Feature Importance Explanations for Temporal Black-Box Models

Akshay Sood, Mark Craven

Department of Computer Sciences
Department of Biostatistics and Medical Informatics
University of Wisconsin-Madison
Madison, Wisconsin, U.S.A.
sood@cs.wisc.edu, craven@biostat.wisc.edu

Abstract

Models in the supervised learning framework may capture rich and complex representations over the features that are hard for humans to interpret. Existing methods to explain such models are often specific to architectures and data where the features do not have a time-varying component. In this work, 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.

Introduction

The last decade has seen an explosion in models that learn rich representations over large, complex parameter spaces. These 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 has thus been a concomitant rise in methods to generate explanations for black-box models. Existing research 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. In this work, we present a method that advances the state of the art in model explanation by being specifically focused on temporal models, being model-agnostic, and providing global explanations.

Most existing explanation methods are designed for tabular, as opposed to temporal, representations. 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 (Karpathy, Johnson, and Fei-Fei 2015; Suresh et al. 2017; Ismail et al. 2019) and attention-based models (Choi et al. 2016; 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, Alvarez-Melis, and Jaakkola 2018). However, these approaches require specific model architectures or training-time alterations, limiting their applicability.

Model-agnostic methods such as LIME (Ribeiro, Singh, and Guestrin 2016) and SHAP (Lundberg 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 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.

Whereas most work in explanation methods 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 falls under the class of perturbation or removal-based methods for model explanation (Covert, Lundberg, and Lee 2020a). This class includes other model-agnostic and global methods such as Feature Occlusion (FO) (Zeiler and Fergus 2014) and CXPlain (Schwab and Karlen 2019), which perturb features by setting their values to zero but which are focused on tabular data, with a few exceptions (Suresh et al. 2017; Tonekaboni et al. 2020). 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 (Strobl et al. 2008; Ojala and Garriga 2010; Altmann et al. 2010; Gregorutti, Michel, and Saint-Pierre 2015; Fisher, Rudin, and Dominici 2019; Zhou and Hooker 2020). The

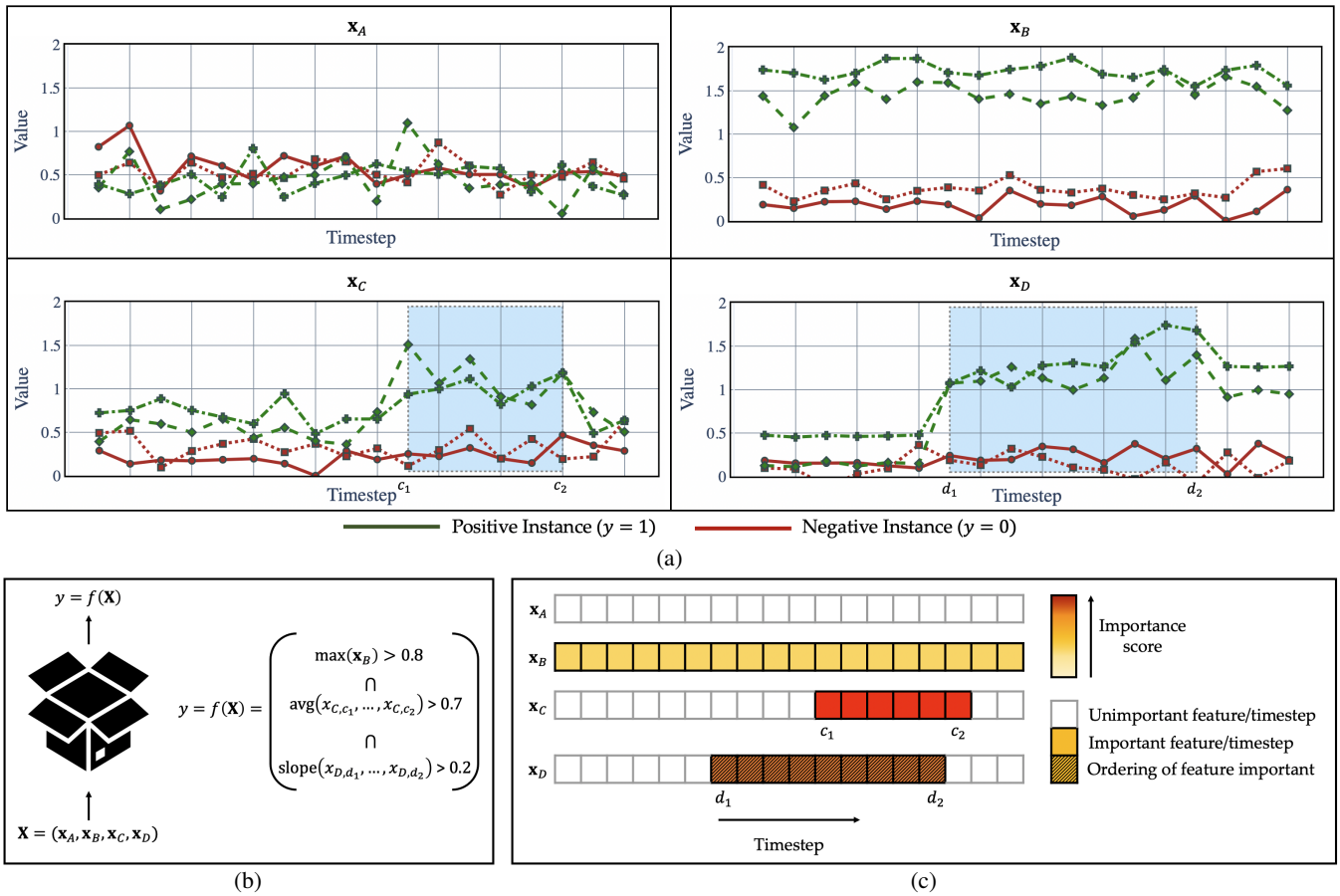


Figure 1: (a) Time series for positive (green) and negative (red) instances for four different features, illustrating temporal properties of the features that a learned model may capture. (b) A trained binary classification model over the four time-varying features, whose underlying function uses the 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.

simplicity and generality of permutations makes them attractive as a tool for model-agnostic explanation. Whereas 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, and does the following global analyses: (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 model types. There are many applications that match the setting we address, such as numerous clinical risk assessment approaches that make predictions relative to an index time: these include hospital readmission, inpatient deterioration, post-hospitalization complications, post-surgery complications, and asthma exacerbations, among others (Ashfaq et al. 2019; Mayampurath et al. 2019; Kawaler et al. 2012; Xue et al. 2021; Cobian et al. 2020). Figure 1 illustrates the setting and our approach.

Methods

Identifying Important Features/Timesteps

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:

$$f\left(\mathbf{x}_j^{(i,l)}\right) = f\left(x_1^{(i)}, x_2^{(i)}, \ldots, x_j^{(l)}, \ldots, x_D^{(i)}\right) \quad (1)$$

where the value of feature j has been replaced by its corresponding value from instance l, as shown in Figure 2a. Then, we can compute the change in loss between the perturbed and original losses as:

$$\Delta\mathcal{L}_j^{(i,l)} = \mathcal{L}\left[y^{(i)}, f\left(\mathbf{x}_j^{(i,l)}\right)\right] - \mathcal{L}\left[y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right]. \quad (2)$$

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:

$$I\left(f, j\right) = \frac{1}{|\mathcal{P}_j|} \sum_{\Pi \in \mathcal{P}_j} \left[\frac{1}{M} \sum_{i=1}^{M} \Delta\mathcal{L}_j^{(i,\pi_i)}\right]. \quad (3)$$

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. Equation 3 characterizes a feature as important if the model's performance degrades on average when the feature is perturbed via permutation, as captured by the effect of the perturbation on the model loss rather than the model output (Covert, Lundberg, and Lee 2020a). We use hypothesis testing to test the significance of this degradation, as outlined in a subsequent section.

Temporal models. We extend the idea of permuting features to assess their 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 for instance i w.r.t. feature j is given by:

$$f\left(\mathbf{X}_{j,[k_1,k_2]}^{(i,l)}\right) = f\left(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_{j,[k_1,k_2]}^{(i,l)}, \ldots, \mathbf{x}_D^{(i)}\right) \quad (4)$$

where $\mathbf{x}_{j,[k_1,k_2]}^{(i,l)}$ 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 2b.

$$\mathbf{x}_{j,[k_1,k_2]}^{(i,l)} = \left\langle x_{j,1}^{(i)}, x_{j,2}^{(i)}, \ldots, x_{j,k_1}^{(l)}, \ldots, x_{j,k_2}^{(l)}, \ldots, x_{j,L}^{(i)} \right\rangle. \quad (5)$$

We compute the perturbed loss $\mathcal{L}\left[y^{(i)}, f\left(\mathbf{X}_{j,[k_1,k_2]}^{(i,l)}\right)\right]$ and the change in loss (Equation 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:

$$I\left(f, j, [k_1, k_2]\right) = \frac{1}{|\mathcal{P}_j|} \sum_{\Pi \in \mathcal{P}_j} \left[\frac{1}{M} \sum_{i=1}^{M} \Delta\mathcal{L}_{j,[k_1,k_2]}^{(i,\pi_i)}\right]. \quad (6)$$

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

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 across the distribution of instances. 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:

$$I(f, j, \tilde{W}) < \left(\frac{1 - \gamma}{2}\right) I\left(f, j, [1, L]\right) \quad (7)$$

where $\gamma : \ 0 < \gamma < 1$ controls the degree to which the model focuses on W^* and affects the size of the identified windows. γ may be tuned by the user based on the desired conciseness of the generated explanations. 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 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 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 7. We select the final boundary estimates k_1 and

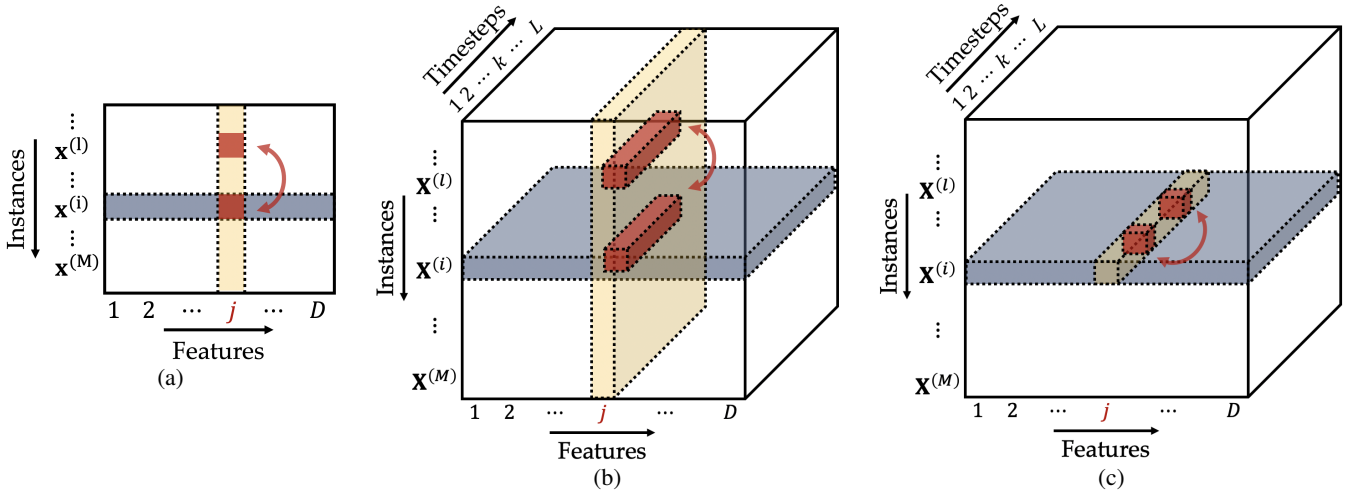


Figure 2: Perturbation for instance i and feature j to compute feature importance. (a) Data matrix showing the replacement of the value of feature j in instance i from instance l. (b) Data tensor showing the replacement of a window of feature j in instance i from the corresponding window of instance l . (c) Time series $\mathbf{x}_j^{(i)}$ showing the exchange of feature values at two timesteps.

$k_2 = L - |\hat{W}_S|$ to characterize the important window W^*. We then compute its importance score using Equation 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 $I(f, j, W^*) > \gamma I(f, j, [1, L])$.

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. 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 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:

$$f\left(\mathbf{X}^{(i)}_{j,\Pi_{[k_1,k_2]}}\right) = f\left(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}^{(i)}_{j,\Pi_{[k_1,k_2]}}, \ldots, \mathbf{x}_D^{(i)}\right) \tag{8}$$

over the permuted time series for instance i and feature j:

$$\mathbf{x}^{(i)}_{j,\Pi_{[k_1,k_2]}} = \langle x^{(i)}_{j,1}, \ldots, x^{(i)}_{j,\pi_{k_1}}, \ldots, x^{(i)}_{j,\pi_{k_2}}, \ldots x^{(i)}_{j,L} \rangle. \tag{9}$$

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.

Hypothesis Testing and False Discovery Rate Control

Existing work has leveraged hypothesis testing in conjunction with permutations (Golland et al. 2005; Ojala and Garriga 2010; Lee, Sood, and Craven 2019; Burns, Thomason, and Tansey 2020) to examine feature importance for models with tabular representations. We extend this approach to temporal models by using permutation tests, a type of widely used non-parametric statistical test, to test the significance of important sequences, 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:

$$\hat{p} = \frac{|\left\{\Pi \in \mathcal{P}_j : \bar{\mathcal{L}}_\Pi \leq \bar{\mathcal{L}}\right\}| + 1}{|\mathcal{P}_j| + 1}. \tag{10}$$

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 10.

Depending on the permuted quantity, we can use Equation 10 to test the overall importance, window importance, and ordering importance of feature j. These tests may be organized as a hierarchy, as shown in Figure 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,

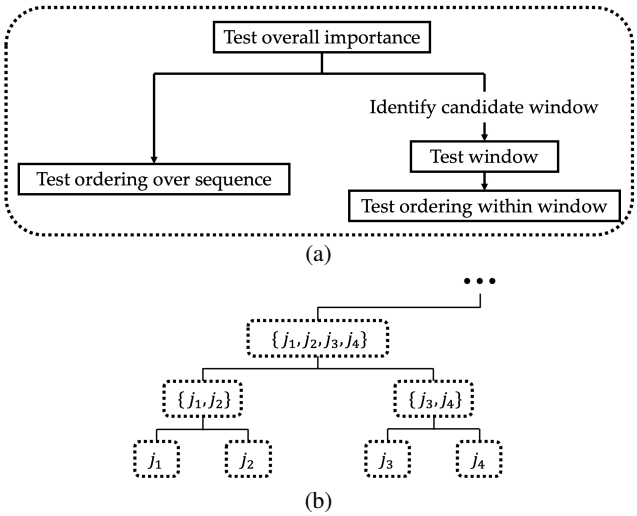


Figure 3: (a) A hierarchy of tests used to check a given feature for its (i) overall importance, (ii) important window and (iii) the importance of ordering within the window. (b) A hierarchy over the features, where each node is tested using the hierarchy shown in (a). Feature groups are tested via joint 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.

as shown in Figure 3b and detailed in Lee, Sood, and Craven (2019).

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. Software as well as supplementary material for TIME are available at https://github.com/Craven-Biostat-Lab/anamod.

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 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). More details of the synthetic data and models along with software are available at https://github.com/Craven-Biostat-Lab/synmod.

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 a given method are indicated in parentheses.

- **LIME** (Ribeiro, Singh, and Guestrin 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 popularity 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 reference values sampled from a uniform distribution to analyze LSTM models (FO-u).
- **CXPlain** (Schwab and Karlen 2019): a global method that trains a surrogate explanation model and perturbs features using reference values to calculate importance scores.
- **SAGE** (Covert, Lundberg, and Lee 2020b): a Shapley value-based 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 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 tabular feature representations, 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 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, 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 impor-

Method	Features		Timesteps		Windows	Runtime (seconds)
	Power	FDR	Power	FDR		
TIME	$\mathbf{0.930 \pm 0.111}$	0.037 ± 0.080	$\mathbf{0.923 \pm 0.138}$	0.054 ± 0.124	$\mathbf{4.87 \pm 0.76}$	371 ± 116
TIME-n	0.922 ± 0.113	$\mathbf{0.018 \pm 0.058}$	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	$\mathbf{29 \pm 8}$
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	$\mathbf{0.008 \pm 0.022}$	2.71 ± 3.92	1478 ± 663

Table 1: Comparison between different explanation methods on synthetic data, indicating the average power and FDR for detecting relevant features and timesteps, the average number of windows, and the median runtime.

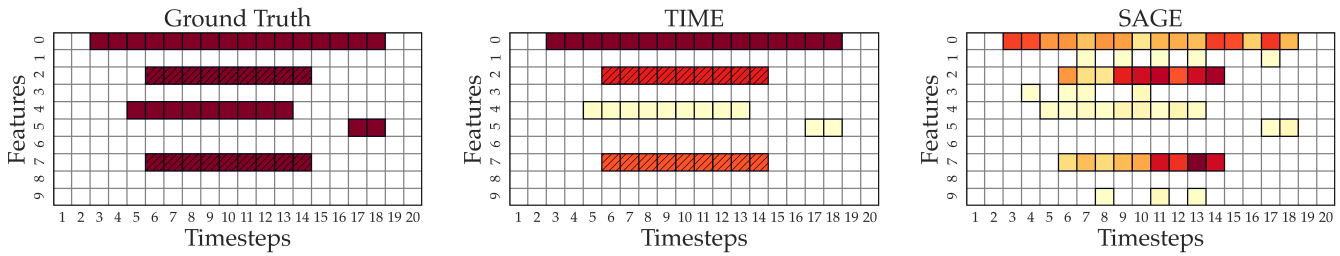
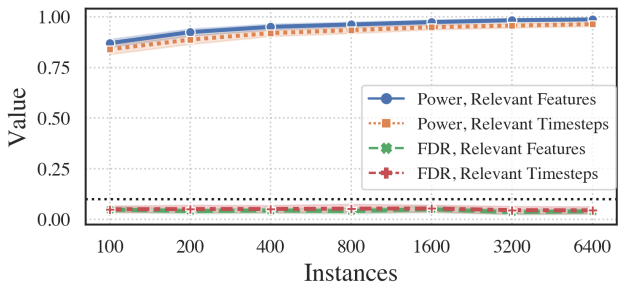


Figure 4: Heat maps showing (a) relevant features, windows and ordering for the ground truth model, (b) TIME importance scores, indicating important features, windows and ordering, and (c) SAGE importance scores. Hatched textures indicate features sensitive to ordering. Darker shades indicate higher scores.

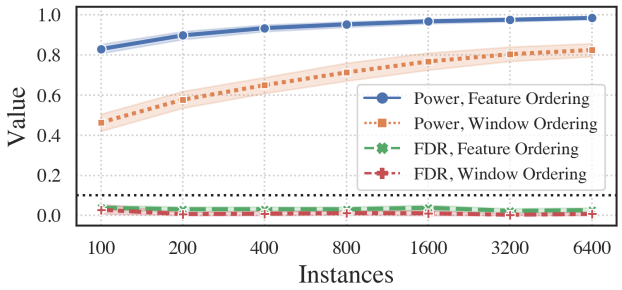
tance by averaging non-zero importance scores across the timesteps belonging to the feature. We sort timesteps in decreasing order of importance scores and report the n features or timesteps with the highest scores, where n is determined by the number of relevant features and timesteps in the ground truth model. Since TIME identifies statistical significance in addition to scores for important features and windows, 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 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 (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.

Figure 4 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 TIME and SAGE. For the ground truth model, boxes corresponding to relevant timesteps are shown in 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 SAGE since it is not able to detect the significance of ordering. TIME assigns importance scores to windows for each feature, while SAGE, as well as other baseline 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. SAGE assigns non-zero importance scores to all the relevant timesteps, but in some cases, irrelevant timesteps are ranked above relevant ones, adversely affecting its power and FDR for detecting important features. It also produces more fragmented explanations due to the larger number of



(a)



(b)

Figure 5: Average power and FDR 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 black horizontal line represents the 0.1 level at which FDR is controlled.

identified windows.

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.

Figure 5 shows the results of this analysis for regression models. Figure 5a 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.

Figure 5b 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 overall 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 FDRs are well-controlled at the 0.1 level. We detect window ordering with lower power compared to feature ordering due to the greater difficulty of the task, and the fact that relevant features that are not detected as important are not assessed for important windows or their ordering.

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 proposed 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 benchmarking paper (Harutyunyan et al. 2017).

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

Table 2: Comparison of baseline methods for MIMIC-III LSTM models retrained after feature selection, using the top-scoring features and timesteps. SAGE is not included since it failed to converge in a reasonable amount of time, and PERM-f is not included since it did not report any important features after performing FDR control.

We use the validation set to examine 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 6 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 unsurprising 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

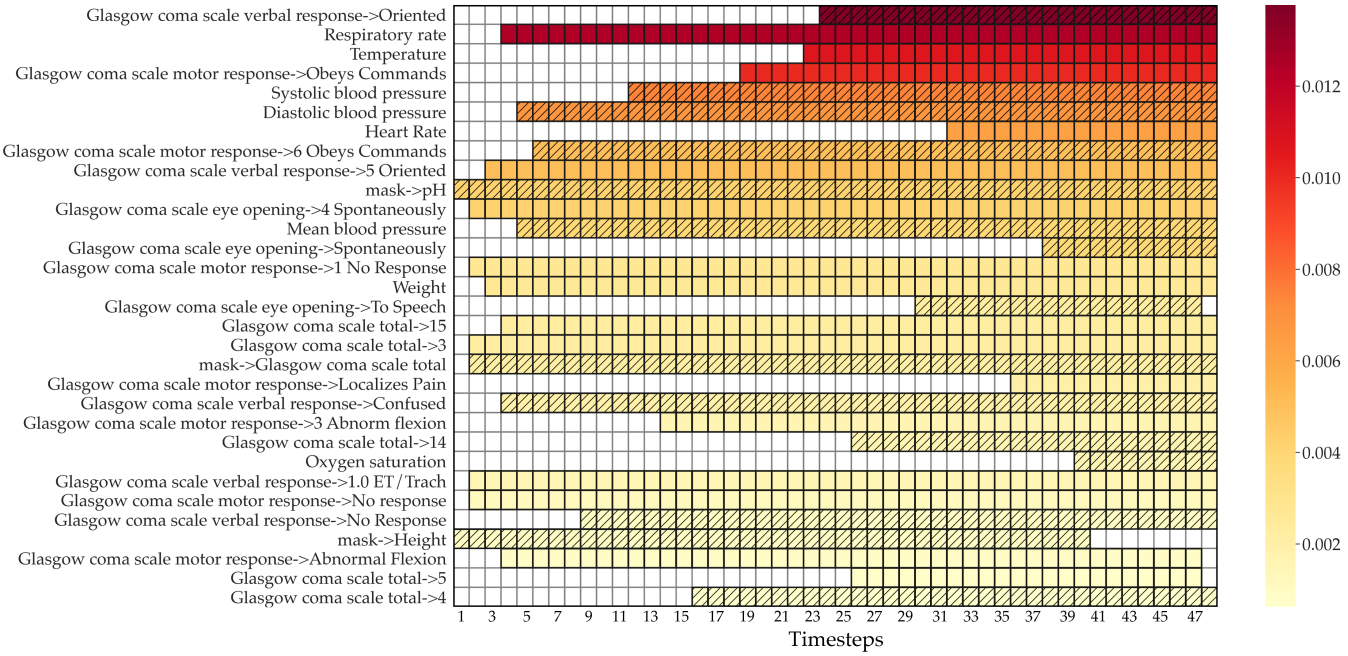


Figure 6: Heat map showing the TIME analysis of a MIMIC-III LSTM model trained to predict in-hospital mortality. Out of a total of 76 features, 31 were identified as important and are shown in decreasing order of their importance scores. Each row corresponds to a single feature and shows the window corresponding 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.

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, 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 2 shows the results of this comparison. The AUROC for the retrained 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 interpretability 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.

Conclusions

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 show 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 isolated timesteps. Like other marginal permutation-based feature importance methods, TIME is fairly efficient to compute (Gregorutti, Michel, and Saint-Pierre 2015). We apply TIME to an LSTM trained to predict risk of in-hospital mortality from ICU data, and we identify salient features, windows and ordering in patients' clinical histories that the model focuses on. We show that a model trained using features and timesteps selected using this analysis performs nearly as well as the original model, and produces more concise explanations than comparable baseline methods.

We plan to address some limitations of TIME in future work. Our approach for permutation across sequences cur-

rently assumes regularly sampled, time-aligned and fixed-length sequences. We can extend the approach to consider windows that are aligned in other ways, such as on an absolute scale (e.g., dates on the calendar) or a relative scale (e.g., patient age). We assume that there exists a single contiguous window that is important, which could be generalized. However, we note that that if the model actually focuses on multiple windows, TIME will degrade gracefully by identifying a single important window that subsumes multiple windows. Like many other explanation methods, TIME may perform out-of-distribution perturbations (Kumar et al. 2020), leading to an inflation of importance scores for correlated features. One approach to ameliorate this problem is by perturbing groups of correlated features together, which is supported by TIME. We also plan to explore the use of conditional permutations (Strobl et al. 2008) for this purpose.

Acknowledgments

This research was supported by NIH grants UL1 TR002373 and U54 AI117924, and the University of Wisconsin Institute for Clinical and Translational Research.

References

Altmann, A.; Toloşi, L.; Sander, O.; and Lengauer, T. 2010. Permutation Importance: A Corrected Feature Importance Measure. *Bioinformatics*, 26(10): 1340–1347.

Ashfaq, A.; Sant'Anna, A.; Lingman, M.; and Nowaczyk, S. 2019. Readmission Prediction Using Deep Learning on Electronic Health Records. *Journal of Biomedical Informatics*, 97: 103256.

Benjamini, Y.; and Hochberg, Y. 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1): 289–300.

Bento, J.; Saleiro, P.; Cruz, A. F.; Figueiredo, M. A. T.; and Bizarro, P. 2020. TimeSHAP: Explaining Recurrent Models through Sequence Perturbations. *arXiv:2012.00073 [cs]*.

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32.

Burns, C.; Thomason, J.; and Tansey, W. 2020. Interpreting Black Box Models via Hypothesis Testing. *arXiv:1904.00045 [cs, stat]*.

Choi, E.; Bahadori, M. T.; Sun, J.; Kulas, J.; Schuetz, A.; and Stewart, W. 2016. RETAIN: An Interpretable Predictive Model for Healthcare Using Reverse Time Attention Mechanism. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 3504–3512. Curran Associates, Inc.

Cobian, A.; Abbott, M.; Sood, A.; Sverchkov, Y.; Hanrahan, L.; Guilbert, T.; and Craven, M. 2020. Modeling Asthma Exacerbations from Electronic Health Records. *AMIA Summits on Translational Science Proceedings*, 2020: 98–107.

Covert, I.; Lundberg, S. M.; and Lee, S.-I. 2020a. Explaining by Removing: A Unified Framework for Model Explanation. *arXiv:2011.14878 [cs, stat]*.

Covert, I.; Lundberg, S. M.; and Lee, S.-I. 2020b. Understanding Global Feature Contributions With Additive Importance Measures. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc.

Doshi-Velez, F.; and Kim, B. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]*.

Fisher, A.; Rudin, C.; and Dominici, F. 2019. All Models Are Wrong, but Many Are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research*, 20(177): 1–81.

Golland, P.; Liang, F.; Mukherjee, S.; and Panchenko, D. 2005. Permutation Tests for Classification. In Auer, P.; and Meir, R., eds., *Learning Theory*, Lecture Notes in Computer Science, 501–515. Berlin, Heidelberg: Springer. ISBN 978-3-540-31892-7.

Gregorutti, B.; Michel, B.; and Saint-Pierre, P. 2015. Grouped Variable Importance with Random Forests and Application to Multiple Functional Data Analysis. *Computational Statistics & Data Analysis*, 90: 15–35.

Harutyunyan, H.; Khachatrian, H.; Kale, D. C.; Steeg, G. V.; and Galstyan, A. 2017. Multitask Learning and Benchmarking with Clinical Time Series Data. *arXiv:1703.07771 [cs, stat]*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Ibrahim, M.; Louie, M.; Modarres, C.; and Paisley, J. 2019. Global Explanations of Neural Networks: Mapping the Landscape of Predictions. *arXiv:1902.02384 [cs, stat]*.

Ismail, A. A.; Gunady, M.; Bravo, H. C.; and Feizi, S. 2020. Benchmarking Deep Learning Interpretability in Time Series Predictions. *arXiv:2010.13924 [cs, stat]*.

Ismail, A. A.; Gunady, M.; Pessoa, L.; Bravo, H. C.; and Feizi, S. 2019. Input-Cell Attention Reduces Vanishing Saliency of Recurrent Neural Networks. *arXiv:1910.12370 [cs, stat]*.

Johnson, A. E. W.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; and Mark, R. G. 2016. MIMIC-III, a Freely Accessible Critical Care Database. *Scientific Data*, 3: 160035.

Karpathy, A.; Johnson, J.; and Fei-Fei, L. 2015. Visualizing and Understanding Recurrent Networks. *arXiv:1506.02078 [cs]*.

Kawaler, E.; Cobian, A.; Peissig, P.; Cross, D.; Yale, S.; and Craven, M. 2012. Learning to Predict Post-Hospitalization VTE Risk from EHR Data. *AMIA Annual Symposium Proceedings*, 2012: 436–445.

Kumar, I. E.; Venkatasubramanian, S.; Scheidegger, C.; and Friedler, S. 2020. Problems with Shapley-value-based Explanations as Feature Importance Measures. *arXiv:2002.11097 [cs, stat]*.

Lee, G.-H.; Alvarez-Melis, D.; and Jaakkola, T. S. 2018. Game-Theoretic Interpretability for Temporal Modeling. *arXiv:1807.00130 [cs, stat]*.

Lee, K.; Sood, A.; and Craven, M. 2019. Understanding Learned Models by Identifying Important Features at the Right Resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4155–4163.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 4765–4774. Curran Associates, Inc.

Mayampurath, A.; Sanchez-Pinto, L. N.; Carey, K. A.; Venable, L.-R.; and Churpek, M. 2019. Combining Patient Visual Timelines with Deep Learning to Predict Mortality. *PLOS ONE*, 14(7): e0220640.

Nicodemus, K. K.; Malley, J. D.; Strobl, C.; and Ziegler, A. 2010. The Behaviour of Random Forest Permutation-Based Variable Importance Measures under Predictor Correlation. *BMC Bioinformatics*, 11(1): 110.

Ojala, M.; and Garriga, G. C. 2010. Permutation Tests for Studying Classifier Performance. *Journal of Machine Learning Research*, 11(Jun): 1833–1863.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 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, 1135–1144. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-4232-2.

Schwab, P.; and Karlen, W. 2019. CXPlain: Causal Explanations for Model Interpretation under Uncertainty. *arXiv:1910.12336 [cs, stat]*.

Strobl, C.; Boulesteix, A.-L.; Kneib, T.; Augustin, T.; and Zeileis, A. 2008. Conditional Variable Importance for Random Forests. *BMC Bioinformatics*, 9(1): 307.

Suresh, H.; Hunt, N.; Johnson, A.; Celi, L. A.; Szolovits, P.; and Ghassemi, M. 2017. Clinical Intervention Prediction and Understanding Using Deep Networks. *arXiv:1705.08498 [cs]*.

Tonekaboni, S.; Joshi, S.; Campbell, K.; Duvenaud, D. K.; and Goldenberg, A. 2020. What Went Wrong and When? Instance-wise Feature Importance for Time-Series Black-Box Models. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc.

Wu, M.; Hughes, M. C.; Parbhoo, S.; Zazzi, M.; Roth, V.; and Doshi-Velez, F. 2017. Beyond Sparsity: Tree Regularization of Deep Models for Interpretability. *arXiv:1711.06178 [cs, stat]*.

Xue, B.; Li, D.; Lu, C.; King, C. R.; Wildes, T.; Avidan, M. S.; Kannampallil, T.; and Abraham, J. 2021. Use of Machine Learning to Develop and Evaluate Models Using Preoperative and Intraoperative Data to Identify Risks of Postoperative Complications. *JAMA Network Open*, 4(3): e212240.

Yekutieli, D. 2008. Hierarchical False Discovery Rate–Controlling Methodology. *Journal of the American Statistical Association*, 103(481): 309–316.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, 818–833. Cham: Springer International Publishing. ISBN 978-3-319-10590-1.

Zhang, Y.; Yang, X.; Ivy, J.; and Chi, M. 2019. ATTAIN: Attention-based Time-Aware LSTM Networks for Disease Progression Modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 4369–4375. Macao, China: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-4-1.

Zhou, Z.; and Hooker, G. 2020. Unbiased Measurement of Feature Importance in Tree-Based Methods. *arXiv:1903.05179 [cs, stat]*.