and game-theoretic characterizations (Lee et al. 2018). However, these approaches require specific model architectures or training-time alterations, limiting their applicability.

Model-agnostic methods such as LIME (Ribeiro et al. 2016) and SHAP (Lund-

berg and Lee 2017) avoid this limitation by treating models as black-boxes but are designed for tabular representations. Recent work has begun to address model-agnostic explanation for temporal models. Tonekaboni et al. (2020) propose FIT, a method to assign importance scores for sequence-sequence models, and Bento et al. (2020) propose TimeSHAP, an extension of SHAP (Lundberg and Lee 2017) to temporal models. Importantly, all these methods focus on local interpretability, which seeks to explain individual predictions in terms of their important features, rather than global interpretability, which seeks to characterize a model's decisions across a population of instances.

Local and global explanations are complementary approaches to interpretability. While local explanations may be used to justify specific decisions, global explanations are often advantageous for model diagnostics, feature engineering, bias detection, trust, and scientific understanding (Doshi-Velez and Kim 2017; Ibrahim et al. 2019).

Our approach falls under the class of perturbation-based methods for model explanation. Some methods, such as Feature Occlusion (Zeiler and Fergus 2014) and CXPlain (Schwab and Karlen 2019), perturb features by setting their values to zero. Our approach is most similar to permutation-based feature importance methods. Breiman (2001) uses permutations to identify important features in random forests, and many variants of feature importance using permutations have since been studied (Altmann et al. 2010; Fisher et al. 2019; Gregorutti et al. 2015; Ojala and Garriga 2010; Strobl et al. 2008). The simplicity and generality of permutations makes them attractive as a tool for model-agnostic explanation. While existing methods focus on permutations of features as part of a tabular representation, we extend permutation-based feature importance to temporal models.

In this work, we propose Temporal Importance Model Explanation (TIME), a method for explaining temporal black-box models. Our approach is model-agnostic, produces global explanations, and elicits specific properties of temporal models. It takes as input a learned model over features representing sequences or time-series, and a test data set used to analyze the model, and does the following: (i) it identifies features that are important for the model's predictions across the

distribution of instances, (ii) for each such feature, it identifies the most important temporal window that the model focuses on, (iii) it determines whether the model's predictions are dependent on the ordering of the values within the window, (iv) it uses hypothesis testing and a false discovery rate control methodology to identify important features and their temporal properties with statistical rigor, and (v) it treats the model as a black-box and thus may be used to analyze a variety of temporal models. There are many applications that match the setting we address, such as numerous clinical risk approaches that make predictions relative to an index time: hospital readmission, inpatient deterioration, post-hospitalization complications, post-surgical complications, and asthma exacerbations, among others (Ashfaq et al. 2019; Cobian et al. 2020; Kawaler et al. 2012; Mayampurath et al. 2019; Xue et al. 2021). Figure 5.1 illustrates the setting and our approach.

5.2 Methods

5.2.1 Identifying Important Features/Timesteps via Permutation

Non-temporal models. We first outline the case of a model trained on a tabular data set where each feature takes a single value per instance. Consider a model f over D features, trained to predict a target y. We are interested in examining the importance of a given feature j for the model in predicting y. We assume that a test set comprising M instances is available to analyze the model's generalization performance. Let $(\mathbf{x}^{(i)}, y^{(i)})$ be the i^{th} instance-target pair, and $\mathcal{L}$ be a loss function linking the model output $f(\mathbf{x})$ to the target y. The *perturbed* output of the model for instance i w.r.t feature j and another instance $l \neq i$ is given by:

$$f\left(\mathbf{x}_j^{(i,l)}\right) = f\left(x_1^{(i)}, x_2^{(i)}, \ldots, x_j^{(l)}, \ldots, x_D^{(i)}\right) \tag{5.1}$$

where the value of feature j is replaced by its corresponding value from instance l, as shown in Figure 5.2a. Then, we can compute the change in loss between the

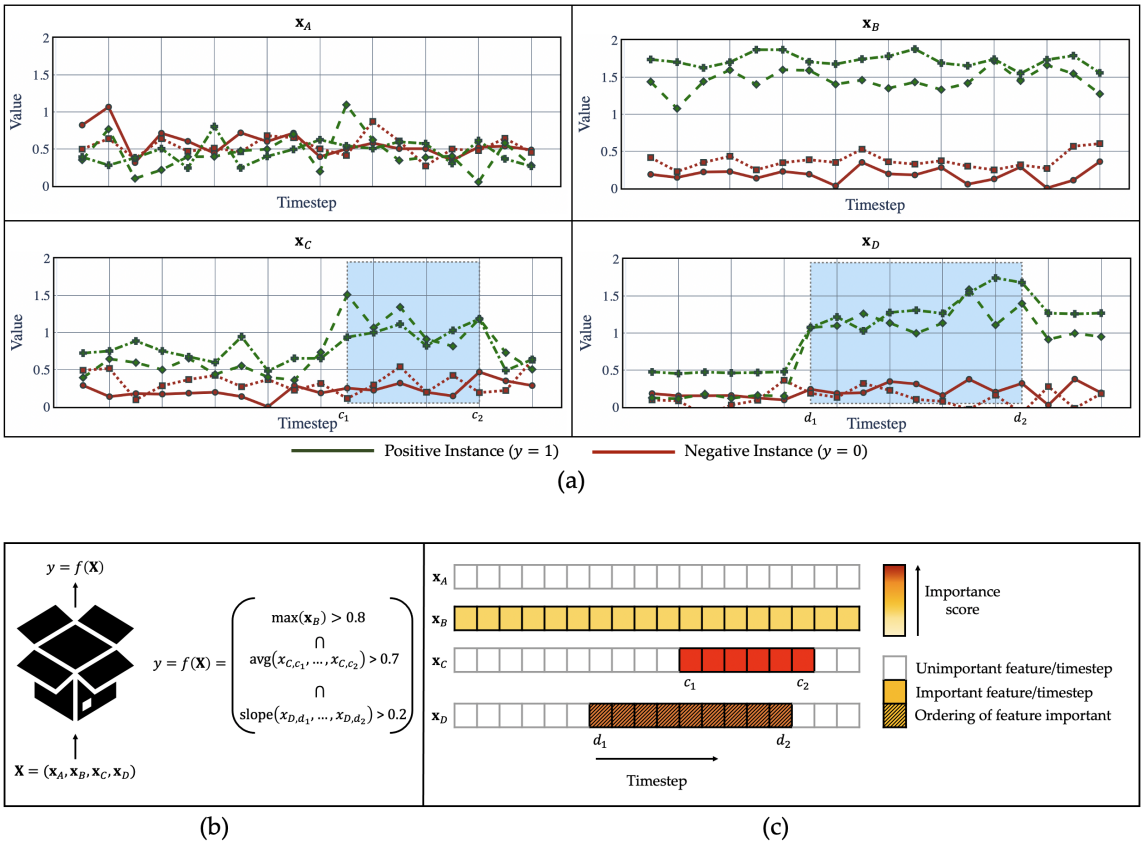


Figure 5.1: An illustration of the task addressed by TIME. (a) Time series for positive (green) and negative (red) instances for four different features, showing temporal properties of the features that a learned model may capture. (b) A trained binary classification model over the four time-varying features, whose underlying function uses the features' temporal properties to capture the target concept. $\mathbf{x}_A$ is not used by the model; all timesteps for $\mathbf{x}_B$ are equally important; the model focuses on windows $[c_1, c_2]$ and $[d_1, d_2]$ for $\mathbf{x}_C$ and $\mathbf{x}_D$ respectively; the ordering of values is important only for $\mathbf{x}_D$. (c) The output of TIME, showing for each feature (i) its overall importance to the model, (ii) the most important window that the model focuses on, and (iii) whether the ordering of the values within the window is important to the model.

perturbed and original losses as:

$$\Delta\mathcal{L}_j^{(i,l)} = \mathcal{L}\left[y^{(i)}, f\left(\mathbf{x}_j^{(i,l)}\right)\right] - \mathcal{L}\left[y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right]. \tag{5.2}$$

Let $\Pi = \langle \pi_1, \pi_2, \ldots, \pi_M \rangle$ be a permutation of the data set sampled from a set of permutations $\mathcal{P}_j$, so that feature j is sampled from instance $l = \pi_i$ for each instance i. Averaging over all instances $i = 1 \ldots M$ and $|\mathcal{P}_j|$ permutations of the data set, we compute the importance score of feature j as:

$$I\left(f, j\right) = \frac{1}{|\mathcal{P}_j|} \sum_{\Pi \in \mathcal{P}_j} \left[\frac{1}{M} \sum_{i=1}^{M} \Delta\mathcal{L}_j^{(i,\pi_i)}\right]. \tag{5.3}$$

A model includes many features, all of which may have some effect on the model's output, but only some of which may be useful in predicting the target. We consider a feature to be important if the model's performance degrades on average when the feature is perturbed via permutation, a notion that is captured by focusing on the model loss rather than the model output. We use hypothesis testing to test the significance of this degradation, as outlined in Section 5.2.4.

Temporal models. We extend the idea of permutation-based feature importance to temporal models. Here, we assume that each feature is represented by a time series of length L, so that the data is represented by an $M \times D \times L$ tensor, with instance i represented by a matrix $\mathbf{X}^{(i)}$ and feature j of instance i by a time series $\mathbf{x}_j^{(i)} = \left\langle x_{j,1}^{(i)}, x_{j,2}^{(i)}, \ldots, x_{j,k}^{(i)}, \ldots x_{j,L}^{(i)} \right\rangle$.

By unrolling in time, this may be viewed as tabular data consisting of M instances and $D \cdot L$ features, so that permutations of individual features in the tabular setting correspond to permutations of individual timesteps in the temporal setting. However, doing so ignores the temporal structure of the data and correlations within time series. Thus, we consider joint permutations of contiguous regions, i.e., windows, in time. Given a time window $[k_1, k_2]$, the perturbed output of the model

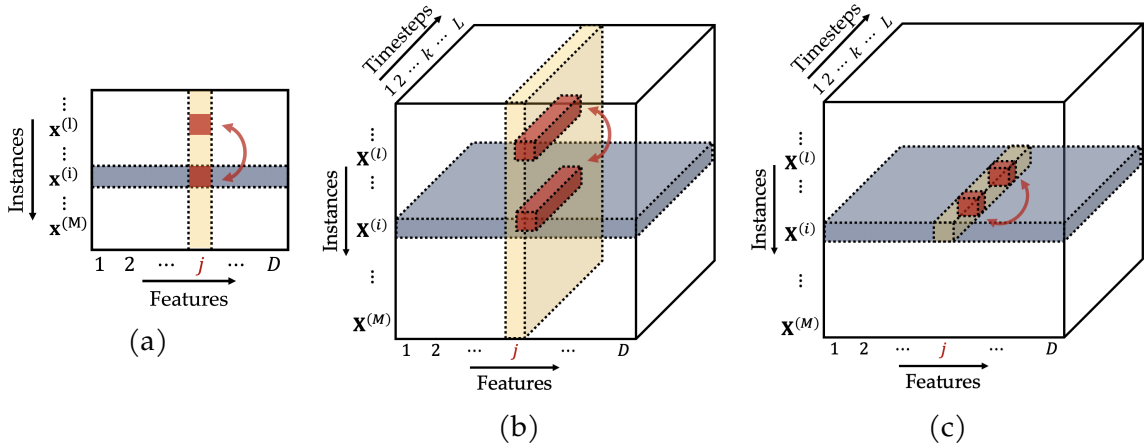


Figure 5.2: Perturbation for instance i and feature j to compute feature importance. (a) Data matrix showing the replacement of the value of feature j in instance i from instance l. (b) Data tensor showing the replacement of a window of feature j in instance i from the corresponding window of instance l . (c) Data tensor showing the exchange of values at two timesteps within the same time series $\mathbf{x}_j^{(i)}$.

for instance i w.r.t. feature j is given by:

$$f\left(\mathbf{X}^{(i,l)}_{j,[k_1,k_2]}\right) = f\left(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}^{(i,l)}_{j,[k_1,k_2]}, \ldots, \mathbf{x}_D^{(i)}\right) \tag{5.4}$$

where $\mathbf{x}^{(i,l)}_{j,[k_1,k_2]}$ is the time series for instance i and feature j with timesteps in the window $[k_1, k_2]$ replaced by the corresponding window from another instance $l \neq i$, as shown in Figure 5.2b.

$$\mathbf{x}^{(i,l)}_{j,[k_1,k_2]} = \left\langle x_{j,1}^{(i)}, x_{j,2}^{(i)}, \ldots, x_{j,k_1}^{(l)}, \ldots, x_{j,k_2}^{(l)}, \ldots, x_{j,L}^{(i)} \right\rangle . \tag{5.5}$$

We compute the perturbed loss $\mathcal{L}\left[y^{(i)}, f\left(\mathbf{X}^{(i,l)}_{j,[k_1,k_2]}\right)\right]$ and the change in loss (Equation 5.2) for instance i. We average this over all instances $i = 1 \ldots M$ and $|\mathcal{P}_j|$ permutations of the data set to compute the importance score corresponding to the

window $[k_1, k_2]$ for feature j:

$$I\left(f, j, [k_1, k_2]\right) = \frac{1}{|\mathcal{P}_j|} \sum_{\Pi \in \mathcal{P}_j} \left[\frac{1}{M} \sum_{i=1}^{M} \Delta \mathcal{L}_{j,[k_1,k_2]}^{(i,\pi_i)} \right]. \tag{5.6}$$

The overall importance score $I\left(f, j, [1, L]\right)$ of feature j is computed by selecting $k_1 = 1$ and $k_2 = L$.

5.2.2 Identifying Important Windows

Given that the features have an explicit temporal structure, we want to localize the timesteps that the model may be focusing on. We assume that for a given feature j, there exists an underlying contiguous time window $W^* = [k_1, k_2] : 1 \leq k_1 < k_2 \leq L$, so that most of the effect of perturbing j derives from W^*. Specifically, we consider a partitioning of the sequence into three windows: *prior* window $W_P = [1, k_1 - 1]$, important window $W^* = [k_1, k_2]$, and *subsequent* window $W_S = [k_2 + 1, L]$ where W_P are W_S both have low importance and a size of zero or more timesteps. In order to pin down the most salient timesteps, we want to find the largest W_P and W_S that satisfy:

$$I(f, j, \tilde{W}) < \left(\frac{1 - \gamma}{2} \right) I\left(f, j, [1, L]\right) \tag{5.7}$$

where $\gamma : \ 0 < \gamma < 1$ controls the degree to which the model focuses on W^* and affects the size of the identified windows. We use a binary search algorithm to identify W_P and W_S, and by exclusion, identify the important window W^*. We start with an initial estimate $\hat{W}_P = [1, \hat{k}_1]$ with $\hat{k}_1 = \frac{L}{2}$. We then perturb $\hat{W}_P$ and observe its importance score $I(f, j, \hat{W}_P)$. If $\hat{W}_P$ contains important timesteps, its importance score is likely to be inflated due to the breakage of correlations between all timesteps of the important window, i.e., predictors strongly associated with the response (Nicodemus et al. 2010), leading the search algorithm to contract $\hat{W}_P$ to exclude these timesteps. On the other hand, if $\hat{W}_P$ has a low importance score that satisfies Equation 5.7, we expand it unless doing so would violate this condition. We expand or contract $\hat{W}_P$ by updating $\hat{k}_1$ and repeat the perturbation until we

find the largest $\hat{W}_P$ that satisfies Equation 5.7, and set $k_1 = |\hat{W}_P| + 1$.

Similarly, to identify k_2, we start from an initial estimate $\hat{W}_S = [\hat{k}_2, L]$ with $\hat{k}_2 = k_1 + 1$, measure its importance score, and iteratively expand or contract it under the constraint $\hat{k}_2 > k_1$, until we identify the largest $\hat{W}_S$ that satisfies Equation 5.7. We select the final boundary estimates k_1 and $k_2 = L - |\hat{W}_S|$ to characterize the important window W^*. We then compute its importance score using Equation 5.6 and use hypothesis testing to test its significance. We note that importance scores are not additive in general, and W^* is not guaranteed to satisfy $I(f, j, W^*) > \gamma I(f, j, [1, L])$.

5.2.3 Identifying the Importance of Feature Ordering

To examine how a feature's ordering affects the model's performance, we consider permutations of timesteps within its time series. Such permutations have previously been used to detect circadian patterns in gene expression data (Ptitsyn et al. 2006; Storch et al. 2002). To determine the importance of the ordering of a feature j within a window $[k_1, k_2]$, we permute its values within the window, as illustrated in Figure 5.2c, and average across instances. Let $\Pi_{[k_1,k_2]} = \langle \pi_{k_1}, \pi_{k_1+1}, \ldots, \pi_{k_2} \rangle$ be a permutation over timesteps within the window. The perturbed model output is given by:

$$f\left(\mathbf{X}^{(i)}_{j,\Pi_{[k_1,k_2]}}\right) = f\left(\mathbf{x}^{(i)}_1, \mathbf{x}^{(i)}_2, \ldots, \mathbf{x}^{(i)}_{j,\Pi_{[k_1,k_2]}}, \ldots, \mathbf{x}^{(i)}_D\right) \tag{5.8}$$

where the permuted time series for instance i and feature j is given by:

$$\mathbf{x}^{(i)}_{j,\Pi_{[k_1,k_2]}} = \langle x^{(i)}_{j,1}, \ldots, x^{(i)}_{j,k_1-1}, x^{(i)}_{j,\pi_{k_1}}, \ldots, x^{(i)}_{j,\pi_{k_2}}, x^{(i)}_{j,k_2+1}, \ldots x^{(i)}_{j,L} \rangle \tag{5.9}$$

As before, we compute the average change between the perturbed and original losses over all instances i and multiple permutations $\Pi_{[k_1,k_2]}$, and use hypothesis testing to test the significance of the change.

5.2.4 Hypothesis Testing and False Discovery Rate Control

Existing work has leveraged hypothesis testing in conjunction with permutations to examine black-box models (Burns et al. 2020; Golland et al. 2005; Ojala and Garriga 2010; Tansey et al. 2019). Our previous work (Chapter 4) uses hypothesis testing to test feature groups in addition to base features, but employs the Wilcoxon signed-rank test, which makes assumptions about the null distribution of the test statistic that may be inappropriate for certain perturbations or models. Here, we perform hypothesis testing using permutation tests, a type of widely used, non-parametric, exact statistical test that makes few assumptions about the null distribution. We use permutation tests to assess the significance of important features and windows as well as time series ordering.

We use importance scores to quantify the degree to which permuting features degrades the model's performance, and use hypothesis testing to test the statistical significance of this degradation. Specifically, we use the formulation of permutation tests in Ojala and Garriga (2010), using the mean loss as the test statistic. The one-sided empirical p-value for feature j is given by:

$$\hat{p} = \frac{|\left\{\Pi \in \mathcal{P}_j : \bar{\mathcal{L}}_\Pi \leq \bar{\mathcal{L}}\right\}| + 1}{|\mathcal{P}_j| + 1} \tag{5.10}$$

where $\mathcal{P}_j$ is a set of permutations of the original data with feature j permuted in some way, $\bar{\mathcal{L}}$ is the mean loss for the original data, and $\bar{\mathcal{L}}_\Pi$ is the mean loss for permuted data. By repeatedly permuting the data, we generate the empirical null distribution of the test statistic (mean loss). The null hypothesis is that the effect of the feature on the model's loss is zero when averaged across instances, so that the test statistic on the original data set comes from this distribution. When the one-sided p-value is sufficiently small, we conclude that permuting the feature degrades the model's performance. This approach may also be used to detect overfitted features by reversing the inequality in Equation 5.10.

Depending on the permuted quantity, we can use Equation 5.10 to test the overall importance, window importance, and ordering importance of feature j.

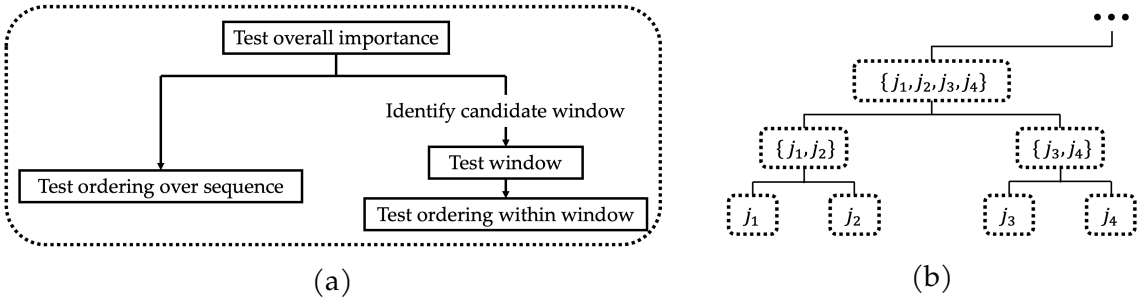


Figure 5.3: (a) A hierarchy of tests used to check a given feature for its (i) overall importance, (ii) important window and (iii) the importance of ordering within the window. (b) A hierarchy over the features, where each node is tested using the testing hierarchy shown in (a). Feature groups are tested via joint permutations of their constituent features. Hierarchical FDR control is used for multiple testing correction, and subtrees rooted at nodes with p-values above a threshold are pruned.

These tests may be organized as a hierarchy, as shown in Figure 5.3a, so that a test is performed only if its parent test returns a significant p-value.

The multiplicity of hypothesis tests for a given feature and across the set of features leads to a multiple comparisons problem. We address this by using a hierarchical false discovery rate (FDR) control methodology (Yekutieli 2008), with the FDR for sibling tests controlled using the Benjamini-Hochberg procedure (Benjamini and Hochberg 1995). This approach also readily extends to features arranged in a hierarchy in order to interpret models in terms of feature groups, as shown in Figure 5.3b.

5.2.5 Rationale for Permutation-based Feature Importance

In this section, we further discuss the rationale behind the permutation-based feature importance approach used by TIME. We elucidate connections of the importance scores to existing permutation-based feature importance measures. We then discuss issues due to out-of-distribution sampling by feature importance methods, including TIME. Finally, we provide an illustrative example showing how our approach may be used to discover relevant features and windows even in the presence

of out-of-distribution sampling and feature correlations.

Equation 5.3 computes the importance score of feature j for a tabular model, and Equation 5.6 computes the importance score for window $[k_1, k_2]$ of feature j for a temporal model. Intuitively, Equation 5.6 captures the notion that a window for a given feature is considered important to the model if it has a positive association with the target, in the sense that perturbing its values via permutation increases the model's loss on average across the distribution of instances. We use permutation tests to assess the statistical significance of this degradation, and compute the empirical *p*-value for feature j using Equation 5.10.

Permutations serve two purposes in our approach: (i) to compute the importance score for a feature j (Equation 5.6), and (ii) to test the significance of feature j using permutation tests (Equation 5.10). While numerous works have examined permutation-based feature importance scores, e.g., Breiman (2001), Fisher et al. (2019), Gregorutti et al. (2015), Henelius et al. (2014), and Strobl et al. (2008), and some works have used hypothesis testing based on permutation tests to assess the statistical significance of important features (Burns et al. 2020; Golland et al. 2005; Ojala and Garriga 2010), combining the two approaches is a novel aspect of our work. In particular, it allows us to improve the value of the explanations (by examining the relative importance of features and windows, rather than just performing feature selection) while providing them with statistical rigor (by using hypothesis testing and controlling for false discoveries). By choosing the mean loss as the test statistic for permutation tests, we can leverage the same computation for both the importance score of a feature and the test for its statistical significance.

5.2.5.1 Theoretical Properties

The importance score computed by TIME (Equation 5.6) is closely related to *model reliance*, a formalization of permutation-based feature importance measures by Fisher et al. (2019) based on the approach to examine feature importance for random forests introduced by Breiman (2001). The *difference-based* model reliance

(MR_{diff}) of a tabular model f on a feature group S is defined as:

$$\begin{aligned} MR_{\text{diff}}(f, S) &= [\text{Expected loss of } f \text{ under noise}] - [\text{Expected loss of } f \text{ without noise}] \\ &= \mathbb{E}\left[\mathcal{L}\left[Y, f(\tilde{\mathbf{X}})\right]\right] - \mathbb{E}\left[\mathcal{L}\left[Y, f(\mathbf{X})\right]\right] \end{aligned}$$

where $\mathbf{X}$ represents the original features and $\tilde{\mathbf{X}}$ represents $\mathbf{X}$ with noise added to the subset $\mathbf{X}_S$ that renders it completely uninformative of the target Y without altering the marginal distribution of features in S. Then, Equations 5.3 and 5.6 correspond to empirical estimates of difference-based model reliance for tabular and temporal models respectively, with S representing a feature for tabular models and a window $[k_1, k_2]$ for a feature for temporal models (generalizable to feature groups in both cases).

Interested readers should refer to Fisher et al. (2019) for a detailed formal treatment of model reliance and related permutation-based feature importance measures, and Gregorutti et al. (2015) for a study of an equivalant formulation of MR_{diff} in the presence of feature correlations. In Proposition 5.1, we use results for model reliance to show that for additive models under certain assumptions, Equation 5.6 is a measure of the positive association between the relevant window of a feature j and the target.

Proposition 5.1. *Let $\mathbf{X}_1 \dots \mathbf{X}_D$ be independent random vectors of size L representing temporal features $\mathcal{F} = \{1, 2, \dots, D\}$ for the additive temporal model $f(\mathbf{X}_1, \dots \mathbf{X}_D) = \sum_{j' \in \mathcal{F}} g_{j'}(\mathbf{X}_{j'})$. Let W be a window for $\mathbf{X}_j$ perturbed according to Equation 5.5, and let $\overline{W}$ represent the timesteps outside the window. Further, assume that $g_j(\mathbf{X}_j)$ decomposes additively over the sequence as:*

$$g_j(\mathbf{X}_j) = g_{j,W}(\mathbf{X}_{j,W}) + g_{j,\overline{W}}(\mathbf{X}_{j,\overline{W}})$$

where $\mathbf{X}_{j,W}$ and $\mathbf{X}_{j,\overline{W}}$ represent subsequences of $\mathbf{X}_j$ inside and outside the window respectively, and $g_{j,W}(\mathbf{X}_{j,W})$ and $g_{j,\overline{W}}(\mathbf{X}_{j,\overline{W}})$ represent arbitrary feature functions over these subsequences. Let Y be the target and $\mathcal{L}$ be the quadratic loss function.

Then, the importance score $I(f, j, W)$ of the window W for feature $\mathbf{X}_j$ as computed by Equation 5.6 satisfies:

$$\mathbb{E}\left[I(f,j,W)\right] = 2\left[\operatorname{cov}\left(Y, g_{j,W}(\mathbf{X}_{j,W})\right) - \operatorname{cov}\left(g_{j,W}(\mathbf{X}_{j,W}), g_{j,\overline{W}}(\mathbf{X}_{j,\overline{W}})\right)\right]. \quad (5.11)$$

Proof. Consider an additive tabular model $f = \sum_{j' \in \mathcal{F}} g_{j'}(X_{j'})$ composed of univariate functions $g_{j'}(X_{j'})$ over tabular features $\mathbf{X}$. Assuming a quadratic loss function $\mathcal{L}$, Proposition 15 from Fisher et al. (2019) [1] gives MR_{diff} of model f on feature group S as:

$$MR_{\text{diff}}(f, S) = 2\left[\operatorname{cov}\left(Y, \mathbf{g}_S(\mathbf{X}_S)\right) - \operatorname{cov}\left(\mathbf{g}_{\bar{S}}(\mathbf{X}_{\bar{S}}), \mathbf{g}_S(\mathbf{X}_S)\right)\right] \quad (5.12)$$

where $\mathbf{X}_S$ and $\mathbf{X}_{\bar{S}}$ correspond to features inside and outside the feature group S respectively, and $\mathbf{g}_S$ and $\mathbf{g}_{\bar{S}}$ are vector-valued functions over $\mathbf{X}_S$ and $\mathbf{X}_{\bar{S}}$ respectively. Since the importance score in Equation 5.3 is an empirical estimate of MR_{diff} for tabular models, we have $\mathbb{E}\left[I(f, S)\right] = MR_{\text{diff}}(f, S)$. Extending to the case of a temporal model f under the assumptions of the proposition, the expected importance score of feature j in window W is given by:

$$\begin{aligned}
\mathbb{E}\left[I(f,j,W)\right] &= 2\Bigg[\operatorname{cov}\left(Y, g_{j,W}(\mathbf{X}_{j,W})\right) - \sum_{j' \in \mathcal{F}\setminus j} \operatorname{cov}\left(g_{j,W}(\mathbf{X}_{j,W}), g_{j'}(\mathbf{X}_{j'})\right) \\
&\qquad - \operatorname{cov}\left(g_{j,W}(\mathbf{X}_{j,W}), g_{j,\overline{W}}(\mathbf{X}_{j,\overline{W}})\right)\Bigg] \\
&= 2\left[\operatorname{cov}\left(Y, g_{j,W}(\mathbf{X}_{j,W})\right) - \operatorname{cov}\left(g_{j,W}(\mathbf{X}_{j,W}), g_{j,\overline{W}}(\mathbf{X}_{j,\overline{W}})\right)\right] \quad (5.13)
\end{aligned}$$

since $\operatorname{cov}\left(g_{j,W}(\mathbf{X}_{j,W}), g_{j'}(\mathbf{X}_{j'})\right) = 0 \;\forall\; \bar{j} \in \mathcal{F}\setminus j$ as $\mathbf{X}_1, \ldots \mathbf{X}_D$ are assumed independent. □

[1] Gregorutti et al. (2015) and Gregorutti et al. (2017) show equivalent formulations under stronger assumptions.

Corollary 5.1. *Let feature j have a relevant window W^*, so that there is no association between the timesteps outside the window and the target, i.e.,* $\text{cov}\left(Y, g_{j,\overline{W}^*}(\mathbf{X}_{j,\overline{W}^*})\right) = 0$. *Then, the expected importance score for the entire sequence for feature j is given by:*

$$\begin{aligned}\mathbb{E}\left[I(f, j, [1, L])\right] &= 2\ \text{cov}\left(Y, g_j(\mathbf{X}_j)\right) \\ &= 2\ \text{cov}\left(Y, g_{j,W^*}(\mathbf{X}_{j,W^*})\right) + 2\ \text{cov}\left(Y, g_{j,\overline{W}^*}(\mathbf{X}_{j,\overline{W}^*})\right) \\ &= 2\ \text{cov}\left(Y, g_{j,W^*}(\mathbf{X}_{j,W^*})\right). \end{aligned} \tag{5.14}$$

Thus, the expected importance score for feature j is a measure of the positive association between the target and its relevant window. Moreover, when the timesteps inside and outside the window are weakly correlated, i.e., $\text{cov}\left(g_{j,W^*}(\mathbf{X}_{j,W^*}), g_{j,\overline{W}^*}(\mathbf{X}_{j,\overline{W}^*})\right) \approx 0$, then using Equations 5.13 and 5.14, we have $\mathbb{E}\left[I(f, j, [1, L])\right] \approx \mathbb{E}\left[I(f, j, W^*)\right]$. In other words, the expected importance score of the entire sequence for feature j is approximately equal to the expected importance score of its relevant window W^*, motivating the window search algorithm (Section 5.2.2).

5.2.5.2 Out-of-distribution Sampling

Since permutations of a given feature break correlations that may exist between that feature and other features, the model may end up being evaluated on 'out-of-distribution' samples that are not representative of the underlying distribution of data in the domain. This is a common concern with many feature importance methods (Kumar et al. 2020), and may lead to distorted importance scores for correlated features (Gregorutti et al. 2017). We note that methods that perform perturbations using reference values, such as feature occlusion (Zeiler and Fergus 2014) and CXPlain (Schwab and Karlen 2019), may also evaluate the model on out-of-distribution samples, and tractable approximations of many Shapley-value-based methods, including KernelSHAP (Lundberg and Lee 2017) and SAGE (Covert et al. 2020b) use marginal in place of conditional distributions, leading to out-of-distribution sampling.

For permutation-based feature importance measures, certain strategies may be employed to mitigate the distortion of importance scores due to out-of-distribution sampling. One approach is to induce approximate independence between the tested and held-out features, using (i) backward selection (Gregorutti et al. 2017), when the goal of feature importance is to perform feature selection, or (ii) feature hierarchies (Chapter 4), when the goal of feature importance is model explanation, which is the case for TIME.

A second approach to avoid out-of-distribution sampling is to use permutations sampled from distributions conditioned on the held-out features (Strobl et al. 2008). However, this is computationally intractable in general, so that significant approximation is necessary, and may also produce importance scores that may (i) differ between equally relevant groups of correlated and uncorrelated features (Nicodemus et al. 2010), and (ii) change based on the choice of correlated features included in the model (Kumar et al. 2020).

5.2.5.3 An Illustrative Example

Using results for model reliance (Equation 5.12), we show an illustrative example where our approach is able to detect the importance of features and windows accurately, even in the presence of out-of-distribution sampling and feature correlations.

Tabular models. Consider an additive model over four features: $f(\mathbf{X}) = \sum_{j=1}^{4} g_j(X_j)$, where X_j is a random variable corresponding to feature j and $g_j(X_j)$ is a univariate function over feature j. Let $h_j(X_j)$ be the standardized version of $g_j(X_j)$, so that $g_j(X_j) = \sigma_j h_j(X_j) + \mu_j$, where μ_j and σ_j represent the mean and standard deviation of $g_j(X_j)$ respectively. Then, $f = \sum_{j=1}^{4} \sigma_j h_j(X_j) + c$ where σ_j may be interpreted as the weight assigned to feature j by the model, and c is a constant. We use g_j and h_j as shorthand for $g_j(X_j)$ and $h_j(X_j)$ respectively.

Let features 1 and 2 be relevant and have covariance ρ with the target, and let features 3 and 4 be irrelevant. Additionally, let features 1 and 2 be highly correlated with each other with covariance ρ_{R}, let features 3 and 4 be highly correlated with each other with covariance ρ_{I}, and let the relevant and irrelevant features be weakly

correlated with each other with covariance ρ_{RI}. Namely:

$$\text{cov}(Y, \mathbf{h}) = \begin{bmatrix} \rho \\ \rho \\ 0 \\ 0 \end{bmatrix} \qquad \text{cov}(\mathbf{h}, \mathbf{h}) = \begin{bmatrix} 1 & \rho_{\text{R}} & \rho_{\text{RI}} & \rho_{\text{RI}} \\ \rho_{\text{R}} & 1 & \rho_{\text{RI}} & \rho_{\text{RI}} \\ \rho_{\text{RI}} & \rho_{\text{RI}} & 1 & \rho_{\text{I}} \\ \rho_{\text{RI}} & \rho_{\text{RI}} & \rho_{\text{I}} & 1 \end{bmatrix}$$

To examine the effect of feature correlations on the importance scores, we select specific values for the coefficients. Let $c = 0$, $\sigma_1 = \sigma_2 = 0.9$, and $\sigma_3 = \sigma_4 = 0.1$, so that $f = 0.9h_1 + 0.9h_2 + 0.1h_3 + 0.1h_4$, i.e., the model places the highest weight on relevant features 1 and 2 but also incorporates irrelevant features 3 and 4. Let $\rho = \rho_{\text{R}} = \rho_{\text{I}} = 0.9$ and $\rho_{\text{RI}} = 0.1$.

The expected importance scores, i.e., model reliances, for different features and feature groups may be computed using Equation 5.12. The model reliance of f on feature 1 is given by:

$$\begin{aligned} MR_{\text{diff}}(f, \{1\}) &= 2\left[\text{cov}\left(Y, g_1\right) - \text{cov}\left(g_2, g_1\right) - \text{cov}\left(g_3, g_1\right) - \text{cov}\left(g_4, g_1\right)\right] \\ &= 2\sigma_1\left[\text{cov}\left(Y, h_1\right) - \sigma_2\text{cov}\left(h_2, h_1\right) - \sigma_3\text{cov}\left(h_3, h_1\right) - \sigma_4\text{cov}\left(h_4, h_1\right)\right] \\ &= 2\sigma_1\left[\rho - \sigma_2\rho_{\text{R}} - \sigma_3\rho_{\text{RI}} - \sigma_4\rho_{\text{RI}}\right] \\ &= 2 \cdot 0.9 \cdot \left[0.9 - 0.9 \cdot 0.9 - 0.1 \cdot 0.1 - 0.1 \cdot 0.1\right] \\ &= 0.126 \end{aligned}$$

Similarly, we can compute the model reliance of f on all features and feature groups:

$$MR_{\text{diff}}(f, \{2\}) = MR_{\text{diff}}(f, \{1\}) = 0.126$$
$$MR_{\text{diff}}(f, \{3\}) = MR_{\text{diff}}(f, \{4\}) = -0.054$$
$$MR_{\text{diff}}(f, \{1, 2\}) = 3.168$$
$$MR_{\text{diff}}(f, \{1, 3\}) = MR_{\text{diff}}(f, \{1, 4\}) = MR_{\text{diff}}(f, \{2, 3\}) = MR_{\text{diff}}(f, \{2, 4\}) = 0.108$$
$$MR_{\text{diff}}(f, \{3, 4\}) = -0.072$$
$$MR_{\text{diff}}(f, \{1, 2, 3\}) = MR_{\text{diff}}(f, \{1, 2, 4\}) = 3.186$$
$$MR_{\text{diff}}(f, \{1, 3, 4\}) = MR_{\text{diff}}(f, \{2, 3, 4\}) = 0.126$$
$$MR_{\text{diff}}(f, \{1, 2, 3, 4\}) = 3.24$$

This leads us to make the following observations:

1. The expected importance scores of feature groups that include all relevant features, i.e., $\{1, 2\}$, $\{1, 2, 3\}$, $\{1, 2, 4\}$, and $\{1, 2, 3, 4\}$ are approximately the same.
2. The sum of expected importance scores of the relevant features (0.252) does not add up to the expected importance score of the relevant feature group (0.3168) due to the correlation between the features. This is consistent with observations made by other authors (Gregorutti et al. 2015; Toloşi and Lengauer 2011).
3. The expected importance scores of features and feature groups composed entirely of irrelevant features, i.e., $\{3\}$, $\{4\}$, and $\{3, 4\}$ have the lowest magnitude.

In summary, out-of-distribution sampling caused by marginal permutations of the features may break feature correlations and distort importance scores and rankings, but permuting feature groups in addition to base features can ameliorate this problem. Since enumerating and permuting all feature groups is intractable for a large feature set, we leverage feature hierarchies to identify feature groups to permute. The hierarchies may represent groupings of conceptually related features derived from domain knowledge, or they may be generated by pre-processing the data, such as using hierarchical clustering (Chapter 4). For the current example, Figure 5.4 shows a feature hierarchy that groups highly correlated features together, thus allowing an accurate assessment of importance scores for relevant features at

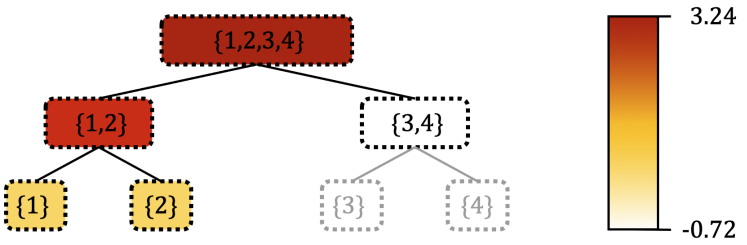


Figure 5.4: A hierarchy over four features for the illustrative example in Section 5.2.5.3, with highly correlated features grouped together. Darker shades indicates higher importance scores, and gray shades indicate pruned tests.

the group level, as well as pruning tests for importance of irrelevant features.

Temporal models. We extend this example to temporal models in order to illustrate the search algorithm used to identify the relevant window for a feature (Section 5.2.2). Instead of representing a tabular model over four features, let f represent an additive temporal model over D independent temporal features having four timesteps each. Namely, let $f(\mathbf{X_1}, \ldots, \mathbf{X_D}) = \sum_{j'=1}^{D} g_{j'}(\mathbf{X}_{j'}) = \sum_{j'=1}^{D} \sum_{t=1}^{4} g_{j',t}(X_{j',t})$. Consider a feature j with each timestep $X_{j,t}$ mapped to tabular feature $X_t : t \in \{1, 2, 3, 4\}$, so that the model reliance calculations and observations from the previous discussion still hold but apply to timesteps instead of features.

Figure 5.5 uses the calculated expected importance scores to walk through the steps of the search algorithm used to identify the relevant window for feature j. A contiguous relevant window may be located at either the edge (Figure 5.5a) or the center of the sequence (Figure 5.5b). At each step of the search, the algorithm expands or contracts $\hat{W}_P$ or $\hat{W}_S$ in order to identify the largest prior and subsequent windows that satisfy Equation 5.7, with γ chosen as 0.95. In both cases, the resulting window includes all the relevant timesteps and excludes all the irrelevant timesteps.

Thus, the window search algorithm can leverage joint permutations of timesteps and correctly identify the relevant window, even in the presence of correlated timesteps. While this example uses a univariate function $g_{j,t}(X_{j,t})$ for each timestep in the sequence, the analysis generalizes to multivariate functions $g_{j,W}(X_{j,W})$ over windows of correlated timesteps, potentially including interaction effects. This is

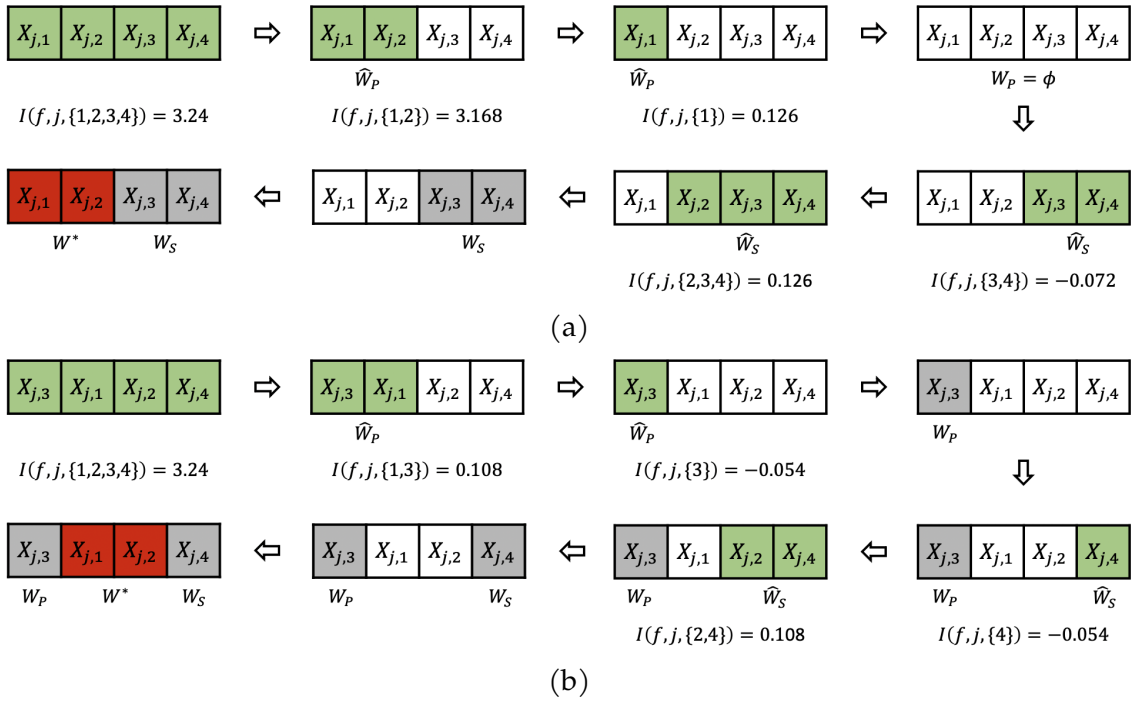


Figure 5.5: Illustration of the window search algorithm for (a) a relevant window comprising the first two timesteps of a feature, and (b) a relevant window comprising the second and third timesteps of a feature. Held-out timesteps are represented in white, permuted timesteps are represented in green, estimated prior and subsequent windows are represented in gray, and the estimated important window is represented in red. The top row for each figure represents the search for the prior window, and the bottom row represents the search for the subsequent window. Expected importance scores are shown below the sequence at each step of the search.

the case for the synthetic model that we use to evaluate our approach in Section 5.3.1. Moreover, as noted in Section 5.2.4, the hierarchical FDR approach used to organize tests of temporal properties can readily be integrated with feature hierarchies, as shown experimentally in Section 5.3.2.

5.2.6 Computational Details

Choice of data set. The classical approach for permutation-based feature importance uses out-of-bag (OOB) samples to examine the model, and avoids in-sample importance estimation (Breiman 2001). However, Fisher et al. (2019) show uniform bounds on the model reliance estimation error for all models in a sufficiently regularized class of models, so that it is possible to train a model and to analyze its feature importance using the same data. For our experiments, we use a validation data set to analyze models trained on real data.

Number of permutations. The size of $\mathcal{P}_j$ (the set of permutations used to examine feature j) may be selected based on available computational resources, the desired precision, or using sequential probability ratio tests (Ojala and Garriga 2010). In our experiments, we select $|\mathcal{P}_j| = 50$ for synthetic data and $|\mathcal{P}_j| = 200$ for real data, but empirically observe minor differences in results beyond $|\mathcal{P}_j| = 20$.

Time complexity. Marginal permutation-based feature importance methods such as TIME are considered an efficient tool for ascertaining feature importance (Gregorutti et al. 2015). The time complexity of our method is $\mathcal{O}(MPL \min\{R \log L, D\})$, where M, D, L, R and $P = |\mathcal{P}_j|$ are the test-set size, number of features, sequence length, number of relevant features and number of permutations respectively. The logarithmic term arises since the window search algorithm partitions the search space in half at each step of the search. In practice, as shown in Table 5.1, TIME is often significantly faster than univariate permutations, since permutations of windows can be performed faster than permutations of their constituent timesteps using a vectorized implementation.

Distributed computing. TIME can be computed significantly faster by leveraging a distributed computing environment, where each node analyzes a feature or a subset of features. Our implementation of TIME uses HTCondor for distributed computing. For a fair comparison, the distributed implementation is disabled while comparing running times of different baseline methods, as shown in Table 5.1.

5.3 Results

We evaluate TIME by analyzing synthetic data sets and models where the ground truth pertaining to relevant features and their temporal properties is known, and by analyzing a long short term memory (LSTM) model (Hochreiter and Schmidhuber 1997) trained to predict in-hospital mortality from intensive care unit (ICU) data.

5.3.1 Synthetic Data Sets and Models

We create synthetic time-series data where we control the generating processes for different features. A set of *feature functions* operate on windows for each feature and are used to generate targets for each instance. These include a mixture of linear, non-linear, ordering-insensitive and ordering-sensitive operators. We also create synthetic models that approximate these functions and serve as the models to be analyzed. We control the features that are relevant to the models, as well as the temporal properties of the models, including relevant windows and dependence on ordering for each feature. We then analyze these models using TIME and evaluate the results in terms of power (the fraction of relevant features correctly identified) and FDR (the fraction of features estimated to be important, but not truly relevant in the underlying function).

Synthetic data. We use Markov chains to generate time series data, as shown in Figure 5.6a. Each feature is associated with a randomly selected window and a pair of Markov chains, one each to generate values for in-window and out-of-window timesteps. The number of states in each chain is sampled uniformly

at random between 2 and 5. The features include a combination of continuous and discrete features. Each state m is associated with a Gaussian random variable $S_m \sim \mathcal{N}(\mu_m, \sigma_m^2)$ (for continuous features) or an integer value (for discrete features) and transition probabilities p_{mn} to other states n within the same chain. The mean, standard deviation, and transition probabilities for each state are sampled uniformly at random. The sequence for a given instance and feature is generated via a random walk through the chains. For example, a sequence i for feature j with 5 timesteps and underlying window [2, 3] may be generated as: $\mathbf{x}_j^{(i)} = \left\langle x_{j,1}^{(i)}, x_{j,2}^{(i)}, x_{j,3}^{(i)}, x_{j,4}^{(i)}, x_{j,5}^{(i)} \right\rangle = \left\langle s'_{0,1}, s'_{2,2}, s_{1,3}, s_{1,4}, s'_{0,5} \right\rangle$, where $s_{m,t}$ is sampled from S_m at timestep t. For some continuous features, sampled values are aggregated over time to model increasing, constant, or decreasing trends, as shown in Figure 5.6b. In this case, for each timestep t, $x_{j,t} = s_{m,t} + \sum_{k=1}^{t-1} x_{j,k}$.

Synthetic models. For each synthetic data set, we create a synthetic model comprising a multi-level function to generate targets for the instances. For each feature j, we apply a feature function g_j that aggregates the values within the window $[k_1, k_2]$ of that feature:

$$g_j\left(\mathbf{x}_j\right) = \hat{g}_j \circ \tilde{g}_j \circ \bar{g}_j\left(x_{j,k_1} \ldots x_{j,k_2}\right) \tag{5.15}$$

where (i) $\bar{g}_j$ is aggregation operator, randomly selected from one of `max`, `average` (both insensitive to temporal ordering), `monotonic-weighted-average` and `random-weighted-average` (both sensitive to temporal ordering) functions, (ii) $\tilde{g}_j$ is randomly selected from one of `identity`, `absolute-value` and `square` functions and serves to potentially induce interactions between the timesteps in the window, and (iii) $\hat{g}_j$ is a standardization operator that yields zero mean and unit variance for feature j across the data set.

We designate a subset of all features as relevant and take a linear combination

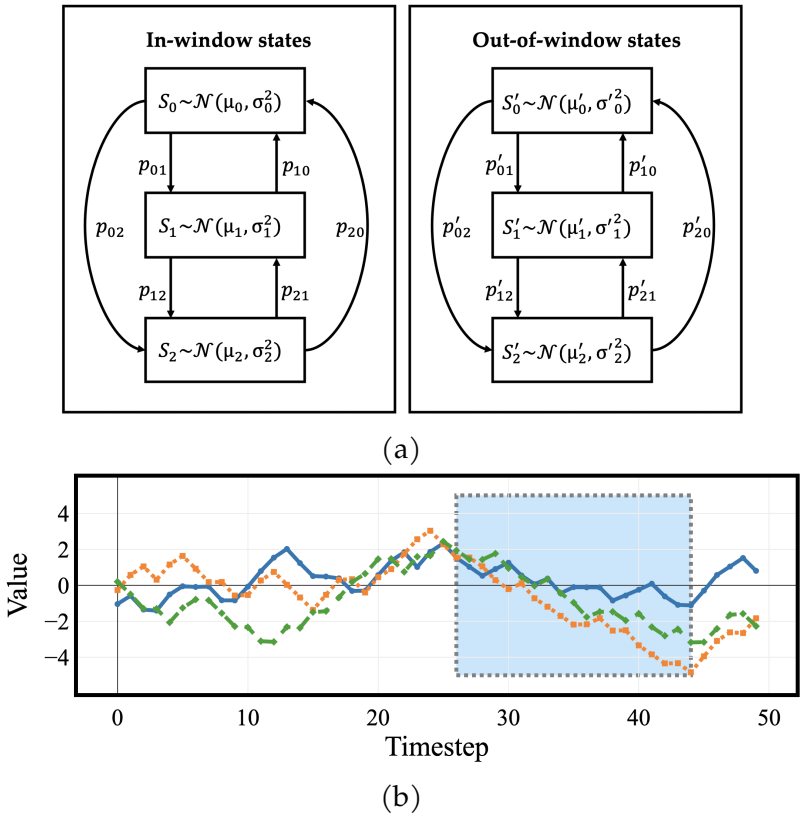


Figure 5.6: (a) Generator for a continuous feature consisting of two Markov chains, one each for in-window and out-of-window states. Here each Markov chain consists of three states, and each state is associated with a Gaussian random variable. (b) Three sequences generated via random walks through the chains, with the sampled values aggregated over time to create trends. The window is represented by blue shading.

of their feature functions to generate the target:

$$y = \sum_{j \in \mathcal{R}} \alpha_j g_j \left(\mathbf{x}_j \right) \tag{5.16}$$

where $\mathcal{R}$ is the set of relevant features, and coefficients α_j are sampled uniformly at random between -1 and 1. This serves to generate responses for a regression task. To emulate a classification task, we choose a threshold such that half the instances are labeled negative and the other half are labeled positive.

The synthetic model represents an approximation of this function and is generated by adding a weighted linear combination of the set of irrelevant features $\mathcal{R}'$ to the function:

$$f\left(\mathbf{X}\right) = \sum_{j \in \mathcal{R}} \alpha_j g_j \left(\mathbf{x}_j \right) + \beta \left[\sum_{j' \in \mathcal{R}'} \alpha_{j'} g_{j'} \left(\mathbf{x}_{j'} \right) \right]. \tag{5.17}$$

The terms corresponding to the irrelevant features represent noise in the model, with the overall level of noise controlled by the multiplier β. The rationale behind the approximation is to have a realistic model that does not perfectly match the underlying function and whose output changes in a small way when irrelevant features are perturbed, but not in a way that consistently affects the loss function $\mathcal{L}$.

Unlike a real model where training may involve optimizing over a loss function, here we use a loss function only to measure the fidelity of the model output f to the target y and compute the importance of each feature. We use quadratic loss for regression models and binary cross-entropy for classification models.

Baseline comparisons. We compare TIME against several model-agnostic baseline methods, covering a range of alternative methodologies: global vs. local, loss vs. output-based, reference value vs. permutation-based. We also attempted to include methods that address model-agnostic interpretability of temporal models, namely, TimeSHAP (Bento et al. 2020) and FIT (Tonekaboni et al. 2020), but were unable to do so due to the lack of a public implementation for TimeSHAP and impractically slow performance of FIT. Acronyms used to refer to variants of the same method are indicated in parentheses.

- **LIME** (Ribeiro et al. 2016): a method for local explanations. We aggregate local feature importance scores to generate global ones, based on the *submodular pick* algorithm described by the authors. We include LIME due to its widespread usage as an explanation method, and as a representative of other methods that focus on the model output rather than loss and generate local explanations.
- **Feature Occlusion** (Zeiler and Fergus 2014): a perturbation-based method that focuses on the model output and perturbs features by replacing them with zero reference values (FO-z). Suresh et al. (2017) use a variant that uses uniformly sampled reference values to analyze LSTM models (FO-u).
- **CXPlain** (Schwab and Karlen 2019): a method that trains a surrogate explanation model and perturbs features using reference (typically zero) values to calculate importance scores.
- **SAGE** (Covert et al. 2020b): a method that generalizes SHAP (Lundberg and Lee 2017) to global explanations. SAGE is intractable to compute exactly, so we use two approximations: sampling held-out features from (i) their marginal distributions (SAGE), or (ii) reference values, namely mean (SAGE-m) or zero (SAGE-z) values.
- **PERM**: a method that uses conventional permutations of individual timesteps rather than sequences to compute importance scores. We also test a variant that also performs hypothesis testing and FDR control using permutation tests and the BH-procedure (Benjamini and Hochberg 1995) over all timesteps (PERM-f).

Since the baseline methods are designed for a tabular feature representation, we unroll the temporal data comprising D features and L timesteps into tabular data with $D \times L$ features. To avoid confusion with temporal features, we refer to tabular features simply as 'timesteps' in the context of evaluation, since each tabular feature corresponds to a single feature-timestep pair in the original representation.

For TIME, we set γ (see Section 5.2.2) to 0.99 and control FDR at the 0.1 level. We sample $|\mathcal{P}_j| = 50$ permutations to compute importance scores and p-values for each feature j.

We generate data sets with 1,000 instances, 10 features and 20 timesteps per feature. Five features are randomly selected as relevant. We create a synthetic

model for each data set, with β tuned to yield a 90% accuracy for classification models or an R^2 value of 0.9 for regression models. We evaluate the methods by examining power and FDR for identifying relevant features as well as timesteps, and average the results over 100 data sets and models.

For the baseline methods, we estimate a feature's importance by averaging non-zero importance scores across the timesteps belonging to the feature. We sort timesteps in decreasing order of importance scores and report the n features or timesteps with the highest scores, where n is determined by the number of relevant features and timesteps in the ground truth. Since TIME identifies specific features and windows as important, we evaluate it based on two metrics: (i) using all the features and timesteps it identifies as important, and (ii) using up to n timesteps with the highest non-zero scores, as we do with the other baselines. We refer to these as TIME and TIME-n respectively.

Table 5.1 shows results from this comparison, averaged across 100 data sets and classification models. Both TIME and TIME-n significantly outperform all baselines in terms of average power and FDR for both features and timesteps, and the average FDR is well-controlled at the 0.1 level. We also include the average number of windows as a measure of the interpretability of the resulting explanations. Each ground truth model has five windows $(\text{one per relevant feature})$, so values closer to five are better. By this metric, TIME and TIME-n are advantaged in the sense that they identify one window per feature, though the high performance of TIME rests on its ability to distinguish relevant and irrelevant features accurately. In contrast, most baseline methods identify a much larger number of windows, leading to more fragmented and less interpretable explanations. Finally, we note that our implementation of TIME supports distributed processing of features, which provides a significant speedup, but which is disabled for these results for a fair comparison of running times.

Figure 5.7 illustrates feature importance explanations for a single model. It shows a set of heat maps indicating relevant timesteps for the ground truth model along with the importance scores returned by different explanation methods. For the ground truth model, boxes corresponding to relevant timesteps are shown in

Table 5.1: Comparison between different explanation methods on synthetic data, indicating sample means and standard deviations for power and FDR for detecting relevant features and timesteps, the number of windows, and the median runtime.

Method	Features				Timesteps				Windows		Runtime (seconds)	
	Power		FDR		Power		FDR					
	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\tilde{x}$	s
TIME	**0.930**	0.111	0.037	0.080	**0.923**	0.138	0.054	0.124	**4.87**	0.76	371	116
TIME-n	0.922	0.113	**0.018**	0.058	0.914	0.141	0.021	0.071	4.83	0.75	371	116
LIME	0.710	0.122	0.290	0.122	0.692	0.146	0.308	0.146	8.49	2.03	572	585
FO-u	0.644	0.135	0.356	0.135	0.637	0.167	0.363	0.167	7.17	1.99	292	88
FO-z	0.676	0.155	0.324	0.155	0.666	0.169	0.334	0.169	8.05	1.87	**29**	8
CXPlain	0.686	0.156	0.314	0.156	0.661	0.157	0.339	0.157	8.36	2.21	45	21
SAGE	0.806	0.129	0.194	0.129	0.786	0.128	0.214	0.128	11.05	3.47	15384	12695
SAGE-m	0.758	0.140	0.242	0.140	0.731	0.153	0.269	0.153	10.26	3.54	128	125
SAGE-z	0.656	0.142	0.344	0.142	0.648	0.163	0.352	0.163	8.21	2.19	44	96
PERM	0.836	0.127	0.164	0.127	0.818	0.135	0.182	0.135	9.28	2.87	1478	663
PERM-f	0.326	0.451	0.024	0.071	0.312	0.430	**0.008**	0.022	2.71	3.92	1478	663

a uniform color. For the explanation methods, colored boxes indicate non-zero importance scores, with higher scores shown in darker shades. Hatched textures are used to show features for which ordering is relevant (ground truth) or identified as important (TIME), but they are not shown for other explanation methods since they are not able to detect the significance of ordering. TIME assigns importance scores to windows for each feature, while the other explanation methods assign importance scores to each timestep, since they operate on a tabular representation. For this model, TIME identifies all the relevant features, timesteps and their ordering correctly. Other explanation methods assign non-zero importance scores to a mix of relevant and irrelevant timesteps, and rank irrelevant timesteps above relevant ones in some cases, adversely affecting their power and FDR for detecting important features. They also generally produce more fragmented explanations due to the larger number of reported windows.

We also perform baseline comparisons on synthetic data using a larger feature set composed of 30 features and 50 timesteps, with 10 features randomly selected as relevant. Table 5.2 shows these results, aggregated over 100 different models.

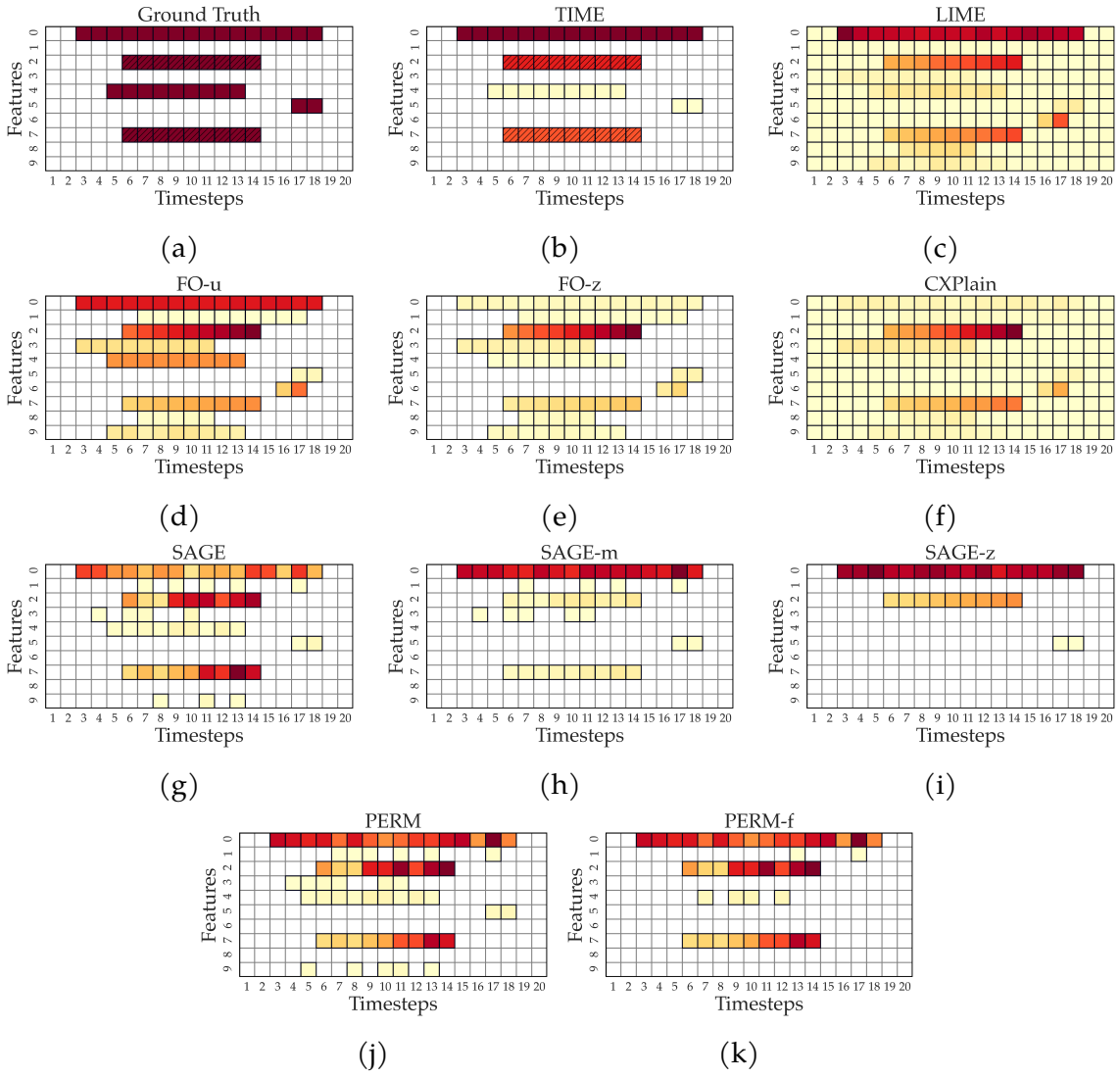


Figure 5.7: Heat maps for a single synthetic model showing (a) relevant features, windows and ordering for the ground truth model, and importance scores for (b) TIME, (c) LIME, (d) FO-u, (e) FO-z, (f), CXPlain, (g), SAGE, (h), SAGE-m, (i) SAGE-z, (j) PERM, and (k) PERM-f. Color indicates non-zero importance scores, and darker shades indicate higher scores. Hatched textures indicate sensitivity to ordering.

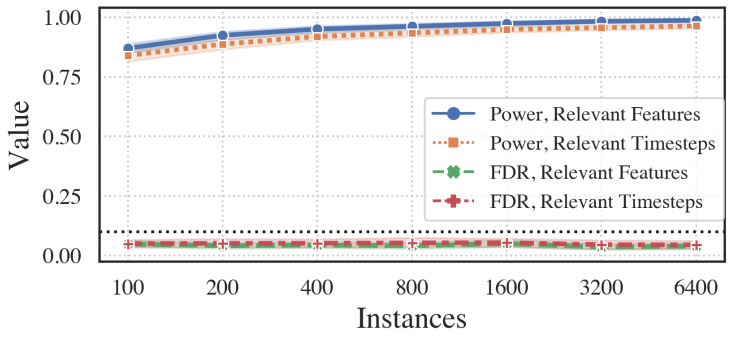
Table 5.2: Comparison between different explanation methods for synthetic models composed of 30 features and 50 timesteps, indicating sample means and standard deviations for power and FDR for detecting relevant features and timesteps, the number of windows, and the median runtime.

Method	Features				Timesteps				Windows		Runtime (seconds)	
	Power		FDR		Power		FDR					
	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s	$\tilde{x}$	s
TIME	**0.914**	0.093	0.033	0.062	**0.909**	0.105	0.037	0.079	**9.51**	1.34	2810	1026
TIME-n	0.909	0.092	**0.015**	0.039	0.905	0.104	**0.011**	0.034	9.36	1.15	2810	1026
LIME	0.728	0.105	0.272	0.105	0.680	0.117	0.320	0.117	17.77	3.87	1704	810
FO-u	0.565	0.124	0.435	0.124	0.564	0.135	0.436	0.135	14.42	2.83	2396	759
FO-z	0.626	0.106	0.374	0.106	0.615	0.113	0.385	0.113	17.38	3.65	261	114
CXPlain	0.675	0.110	0.325	0.110	0.636	0.105	0.364	0.105	17.64	3.57	**169**	62
SAGE	0.804	0.094	0.196	0.094	0.750	0.081	0.250	0.081	27.87	5.32	207241	183574
SAGE-m	0.692	0.107	0.308	0.107	0.642	0.114	0.358	0.114	22.43	6.51	2463	3037
SAGE-z	0.609	0.111	0.391	0.111	0.583	0.127	0.417	0.127	17.31	3.77	667	8747
PERM	0.830	0.092	0.170	0.092	0.792	0.097	0.208	0.097	22.73	5.37	11365	7444
PERM-f	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	11365	7444

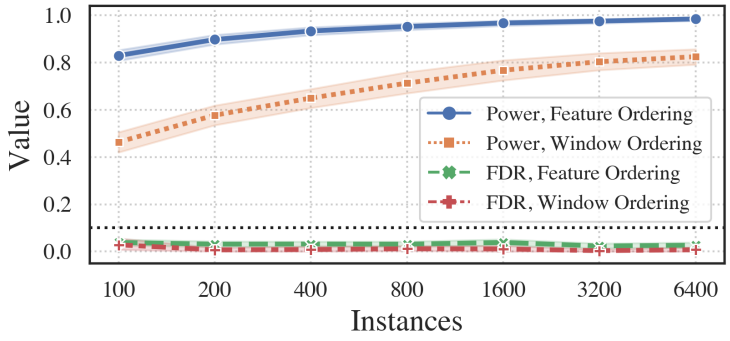
In case of SAGE and SAGE-m, these results are aggregated over 99 instead of 100 models each due to convergence issues. PERM-f does not identify any features as important. The results corroborate the conclusions drawn from Table 5.1.

Performance vs. test set size. In addition to baseline comparisons, we examine the performance of our method as a function of the size of the test set used to analyze the model. We generate data sets with 6,400 instances, 30 features and 50 timesteps per feature, and increase the size of the test set available to the model in multiples of two. Ten features are randomly selected as relevant. For each test set size, we aggregate the results over 100 different models.

Figures 5.8 and 5.9 show the results of this analysis for regression and classification models respectively. Figure 5.8a shows average power and FDR for relevant features and timesteps as a function of test set size. The power increases as the test set size increases and has high terminal values, indicating that our approach is successful at identifying most of the relevant features and windows. The average FDRs are well-controlled at the 0.1 level.



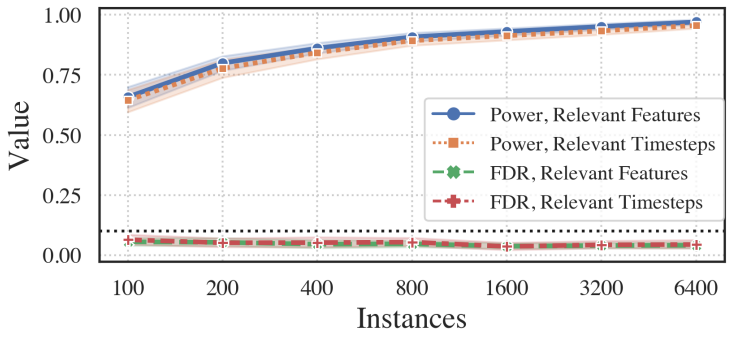
(a)



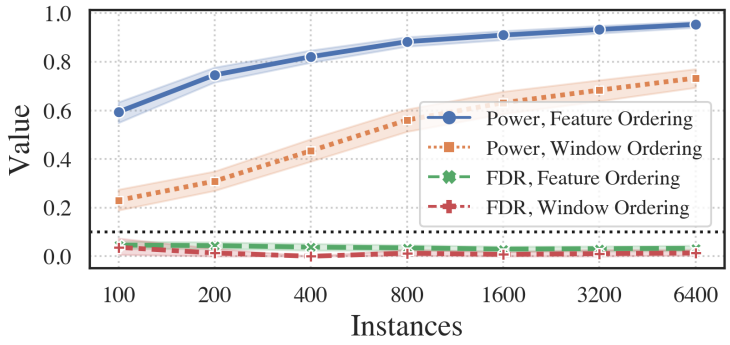
(b)

Figure 5.8: Average power and FDR for synthetic regression models for detecting (a) relevant features and timesteps, and (b) ordering relevance for features and windows, as a function of test set size. The bands represent 95% confidence intervals, and the dotted horizontal line represents the 0.1 level at which FDR is controlled.

Figure 5.8b shows average power and FDR for detecting features and windows for which the ordering of values is important. Feature ordering refers to the ordering of a feature's values across its entire sequence. Since the distribution of values inside the window is different from that outside the window, the model is sensitive to the ordering of all features having windows smaller than the sequence length. However, the model is sensitive to the ordering of values within the window only for certain feature functions. At the largest test set size, TIME is able to detect ordering with high accuracy while the FDRs are well-controlled at the 0.1 level. We detect window ordering at lower power compared to feature ordering due to



(a)



(b)

Figure 5.9: Average power and FDR for synthetic classification models for detecting (a) relevant features and timesteps, and (b) ordering relevance for features and windows, as a function of test set size. The bands represent 95% confidence intervals, and the dotted horizontal line represents the 0.1 level at which FDR is controlled.

the greater difficulty of the task, and the fact that relevant features that are not identified as important are not assessed for important windows or their ordering. Similar conclusions may be drawn from Figure 5.9.

5.3.2 MIMIC-III Benchmark LSTM Model

To consider a challenging, real-world task, we analyze an LSTM trained on MIMIC-III, a publicly available critical care database consisting of records of 58,976 intensive care unit (ICU) admissions (Johnson et al. 2016). The model is one of several pro-

posed as part of a benchmark suite for four different clinical prediction tasks over MIMIC-III (Harutyunyan et al. 2017), trained to predict in-hospital mortality of patients given the first 48 hours of their ICU stay observations. The data comprises training, validation and test sets of 14,682, 3,221 and 3,236 stays respectively, with 13.23% of the labels being positive. There are 76 features, each represented by a sequence of length 48. The features are derived from chart and laboratory measurements, and include 'mask' features indicating interpolated values. Further details on the model and features may be found in the benchmarks paper (Harutyunyan et al. 2017).

We use the validation set to analyze the LSTM and identify important features and windows, and whether or not their ordering is important to the model. We set γ as 0.9 and control FDR at the 0.1 level. We sample 200 permutations to compute importance scores and p-values. Figure 5.10 shows the results of this analysis. TIME identifies a set of 31 features that are important for the model's predictions, as well as important windows for these features. The windows almost always focus on the more recent part of the patients' histories, which is expected since death is more likely to be predicted by abnormalities in the later stages of the ICU stay. We also note that the ordering of timesteps is found to be important for some features, suggesting that the model may be picking up on trends for these features.

Since ground truth is not available for this data, we cannot compute power and FDR. Instead, to validate that our analysis has identified truly important factors, we use the set of features and windows estimated to be important to perform feature selection. We prune the features that are not estimated to be important and set the out-of-window timesteps for important features to zero. We then retrain the LSTM on the pruned data set and compare its area under the ROC curve (AUROC) to the original model on the held-aside test set. We repeat this pruning and retraining procedure for the baseline methods in Section 5.3.1, while limiting the number of features and timesteps to the numbers reported by TIME (since the baselines report non-zero importance scores for every feature and timestep). We also train and test 20 feature-selected models with 31 features and windows chosen at random.

Table 5.3 shows the results of this comparison. The AUROC for the retrained

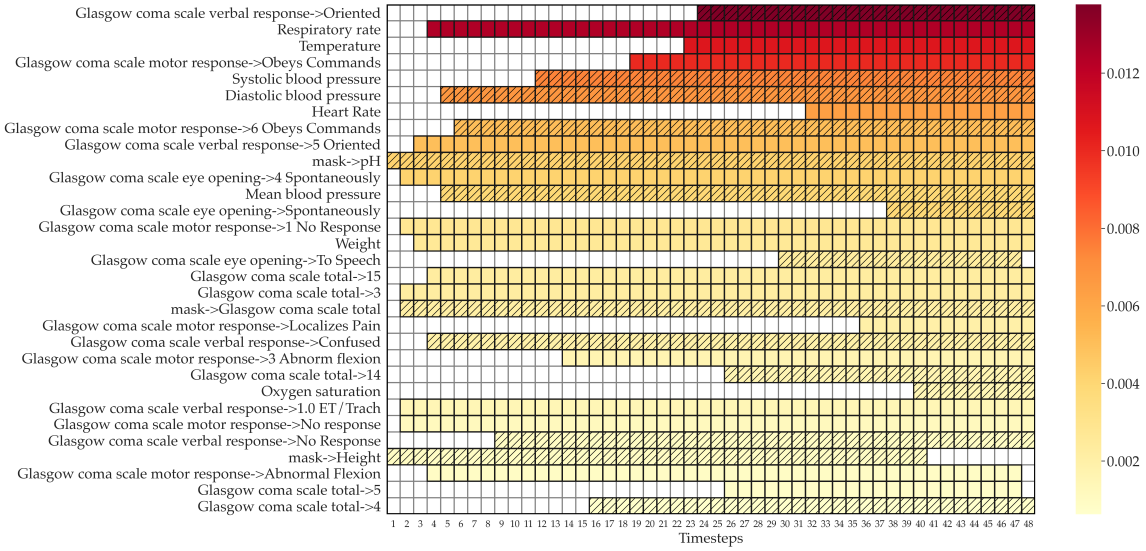


Figure 5.10: Heat map showing the TIME analysis of a MIMIC-III LSTM model trained to predict in-hospital mortality. Out of a total of 76 features, 31 were identified as important and are shown in decreasing order of their importance scores. Each row corresponds to a single feature and shows the window corresponding to important timesteps in color. The importance score is indicated by the color bar, and hatched textures show windows that were found to be significant in relation to ordering.

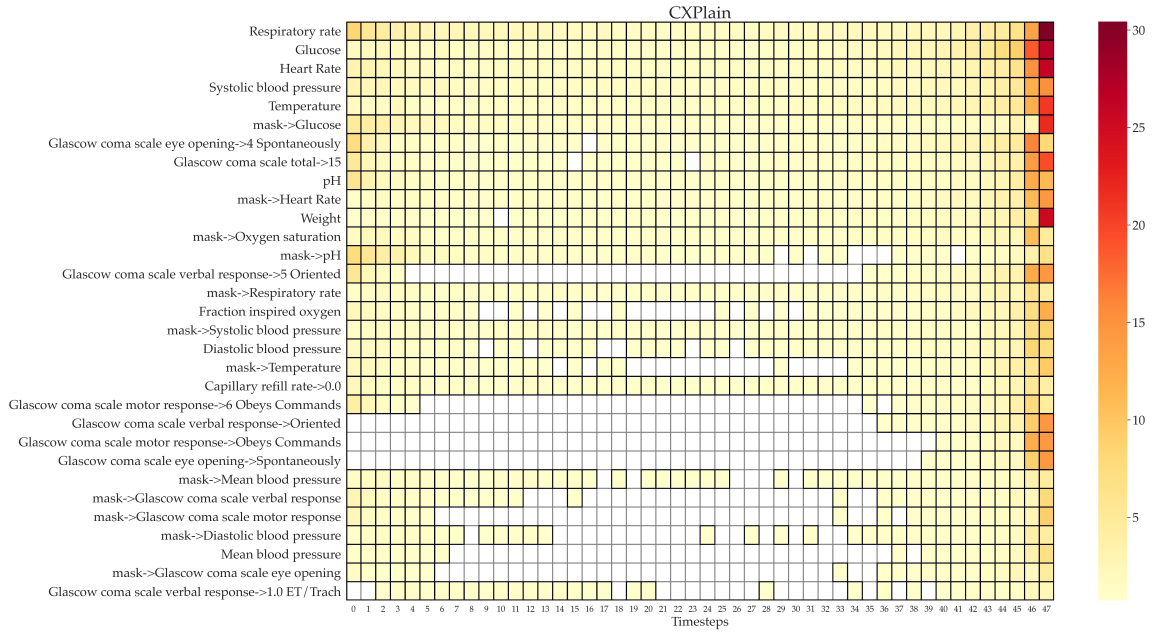
model pruned using TIME is close to that of the original model but significantly higher than the models using randomly selected features, suggesting that TIME is able to identify a salient subset of features and windows for this model. Baseline methods are advantaged in this evaluation since they assign non-zero importance scores to each timestep, whereas TIME is constrained to select features as important after performing FDR control and hence affected by the chosen FDR control rate. While AUROC serves as an imperfect surrogate of the performance of the methods in identifying important features and timesteps, it does not assess the comprehensibility of the resulting explanations, which is better represented by the number of contiguous windows identified. The results show that TIME performs competitively with the best-performing baselines while reporting significantly fewer contiguous windows, leading to concise yet accurate explanations of temporal models.

Table 5.3: Comparison of baseline methods for MIMIC-III LSTM models retrained after feature selection, using the number of features and timesteps reported by TIME to select the top-scoring features and timesteps for each method. PERM-f is not included since it does not identify any important features after FDR control.

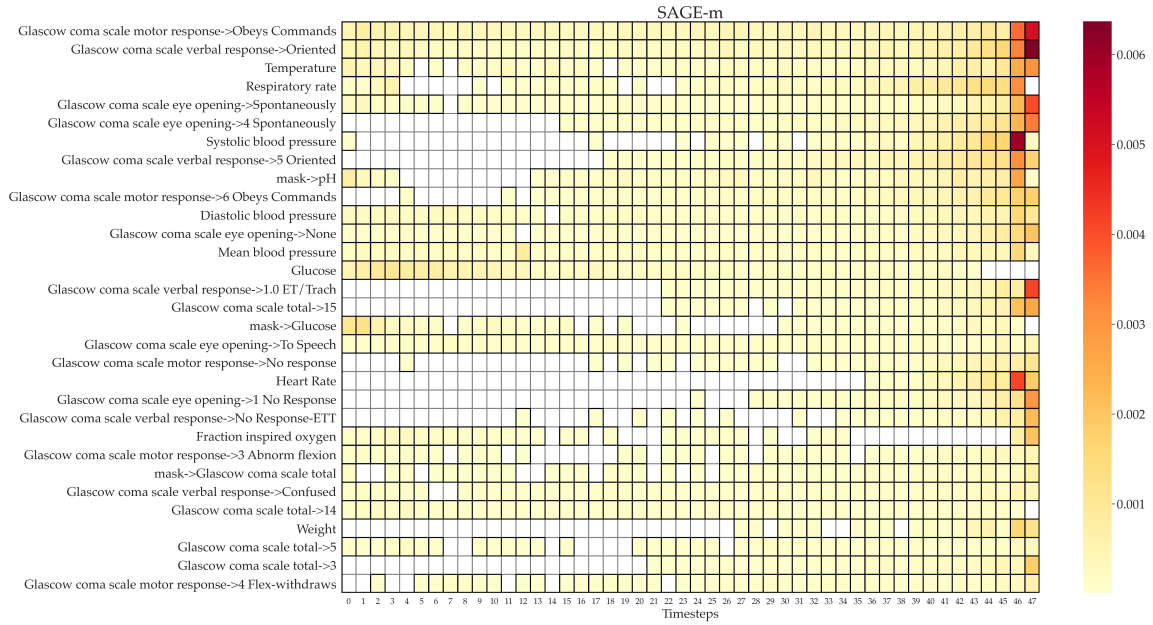
	Original	TIME	Random	LIME	FO-u	FO-z	CXPlain	SAGE-m	SAGE-z	PERM
AUROC	0.838	0.835	0.801 ± 0.015	0.784	0.805	0.818	0.834	**0.840**	0.834	0.837
Windows	-	**31**	31	38	61	61	85	101	135	225

Figure 5.11 shows heat maps for the analysis of the MIMIC-III model for three competitive baseline methods (CXPlain, SAGE-m and PERM). The heat maps show higher fragmentation and dispersion of important windows compared to TIME (Figure 5.10), leading to less interpretable explanations.

MIMIC-III LSTM analysis with feature hierarchy. Recall that the hypothesis tests performed by TIME for a given feature are arranged in a hierarchy, which may be extended to test feature groups (Figure 5.3), and for which we use a hierarchical FDR control methodology (Section 5.2.4). Using feature hierarchies can mitigate out-of-distribution sampling (Section 5.2.5.2), as well as provide explanations at multiple resolutions (Chapter 4). Figure 5.12 shows a feature hierarchy created by grouping together conceptually related features included in the MIMIC-III LSTM model. TIME is used to explain the model in conjunction with the hierarchy. Feature groups are permuted in addition to features, and important windows for feature groups are used to prune important windows for their constituent features. Features belonging to unimportant feature groups are not tested. Figure 5.13 shows a subset of the hierarchy comprising important features and feature groups identified by TIME. Figure 5.14 shows the corresponding heat map for base features. The analysis identifies fewer and more compact windows (27 windows, 859 timesteps) compared to Figure 5.10, where no hierarchy is used (31 windows, 1111 timesteps). Table 5.4 shows baseline comparisons for feature selection using this analysis, analogous to Table 5.3 (random feature selection is not included). The retrained model for TIME performs nearly as well (AUROC 0.833) as when not using a hierarchy (AUROC 0.835), and performs competitively with the best-performing baseline methods.

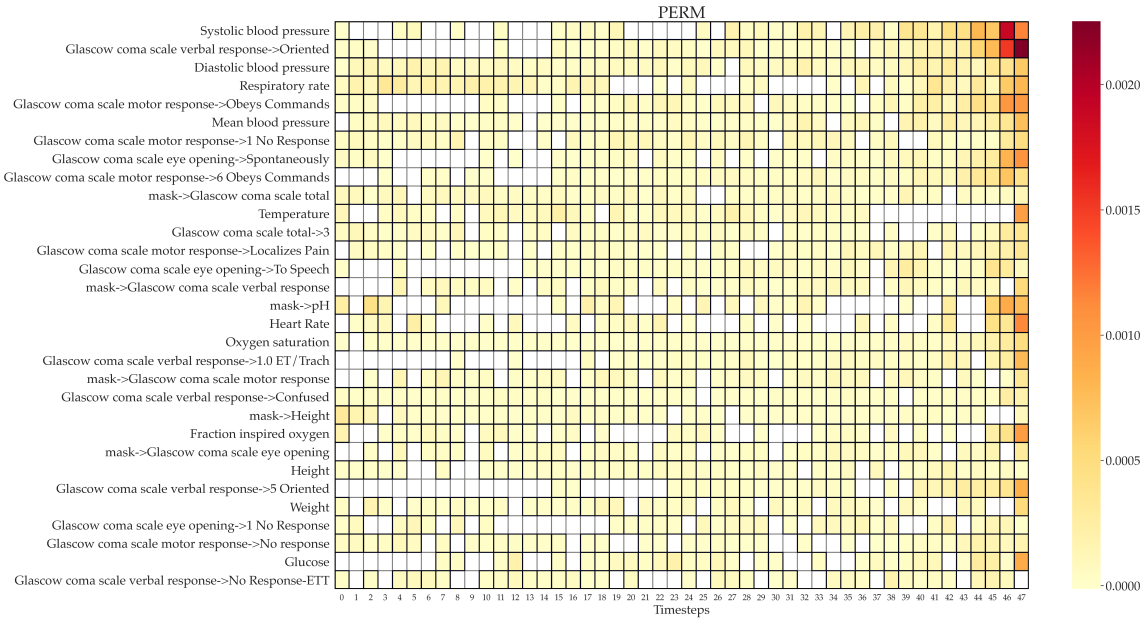


(a)



(b)

Figure 5.11: Heat maps showing explanations for the MIMIC-III LSTM model generated by (a) CXPlain and (b) SAGE-m.



(c)

Figure 5.11: Heat maps showing explanations for the MIMIC-III LSTM model (cont.) generated by (c) PERM. The number of important timesteps is selected to match the number reported by TIME. Different methods use different importance scales, as indicated by the color bars.

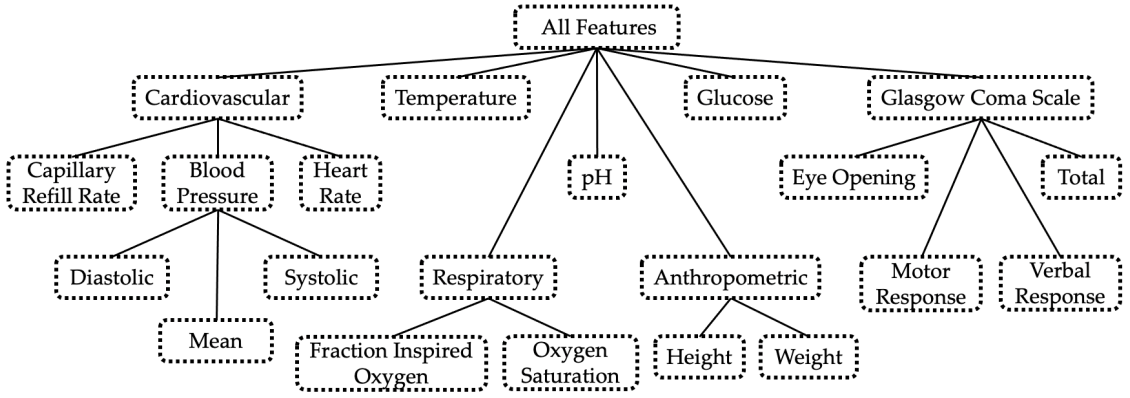


Figure 5.12: Hierarchy over features included in the MIMIC-III LSTM model, created by grouping together conceptually related categories of features. Only feature groups are shown, with each leaf node containing two or more individual features, including a mask feature.

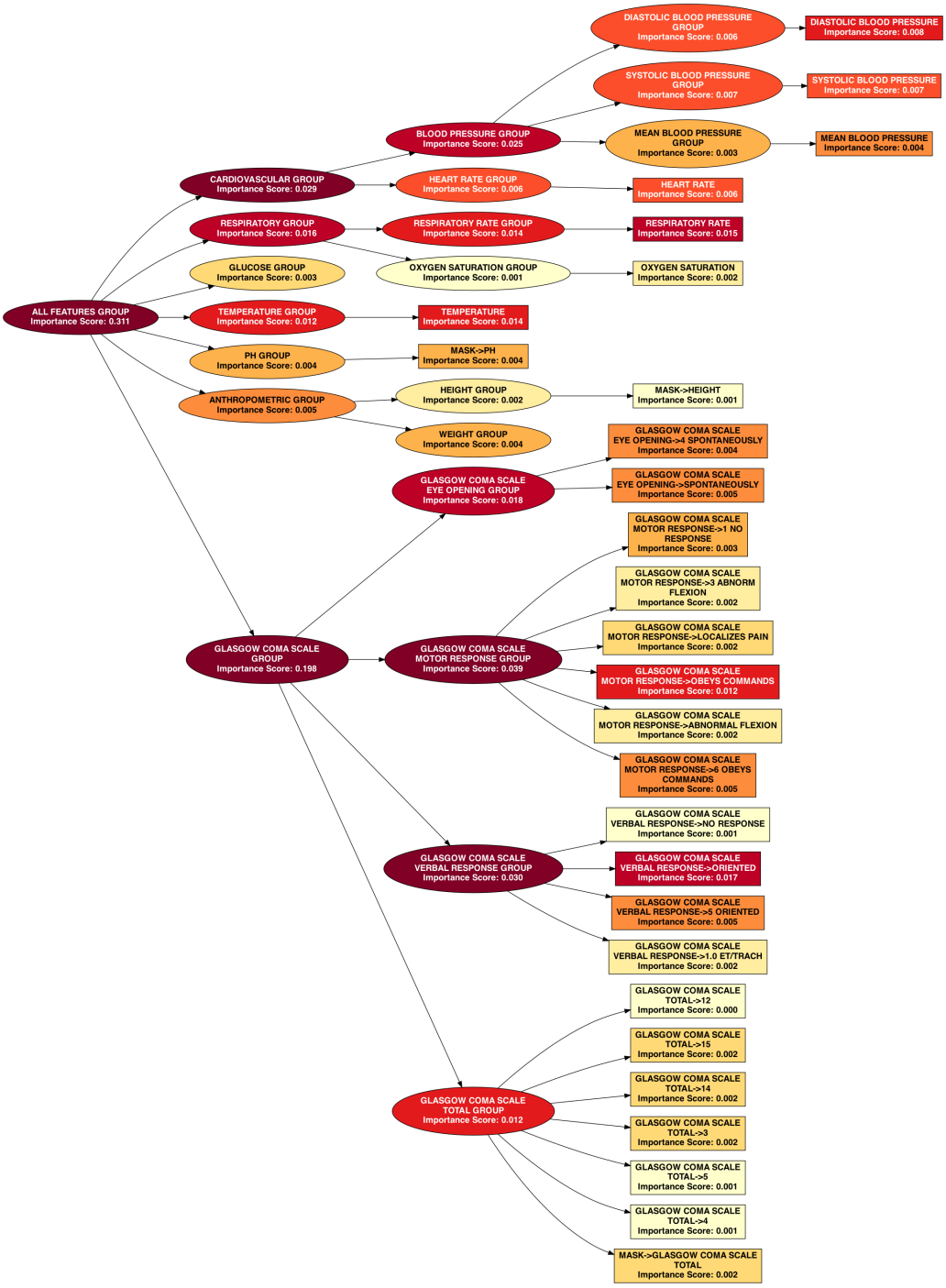


Figure 5.13: Hierarchy showing features and feature groups identified as important by TIME. Rectangles and ovals correspond to base features and feature groups respectively, and darker shades represent higher importance scores.

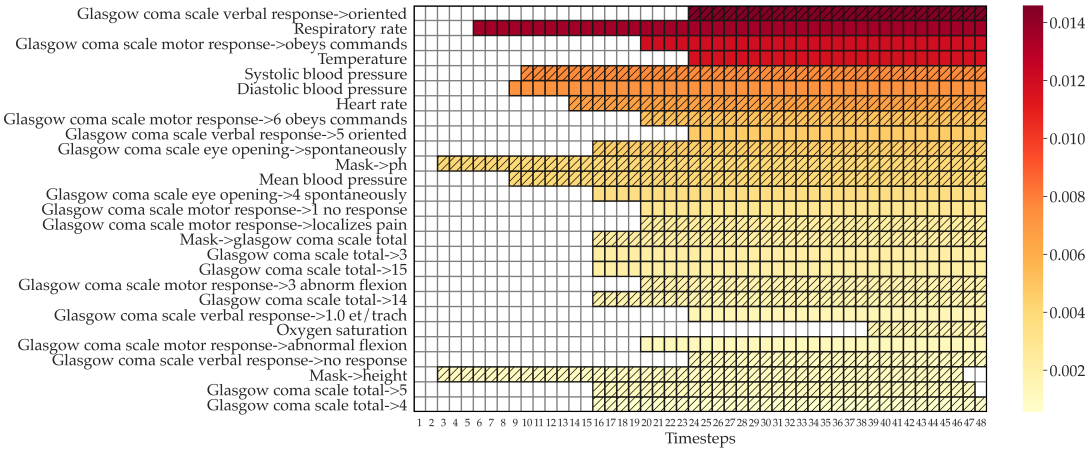


Figure 5.14: Heat map showing the analysis of the MIMIC-III LSTM model using a feature hierarchy (Figure 5.12). 27 out of 76 features are identified as important and are shown in decreasing order of importance score. Each row corresponds to a single feature and shows the window corresponding to important timesteps in color. The importance score is indicated by the color bar, and hatched textures show windows that were found to be significant in relation to ordering.

Table 5.4: Comparison of baseline methods for MIMIC-III LSTM models retrained after feature selection, using the number of features and timesteps reported by TIME in conjunction with a feature hierarchy to select the top-scoring features and timesteps for each method. PERM-f is not included since it does not identify any features as important after performing FDR control.

	Original	TIME	LIME	FO-u	FO-z	CXPlain	SAGE-m	SAGE-z	PERM
AUROC	0.838	0.833	0.780	0.784	0.806	0.833	**0.839**	0.836	0.833
Windows	-	**27**	41	59	59	73	90	129	219

5.4 Discussion

We have presented TIME, a method to explain black-box models having an explicit sequential or temporal structure. TIME identifies the set of important features and their degree of importance, and for each important feature, it identifies the window that the model focuses on and the significance of ordering within the window. It uses hypothesis testing and an FDR control methodology to detect these with statistical rigor.

Our experiments showed that on synthetic data, TIME performs significantly better than baseline methods at identifying relevant features and timesteps, and is potentially more interpretable, since it identifies important features in terms of contiguous windows rather than scattered fragments. Moreover, TIME identifies the significance of feature ordering and controls for false discoveries. Like other marginal permutation-based feature importance methods, TIME is fairly efficient to compute. We applied TIME to an LSTM trained to predict risk of in-hospital mortality from ICU data, and we identified salient features, windows and ordering in patients' clinical histories that the model focuses on. Using the important features and timesteps identified by this analysis to perform feature selection, we showed that TIME performs competitively with the best-performing baselines while yielding more comprehensible explanations.

6 CONCLUSIONS AND FUTURE WORK

Machine learning continues to have a growing impact on society, and the need for better methods to explain the decision-making of black-box models has never been greater. Interpretability in machine learning has evolved into a dynamic field of research with many areas of inquiry. Yet it remains a subject that often poses more questions than answers and presents challenges with no simple or direct solutions.

In this thesis, we explored issues of comprehensibility arising from the use of complex models for tasks characterized by large, structured feature representations. We focused on the development and analysis of black-box models that address tasks in challenging biomedical domains where interpretability is essential. We developed model-agnostic methods that can interpret these and other models by leveraging feature abstractions to expand the explanation vocabulary, while providing statistically grounded characterizations of population-level model behavior.

6.1 Summary of Contributions

1. **Modeling asthma exacerbations from electronic health records (Chapter 3).** We presented research on modeling asthma exacerbations, a prevalent acute respiratory condition, from EHRs. We developed an algorithm for phenotyping asthma excarbations from EHRs using a set of expert-curated features. We used this algorithm to identify exacerbations in our patient cohort, which we then utilized for two key tasks in modeling exacerbations. First, we considered the task of predicting exacerbations from a patient's clinical history as represented in their EHR, and performed a comparative study of supervised learning approaches for predicting near-term exacerbations. We were able to learn models that predict exacerbations with a moderately high degree of accuracy, serving as proof-of-concept for models that can enable better care for patients suffering acute asthma. Second, we considered the task

of inferring temporal exacerbation phenotypes from EHRs using a mixture of semi-Markov models. We showed that our approach is able to identify subpopulations of asthma patients sharing distinct temporal and seasonal patterns in their exacerbation susceptibility.

2. **Interpreting black-box models at multiple resolutions (Chapter 4).** We proposed a model-agnostic approach to interpreting learned models at various levels of resolution. Given a learned model and a hierarchy over the features extracted from domain knowledge or from data, our approach identifies features and feature groups important to the model's predictions. It uses a hypothesis testing methodology and a novel application of hierarchical FDR control to assess the statistical significance of features and feature groups. We also presented an approach to identify important feature interactions. We validated our approach by analyzing models using synthetic data, as well as real data from two biomedical domains, and demonstrated how our approach lends insight into complex learned models.

   The analysis of the asthma exacerbation prediction model showed the differential impact of EHR categories on the predicted outcome. We also examined which diagnoses, as defined by the ICD-9 hierarchy, are important in determining the model's predictions. Our analysis highlighted several known and some unknown (but potentially important) diagnoses associated with asthma exacerbations, as identified by the learned model.

3. **Interpreting temporal black-box models (Chapter 5).** We presented TIME, a method to explain black-box models having an explicit sequential or temporal structure. Our approach identifies the set of features important to the models' predictions as well as importance scores to indicate their degree of importance. For each important feature, it identifies the window that the models focus on and the significance of ordering within the window. TIME uses permutations both for assessing the importance of features and for hypothesis testing using permutations tests, followed by hierarchical FDR control. We showed the close connections of our method to existing permutation-based, theoretically

grounded feature importance measures, and illustrated how it is able to identify important features and windows even in the presence of out-of-distribution sampling and feature correlations.

Our experiments showed that on synthetic data, TIME performs significantly better than baseline methods at identifying relevant features and timesteps and produces more interpretable explanations, since it identifies important features in terms of contiguous windows rather than isolated timesteps. Moreover, it identifies the significance of feature ordering and controls for false discoveries. We showed that TIME is fairly efficient to compute. We used TIME to analyze an LSTM trained to predict risk of in-hospital mortality from ICU data, and we identified salient features, windows and ordering in patients' clinical histories that the model focuses on. We showed that a model trained on features and timesteps selected using this analysis performs nearly as well as the original model, and produces more concise explanations than comparable baseline methods.

Finally, we provided a software package that includes an efficient, distributed implementation of TIME as well as tools to readily visualize the model explanations generated by the method.

6.2 Future Directions

In this section, we present some promising directions for future work based on the contributions made by this thesis.

6.2.1 Predictive Modeling using Electronic Health Records

Other sources of data. *Population and environmental data*: Population data such as census tract characteristics and the area deprivation index can provide information about the epidemiological characteristics of diseases being modeled. Environmental data such as pollution and weather data can also be valuable, particularly for diseases with significant seasonal and environmental factors such as asthma.

Unstructured data: Our work uses structured data present in EHRs, including diagnoses, medications, and procedures, for predictive modeling. However, EHRs also include a significant amount of rich, unstructured data in the form of clinical notes, which may be analyzed using natural language processing techniques.

Genomic data: Genetic profiles of patients, if available, present a significant opportunity for augmenting the information embedded in EHRs and improving predictions, particularly for personalized and precision medicine.

Unsupervised and semi-supervised approaches. Unsupervised and semi-supervised approaches have the potential to significantly improve predictive models trained for specific tasks of interest. In case of asthma exacerbations, our approach based on a mixture of semi-Markov models could lend value to the supervised learning approach for predicting exacerbations. One way in which we might do this is by using the mixture model to cluster each patient based on their past exacerbation history and then computing a seasonally varying, cluster-specific risk score to use as another input feature for the exacerbation prediction models.

Institutional variability. Conventions, practices, ontologies, and systems for recording EHRs can vary significantly across healthcare institutions and states. The robustness and practical usefulness of the approach may be significantly improved by developing and evaluating models across multiple institutions.

6.2.2 Black-box Model Explanation

Local explanations. Our work focuses on global explanations, i.e., explanations of model behavior across the distribution of instances. While global explanations are valuable for many reasons, such as feature engineering and scientific understanding, local explanations are useful for explaining specific decisions made by the model. One way to extend our permutation-based approach to local explanations would be to use a generative model to generate instances in the neighborhood of the instance being explained, and to constrain perturbations to this neighborhood.

Model translation. Model translation methods learn interpretable explanatory models such as decision trees to approximate the black-box model. The important features, feature groups, and temporal properties of the features identified by our approach could be used to augment or replace the representation used by such an explanatory model in order to produce more comprehensible explanations, while maintaining high fidelity to the black-box model.

User studies. The effect of different feature abstractions on the comprehensibility of explanations and their inducement of trust could be empirically evaluated with user studies. Users could be asked to evaluate explanations using progressively granular feature sets, presented in contrast with important features resolved using feature hierarchies. Temporal explanations indicating windows of importance could be evaluated visually, such as using heat maps, or by translating them into descriptive decision rules. To isolate the effect of explanation accuracy on trust, explanations could be grouped based on their accuracy as measured by retraining and retesting the model after performing feature selection.

Other feature organizations. We use a feature hierarchy to identify groupings of related features in order to interpret models at multiple resolutions. In place of a hierarchy, however, we could consider other organizations of the features, such as using a directed acyclic graph where a feature may belong to multiple groups.

Other temporal alignments. Our approach for permutation across sequences currently assumes regularly sampled, time-aligned and fixed-length sequences. This assumption could be generalized by considering windows that are aligned in other ways, such as on an absolute scale (e.g., dates on the Gregorian calendar) or a relative scale (e.g., patient age). Where applicable, irregularly sampled time series could be transformed to a single time scale.

Other temporal properties. TIME makes the assumption that there exists a single contiguous window that is important, which could be generalized at the cost of

increased computational and explanation complexity. Additionally, we could identify other temporal properties of the model by perturbing occurrences of frequently observed temporal patterns in addition to contiguous windows. Another approach might be to identify important pairwise interactions between windows by testing pairs of non-correlated windows for non-additivity (Equation 4.2). Candidate interactions could be selected by first identifying important windows for each feature using TIME and then searching for pairs of windows having low cross-correlation.

Conditional permutations. Like many other explanation methods, TIME may perform out-of-distribution sampling, potentially breaking correlations between features, due to its use of marginal permutations for feature perturbation. Conditional permutations (Strobl et al. 2008) could be used to ameliorate this problem at the cost of increased computational complexity.

BIBLIOGRAPHY

Adadi, A. and M. Berrada (2018). "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6, pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052 (cit. on p. 12).

Adler-Milstein, J., C. M. DesRoches, P. Kralovec, et al. (2015). "Electronic Health Record Adoption in US Hospitals: Progress Continues, but Challenges Persist". In: *Health Affairs* 34.12, pp. 2174–2180 (cit. on p. 23).

Altmann, A., L. Toloşi, O. Sander, and T. Lengauer (2010). "Permutation Importance: A Corrected Feature Importance Measure". In: *Bioinformatics* 26.10, pp. 1340–1347. DOI: 10.1093/bioinformatics/btq134 (cit. on pp. 22, 69).

Alvarez-Melis, D. and T. S. Jaakkola (2017). *A Causal Framework for Explaining the Predictions of Black-Box Sequence-to-Sequence Models*. arXiv: 1707.01943 [cs]. URL: http://arxiv.org/abs/1707.01943 (cit. on pp. 17, 18).

Alvarez-Melis, D. and T. S. Jaakkola (2018a). *On the Robustness of Interpretability Methods*. arXiv: 1806.08049. URL: http://arxiv.org/abs/1806.08049 (cit. on p. 20).

Alvarez-Melis, D. and T. S. Jaakkola (2018b). "Towards Robust Interpretability with Self-Explaining Neural Networks". In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., pp. 7785–7794 (cit. on pp. 15, 19, 20, 48).

Ashfaq, A., A. Sant'Anna, M. Lingman, and S. Nowaczyk (2019). "Readmission Prediction Using Deep Learning on Electronic Health Records". In: *Journal of Biomedical Informatics* 97, p. 103256. DOI: 10.1016/j.jbi.2019.103256 (cit. on p. 70).

Ba, J. and R. Caruana (2014). "Do Deep Nets Really Need to Be Deep?" In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., pp. 2654–2662 (cit. on p. 17).

Barocas, S., A. D. Selbst, and M. Raghavan (2020). "The Hidden Assumptions behind Counterfactual Explanations and Principal Reasons". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. FAT* '20. Barcelona,

Spain: Association for Computing Machinery, pp. 80–89. DOI: 10.1145/3351095.3372830 (cit. on p. 4).

Barredo Arrieta, A., N. Díaz-Rodríguez, J. Del Ser, et al. (2020). "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI". In: *Information Fusion* 58, pp. 82–115. DOI: 10.1016/j.inffus.2019.12.012 (cit. on p. 12).

Bateman, E. D., R. Buhl, P. M. O'Byrne, et al. (2015). "Development and Validation of a Novel Risk Score for Asthma Exacerbations: The Risk Score for Exacerbations". In: *Journal of Allergy and Clinical Immunology* 135.6, 1457–1464.e4. DOI: 10.1016/j.jaci.2014.08.015 (cit. on p. 28).

Bau, D., B. Zhou, A. Khosla, A. Oliva, and A. Torralba (2017). *Network Dissection: Quantifying Interpretability of Deep Visual Representations*. arXiv: 1704.05796 [cs]. URL: http://arxiv.org/abs/1704.05796 (cit. on pp. 18, 20, 48).

Bau, D., J.-Y. Zhu, H. Strobelt, et al. (2018). *GAN Dissection: Visualizing and Understanding Generative Adversarial Networks*. arXiv: 1811.10597 [cs]. URL: http://arxiv.org/abs/1811.10597 (cit. on p. 19).

Benjamini, Y. and Y. Hochberg (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1, pp. 289–300. JSTOR: 2346101 (cit. on pp. 53, 77, 92).

Bento, J., P. Saleiro, A. F. Cruz, M. A. T. Figueiredo, and P. Bizarro (2020). *TimeSHAP: Explaining Recurrent Models through Sequence Perturbations*. arXiv: 2012.00073 [cs]. URL: http://arxiv.org/abs/2012.00073 (cit. on pp. 21, 69, 91).

Bousquet, J. (2000). "Global Initiative for Asthma (GINA) and Its Objectives". In: *Clinical and experimental allergy* 30 Suppl 1, pp. 2–5. DOI: 10.1046/j.1365-2222.2000.00088.x. pmid: 10849466 (cit. on p. 30).

Breiman, L. (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324 (cit. on pp. 18–20, 22, 32, 48, 49, 69, 78, 87).

Breiman, L. and N. Shang (1996). "Born Again Trees". In: *University of California, Berkeley, Berkeley, CA, Technical Report* (cit. on p. 19).

Burns, C., J. Thomason, and W. Tansey (2020). *Interpreting Black Box Models via Hypothesis Testing*. arXiv: 1904.00045 [cs, stat]. URL: http://arxiv.org/abs/1904.00045 (cit. on pp. 22, 76, 78).

Burrell, J. (2016). "How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms". In: *Big Data & Society* (cit. on p. 13).

Busse, W. W., W. J. Morgan, V. Taggart, and A. Togias (2012). "Asthma Outcomes Workshop: Overview". In: *Journal of Allergy and Clinical Immunology*. Standardizing Asthma Outcomes in Clinical Research: Report of the Asthma Outcomes Workshop 129 (3, Supplement), S1–S8. DOI: 10.1016/j.jaci.2011.12.985 (cit. on p. 30).

Carter, B., J. Mueller, S. Jain, and D. Gifford (2018). *What Made You Do This? Understanding Black-Box Decisions with Sufficient Input Subsets*. arXiv: 1810.03805. URL: http://arxiv.org/abs/1810.03805 (cit. on p. 19).

Caruana, R., H. Kangarloo, J. D. Dionisio, U. Sinha, and D. Johnson (1999). "Case-Based Explanation of Non-Case-Based Learning Methods." In: *Proceedings of the AMIA Symposium*, pp. 212–215. pmid: 10566351 (cit. on p. 19).

Caruana, R., Y. Lou, J. Gehrke, et al. (2015). "Intelligible Models for Healthcare: Predicting Pneumonia Risk and Hospital 30-Day Readmission". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia). KDD '15. New York, NY, USA: ACM, pp. 1721–1730. DOI: 10.1145/2783258.2788613 (cit. on p. 14).

Charles, D., M. Gabriel, and M. F. Furukawa (2013). "Adoption of Electronic Health Record Systems among US Non-Federal Acute Care Hospitals: 2008-2012". In: *ONC Data Brief* 9, pp. 1–9 (cit. on p. 23).

Che, Z., D. Kale, W. Li, M. Bahadori, and Y. Liu (2015). "Deep Computational Phenotyping". In: *Proceedings of the 21th ACM* (cit. on pp. 25, 26).

Che, Z., S. Purushotham, K. Cho, D. Sontag, and Y. Liu (2016). *Recurrent Neural Networks for Multivariate Time Series with Missing Values*. arXiv: 1606.01865 [cs, stat]. URL: http://arxiv.org/abs/1606.01865 (cit. on p. 25).

Che, Z., S. Purushotham, R. Khemani, and Y. Liu (2017). "Interpretable Deep Models for ICU Outcome Prediction". In: *AMIA Annual Symposium Proceedings* 2016, pp. 371–380. pmid: 28269832 (cit. on pp. 25, 26).

Cheng, Y., F. Wang, P. Zhang, and J. Hu (2016). "Risk Prediction with Electronic Health Records: A Deep Learning Approach". In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. 0 vols. Proceedings. Society for Industrial and Applied Mathematics, pp. 432–440. DOI: 10.1137/1.9781611974348.49 (cit. on pp. 25, 26).

Choi, E., M. Bahadori, L. Song, and W. Stewart (2016a). "GRAM: Graph-Based Attention Model for Healthcare Representation Learning". In: *arXiv preprint arXiv:* (cit. on p. 25).

Choi, E., M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun (2015). *Doctor AI: Predicting Clinical Events via Recurrent Neural Networks*. arXiv: 1511.05942 [cs]. URL: http://arxiv.org/abs/1511.05942 (cit. on pp. 24, 25).

Choi, E., M. T. Bahadori, E. Searles, et al. (2016b). "Multi-Layer Representation Learning for Medical Concepts". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA). KDD '16. New York, NY, USA: ACM, pp. 1495–1504. DOI: 10.1145/2939672.2939823 (cit. on pp. 15, 24–26, 33, 61).

Choi, E., M. T. Bahadori, J. Sun, et al. (2016c). "RETAIN: An Interpretable Predictive Model for Healthcare Using Reverse Time Attention Mechanism". In: *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., pp. 3504–3512 (cit. on pp. 21, 25, 26, 68).

Choi, E., A. Schuetz, W. F. Stewart, and J. Sun (2016d). *Medical Concept Representation Learning from Electronic Health Records and Its Application on Heart Failure Prediction*. arXiv: 1602.03686 [cs]. URL: http://arxiv.org/abs/1602.03686 (cit. on pp. 24, 25).

Choi, Y., C. Y.-I. Chiu, and D. Sontag (2016e). "Learning Low-Dimensional Representations of Medical Concepts". In: *AMIA Summits on Translational Science Proceedings* 2016, pp. 41–50. pmid: 27570647 (cit. on p. 24).

Cobian, A., M. Abbott, A. Sood, et al. (2020). "Modeling Asthma Exacerbations from Electronic Health Records". In: *AMIA Summits on Translational Science Proceedings* 2020, pp. 98–107. pmid: 32477628 (cit. on pp. 27, 70).

Covert, I., S. M. Lundberg, and S.-I. Lee (2020a). *Explaining by Removing: A Unified Framework for Model Explanation*. arXiv: 2011.14878 [cs, stat]. URL: http://arxiv.org/abs/2011.14878 (cit. on pp. 21, 22).

Covert, I., S. M. Lundberg, and S.-I. Lee (2020b). "Understanding Global Feature Contributions With Additive Importance Measures". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc. (cit. on pp. 20, 22, 81, 92).

Craven, M. and J. Shavlik (1994). "Using Sampling and Queries to Extract Rules from Trained Neural Networks". In: *Machine Learning Proceedings 1994*. Elsevier, pp. 37–45 (cit. on p. 19).

Craven, M. and J. W. Shavlik (1996). "Extracting Tree-Structured Representations of Trained Networks". In: *Advances in Neural Information Processing Systems 8*. MIT Press, pp. 24–30 (cit. on pp. 17–19).

Datta, A., S. Sen, and Y. Zick (2016). "Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, pp. 598–617. DOI: 10.1109/SP.2016.42 (cit. on pp. 18, 19).

De Vine, L., G. Zuccon, B. Koopman, L. Sitbon, and P. Bruza (2014). "Medical Semantic Similarity with a Neural Language Model". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (Shanghai, China). CIKM '14. New York, NY, USA: ACM, pp. 1819–1822. DOI: 10.1145/2661829.2661974 (cit. on p. 24).

Dombrowski, A.-K., M. Alber, C. Anders, et al. (2019). "Explanations Can Be Manipulated and Geometry Is to Blame". In: *Advances in Neural Information Processing Systems* 32, pp. 13589–13600 (cit. on p. 20).

Doshi-Velez, F. and B. Kim (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. arXiv: 1702.08608. URL: http://arxiv.org/abs/1702.08608 (cit. on pp. 4, 6, 12, 19, 69).

Dougherty, R. H. and J. V. Fahy (2009). "Acute Exacerbations of Asthma: Epidemiology, Biology and the Exacerbation-Prone Phenotype". In: *Clinical & Experimental Allergy* 39.2, pp. 193–202 (cit. on p. 27).

Esteban, C., O. Staeck, S. Baier, Y. Yang, and V. Tresp (2016). "Predicting Clinical Events by Combining Static and Dynamic Information Using Recurrent Neural Networks". In: *Healthcare Informatics (ICHI), 2016 IEEE International Conference On*. Ieee, pp. 93–101 (cit. on p. 25).

Faruqui, M., Y. Tsvetkov, D. Yogatama, C. Dyer, and N. Smith (2015). *Sparse Overcomplete Word Vector Representations*. arXiv: 1506.02004. URL: http://arxiv.org/abs/1506.02004 (cit. on pp. 15, 17, 18).

Fisher, A., C. Rudin, and F. Dominici (2019). "All Models Are Wrong, but Many Are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously". In: *Journal of Machine Learning Research* 20.177, pp. 1–81. arXiv: 1801.01489 (cit. on pp. 19, 22, 69, 78–80, 87).

Fong, R. C. and A. Vedaldi (2017). "Interpretable Explanations of Black Boxes by Meaningful Perturbation". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, pp. 3449–3457. DOI: 10.1109/ICCV.2017.371 (cit. on pp. 11, 18, 20, 48, 49, 52).

Freitas, A. A. (2014). "Comprehensible Classification Models: A Position Paper". In: *SIGKDD Explor. Newsl.* 15.1, pp. 1–10. DOI: 10.1145/2594473.2594475 (cit. on pp. 14, 15).

Friedman, J. H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29.5, pp. 1189–1232. JSTOR: 2699986 (cit. on p. 19).

Friedman, J. H. and B. E. Popescu (2008). "Predictive Learning via Rule Ensembles". In: *The Annals of Applied Statistics* 2.3, pp. 916–954. DOI: 10.1214/07-AOAS148 (cit. on p. 19).

Frosst, N. and G. Hinton (2017). *Distilling a Neural Network Into a Soft Decision Tree*. arXiv: 1711.09784 [cs, stat]. URL: http://arxiv.org/abs/1711.09784 (cit. on p. 19).

Ghassemi, M., T. Naumann, P. Schulam, A. L. Beam, and R. Ranganath (2018). *Opportunities in Machine Learning for Healthcare*. arXiv: 1806.00388 [cs, stat]. URL: http://arxiv.org/abs/1806.00388 (cit. on p. 24).

Ghorbani, A., A. Abid, and J. Zou (2017). *Interpretation of Neural Networks Is Fragile*. arXiv: 1710.10547. URL: http://arxiv.org/abs/1710.10547 (cit. on p. 20).

Gleicher, M. (2016). "A Framework for Considering Comprehensibility in Modeling". In: *Big Data* 4.2, p. 75. DOI: 10.1089/big.2016.0007. pmid: 27441712 (cit. on pp. 4, 11, 12, 17).

Goldstein, B. A., A. M. Navar, M. J. Pencina, and J. P. A. Ioannidis (2017). "Opportunities and Challenges in Developing Risk Prediction Models with Electronic Health Records Data: A Systematic Review". In: *Journal of the American Medical Informatics Association* 24.1, pp. 198–208. DOI: 10.1093/jamia/ocw042 (cit. on pp. 23, 24).

Golland, P., F. Liang, S. Mukherjee, and D. Panchenko (2005). "Permutation Tests for Classification". In: *Learning Theory*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 501–515. DOI: 10.1007/11503415_34 (cit. on pp. 22, 76, 78).

Good, P. (2013). *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer Science & Business Media. 288 pp. Google Books: pK3hBwAAQBAJ (cit. on p. 22).

Gregorutti, B., B. Michel, and P. Saint-Pierre (2015). "Grouped Variable Importance with Random Forests and Application to Multiple Functional Data Analysis". In: *Computational Statistics & Data Analysis* 90, pp. 15–35. DOI: 10.1016/j.csda.2015.04.002 (cit. on pp. 22, 69, 78–80, 84, 87).

Gregorutti, B., B. Michel, and P. Saint-Pierre (2017). "Correlation and Variable Importance in Random Forests". In: *Statistics and Computing* 27.3, pp. 659–678. DOI: 10.1007/s11222-016-9646-1. arXiv: 1310.5726 (cit. on pp. 19, 20, 80–82).

Guidotti, R., A. Monreale, S. Ruggieri, et al. (2018). "A Survey of Methods for Explaining Black Box Models". In: *ACM Comput. Surv.* 51.5, 93:1–93:42. DOI: 10.1145/3236009 (cit. on pp. 12, 17, 18).

Gunning, D. (2017). "Explainable Artificial Intelligence (Xai)". In: *Defense Advanced Research Projects Agency (DARPA)* (cit. on p. 11).

Hara, S. and K. Hayashi (2016). *Making Tree Ensembles Interpretable*. arXiv: 1606.05390. URL: http://arxiv.org/abs/1606.05390 (cit. on pp. 17, 18).

Harutyunyan, H., H. Khachatrian, D. C. Kale, G. V. Steeg, and A. Galstyan (2017). *Multitask Learning and Benchmarking with Clinical Time Series Data*. arXiv: 1703.07771 [cs, stat]. URL: http://arxiv.org/abs/1703.07771 (cit. on p. 99).

Henelius, A., K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou (2014). "A Peek into the Black Box: Exploring Classifiers by Randomization". In: *Data Mining and Knowledge Discovery* 28.5, pp. 1503–1529. DOI: 10.1007/s10618-014-0368-8 (cit. on pp. 18, 78).

Hinton, G., O. Vinyals, and J. Dean (2015). *Distilling the Knowledge in a Neural Network*. arXiv: 1503.02531 [cs, stat]. URL: http://arxiv.org/abs/1503.02531 (cit. on p. 17).

Ho, J., J. Ghosh, and J. Sun (2014). "Marble: High-Throughput Phenotyping from Electronic Health Records via Sparse Nonnegative Tensor Factorization". In: *Proceedings of the 20th ACM SIGKDD* (cit. on p. 15).

Hoch, H. E., A. Calatroni, J. B. West, et al. (2017). "Can We Predict Fall Asthma Exacerbations? Validation of the Seasonal Asthma Exacerbation Index". In: *Journal of Allergy and Clinical Immunology* 140.4, 1130–1137.e5. DOI: 10.1016/j.jaci.2017.01.026 (cit. on p. 28).

Hochreiter, S. and J. Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735 (cit. on pp. 32, 61, 88).

Hoerl, A. E. and R. W. Kennard (1970). "Ridge Regression: Biased Estimation for Nonorthogonal Problems". In: *Technometrics* 12.1, pp. 55–67. DOI: 10.1080/00401706.1970.10488634 (cit. on p. 32).

Hripcsak, G. and D. J. Albers (2013). "Next-Generation Phenotyping of Electronic Health Records". In: *Journal of the American Medical Informatics Association* 20.1, pp. 117–121. DOI: 10.1136/amiajnl-2012-001145 (cit. on pp. 23, 24).

Huysmans, J., K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens (2011). "An Empirical Evaluation of the Comprehensibility of Decision Table, Tree and Rule Based Predictive Models". In: *Decision Support Systems* 51.1, pp. 141–154. DOI: 10.1016/j.dss.2010.12.003 (cit. on p. 14).

Ibrahim, M., M. Louie, C. Modarres, and J. Paisley (2019). *Global Explanations of Neural Networks: Mapping the Landscape of Predictions*. arXiv: 1902.02384 [cs, stat]. URL: http://arxiv.org/abs/1902.02384 (cit. on pp. 6, 18, 69).

Ismail, A. A., M. Gunady, H. C. Bravo, and S. Feizi (2020). *Benchmarking Deep Learning Interpretability in Time Series Predictions*. arXiv: 2010.13924 [cs, stat]. URL: http://arxiv.org/abs/2010.13924 (cit. on pp. 21, 68).

Ismail, A. A., M. Gunady, L. Pessoa, H. C. Bravo, and S. Feizi (2019). *Input-Cell Attention Reduces Vanishing Saliency of Recurrent Neural Networks*. arXiv: 1910.12370 [cs, stat]. URL: http://arxiv.org/abs/1910.12370 (cit. on pp. 21, 68).

Jagannatha, A. N. and H. Yu (2016). "Bidirectional RNN for Medical Event Detection in Electronic Health Records". In: *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting* 2016, pp. 473–482. pmid: 27885364 (cit. on p. 25).

Jeyakumar, J. V., J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava (2020). "How Can I Explain This to You? An Empirical Study of Deep Neural Network Explanation Methods". In: *Advances in Neural Information Processing Systems* 33 (cit. on p. 19).

Johnson, A. E. W., T. J. Pollard, L. Shen, et al. (2016). "MIMIC-III, a Freely Accessible Critical Care Database". In: *Scientific Data* 3, p. 160035. DOI: 10.1038/sdata.2016.35 (cit. on p. 98).

Karpathy, A., J. Johnson, and L. Fei-Fei (2015). *Visualizing and Understanding Recurrent Networks*. arXiv: 1506.02078 [cs]. URL: http://arxiv.org/abs/1506.02078 (cit. on pp. 18, 19, 21, 68).

Kawaler, E., A. Cobian, P. Peissig, et al. (2012). "Learning to Predict Post-Hospitalization VTE Risk from EHR Data". In: *AMIA Annual Symposium Proceedings* 2012, pp. 436–445. pmid: 23304314 (cit. on p. 70).

Kim, B., M. Wattenberg, J. Gilmer, et al. (2018). "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)". In: *International Conference on Machine Learning*, pp. 2668–2677 (cit. on pp. 20, 48).

Koh, P. W. and P. Liang (2017). *Understanding Black-Box Predictions via Influence Functions*. arXiv: 1703.04730 [cs, stat]. URL: http://arxiv.org/abs/1703.04730 (cit. on pp. 11, 18).

Kolb, A. W., K. Lee, I. Larsen, M. Craven, and C. R. Brandt (2016). "Quantitative Trait Locus Based Virulence Determinant Mapping of the HSV-1 Genome in Murine Ocular Infection: Genes Involved in Viral Regulatory and Innate Immune Networks Contribute to Virulence". In: *PLoS pathogens* 12.3, e1005499 (cit. on pp. 60, 63).

Kumar, I. E., S. Venkatasubramanian, C. Scheidegger, and S. Friedler (2020). *Problems with Shapley-Value-Based Explanations as Feature Importance Measures*. arXiv: 2002.11097 [cs, stat]. URL: http://arxiv.org/abs/2002.11097 (cit. on pp. 4, 22, 81, 82).

Lakkaraju, H., E. Kamar, R. Caruana, and J. Leskovec (2019). "Faithful and Customizable Explanations of Black Box Models". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (Honolulu, HI, USA). AIES '19. New York, NY, USA: ACM, pp. 131–138. DOI: 10.1145/3306618.3314229 (cit. on p. 19).

LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep Learning". In: *Nature* 521.7553, pp. 436–444. DOI: 10.1038/nature14539 (cit. on p. 24).

Lee, G.-H., D. Alvarez-Melis, and T. S. Jaakkola (2018). *Game-Theoretic Interpretability for Temporal Modeling*. arXiv: 1807.00130 [cs, stat]. URL: http://arxiv.org/abs/1807.00130 (cit. on pp. 21, 68).

Lee, G.-H., W. Jin, D. Alvarez-Melis, and T. S. Jaakkola (2019a). *Functional Transparency for Structured Data: A Game-Theoretic Approach*. arXiv: 1902.09737 [cs, stat]. URL: http://arxiv.org/abs/1902.09737 (cit. on p. 15).

Lee, K., A. Sood, and M. Craven (2019b). "Understanding Learned Models by Identifying Important Features at the Right Resolution". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 4155–4163. DOI: 10.1609/aaai.v33i01.33014155 (cit. on p. 47).

Lei, T., R. Barzilay, and T. Jaakkola (2016). *Rationalizing Neural Predictions*. arXiv: 1606.04155. URL: http://arxiv.org/abs/1606.04155 (cit. on pp. 15, 18).

Li, J., W. Monroe, and D. Jurafsky (2016). *Understanding Neural Networks through Representation Erasure*. arXiv: 1612.08220. URL: http://arxiv.org/abs/1612.08220 (cit. on pp. 48, 49).

Lipton, Z. C. (2016). *The Mythos of Model Interpretability*. arXiv: 1606.03490 [cs, stat]. URL: http://arxiv.org/abs/1606.03490 (cit. on pp. 11, 12, 25).

Lipton, Z. C., D. C. Kale, C. Elkan, and R. Wetzel (2015). *Learning to Diagnose with LSTM Recurrent Neural Networks*. arXiv: 1511.03677 [cs]. URL: http://arxiv.org/abs/1511.03677 (cit. on p. 25).

Lipton, Z. C., D. Kale, and R. Wetzel (2016). "Directly Modeling Missing Data in Sequences with RNNs: Improved Classification of Clinical Time Series". In: *Machine Learning for Healthcare Conference*. Machine Learning for Healthcare Conference, pp. 253–270 (cit. on p. 25).

Lou, Y., R. Caruana, and J. Gehrke (2012). "Intelligible Models for Classification and Regression". In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 150–158 (cit. on p. 13).

Louppe, G., L. Wehenkel, A. Sutera, and P. Geurts (2013). "Understanding Variable Importances in Forests of Randomized Trees". In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., pp. 431–439 (cit. on p. 18).

Loymans, R. J. B., T. P. A. Debray, P. J. Honkoop, et al. (2018). "Exacerbations in Adults with Asthma: A Systematic Review and External Validation of Prediction Models". In: *The Journal of Allergy and Clinical Immunology: In Practice* 6.6, 1942–1952.e15. DOI: 10.1016/j.jaip.2018.02.004 (cit. on p. 28).

Lundberg, S. M. and S.-I. Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 4765–4774 (cit. on pp. 17–22, 68, 69, 81, 92).

Maaten, L. van der and G. Hinton (2008). "Visualizing Data Using T-SNE". In: *Journal of Machine Learning Research* 9 (Nov), pp. 2579–2605 (cit. on p. 19).

Mahendran, A. and A. Vedaldi (2014). *Understanding Deep Image Representations by Inverting Them*. arXiv: 1412.0035. URL: http://arxiv.org/abs/1412.0035 (cit. on p. 18).

Mayampurath, A., L. N. Sanchez-Pinto, K. A. Carey, L.-R. Venable, and M. Churpek (2019). "Combining Patient Visual Timelines with Deep Learning to Predict Mortality". In: *PLOS ONE* 14.7, e0220640. DOI: 10.1371/journal.pone.0220640 (cit. on p. 70).

Miller, T. (2018). *Explanation in Artificial Intelligence: Insights from the Social Sciences*. arXiv: 1706.07269 [cs]. URL: http://arxiv.org/abs/1706.07269 (cit. on pp. 4, 11, 12).

Miotto, R., L. Li, B. A. Kidd, and J. T. Dudley (2016). "Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records". In: *Scientific Reports* 6, p. 26094. DOI: 10.1038/srep26094 (cit. on pp. 24, 25).

Mohseni, S., N. Zarei, and E. D. Ragan (2018). *A Survey of Evaluation Methods and Measures for Interpretable Machine Learning*. arXiv: 1811.11839 [cs]. URL: http://arxiv.org/abs/1811.11839 (cit. on pp. 12, 17).

Montavon, G., W. Samek, and K.-R. Müller (2018). "Methods for Interpreting and Understanding Deep Neural Networks". In: *Digital Signal Processing* 73, pp. 1–15. DOI: 10.1016/j.dsp.2017.10.011 (cit. on p. 12).

Mueller, S. T., R. R. Hoffman, W. Clancey, A. Emrey, and G. Klein (2019). *Explanation in Human-AI Systems: A Literature Meta-Review, Synopsis of Key Ideas and Publications, and Bibliography for Explainable AI*. arXiv: 1902.01876 [cs]. URL: http://arxiv.org/abs/1902.01876 (cit. on p. 12).

Nguyen, D., W. Luo, D. Phung, and S. Venkatesh (2016). *Control Matching via Discharge Code Sequences*. arXiv: 1612.01812. URL: http://arxiv.org/abs/1612.01812 (cit. on p. 25).

Nguyen, P., T. Tran, N. Wickramasinghe, and S. Venkatesh (2017). "Deepr: A Convolutional Net for Medical Records". In: *IEEE Journal of Biomedical and Health Informatics* 21.1, pp. 22–30. DOI: 10.1109/JBHI.2016.2633963 (cit. on p. 26).

Nicodemus, K. K., J. D. Malley, C. Strobl, and A. Ziegler (2010). "The Behaviour of Random Forest Permutation-Based Variable Importance Measures under Predictor Correlation". In: *BMC Bioinformatics* 11.1, p. 110. DOI: 10.1186/1471-2105-11-110 (cit. on pp. 74, 82).

Ojala, M. and G. C. Garriga (2010). "Permutation Tests for Studying Classifier Performance". In: *Journal of Machine Learning Research* 11 (Jun), pp. 1833–1863 (cit. on pp. 22, 69, 76, 78, 87).

Olden, J. D. and D. A. Jackson (2002). "Illuminating the "Black Box": A Randomization Approach for Understanding Variable Contributions in Artificial Neural Networks". In: *Ecological Modelling* 154.1, pp. 135–150. DOI: 10.1016/S0304-3800(02)00064-9 (cit. on p. 19).

Pastor, E. and E. Baralis (2019). "Explaining Black Box Models by Means of Local Rules". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC '19. Limassol, Cyprus: Association for Computing Machinery, pp. 510–517. DOI: 10.1145/3297280.3297328 (cit. on p. 19).

Pham, T., T. Tran, D. Phung, and S. Venkatesh (2016). *DeepCare: A Deep Dynamic Memory Model for Predictive Medicine*. arXiv: 1602.00357 [cs, stat]. URL: http://arxiv.org/abs/1602.00357 (cit. on p. 25).

Plumb, G., M. Al-Shedivat, Á. A. Cabrera, et al. (2020). "Regularizing Black-Box Models for Improved Interpretability". In: *Advances in Neural Information Processing Systems* 33 (cit. on p. 15).

Preece, A., D. Harborne, D. Braines, R. Tomsett, and S. Chakraborty (2018). *Stakeholders in Explainable AI*. arXiv: 1810.00184. URL: https://arxiv.org/abs/1810.00184 (cit. on p. 4).

Ptitsyn, A. A., S. Zvonic, S. A. Conrad, et al. (2006). "Circadian Clocks Are Resounding in Peripheral Tissues". In: *PLOS Computational Biology* 2.3, e16. DOI: 10.1371/journal.pcbi.0020016 (cit. on p. 75).

Rabiner, L. (1989). "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286. DOI: 10.1109/5.18626 (cit. on p. 35).

Rajkomar, A., E. Oren, K. Chen, et al. (2018). "Scalable and Accurate Deep Learning with Electronic Health Records". In: *npj Digital Medicine* 1.1, p. 18. DOI: 10.1038/s41746-018-0029-1 (cit. on pp. 15, 24–26).

Razavian, N. and D. Sontag (2015). *Temporal Convolutional Neural Networks for Diagnosis from Lab Tests*. arXiv: 1511.07938 [cs]. URL: http://arxiv.org/abs/1511.07938 (cit. on p. 25).

Reddel, H. K., D. R. Taylor, E. D. Bateman, et al. (2009). "An Official American Thoracic Society/European Respiratory Society Statement: Asthma Control and Exacerbations". In: *American Journal of Respiratory and Critical Care Medicine* 180.1, pp. 59–99. DOI: 10.1164/rccm.200801-060ST (cit. on p. 30).

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: Association for Computing Machinery, pp. 1135–1144. DOI: 10.1145/2939672.2939778. arXiv: 1602.04938 (cit. on pp. 17–20, 22, 48, 49, 68, 92).

Ribeiro, M. T., S. Singh, and C. Guestrin (2018). "Anchors: High-Precision Model-Agnostic Explanations". In: *Thirty-Second Proceedings of AAAI Conference on Artificial Intelligence*. Vol. 18, pp. 1527–1535 (cit. on pp. 17–19).

Ross, A. S., M. C. Hughes, and F. Doshi-Velez (2017). *Right for the Right Reasons: Training Differentiable Models by Constraining Their Explanations*. arXiv: 1703.03717. URL: http://arxiv.org/abs/1703.03717 (cit. on p. 15).

Schwab, P. and W. Karlen (2019). *CXPlain: Causal Explanations for Model Interpretation under Uncertainty*. arXiv: 1910.12336 [cs, stat]. URL: http://arxiv.org/abs/1910.12336 (cit. on pp. 17, 20, 22, 69, 81, 92).

Scott, K. M., M. Von Korff, J. Ormel, et al. (2007). "Mental Disorders among Adults with Asthma: Results from the World Mental Health Survey". In: *General Hospital Psychiatry* 29.2, pp. 123–133. DOI: 10.1016/J.GENHOSPPSYCH.2006.12.006 (cit. on p. 65).

Selbst, A. D. and S. Barocas (2018). *The Intuitive Appeal of Explainable Machines*. SSRN Scholarly Paper ID 3126971. Rochester, NY: Social Science Research Network. DOI: 10.2139/ssrn.3126971 (cit. on p. 12).

Shickel, B., P. J. Tighe, A. Bihorac, and P. Rashidi (2017). "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis". In: *IEEE Journal of Biomedical and Health Informatics*. DOI: 10.1109/JBHI.2017.2767063. arXiv: 1706.03446 (cit. on pp. 23, 24, 26).

Shrikumar, A., P. Greenside, and A. Kundaje (2017). *Learning Important Features Through Propagating Activation Differences*. arXiv: 1704.02685. URL: http://arxiv.org/abs/1704.02685 (cit. on p. 19).

Simon, H. A. (1992). "What Is an "Explanation" of Behavior?" In: *Psychological Science* 3.3, pp. 150–161. DOI: 10.1111/j.1467-9280.1992.tb00017.x (cit. on p. 11).

Simonyan, K., A. Vedaldi, and A. Zisserman (2013). *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. arXiv: 1312.6034 [cs]. URL: http://arxiv.org/abs/1312.6034 (cit. on pp. 18, 19, 21).

Storch, K.-F., O. Lipan, I. Leykin, et al. (2002). "Extensive and Divergent Circadian Gene Expression in Liver and Heart". In: *Nature* 417.6884 (6884), pp. 78–83. DOI: 10.1038/nature744 (cit. on p. 75).

Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis (2008). "Conditional Variable Importance for Random Forests". In: *BMC Bioinformatics* 9.1, p. 307. DOI: 10.1186/1471-2105-9-307 (cit. on pp. 22, 69, 78, 82, 112).

Štrumbelj, E. and I. Kononenko (2014). "Explaining Prediction Models and Individual Predictions with Feature Contributions". In: *Knowledge and Information Systems* 41.3, pp. 647–665. DOI: 10.1007/s10115-013-0679-x (cit. on pp. 19, 22).

Suk, H.-I. and D. Shen (2013). "Deep Learning-Based Feature Representation for AD/MCI Classification". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 583–590 (cit. on pp. 24, 25).

Sundararajan, M., A. Taly, and Q. Yan (2017). "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning*

- *Volume 70* (Sydney, NSW, Australia). ICML'17. JMLR.org, pp. 3319–3328 (cit. on pp. 19, 21).

Suresh, H., N. Hunt, A. Johnson, et al. (2017). *Clinical Intervention Prediction and Understanding Using Deep Networks*. arXiv: 1705.08498 [cs]. URL: http://arxiv.org/abs/1705.08498 (cit. on pp. 21, 22, 49, 68, 92).

Sutskever, I., O. Vinyals, and Q. V. Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., pp. 3104–3112 (cit. on p. 25).

Tansey, W., V. Veitch, H. Zhang, R. Rabadan, and D. M. Blei (2019). *The Holdout Randomization Test: Principled and Easy Black Box Feature Selection*. arXiv: 1811.00645 [stat]. URL: http://arxiv.org/abs/1811.00645 (cit. on p. 76).

Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288. JSTOR: 2346178 (cit. on p. 32).

To, T., S. Stanojevic, G. Moores, et al. (2012). "Global Asthma Prevalence in Adults: Findings from the Cross-Sectional World Health Survey". In: *BMC Public Health* 12.1, p. 204. DOI: 10.1186/1471-2458-12-204 (cit. on p. 27).

Toloşi, L. and T. Lengauer (2011). "Classification with Correlated Features: Unreliability of Feature Ranking and Solutions". In: *Bioinformatics* 27.14, pp. 1986–1994. DOI: 10.1093/bioinformatics/btr300 (cit. on p. 84).

Tonekaboni, S., S. Joshi, K. Campbell, D. K. Duvenaud, and A. Goldenberg (2020). "What Went Wrong and When? Instance-Wise Feature Importance for Time-Series Black-Box Models". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc. arXiv: 2003.02821 (cit. on pp. 21, 69, 91).

Tran, T., T. D. Nguyen, D. Phung, and S. Venkatesh (2015). "Learning Vector Representation of Medical Objects via EMR-Driven Nonnegative Restricted Boltzmann Machines (eNRBM)". In: *Journal of Biomedical Informatics* 54, pp. 96–105. DOI: 10.1016/j.jbi.2015.01.012 (cit. on pp. 24, 26).

Wan, C. and A. A. Freitas (2018). "An Empirical Evaluation of Hierarchical Feature Selection Methods for Classification in Bioinformatics Datasets with Gene

Ontology-Based Features". In: *Artificial Intelligence Review* 50.2, pp. 201–240. DOI: 10.1007/s10462-017-9541-y (cit. on p. 52).

Wark, P. a. B. and P. G. Gibson (2006). "Asthma Exacerbations · 3: Pathogenesis". In: *Thorax* 61.10, pp. 909–915. DOI: 10.1136/thx.2005.045187. pmid: 17008482 (cit. on p. 27).

Weld, D. S. and G. Bansal (2018). *The Challenge of Crafting Intelligible Intelligence*. arXiv: 1803.04263 [cs]. URL: http://arxiv.org/abs/1803.04263 (cit. on p. 11).

Winer, R. A., X. Qin, T. Harrington, J. Moorman, and H. Zahran (2012). "Asthma Incidence among Children and Adults: Findings from the Behavioral Risk Factor Surveillance System Asthma Call-Back Survey—United States, 2006–2008". In: *Journal of Asthma* 49.1, pp. 16–22. DOI: 10.3109/02770903.2011.637594 (cit. on p. 27).

Wu, M., M. C. Hughes, S. Parbhoo, et al. (2017). *Beyond Sparsity: Tree Regularization of Deep Models for Interpretability*. arXiv: 1711.06178 [cs, stat]. URL: http://arxiv.org/abs/1711.06178 (cit. on pp. 15, 21, 68).

Xiao, C., E. Choi, and J. Sun (2018). "Opportunities and Challenges in Developing Deep Learning Models Using Electronic Health Records Data: A Systematic Review". In: *Journal of the American Medical Informatics Association*. DOI: 10.1093/jamia/ocy068 (cit. on pp. 23–25).

Xue, B., D. Li, C. Lu, et al. (2021). "Use of Machine Learning to Develop and Evaluate Models Using Preoperative and Intraoperative Data to Identify Risks of Postoperative Complications". In: *JAMA Network Open* 4.3, e212240. DOI: 10.1001/jamanetworkopen.2021.2240 (cit. on p. 70).

Yang, Z., A. Zhang, and A. Sudjianto (2020). "Enhancing Explainability of Neural Networks Through Architecture Constraints". In: *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12. DOI: 10.1109/TNNLS.2020.3007259 (cit. on p. 15).

Yekutieli, D. (2008). "Hierarchical False Discovery Rate–Controlling Methodology". In: *Journal of the American Statistical Association* 103.481, pp. 309–316. DOI: 10.1198/016214507000001373 (cit. on pp. 53, 77).

Zeiler, M. D. and R. Fergus (2014). "Visualizing and Understanding Convolutional Networks". In: Springer, Cham, pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53 (cit. on pp. 19, 20, 22, 48, 49, 69, 81, 92).

Zhang, Q., Y. N. Wu, and S.-C. Zhu (2018). "Interpretable Convolutional Neural Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8827–8836. DOI: 10.1109/CVPR.2018.00920 (cit. on p. 18).

Zhang, Y., X. Yang, J. Ivy, and M. Chi (2019). "ATTAIN: Attention-Based Time-Aware LSTM Networks for Disease Progression Modeling". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}. Macao, China: International Joint Conferences on Artificial Intelligence Organization, pp. 4369–4375. DOI: 10.24963/ijcai.2019/607 (cit. on pp. 21, 68).

Zhou, B., Y. Sun, D. Bau, and A. Torralba (2018). "Interpretable Basis Decomposition for Visual Explanation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 119–134 (cit. on pp. 20, 48).