Learning Optimal Summaries of Clinical Time-series with Concept Bottleneck Models

Carissa Wu CARISSAWU@COLLEGE.HARVARD.EDU
Harvard University

Sonali Parbhoo S.PARBHOO@IMPERIAL.AC.UK
Imperial College London, Harvard University

Marton Havasi MHAVASI@SEAS.HARVARD.EDU
Harvard University

Finale Doshi-Velez FINALE@SEAS.HARVARD.EDU
Harvard University

Abstract

Despite machine learning models' state-of-the-art performance in numerous clinical prediction and intervention tasks, their complex black-box processes pose a great barrier to their real-world deployment. Clinical experts must be able to understand the reasons behind a model's recommendation before taking action, as it is crucial to assess for criteria other than accuracy, such as trust, safety, fairness, and robustness. In this work, we enable human inspection of clinical timeseries prediction models by learning *concepts*, or groupings of features into high-level clinical ideas such as illness severity or kidney function. We also propose an optimization method which then selects the most important features within each concept, learning a collection of sparse prediction models that are sufficiently expressive for examination. On a real-world task of predicting vasopressor onset in ICU units, our algorithm achieves predictive performance comparable to state-of-the-art deep learning models while learning concise groupings conducive for clinical inspection.[1]

1. Introduction

State-of-the-art machine learning models have demonstrated strong predictive performance in several high-stakes clinical contexts, such as predicting in-hospital mortality risk (Awad et al., 2017; Ghassemi et al., 2015; Rajkomar et al., 2018; Purushotham et al., 2018), sepsis onset (Fleuren et al., 2020; Nemati et al., 2018; Futoma et al., 2017), respiratory distress (Zeiberg et al., 2019; Le et al., 2020), and deterioration due to COVID-19 (Wynants et al., 2020). However, for such models to be adopted in practice, it is crucial to assess performance with criteria other than accuracy, such as safety, fairness, generalizability, and robustness (Hoffman et al., 2016; Bolukbasi et al., 2016; Strakowski et al., 2003; Futoma et al., 2020; Beam et al., 2020; Szegedy et al., 2014).

[1] The code is publicly available at https://github.com/dtak/optimal-summaries-public

In particular, interpretable models (those which can be inspected by a human) enable not only statistical validation, but also provide an extra layer of human-based validation that can be helpful in high-stake settings such as healthcare. Yet, making interpretable predictions from clinical timeseries, especially those that have moderate dimensionality and missingness, remains an open challenge (Tomašev et al., 2019; Rajkomar et al., 2018). Johnson et al. (2021) tackle this problem by first converting a timeseries into a set of aggregate features akin to the kind of features that clinicians typically use, and use these to make predictions. For example, a patient's ICU timeseries could be turned into summary features per predictor, such as the mean across the entire LOS and the proportion of time spent above threshold. However, while each of these features is interpretable, the number of features required to make an accurate prediction can be quite large. As a result, the reason behind the model's prediction may still be difficult for experts to understand.

In our work, we improve the interpretability (while maintaining the prediction quality) of clinical timeseries prediction models by introducing one more stage into the prediction pipeline: the human-interpretable timeseries features are now grouped into *concepts* that allow for organized inspection of the model and can correspond to semantically meaningful clinical ideas. In doing so, our approach facilitates meaningful human-inspection by learning feature groupings which aren't too dense (too many correlated features are hard to interpret) or too sparse (the features themselves then become the concept). Examples of these high-level concepts may include illness severity or respiratory function, which could be used to explain a model in a decomposable, concise manner. Moreover, we introduce a new method for optimizing such concept-based models that selects only the most predictive features to add to each concept, thereby enabling simultaneously sparse and accurate interpretations of the downstream prediction task.

On a task of predicting vasopressor onset in the ICU (based on MIMIC-III database (Johnson et al. (2016))), we demonstrate that our algorithm achieves state-of-the-art predictive performance while presenting interpretable and concise feature groupings for inspection. When examined by an expert intensivist, we show that our concepts allow for identification of high-level clinical concepts that made sense for the prediction. Even for concepts that don't possess medical meaning at face-value, the concept framework provides a better evaluation than would have otherwise been possible. Thus, our method enables users such as clinical experts to better inspect model predictions and validate its reasoning, addressing important desiderata such as safety and fairness of ML models in high-risk clinical settings.

Generalizable Insights about Machine Learning in the Context of Healthcare

While our core application in this paper is predicting vasopressor onset, the architecture and optimization algorithm will be valuable in other clinical applications that involve making predictions from timeseries data. Specifically:

1. Model: We introduce a bottleneck architecture which automatically learns high-level concepts, or clinically-intuitive groups of features, as explanations for the model's prediction. In doing so, we allow for sufficiently expressive representations that organize the data as clinicians might.

2. Optimization: The above model does not optimize well with standard gradient descent; however, we present a simple, easy-to-implement optimization procedure that robustly identifies these solutions. This allows us to not only optimize the model once, but to get a collection of possible prediction models for further exploration. This is important in health domains because models cannot be expected to know what is causal.
3. Demonstration: We show that this architecture and optimization method combination produces a generalizable framework for clinicians to more easily inspect and intervene on machine learning predictions in high-stakes scenarios while maintaining predictive accuracy.

The combination of our architecture and optimization innovations let us take one step closer to working with clinical experts to build models we can trust.

2. Related Work

2.1. Timeseries Summaries

A common method to improve the interpretability and predictive performance of ML models on clinical timeseries data is to learn representative, understandable statistical summaries as the models input (Johnson et al. (2021); Guo et al. (2020); Awad et al. (2017); Harutyunyan et al. (2019)). These summary statistics provide an efficient, easy-to-calculate representation of physiological time series, some examples of which include mean, minimum, maximum, first time measured, and proportion of time spent above threshold (see Table 1). The identification of important summary features or combinations of summaries subsequently can provide interpretable explanations of the prediction. On various clinical risk stratification tasks, the summary statistics were found to outperform traditional scoring systems and achieve similar performance to baseline deep learning models: Awad et al. (2017) uses the timeseries summaries as features for various data mining models (e.g. Random Forest, Decision Trees, Naive Bayes) and Harutyunyan et al. (2019) and Johnson et al. (2021) use them as features for a logistic regression model.

However, using summary statistics themselves as explanations poses various challenges. First, numerous features are needed for the prediction task, as summary statistics must be calculated for each feature, which significantly decreases a clinician's ability to understand the explanation of the prediction. Next, some summary statistics lack semantic meaning and therefore decrease the model's interpretability, in particular measures of dispersion and distribution tendency such as skewness and kurtosis. This work is the first to propose a method to improve sparsity, semantic meaning, and therefore interpretability, through concepts learned by concept bottleneck models.

2.2. Concept Bottleneck Models

While most deep learning models are considered "black-box" models because they go from raw input to final prediction, concept bottleneck models learn intermediate human-understandable representations of the raw input, i.e. concepts, and then use them to make predictions (Koh et al. (2020)). This architecture has been gaining interest in the deep learning community because of its competitive predictive performance and interpretability advantages. In particular, concept bottleneck models have become very useful in clinical prediction tasks, which tend to be high-risk and therefore require any ML-based models to be both accurate and interpretable (Clough et al. (2019); De Fauw et al. (2018)).

However, one major drawback of concept bottleneck models is that they require a set of *pre-defined concepts and concept labels* for each data point during training time. There is no guarantee that raw inputs will naturally align with human concepts, and human judgement may lead to concepts that are poor representatives of the original data (Chen et al. (2020)). In addition, the process of labelling concepts for thousands of data points requires extensive labor from an expert on the prediction domain, which is infeasible in many contexts (De Fauw et al. (2018)). Our work avoids these challenges by simultaneously learning meaningful concepts while training for the downstream prediction task.

2.3. Interpretability

Recently, interpretability has arisen as necessary desiderata of an applicable machine learning model. In many high-stakes scenarios, it is not sufficient for a machine learning model to simply be accurate at prediction, but it must also be easily understood by the deployer.

Specific to clinical data, one main approach towards bridging the gap between clinician expertise and black-box predictions is to develop simple, clear, interpretable machine learning models that achieve similar accuracy as complex black-box models. For example, attention mechanism models use attention scores to quantify importance of features towards the final prediction, thereby providing an explanation in the form of important predictors (Sha and Wang (2017); Choi et al. (2016)). However, attention scores have been proven to be misleading, as they are frequently uncorrelated with gradient-based measures of feature importance, and there can exist very different attention configurations that nonetheless yield similar predictions (Serrano and Smith (2019); Jain and Wallace (2019)). Sparse logistic regression similarly regularizes feature weights to identify the most important features used during prediction (Poursabzi-Sangdeh et al. (2021)). However, multi-collinearity amongst predictors is a frequent phenomenon in clinical timeseries data, which causes unstable estimates of feature weights and thereby renders them to be unreliable as feature importance measures. Our work proposes a greedy optimization method which directly selects important features to the prediction, thereby avoiding misleading information from attention mechanisms and feature coefficients.

Another family of interpretable models is rule-based models, such as decision sets (Lakkaraju et al. (2016); Lage et al. (2019)) and rule lists (Ustun and Rudin (2015); Letham et al. (2015)), which identify sets of rules based on specific feature values to describe the relationships among variables and explain predictions. One limitation of this approach is that explanations are only presented in terms of raw features, which often may not be meaningful on their own. In our work, we aim to automatically learn concepts which meaningfully group features together, thereby increasing the scope of feature space and enhancing the ability of a clinical expert to understand the prediction.

3. Background and Notation

Notation We consider concept bottleneck models with the following architecture: let $\{\boldsymbol{X}, \boldsymbol{M}\}$ be the inputs (timeseries clinical variables and measurement indicators). First, the inputs $\{\boldsymbol{X}, \boldsymbol{M}\}$ are turned into semantically meaningful summary statistics $\boldsymbol{H}$, which contains each of the summary statistics calculated for each of the clinical variables for each patient. Examples of summary functions include the mean of the timeseries variable and the number of hours a measurement has been above some threshold. Next, those features $\boldsymbol{H}$ are converted into *concepts* $\boldsymbol{C}$, or collections of features that may have clinical meaning that are also useful for prediction. Finally, predictions are made from the concepts $\boldsymbol{C}$ to the outcome $\boldsymbol{Y}$.

Interpretable Timeseries Summaries In this work, we build on the prior work of Johnson et al. (2021), which used the following approach to construct the summary statistics listed in Table 1. However, this prior work did not have a bottleneck layer to connect these features into high-level, meaningful concepts.

The summary functions utilize a weight matrix $\boldsymbol{W}$ as a duration parameter to model how much of each timeseries is needed to compute each of the summaries. Let $D_{i,v}$ represent a duration cutoff for variable v and summary function i where only the variable's timeseries data that occurred within the previous $D_{i,v}$ hours before time of prediction is used. We define the weight tensor as followed:

$$w_{t,i,v} = \sigma((t - T + D_{i,v})/\tau)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ and τ controls the harshness of the weight tensor through the sigmoid function. Here we see that $t > T - D_{i,v}$ results in a $w_{t,i,v}$ value closer to 1, and $t < T - D_{i,v}$ results in a $w_{t,i,v}$ value closer to 0, with the temperature τ controlling how far the sigmoid function is pushed towards its edges. The vector of duration parameters $\boldsymbol{D}$ are included in $\boldsymbol{\beta_H}$, the set of parameters for summary functions learned during the downstream prediction task.

Some summary functions also utilize additional threshold parameters, which correspond to clinically-intuitive ideas such as how long a patient's clinical variables have exceeded above or dropped below some critical value. Let ϕ^+ and ϕ^- represent the upper threshold and lower threshold parameters respectively. These thresholds are also included in $\boldsymbol{\beta_H}$, the set of parameters for summary functions learned in the downstream prediction model.

Lastly, each of the summary statistics uses measurement indicators $\boldsymbol{M}$ in order to ensure that summaries are only computed using time steps where the variable is measured.

Description	Function
Mean of time-series	$(\sum_{t=1}^{T}(\boldsymbol{w_{t,i}} \odot \boldsymbol{X_t} \odot \boldsymbol{M_t}))/(\sum_{t=1}^{T} \boldsymbol{M_t} \odot \boldsymbol{w_{t,i}})$
Variance of time-series	$\frac{(\sum_{t=1}^{T} \boldsymbol{M_t} \odot \boldsymbol{w_{t,i}})^2}{(\sum_{t=1}^{T} \boldsymbol{M_t} \odot \boldsymbol{w_{t,i}})^2 - \sum_{t=1}^{T} \boldsymbol{M_t} \odot \boldsymbol{W_t^2}} \odot \sum_{t=1}^{T} \boldsymbol{M_t} \odot \boldsymbol{w_{t,i}} \odot (\boldsymbol{X}_t - \bar{\boldsymbol{X}})^2$
Feature ever measured indicator	$\sigma((\sum_{t=1}^{T} \boldsymbol{w_{t,i}} \odot \boldsymbol{M_t})/(\tau \odot \sum_{t=1}^{T} \boldsymbol{w_{t,i}}))$
Mean of indicators	$(\sum_{t=1}^{T} \boldsymbol{w_{t,i}} \odot \boldsymbol{M_t})/(\sum_{t=1}^{T} \boldsymbol{w_{t,i}})$
Variance of indicators	$(\frac{(\sum_{t=1}^{T} \boldsymbol{w_{t,i}})^2}{(\sum_{t=1}^{T} \boldsymbol{w_{t,i}})^2 - \sum_{t=1}^{T} \boldsymbol{W_t^2}}) \sum_{t=1}^{T} \boldsymbol{w_{t,i}} \odot (\boldsymbol{M}_t - \bar{\boldsymbol{M}})^2$
# switches from missing to measured	$(\sum_{t=1}^{T} \boldsymbol{w_{t,i}} \odot \lvert \boldsymbol{M_{t+1}} - \boldsymbol{M_t})/(\sum_{t=1}^{T} \boldsymbol{w_{t,i}})$
First time feature is measured	$\min t$ s.t. $M_t = 1$
Last time feature is measured	$\max t$ s.t. $M_t = 1$
Proportion of time above threshold ϕ^+	$(\sum_{t=1}^{T}(\boldsymbol{w_{t,i}} \odot \boldsymbol{M_t} \odot \sigma(\frac{\boldsymbol{X_t} - \phi^+}{\tau})))/(\sum_{t=1}^{T} \boldsymbol{M_t} \odot \boldsymbol{w_{t,i}})$
Proportion of time below threshold ϕ^-	$(\sum_{t=1}^{T}(\boldsymbol{w_{t,i}} \odot \boldsymbol{M_t} \odot \sigma(\frac{\phi^- - \boldsymbol{X_t}}{\tau})))/(\sum_{t=1}^{T} \boldsymbol{M_t} \odot \boldsymbol{w_{t,i}})$
Slope of a L2 line	$\frac{\sum_{t=1}^{T} \boldsymbol{w_{t,i}}(t - \bar{t}_w)(\boldsymbol{X}_t - \bar{\boldsymbol{X}}_{\boldsymbol{w}})}{\sum_{t=1}^{T} \boldsymbol{w_{t,i}}(t - \bar{t}_w)^2}$, where $\bar{t}_w = \frac{\sum_t \boldsymbol{w}_{t,i} \cdot t}{\sum_t \boldsymbol{w}_{t,i}}$, $\bar{\boldsymbol{X}}_{\boldsymbol{w}} = \frac{\sum_t \boldsymbol{w}_{t,i} \cdot \boldsymbol{X}_t}{\sum_t \boldsymbol{w}_{t,i}}$
Standard error of the L2 line slope	$\frac{1}{\sum_{t=1}^{T} \boldsymbol{w_{t,i}}(t - \bar{t}_w)^2}$

Table 1: Summary functions $\boldsymbol{H}$. $\odot$ denotes element-wise matrix multiplication and division is performed element-wise.

4. Methods

4.1. Concept Bottleneck Model Architecture

Consider predicting target $y \in \{0, 1\}$ from input $x \in \mathbb{R}^d$ through a vector of k concepts c $\in \mathbb{R}^k$. We define bottleneck models of the form $f(g(x))$, where $g : \mathbb{R}^d \to \mathbb{R}^k$ maps inputs x to concepts c and $f : \mathbb{R}^k \to \mathbb{R}$ maps concepts c to labels y. The term *concept bottleneck model* comes from the design of the architecture such that the prediction $\hat{y} = f(g(x))$ relies on input x solely through $\hat{c} = g(x)$. We consider the task of learning concepts $\hat{c}$ in an unsupervised way, without access to prior labels. In our concept bottleneck model, we follow the same architecture with an additional feature pre-processing step $h(x)$, i.e. the computation of summary statistics. Thus, our bottleneck model follows the form $f(g(x, h(x))$.

Figure 1 visually depicts our model's architecture and learning process as described below. First, we compute summary statistics $\boldsymbol{H}$ from clinical timeseries inputs $\{\boldsymbol{X}, \boldsymbol{M}\}$, and then concatenate them with the original inputs to form pre-processed input $\boldsymbol{Z}$. We then feed $\boldsymbol{Z}$ as input to function g, which outputs concepts $\boldsymbol{C}$. Logistic regression model f outputs predicted probabilities $\hat{\boldsymbol{y}} = p(\boldsymbol{y} = 1 \vert \boldsymbol{C})$.

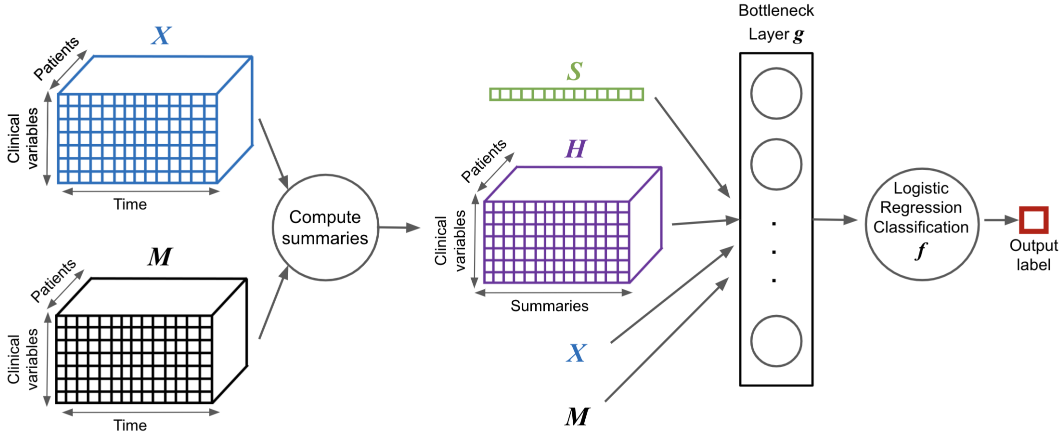


Figure 1: Our Bottleneck Model Architecture. We compute summaries $\boldsymbol{H}$ from clinical timeseries $\{\boldsymbol{X}, \boldsymbol{M}\}$, and then concatenate them with the original inputs to form $\boldsymbol{Z}$. We then feed $\boldsymbol{Z}$ as input to function g, which outputs concepts $\boldsymbol{C}$. Lastly, logistic regression model f uses $\boldsymbol{C}$ to output predicted probabilities for the output label $\boldsymbol{y}$.

4.2. Objective with Interpretability Regularization

Our objective is to jointly learn the logistic regression, bottleneck layer, and summary function parameters $\boldsymbol{\beta} = \{\boldsymbol{\beta}_f, \boldsymbol{\beta}_g, \boldsymbol{\beta}_{\boldsymbol{H}}\}$. We minimize the below loss function for N patients using K concepts:

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{y}) = -\frac{1}{N}\sum_{n=1}^{N} \omega_n(\boldsymbol{y_n} \cdot log[f(g(\boldsymbol{Z_n}, \boldsymbol{\beta}))] + (1 - \boldsymbol{y_n}) log[1 - f(g(\boldsymbol{Z_n}, \boldsymbol{\beta}))]) \\ + \lambda_1 \cdot \sum_{k=1}^{K} |\boldsymbol{\beta}_{g,k}| + \lambda_2 \cdot \sum_{k=1}^{K}\sum_{j\neq k} \left| \frac{\boldsymbol{\beta}_{g,j} \cdot \boldsymbol{\beta}_{g,k}}{\|\boldsymbol{\beta}_{g,j}\| \|\boldsymbol{\beta}_{g,k}\|} \right| \end{aligned} \quad (1)$$

Our loss function builds off the weighted binary cross-entropy loss using predictive model g. We address class imbalance by weighting the contribution of each training example inverse proportionally to its class frequency, or ω_n.

We then add regularizations to increase the interpretability of learned concepts, enforcing sparsity of features within each concept and distinction of features across concepts. Specifically, we include an L1 regularization penalty on $\boldsymbol{\beta_g}$ and a cosine similarity regularization penalty across all combinations of $\{\boldsymbol{\beta}_{g,k}, \boldsymbol{\beta}_{g,j\neq k}\}$.

In particular, we provide a method to fairly ensure sparsity across models with different numbers of concepts, as the strength of L1 regularization is inherently positively correlated with the number of concepts since it is the cumulative sum of coefficient magnitudes. Instead of comparing $k-$concept models with the same λ_1 value, we perform a hyperparameter search for λ_1 such that the sum of the magnitude of the feature weights within each concept are below a certain threshold γ, or $\sum_i |\boldsymbol{\beta}_{g,k,i}| < \gamma \;\; \forall k$.

4.3. Greedy optimization method

Given a trained concept bottleneck model according to the objective in Equation 1, we define a procedure for selecting the most interpretable and predictive concept definitions. We aim to build compact groups of features to serve as concepts, ones that can achieve comparable predictive performance to a full model, but without hundreds of low-weight, unpredictive features. The steps of the algorithm are described below, and the full algorithm is described in Algorithm 1.

Initialization We first select a subset of features $\boldsymbol{A}_k$ for each concept c_k as input for the optimization method. We select the 90th percentile of features by weight:

$$\sum_i |\boldsymbol{\beta}_{g,k,\boldsymbol{A}_{i,k}}| \geq 0.9 \sum_i \boldsymbol{\beta}_{g,k,i} \;\; \forall k$$

Using this selection process, we choose a small enough subset of important features such that speed during runtime is optimized, yet a large enough subset such that a holistic group of features is considered for the concept definition. This initialization also allows for subsets of input features to be different lengths across various concepts, enabling flexibility amongst concepts sizes.

Feature Budget We then define a feature budget ϕ_b across all concepts to ensure the sparsity, i.e. we select ϕ_b features across all k concepts and use the groups of features as concept definitions. In our case, we set:

$$\phi_b = 10k$$

Note that this does not mean each concept has 10 features, rather each concept can have a different number of features as long as the sum across all concepts is not more than ϕ_b. This addresses the challenge of diverse concept definitions, where one concept may only need 2 features versus another concept which may require 10 features for complete context.

Greedy Selection The optimization then proceeds by, at each step, greedily selecting the feature and concept tuple (z^*, c_{k^*}) which provides the greatest increase in AUC. The method iteratively explores all (z, c_k) from the set of unselected features $\boldsymbol{u}_k$ for each concept c_k, finds the (z^*, c_{k^*}) which results in the maximum AUC, then adds z^* to the set of selected features for concept c_{k^*}: $\boldsymbol{s}_{k^*}$. This selection process repeats until we reach the feature budget, or until we have $\boldsymbol{s}_k : \mathbb{R}^{l_k} \; \forall k$ such that $\sum_k l_k = \phi_b$. $\boldsymbol{s}_k$ is initialized as $\emptyset$ and u_k is initialized as $\boldsymbol{A_k}$.

Additional details After $\boldsymbol{s}_k \; \forall k$ is determined, we fine tune f so that it is optimally predictive given the updated concept definitions. We fix the concept definitions according to $\boldsymbol{s}_k$ in $\boldsymbol{\beta}_{g,k} \; \forall k$ and fine-tune logistic regression parameters $\boldsymbol{\beta}_f$.

Algorithm 1 Greedy Optimization Method

```
Require: $\boldsymbol{A}$, $\boldsymbol{s}, \boldsymbol{u}, \phi_b, f$
  for $i \in 1 : \phi_b$ do
    maxAUC = 0
    $z^* = -1$
    for $k \in K$ do
      for $z \in u_k$ do
        if AUC($f(z)$) > maxAUC then
          $z^* = z$
          $c_{k^*} = c_k$
        end if
      end for
    end for
    $s_{k^*} = s_{k^*} \cup z^*$
    $u_{k^*} = u_{k^*} \setminus z^*$
  end for
  Finetune $f$
```

5. Cohort and Data Processing

5.1. Cohort Selection

We use EHR data from the MIMIC-III care database, which contains deidentified, comprehensive clinical data of patients admitted to the Beth Israel Deaconess Medical Center ICU unit Johnson et al. (2016). We extract data from MIMIC-III PhysioNet version 1.4, which contains 30,232 patients. We restrict analysis to adult patients (age greater than 18 years old) with an ICU or overall length of stay between 6 and 600 hours. We also exclude cases where vasopressors were administered within the first 6 hours of admission. After applying these restrictions, our final cohort contained 15,552 patients, 9.83% or 1528 of whom required vasopressor intervention. A summary of cohort characteristics and demographics can be found in Table 2.

Cohort	Age	% Female	% Urgent	% Emergency	% SICU	% TSICU	% MICU
All	63.4	45.1	1.05	88.8	20.1	13.3	46.7
+	66.5	43.6	1.24	86.4	16.2	13.2	33.6
-	63.1	45.3	1.03	89.1	20.5	13.4	48.1

Table 2: Mean Background Characteristics of Cohort

5.2. Feature Choices

Let N be the total number of patients in our cohort; we extracted both static and hourly physiological data for each patient n of N. All variables were normalized to have zero mean and unit variance.

Static Data $\boldsymbol{S}$. The static data matrix $\boldsymbol{S}$: $(N \times 8)$ contains 8 fixed variables for each patient n, including demographic variables (age at admission and gender), ICU service type (medical, surgical, trauma surgical, or cardiac surgery) and admission type (urgent, emergency, or elective).

Per-timestep Clinical Data $\boldsymbol{X}$. The clinical variable matrix $\boldsymbol{X}$: $(N \times V \times T)$ contains $V = 28$ clinical variable measurements for each patient n at hour t for $T = 6$ hours of data. The following 28 clinical variables measure vital signs and labs: `dbp` (diastolic blood pressure), `fio2` (fraction of inspired oxygen), `GCS` (Glasgow Coma Scale), `hr` (heartrate), `map` (mean arterial pressure), `sbp` (systolic blood pressure), `spontaneousrr` (spontaneous respiratory rate), `spo2` (oxygen saturation), `temp` (body temperature), `urine` (urine output), `bun` (blood urea nitrogen), `magnesium`, `platelets`, `sodium`, `alt` (alanine transaminase), `hct` (hematocrit), `po2` (partial pressure of oxygen), `ast` (aspartate aminotransferase), `potassium`, `wbc` (white blood cell count), `bicarbonate`, `creatinine`, `lactate`, `pco2` (partial pressure of carbon dioxide), `glucose`, `inr` (international normalised ratio), `hgb` (hemaglobin), `bilirubin_total`.

Per-timestep Clincal Data Indicators $\boldsymbol{M}$. The indicator matrix $\boldsymbol{M}$: $(N \times V \times T)$ contains indicators $M_{n,v,t}$ which equals 1 if clinical variable d was measured at time t for patient n and 0 otherwise.

Outcome labels $\boldsymbol{y}$. The outcome label vector $\boldsymbol{y}$ of length N contains indicators y_n which equals 1 if patient n required vasopressor therapy during their stay and 0 otherwise.

6. Evaluation Approach

We compare models using our bottleneck architecture and greedy optimization to other interpretable models and deep learning models. We show that our models achieve performance comparable to these baseline models, while simultaneously providing interpretable concepts that greatly enhance human-understanding of predictions.

6.1. Baseline Models

For our first baseline model, we use a Logistic Regression model on the interpretable timeseries summary statistics, as previous work document their comparable performance to state-of-the-art deep models and their increased interpretability due to clinical sensible summaries (Johnson et al. (2021)). We also use a LSTM model that takes as input all of the patient timeseries as our traditional deep baseline architecture, as prior work show their superior performance at mortality prediction from clinical timeseries data (Harutyunyan et al. (2019)). We train all Logistic Regression models for 1,000 epochs and LSTM models for 10,000 epochs using early stopping. All hyperparameters, including LSTM hidden state and layer dimensions, optimizer learning rate, and regularization parameters (eg. horseshoe shrinkage parameter), were chosen via a non-exhaustive random hyperparameter search.

6.2. Metric

Our main metric for comparing model performance is the Area Under the Receiver Operating Characteristic Curve (ROC AUC). We chose this metric due to the binary reponse variable of vasopressor onset, as AUC best measures how the true positive rate and false positive rate trade off.

6.3. Optimization Details for Concept Bottleneck

We construct our concept bottleneck models by splitting the cohort of N patients into train/test sets using an 80/20 split. All performance metrics are averaged across 6 train-test splits. We trained our models for 1,000 epochs with the Adam optimizer at a batch size of 256. Three hyperparameters (optimizer learning rate, weight decay, and cosine similarity λ_2), were selected via random hyperparameter search. For the L1 hyperparameter λ_1, we performed a hyperparameter search such that the sum of the magnitudes of feature weights across all concepts were below a certain threshold γ. All temperature parameters τ were set to 0.1.

In order to determine the optimal number of concepts, we compared k-concept models at a given γ (holding all other hyperparameters constant) and selected the model with the smallest number of concepts and greatest AUC improvement.

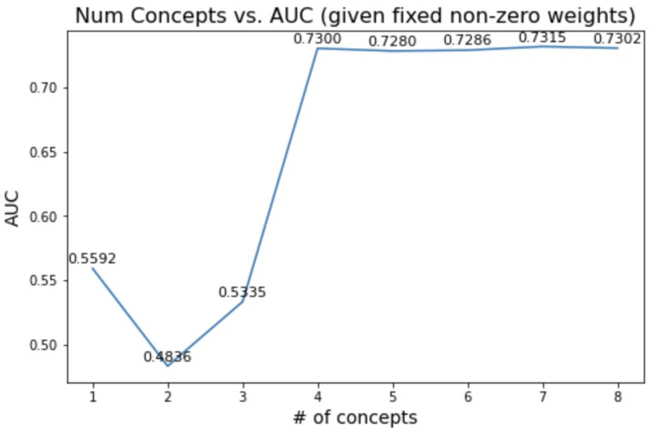


Figure 2: L1 regularization hyperparameter search to determine optimal number of concepts. This plot shows that a 4-concept bottleneck model is the smallest model that provides a high AUC.

From Figure 2, we see that the optimal number of concepts at $\gamma =$ is 4 concepts, as it the smallest k to achieve similar performance to $k > 4$ concept models.

7. Results

Our optimization method provides a superior method of selecting important features for predictive accuracy compared to traditional feature selection methods based on coefficients. We applied our optimization strategy of greedily selecting the feature and concept tuples in our concept bottleneck model (shown in red in 5a), which demonstrates better performance than the baseline strategy (where features are selected based on their weights) on the Johnson et al. (2021) model (shown in green in 5a). Next, in order to separate the effect of our newly introduced optimization from our newly introduced hierarchical bottleneck model, we include an additional baseline where we apply our optimization to the model in Johnson et al. (2021). These results are shown in blue. Our greedy optimization strategy results in significantly higher AUCs for both CBMs and Johnson et al. (2021)'s model.

In particular, the improvement in AUC increases as the sparsity of features increases, which implies that our greedy optimization process is able to better identify the set of important features which achieve high predictive performance. Thus, our optimization method overcomes the challenge of misleading regression coefficients and identifies the predictive value in explicitly and greedily selecting features based on AUC.

We verify these results by showing the concise concept definitions learned by our bottleneck models and optimization method. Figure 3 shows the magnitude of top 100 feature weights for the Johnson et al. baseline model (Johnson et al. (2021)) and Figure 4 shows that of a 4-concept bottleneck model, where the blue represents pre-optimization method and red represents the features selected post-optimization method. First, we see that for most concepts, the magnitude of the weights of the top 100 features quickly converges to 0, whereas for the baseline logistic regression model, all top 100 features have non-zero coefficients, and there is no clear indication of which features are the most important to the prediction. Next, we observe that the selected important features by the optimization method are sparse, thus allowing for more concise and interpretable explanations.

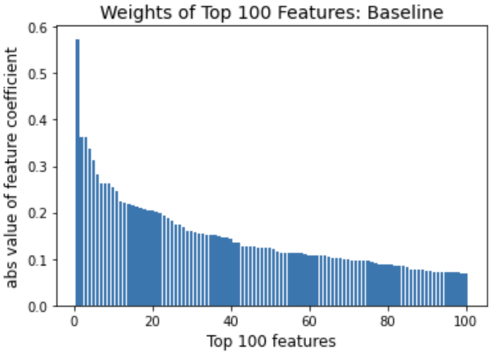
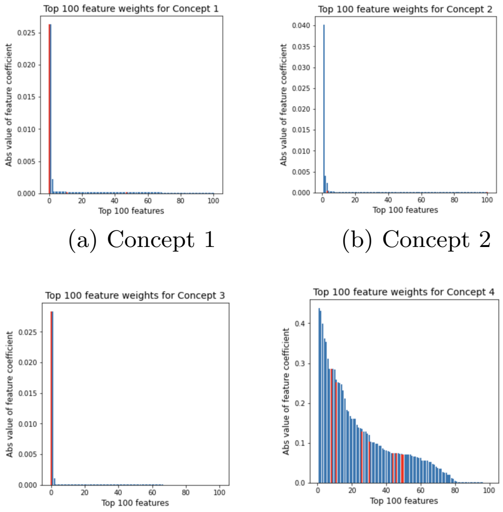


Figure 3: Weights of Top 100 Features for Johnson et al. model



(a) Concept 1

(b) Concept 2

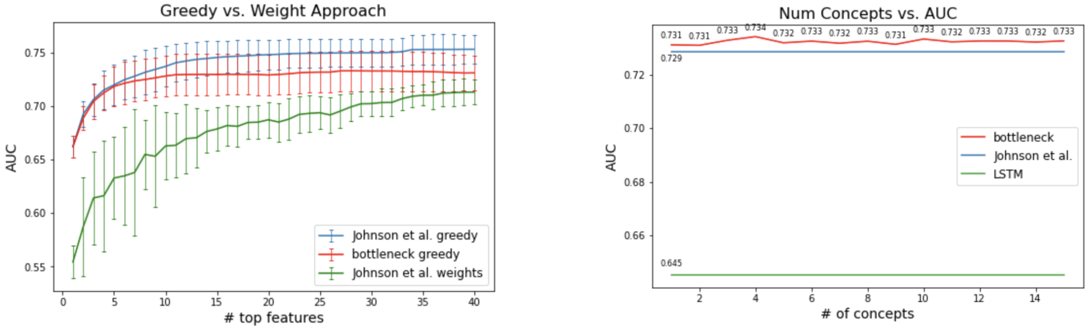
(c) Concept 3

(d) Concept 4

Figure 4: Weights of top 100 features for 4-concept bottleneck model

Our interpretable bottleneck model achieves performance comparable to (less interpretable) state-of-the-art baselines. Figure 5a shows the comparable performance of our bottleneck model architecture (in red) vs. a baseline Logistic Regression model (in blue), both using the greedy optimization method. Notably, our bottleneck model demonstrates equivalent performance with the baseline Logistic Regression model with a sparse feature set, demonstrating the bottleneck model's strong predictive ability under feature constraints.

When regularizations were not added to our objective, we found that our bottleneck models achieved higher accuracy than the baseline Johnson et al. (2021) Logistic Regression model and significantly outperformed the baseline LSTM model, as shown in Figure 5b. As expected, we observe that inserting an intermediate bottleneck layer to a baseline logistic regression model adds increasing complexity in the model architecture, thereby allowing the model to better learn patterns in elaborate clinical timeseries data. Surprisingly, we also see that this neural network architecture achieves significantly better performance the state-of-the-art LSTM model, which may speak to the utility of our timeseries summary functions or may be attributed to the difficult training process for LSTMs (they are extremely prone to random initializations and overfitting). However, it is also important to note that without any regularizations on our bottleneck models, the concept definitions learned during the prediction process may not necessarily be meaningful/interpretable, so we do not employ these non-regularized models in our concept analysis.



(a) Greedy vs. Weights Feature Selection Approach

(b) Bottleneck Model vs. Johnson et al. Model vs. LSTM Model AUC

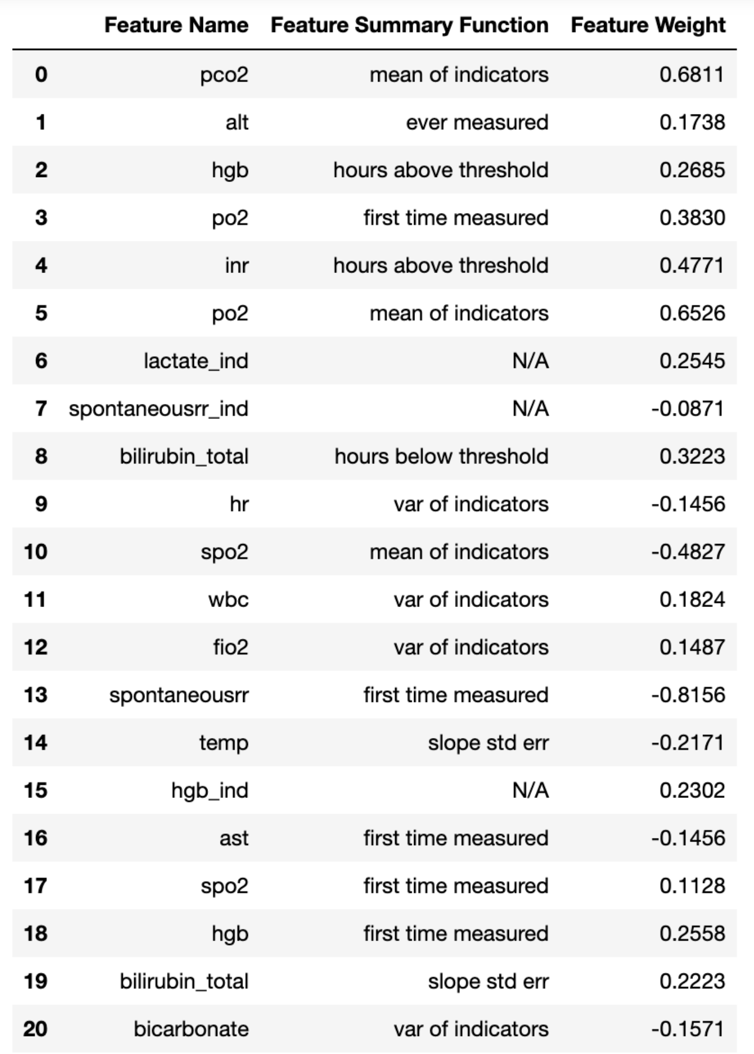
Figure 5: AUC plots comparing our bottleneck model, Johnson et al. model, and LSTM model. The first plot shows the superior selection of predictive features of our greedy optimization method and the second plot demonstrates the greater predictive accuracy of our bottleneck model (without regularizations) compared to Johnson et al. and LSTM baseline models.

Our approach allows for a more meaningful and clinically-sensible inspection of features.

Rather than outputting a raw list of important features and their coefficients (Figure 6 shows a truncated version), our bottleneck architecture learns distinct concepts, i.e. groups of features which lend themselves to high-level clinical conditions that help interpret the prediction (Figure 7). In order to interpret the clinical meaning of our concept definitions, we had a discussion with an intensivist who is an expert on the MIMIC database and the critical care field. We found that models of similar predictive performance learned various concept definitions, some of which had face validity, some of which were more confusing.

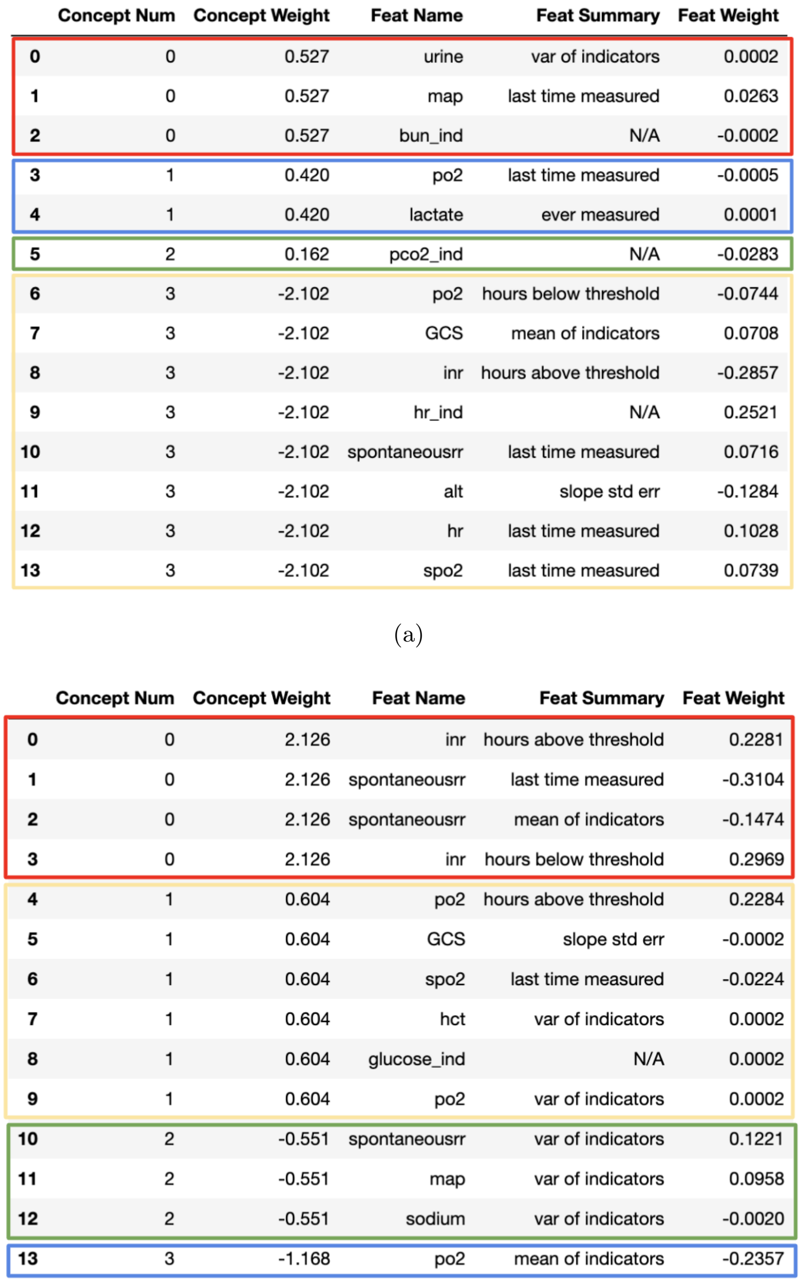
Figure 7a) shows a clinically sensible list of concepts. Concept 0 (outlined in red) generally maps to kidney function: it is widely accepted that a rapid reduction of urine output is a early indicator of decreased kidney function, a low MAP can cause low renal blood flow, and high levels of blood urea nitrogen (bun) are a sign that ones kidneys are unable to remove waste products from blood (Bellomo et al. (2004); Verdecchia et al. (2001); Seki et al. (2019)). Next, the group of features learned by concept 1 (outlined in blue) are common indicators of sepsis. There is overwhelming medical evidence that sepsis and septic shock are associated hyperlactatemia (high levels of lactate concentration), which is also related to po2, a measurement of the effectiveness of the lungs in extracting oxygen into the blood stream (Garcia-Alvarez et al. (2014)). Furthermore, concept 4 (outlined in yellow) corresponds to a general evaluation of illness severity. Variables such as GCS, which measures a patient's consciousness levels, and vital signs, such as heart rate and spontaneous respiratory rate, are common metrics tracked by clinicians that may indicate failing health conditions. Lasty, concept 3 (outlined in green) is defined by a single feature, pCO2, which corresponds to oxygenation of cells or respiratory condition. pCO2 typically reflects the amount of CO2 gas dissolved in the blood and is inversely related with cardiac output, thus decreased pCO2 levels are typically symptoms of hyperventilation or hypoxia (Mallat et al. (2016); Bitar et al. (2020)).

Figure 7b) shows a list of concept definitions learned by another one of our models that are less obviously sensible to a clinician. On one hand, we see that there is a significant amount of overlap in concept definitions between the two lists, with the concepts outlined in yellow and blue having many of the same features. This concept similarity increases our faith in these two concept definitions, as they remain robust to different model runs. On the other hand, the concepts outlined in red and green in Figure 7b) did not map to intuitive clinical conditions at first glance. By way of example, our clinical expert did not find it obvious to group inr, a variable related to blood clotting, and spontaneous respiratory rate together in a concept. This suggests various avenues for future work, specifically exploring the issue of nonidentifiability within our models and improving upon the robustness of concept definitions. Interestingly, we see the utility of the concept bottleneck model in these situations of ambiguous concepts, as the concept framework allows researchers to easily intervene on a concept during prediction according to the clinical expert's advice. That is, a clinician may decide that they do not want the model to associate a particular term associated with a particular concept when vetting the model for validity. Having the ability for a clinical expert to understand and adjust a machine-learned model is crucial in high-stakes applications, and the model we introduce in our work has that essential property. See Appendix A for a series of further experiments, such as augmenting the bottleneck model with concepts that hold no information and changing concept definitions before prediction, in order to sanity check the validity of the concept definitions.



	Feature Name	Feature Summary Function	Feature Weight
0	pco2	mean of indicators	0.6811
1	alt	ever measured	0.1738
2	hgb	hours above threshold	0.2685
3	po2	first time measured	0.3830
4	inr	hours above threshold	0.4771
5	po2	mean of indicators	0.6526
6	lactate_ind	N/A	0.2545
7	spontaneousrr_ind	N/A	-0.0871
8	bilirubin_total	hours below threshold	0.3223
9	hr	var of indicators	-0.1456
10	spo2	mean of indicators	-0.4827
11	wbc	var of indicators	0.1824
12	fio2	var of indicators	0.1487
13	spontaneousrr	first time measured	-0.8156
14	temp	slope std err	-0.2171
15	hgb_ind	N/A	0.2302
16	ast	first time measured	-0.1456
17	spo2	first time measured	0.1128
18	hgb	first time measured	0.2558
19	bilirubin_total	slope std err	0.2223
20	bicarbonate	var of indicators	-0.1571

Figure 6: Example explanation of prediction from Johnson et al. logistic regression model. As shown, this long, unordered list of features is difficult to interpret/to quickly parse for important reasons behind prediction.



	Concept Num	Concept Weight	Feat Name	Feat Summary	Feat Weight
0	0	0.527	urine	var of indicators	0.0002
1	0	0.527	map	last time measured	0.0263
2	0	0.527	bun_ind	N/A	-0.0002
3	1	0.420	po2	last time measured	-0.0005
4	1	0.420	lactate	ever measured	0.0001
5	2	0.162	pco2_ind	N/A	-0.0283
6	3	-2.102	po2	hours below threshold	-0.0744
7	3	-2.102	GCS	mean of indicators	0.0708
8	3	-2.102	inr	hours above threshold	-0.2857
9	3	-2.102	hr_ind	N/A	0.2521
10	3	-2.102	spontaneousrr	last time measured	0.0716
11	3	-2.102	alt	slope std err	-0.1284
12	3	-2.102	hr	last time measured	0.1028
13	3	-2.102	spo2	last time measured	0.0739

(a)

	Concept Num	Concept Weight	Feat Name	Feat Summary	Feat Weight
0	0	2.126	inr	hours above threshold	0.2281
1	0	2.126	spontaneousrr	last time measured	-0.3104
2	0	2.126	spontaneousrr	mean of indicators	-0.1474
3	0	2.126	inr	hours below threshold	0.2969
4	1	0.604	po2	hours above threshold	0.2284
5	1	0.604	GCS	slope std err	-0.0002
6	1	0.604	spo2	last time measured	-0.0224
7	1	0.604	hct	var of indicators	0.0002
8	1	0.604	glucose_ind	N/A	0.0002
9	1	0.604	po2	var of indicators	0.0002
10	2	-0.551	spontaneousrr	var of indicators	0.1221
11	2	-0.551	map	var of indicators	0.0958
12	2	-0.551	sodium	var of indicators	-0.0020
13	3	-1.168	po2	mean of indicators	-0.2357

(b)

Figure 7: Example explanation of prediction from our bottleneck model. Our concept definitions allow for easier inspection by grouping features together into higher-level clinical conditions.

8. Discussion

Our work offers various advantages in the domain of interpretable machine learning in healthcare settings. Our bottleneck models demonstrate the ability to learn high-level, semantically-meaningful concepts, providing contextualized explanations for a traditionally black-box prediction process. Furthermore, our optimization method then identifies a compact yet predictive subset of features within each concept, enhancing the interpretability of our method compared to traditional lengthy lists of features that might not necessarily have clinical meaning on their own. However, it also has limitations that suggest interesting future work.

General Discussion Our work initiates interesting, open-ended discussions on the application of concept bottleneck models in healthcare settings, and more broadly the role of interpretable ML in any real-world setting.

First, how do we address fundamental statistical limitations of machine learning methods in terms of robustness? In our experiments, we found that multiple models (with different concept definitions) achieve similar levels of accuracy, which may be inevitable because choosing the feature which produced the local maximum at every step is not guaranteed to produce the global maximum solution. This issue of non-identifiability across similar models could be minimized by including additional regularizations in the objective function, exploring the sensitivity of our greedy method to the data (perhaps through controlled perturbations to different models), or exploring non-greedy based optimizations for feature selection. However, in scenarios where non-identifiability is unavoidable, human inspection is needed to gain the trust of the clinician. Whether meaningful concepts explanations were produced or not, the clinician must be able to inspect the reasoning to accurately assess the usability of the model.

Thus, in the current scenario where statistical machine learning methods may not be robust enough, we see that the benefit of our interpretable concept framework is that when nonsense concepts are presented, experts can reject them or amend them. However, the best way to solicit this feedback remains a question for further study. In our project, would it have been most helpful for clinicians to define and label concepts in order to provide the model some ground truth notion? Or is it more realistic for clinicians to intervene during prediction time and modify concepts as the need arises? One direction of future work that addresses this problem would be to conduct a user study with clinical experts to provide feedback on the interpretability of our proposed concepts. Overall, the interaction between users and ML models, especially in high-risk scenarios such as hopsitals, must be further explored in order to achieve the most predictive and safe results.

Limitations First, our concept bottleneck model relies on significantly preprocessed data, specifically the manually-defined summary statistics calculated for each timeseries variable. We depend on knowing what might be human-interpretable base features, as we selected functions such as mean, min, max, etc. If our bottleneck architecture and optimization method were applied to the raw data alone (static variables, clinical timeseries variables, and measurement indicators), it is unclear whether the interpretability of the concepts or the accuracy of the model would be as strong as those of our current approach. Future work could address this by experimenting with our methods on the raw data, or using additional non-parametric methods to identify clusters and patterns within the data to enhance interpretability.

Furthermore, the amount of preprocessing required in our project poses a potential barrier to the generalizability of our method. Our learned concepts may only be specific to the cohort of patients we selected from MIMIC or to the vasopressor target we performed our experiments on. Further experimentation on different cuts of the clinical data or different prediction targets would better elucidate the scope of this challenge.

Conclusion In this work, we utilize a bottleneck model architecture and propose a greedy optimization algorithm, simultaneously learning inspectable explanations and achieving high predictive performance. Our bottleneck architecture learns concepts, or high-level clinical conditions, which enable clinicians to better understand predictions by abstracting away raw input data. Our optimization method then selects the most important features within each concept, learning sparse, semantically-meaningful definitions for each concept. We present this architecture and optimization combination as a potentially generalizable methodology that can be applied to other clinical prediction tasks in the future, such as predicting mortality or other interventions (e.g. ventilation). From a technical standpoint, our algorithm enables the automatic learning of concept definitions, providing an inherently interpretable framework for humans to inspect while maintaining predictive accuracy. From a clinical standpoint, our work facilitates a realistic framework for the application of machine learning in high-stakes scenarios such as ICU units: we enable models to learn a set of concise, meaningful clinical explanations to transparentize its prediction process.

Acknowledgments

CW, MH, and FDV acknowledge support from NIH project R01MH123804. The authors thank members of the Data to Actionable Knowledge lab for their feedback on the manuscript.

References

Aya Awad, Mohamed Bader-El-Den, James McNicholas, and Jim Briggs. Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach. *International journal of medical informatics*, 108:185–195, 2017.

Andrew L Beam, Arjun K Manrai, and Marzyeh Ghassemi. Challenges to the reproducibility of machine learning models in health care. *Jama*, 323(4):305–306, 2020.

Rinaldo Bellomo, Claudio Ronco, John A Kellum, Ravindra L Mehta, and Paul Palevsky. Acute renal failure–definition, outcome measures, animal models, fluid therapy and information technology needs: the second international consensus conference of the acute dialysis quality initiative (adqi) group. *Critical care*, 8(4):1–9, 2004.

Zouheir Ibrahim Bitar, Ossama Sajeh Maadarani, AlAsmar Mohammed El-Shably, Ragab Desouky Elshabasy, and Tamer Mohamed Zaalouk. The forgotten hemodynamic (pco2 gap) in severe sepsis. *Critical care research and practice*, 2020, 2020.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems*, 29, 2016.

James R Clough, Ilkay Oksuz, Esther Puyol-Antón, Bram Ruijsink, Andrew P King, and Julia A Schnabel. Global and local interpretability for cardiac mri classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 656–664. Springer, 2019.

Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350, 2018.

Lucas M Fleuren, Thomas LT Klausch, Charlotte L Zwager, Linda J Schoonmade, Tingjie Guo, Luca F Roggeveen, Eleonora L Swart, Armand RJ Girbes, Patrick Thoral, Ari Ercole, et al. Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy. *Intensive care medicine*, 46(3):383–400, 2020.

Joseph Futoma, Sanjay Hariharan, and Katherine Heller. Learning to detect sepsis with a multitask gaussian process rnn classifier. In *International Conference on Machine Learning*, pages 1174–1182. PMLR, 2017.

Joseph Futoma, Morgan Simons, Trishan Panch, Finale Doshi-Velez, and Leo Anthony Celi. The myth of generalisability in clinical research and machine learning in health care. *The Lancet Digital Health*, 2(9):e489–e492, 2020.

Mercedes Garcia-Alvarez, Paul Marik, and Rinaldo Bellomo. Sepsis-associated hyperlactatemia. *Critical care*, 18(5):1–11, 2014.

Marzyeh Ghassemi, Marco Pimentel, Tristan Naumann, Thomas Brennan, David Clifton, Peter Szolovits, and Mengling Feng. A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Chonghui Guo, Menglin Lu, and Jingfeng Chen. An evaluation of time series summary statistics as features for clinical prediction tasks. *BMC medical informatics and decision making*, 20(1):1–20, 2020.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):1–18, 2019.

Kelly M Hoffman, Sophie Trawalter, Jordan R Axt, and M Norman Oliver. Racial bias in pain assessment and treatment recommendations, and false beliefs about biological differences between blacks and whites. *Proceedings of the National Academy of Sciences*, 113(16):4296–4301, 2016.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Nari Johnson, Sonali Parbhoo, Andrew Slavin Ross, and Finale Doshi-Velez. Learning predictive and interpretable timeseries summaries from icu data, 2021.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation, 2019.

Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1675–1684, 2016.

Sidney Le, Emily Pellegrini, Abigail Green-Saxena, Charlotte Summers, Jana Hoffman, Jacob Calvert, and Ritankar Das. Supervised machine learning for the early prediction of acute respiratory distress syndrome (ards). *Journal of Critical Care*, 60:96–102, 2020.

Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3), Sep 2015. ISSN 1932-6157. doi: 10.1214/15-aoas848. URL http://dx.doi.org/10.1214/15-AOAS848.

Jihad Mallat, Malcolm Lemyze, Laurent Tronchon, Benoît Vallet, and Didier Thevenin. Use of venous-to-arterial carbon dioxide tension difference to guide resuscitation therapy in septic shock. *World journal of critical care medicine*, 5(1):47, 2016.

Shamim Nemati, Andre Holder, Fereshteh Razmi, Matthew D Stanley, Gari D Clifford, and Timothy G Buchman. An interpretable machine learning model for accurate prediction of sepsis in the icu. *Critical care medicine*, 46(4):547, 2018.

Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability, 2021.

Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmarking deep learning models on large healthcare datasets. *Journal of biomedical informatics*, 83:112–134, 2018.

Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):1–10, 2018.

Makiko Seki, Masaru Nakayama, Teppei Sakoh, Ryota Yoshitomi, Akiko Fukui, Eisuke Katafuchi, Susumu Tsuda, Toshiaki Nakano, Kazuhiko Tsuruya, and Takanari Kitazono. Blood urea nitrogen is independently associated with renal outcomes in japanese patients with stage 3–5 chronic kidney disease: A prospective observational study. *BMC nephrology*, 20(1):1–10, 2019.

Sofia Serrano and Noah A. Smith. Is attention interpretable?, 2019.

Ying Sha and May D Wang. Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 233–240, 2017.

Stephen M Strakowski, Paul E Keck, Lesley M Arnold, Jacqueline Collins, Rodgers M Wilson, David E Fleck, Kimberly B Corey, Victor R Adebimpe, et al. Ethnicity and diagnosis in patients with affective disorders. *The Journal of clinical psychiatry*, 64(7): 6701, 2003.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

Nenad Tomašev, Xavier Glorot, Jack W Rae, Michal Zielinski, Harry Askham, Andre Saraiva, Anne Mottram, Clemens Meyer, Suman Ravuri, Ivan Protsyuk, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*, 572(7767):116–119, 2019.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, Nov 2015. ISSN 1573-0565. doi: 10.1007/s10994-015-5528-6. URL http://dx.doi.org/10.1007/s10994-015-5528-6.

Paolo Verdecchia, Giuseppe Schillaci, Gianpaolo Reboldi, Stanley S Franklin, and Carlo Porcellati. Different prognostic impact of 24-hour mean blood pressure and pulse pressure on stroke and coronary artery disease in essential hypertension. *Circulation*, 103(21): 2579–2584, 2001.

Laure Wynants, Ben Van Calster, Gary S Collins, Richard D Riley, Georg Heinze, Ewoud Schuit, Marc MJ Bonten, Darren L Dahly, Johanna A Damen, Thomas PA Debray, et al. Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. *bmj*, 369, 2020.

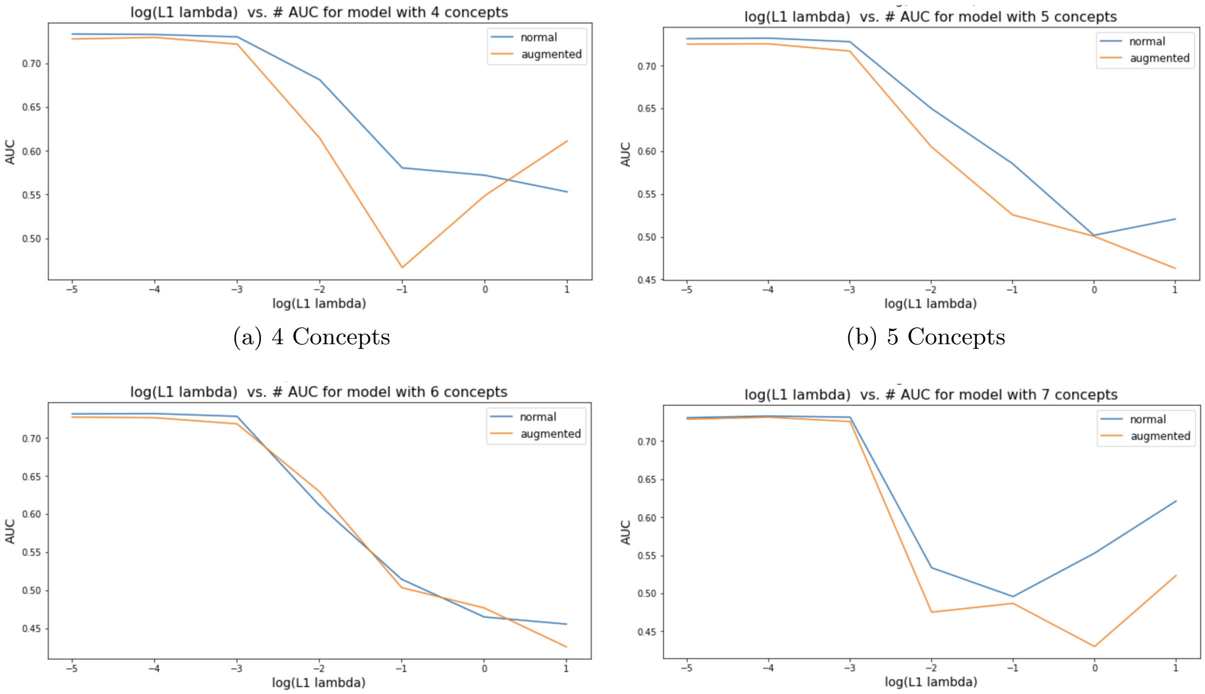
Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565, 2020.

Daniel Zeiberg, Tejas Prahlad, Brahmajee K Nallamothu, Theodore J Iwashyna, Jenna Wiens, and Michael W Sjoding. Machine learning for patient risk stratification for acute respiratory distress syndrome. *PloS one*, 14(3):e0214465, 2019.

Appendix A. Appendix A.

A.1. Zero Weight Concept Experiment

Intuitively, adding an extra concept with no information (which we call a 0-weight concept) should have no bearing on the predictive performance of the model. We compare the AUCs of normal bottleneck models with a 0-weight concept augmented model in order to sanity check our concept framework (shown in Figure 8) during our L1 hyperparameter λ_1 search. We see that for small regularization strengths, the two models perform very similarly and achieve high predictive accuracy. As the regularization strength increases, we observe that the performance of both models decreases as expected, and that the AUCs diverge slightly at some points. We attribute this to the fact that when the regularization strength is too high, the models are unable to learn meaningful patterns and effectively begin to randomly guess predictions. In our analysis, we consider $\lambda_1 = 0.001$, which demonstrates the expected behavior.



(a) 4 Concepts

(b) 5 Concepts

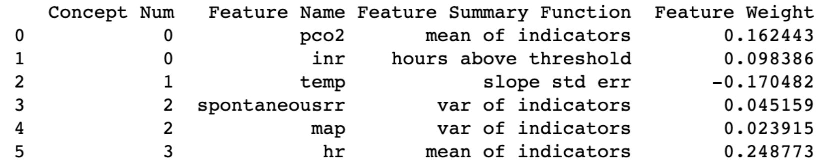
(c) 6 Concepts

(d) 7 Concepts

Figure 8: 0 weight concept AUC

A.2. Changing Concept Definitions Experiment

Intuitively, if our bottleneck models learn concept definitions that are meaningful and predictive, changing features across concepts, thereby changing concept definitions, before prediction should cause a drop in predictive accuracy. Shown in Figure 9, we consider a simple 4-concept bottleneck model, trained and optimized using our algorithm, that contains 1-2 features per concept for this experiment. Table 3 shows the significant decrease in AUC that results from swapping various features across different concepts. This is the expected result, verifying that our models are indeed learning concept definitions that are faithful to the downstream prediction.



```
    Concept Num   Feature Name Feature Summary Function  Feature Weight
0             0           pco2       mean of indicators        0.162443
1             0            inr    hours above threshold        0.098386
2             1           temp            slope std err       -0.170482
3             2  spontaneousrr        var of indicators        0.045159
4             2            map        var of indicators        0.023915
5             3             hr       mean of indicators        0.248773
```

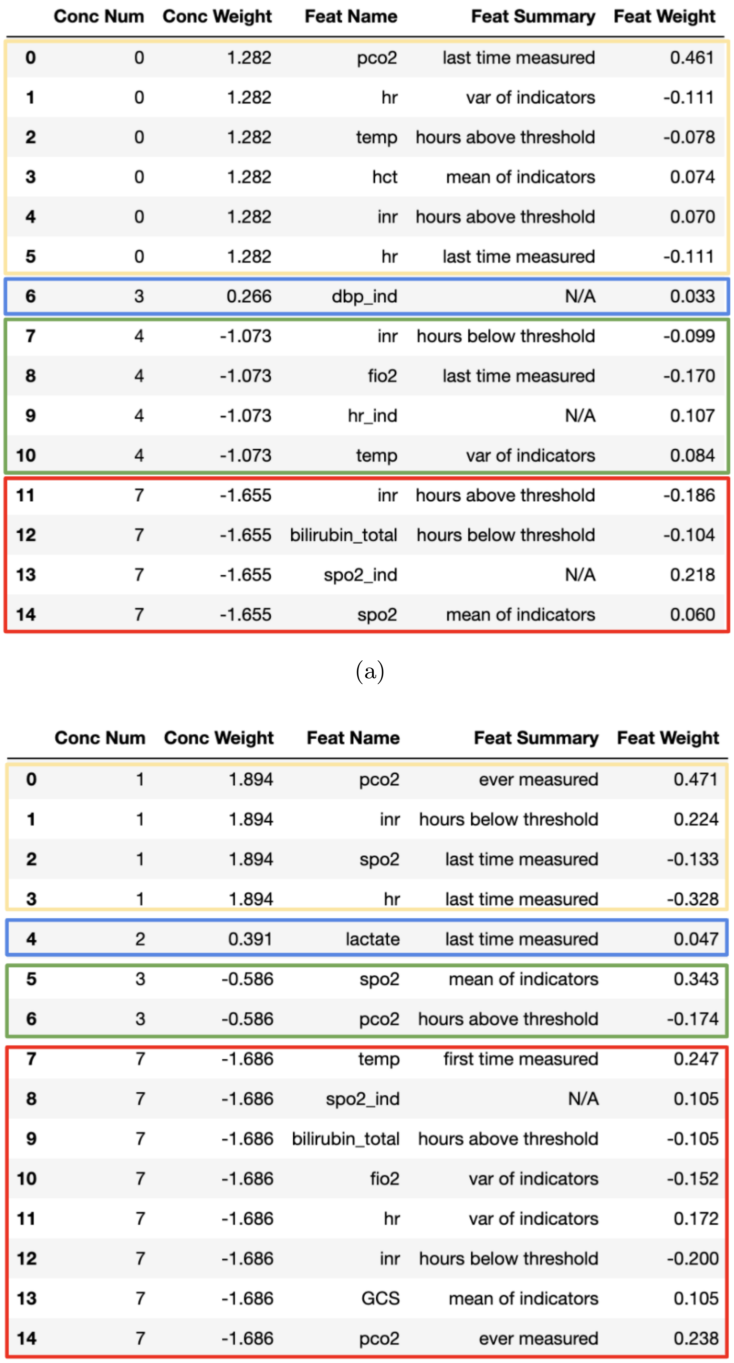
Figure 9: Simple Concept Definitions for Swapping Concept Experiment

Feature Change	AUC before	% AUC after
spontaneuousrr to concept 0	0.721	0.585
map to concept 0	0.721	0.608
pco2 to concept 3	0.721	0.635
inr to concept 3	0.721	0.634

Table 3: Changing Concept Definition AUCs

A.3. 8 Concept Experiment

We conducted an experiment to examine our greedy optimization method when run on a model with a larger number of concepts. Would all the concepts be necessary to achieve similar performance as our baseline 4-concept model? Would the selected features within each concept be similar? Figure 10 shows the concept definitions learned by our optimization method on pre-trained 8-concept models. Interestingly, we found that our initial conclusion of only using 4 concepts was verified, as the optimization method ultimately only chose features from 4 concepts for the top 15 features. The concept definitions themselves are slightly different compared to those from our original 4-concept model experiments, but this is expected because the most important/predictive features from our original 4-concept model are now spread across 8 concepts, and each of the 8 concepts may learn other auxiliary features.



	Conc Num	Conc Weight	Feat Name	Feat Summary	Feat Weight
0	0	1.282	pco2	last time measured	0.461
1	0	1.282	hr	var of indicators	-0.111
2	0	1.282	temp	hours above threshold	-0.078
3	0	1.282	hct	mean of indicators	0.074
4	0	1.282	inr	hours above threshold	0.070
5	0	1.282	hr	last time measured	-0.111
6	3	0.266	dbp_ind	N/A	0.033
7	4	-1.073	inr	hours below threshold	-0.099
8	4	-1.073	fio2	last time measured	-0.170
9	4	-1.073	hr_ind	N/A	0.107
10	4	-1.073	temp	var of indicators	0.084
11	7	-1.655	inr	hours above threshold	-0.186
12	7	-1.655	bilirubin_total	hours below threshold	-0.104
13	7	-1.655	spo2_ind	N/A	0.218
14	7	-1.655	spo2	mean of indicators	0.060

(a)

	Conc Num	Conc Weight	Feat Name	Feat Summary	Feat Weight
0	1	1.894	pco2	ever measured	0.471
1	1	1.894	inr	hours below threshold	0.224
2	1	1.894	spo2	last time measured	-0.133
3	1	1.894	hr	last time measured	-0.328
4	2	0.391	lactate	last time measured	0.047
5	3	-0.586	spo2	mean of indicators	0.343
6	3	-0.586	pco2	hours above threshold	-0.174
7	7	-1.686	temp	first time measured	0.247
8	7	-1.686	spo2_ind	N/A	0.105
9	7	-1.686	bilirubin_total	hours above threshold	-0.105
10	7	-1.686	fio2	var of indicators	-0.152
11	7	-1.686	hr	var of indicators	0.172
12	7	-1.686	inr	hours below threshold	-0.200
13	7	-1.686	GCS	mean of indicators	0.105
14	7	-1.686	pco2	ever measured	0.238

(b)

Figure 10: Concept definitions from our bottleneck models

A.4. Completeness Scores

Recently, Yeh et al. (2020) introduced the notion of completeness, which quantifies the sufficiency of a particular set of concepts in explaining a model's prediction behavior. This metric can be applied to a set of concept vectors that lie in a subspace of some intermediate DNN activations. Below is the formal definition introduced by Yeh et al.:

Given a prediction model $f(\boldsymbol{x}) = h(\phi(\boldsymbol{x}))$, where $\phi(\cdot)$ represents the intermediate concept layer and $h(\cdot)$ maps the intermediate layer to the output, a set of concept vectors $\boldsymbol{c_1}, ..., \boldsymbol{c_m}$, we define the completeness score $\eta_f(\boldsymbol{c_1}, ..., \boldsymbol{c_m})$ as:

$$\eta_f(\boldsymbol{c_1}, ..., \boldsymbol{c_m}) = \frac{\sup_g \mathbb{P}_{\boldsymbol{x},y\sim V}[y = \operatorname{argmax}_{y'} h_{y'}(g(v_{\boldsymbol{c}}(\boldsymbol{x})))] - a_r}{\mathbb{P}_{\boldsymbol{x},y\sim V}[y = \operatorname{argmax}_{y'} f_{y'}(\boldsymbol{x})] - a_r}$$

where V is the set of validation data, $\sup_g \mathbb{P}_{\boldsymbol{x},y\sim V}[y = \operatorname{argmax}_{y'} h_{y'}(g(v_{\boldsymbol{c}}(\boldsymbol{x})))]$ is the best accuracy by predicting the label just given to the concept scores $v_{\boldsymbol{c}}(\boldsymbol{x})$, and a_r is the accuracy of random prediction to equate the lower bound of completeness score to 0.

We conducted an additional experiment to compare the completeness scores of CBMs with different numbers of concepts to verify that our selection of a 4-concept model best balanced the trade-off between sparsity and completeness.

Num of Concepts	Completeness Score
2	0.4637
3	0.6687
4	0.9445
5	0.9462
6	0.9441
7	0.9551
8	0.9526

Table 4: Completeness Scores for CBMs with 2-8 concepts

Based on the completeness scores above in Table 4, we see that our model choice of using 4 concepts is validated, as 4 concepts provides high completeness as a sufficient statistics for model prediction while best maintaining sparseness.