# BOXHED FUSE: CLINICAL NOTE EMBEDDINGS FOR SURVIVAL ANALYSIS

An Undergraduate Research Scholars Thesis

by

AARON SU

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Dr. Bobak Mortazavi

May 2024

Major: Computer Science

# RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Aaron Su, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Faculty Research Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

# TABLE OF CONTENTS

Page

# ABSTRACT

BoXHED Fuse: Clinical Note Embeddings For Survival Analysis

Aaron Su
Department of Computer Science
Texas A&M University

Faculty Research Advisor: Dr. Bobak Mortazavi
Department of Computer Engineering
Texas A&M University

Many works in clinical machine learning use data from Electronic Health Records (EHR) to model patient outcomes and predict medical events. EHR consists of patient demographics, lab measurements, and physician/ nursing notes. While the majority of machine learning has utilized numerical measurements, different modalities such as clinical notes have been underutilized in machine learning applications despite their informativeness. Clinical notes contain latent medical information not captured in lab measurements, including patient history, qualitative observations, and insights from medical professionals, which can improve clinical machine learning models' abilities to provide patient risk monitoring and decision support. The under-utilization of clinical notes in clinical models was largely due to the high computational cost and lack of long-range dependencies of older recurrent language models. However, the development of attention-based transformer models with faster, parallelized training and the ability to capture long range dependencies and deep semantic understanding has made clinical notes a more viable data source for deep learning. Transformer models have led to better clinical note embeddings, or high-dimensional vector representation of unstructured notes. These embeddings capture semantic information from clinical narratives and facilitate downstream tasks such as event prediction and survival analysis.

Survival analysis is the study of time-to-event for an event of interest, such as mortality or invasive-ventilation. BoXHED 2.0 is a state-of-the-art survival analysis model with the unique ability to model recurrent events with time-varying covariates. Invasive-ventilation, a high-risk medical intervention with well-established life saving capabilities, is one such recurrent event. Recent clinical machine learning literature trends show that clinical models can have performance improvements from the addition of clinical note embeddings. This work incorporates clinical note embeddings into the BoXHED 2.0 survival dataset, leading to performance gains in measures of AUROC and AUPRC on invasive-ventilation risk monitoring .

# ACKNOWLEDGMENTS

## Contributors

I would like to thank my faculty advisor, Dr. Mortazavi for his patience, guidance, and support throughout my 3 semesters working under him.

I would also like to thank my research mentor Arash Pakbin, who I consulted frequently for both big-picture questions and technical details.

Thanks also to Nick Robert, my thesis reviewer, my colleagues at the STMI lab, and my friends who gave me feedback and support during the research process.

Finally, thanks to my parents for their encouragement, patience, and love.

The data analyzed/used for BoXHED_Fuse were sourced from The Medical Information Mart for Intensive Care IV (MIMIC IV), a freely accessible electronic health record dataset. All other work conducted for the thesis was completed by the student independently.

## Funding Sources

This undergraduate research received no funding.

# NOMENCLATURE

| | |
|---|---|
| BoXHED | Boosted eXact Hazard Estimator with Dynamic covariates |
| MIMIC | Medical Information Mart for Intensive Care |
| AUROC | Area Under the Receiver Operating Characteristic Curve |
| AUPRC | Area under Precision-Recall Curve |
| NLP | Natural Language Processing |
| EHR | Electronic Health Records |
| LSTM | Long Short-Term Memory network |

# 1. INTRODUCTION

## 1.1 Motivation

With the increasing efforts to bring machine learning technology to healthcare, many healthcare providers are turning towards clinical machine learning models to analyze medical data, extract meaningful insights, predict patient outcomes, and develop accurate and personalized healthcare interventions. These clinical machine learning models rely on Electronic Health Records (EHR). EHR contain multiple data modalities, such as numerical time-series data and unstructured clinical notes.

Numerical time-series data consists of lab measurements such as temperature, glucose levels, and respiration. Time-series data are simple for clinical models to interpret and have been used in clinical machine learning for mortality prediction, classification, and decompensation.

Clinical notes contain latent medical information and insights not captured in lab measurements. However, unstructured clinical notes have seen less use in machine learning applications due to challenges with extracting usable note information for machine learning models. Recent advances in Natural Language Processing have made unstructured clinical notes a viable data source, with language models becoming faster and gaining deeper semantic understanding. The advent of note embeddings, or high-dimensional vector representations of unstructured notes, have made it possible for clinical models to utilize the rich note information without relying on naive heuristics or manual feature engineering. Thus, there is a significant gap between the information potential of clinical notes and the incorporation of notes in clinical machine learning and decision support.

The use of clinical notes are beginning to be explored in machine learning tasks such as mortality prediction [3][2], decompensation, and length-of-stay forecasting[2]. Survival analysis, the study of time-to-event of an event of interest such as mortality, is another domain which can benefit from the incorporation of clinical note data. While early works have begun to explore the use of clinical notes in survival analysis [6], this area of note-augmented survival analysis is much less explored than note augmentation in prediction models.

Real-time risk monitoring of invasive-ventilation presents unique challenges which are not answered by traditional prediction or survival analysis models. Traditional survival analysis typically relies on assumptions of time-static data and non-recurrent events such as mortality, which limit the ability of survival models to use rich time-varying covariates from EHR and model the recurrent nature of invasive-ventilation. Invasive-ventilation is a high-risk procedure with proven life-saving capabilities. Patient risk monitoring can help healthcare professionals predict occurrences or recurrences of invasive-ventilation.

Unlike traditional survival models, Boosted eXact Hazard Estimator with Dynamic covariates v2.0 (BoXHED2.0) [4], a non-parametric hazard estimator, is well suited to modeling recurrent events with time-varying covariates. Thus, the incorporation of clinical notes in BoXHED2.0, can explore how the added richness of clinical note data can improve a state-of-the-art survival analysis model on the unique task of invasive-ventilation risk modelling.

In this work we use publicly available EHR data from MIMIC IV (Medical Information Mart for Intensive Care, version IV.) [12], a publicly available EHR dataset containing clinical notes and numerical data collected from patients at Beth Israel Deaconess Medical Center in Boston, MA, USA. This work builds off of BoXHED 2.0 by augmenting numerical ICU data with clinical note embeddings, leading to performance improvements in risk monitoring of invasive-ventilation.

### 1.2 Note Embeddings

While unstructured notes can be hard for machine learning models to train on, note embeddings offer an elegant solution. To incorporate clinical notes in machine learning models, language models can be used to transform clinical notes into document-level note embeddings. Clinical note embeddings are high-dimensional vectors which represent condensed numerical forms of textual information extracted from electronic health records. These embeddings are then used in downstream machine learning models such as Neural Nets or Tree-Boosted Hazard Estimators. Attention-based transformers, a recent advancement in natural language processing (NLP), enhance the power of note embeddings by capturing intricate semantic relationships and dependencies in notes. These embeddings enable more robust analysis, retrieval, and decision-making in healthcare applications.

## 1.3 Invasive-Ventilation

Invasive-ventilation is a high-risk procedure with proven life-saving capabilities. One epidemiological study found that from 1993 to 2009, nonsurgical Invasive Mechanical Ventilation usage increased from 179 to 310 occurrences per 100000 US adults, with an overall decrease in adjusted hospital mortality, particularly for Chronic Obstructive Pulmonary Disease and Pneumonia [22]. Invasive ventilation is commonly associated with complications such as respiratory muscle dysfunction and lung injury. Cases of mechanical ventilation lasting at least one week are associated with adverse long term physical and mental health effects such as an increased rate of in-hospital-mortality and reduced cognitive ability [20]. Due to the high risk of invasive-ventilation and the potential harm of misdiagnosis, invasive-ventilation risk monitoring models must be able to properly distinguish between high-risk and low-risk patients, including patients who have had previous invasive-ventilation interventions. Patient risk monitoring can help healthcare professionals predict occurrences or recurrences of invasive-ventilation, promoting early and accurate interventions. To provide early prediction and decision support for mechanical ventilation interventions, various machine learning methods have been developed, with XGBoost models being able to perform mechanical ventilation prediction with high accuracy. [21]. With its ability to model recurrent events, BoXHED 2.0 is well suited for this task. The added richness of clinical notes for recurrent invasive-ventilation has potential to increase BoXHED 2.0's invasive-ventilation risk monitoring capabilities.

## 1.4 Survival Analysis

Survival analysis is the analysis of survival time, or time-to-event. The theoretical basis of survival analysis is defined by survival data, where within a given cohort, the observed event time, $\tilde{T}$ is given by the smaller of two independent random variables: survival time or "event time" ($T$)

and censoring time ($C$). To avoid confusion, we will use "event time", which is the time when the patient experiences the event of interest. Censoring time is the time when a patient can no longer be observed due to mortality or leaving the hospital. This censoring is considered "right censoring" because the patient is censored at the end of their period of observation.

$$\tilde{T} = min(T, C) \tag{1}$$

By defining the cumulative distribution, $F\_surv$, survival time can be modeled with respect to time.

$$F_{surv}(t) = P(T \leq t) \tag{2}$$

From (2), we derive the complementary survival function $S(t)$, which is easier to interpret in the context of time-to-event. The survival function represents the probability that a subject survives beyond a specified time point

$$S(t) = 1 - F_{surv}(t) = P(T > t) \tag{3}$$

In survival analysis with time-varying covariates, the survival function cannot be directly estimated. Instead, the instantaneous measurement of risk is estimated, defined by the hazard function, $\lambda(t)$. The hazard function, or "force of mortality" is defined as the probability that an event will occur at some time, t.

$$\lambda(t)dt = P(\tilde{T} < t + dt | \tilde{T} \geq t) \tag{4}$$

or, using the survival function,

$$\lambda(t)dt = -S'(t)/S(t) \tag{5}$$

and

$$S(t) = \exp\left(-\int_0^t \lambda(u)du\right). \tag{6}$$

The hazard function is an approximation of the instantaneous risk of an event occurrence. Unlike survival models, hazard estimators have the ability to model time-varying covariates. For example, the conditional survival function can be defined by

$$S(t|X) = \exp\left(-\int_0^t \lambda(u, X)du\right). \tag{7}$$

However, in the case of time-varying covariates, $S(t|X)$ is undefined, as the future trajectory of the time-varying covariate, $X$, is unknown.

### 1.5 BoxHED 2.0

BoXHED 2.0, a Tree-Boosted Hazard Estimator builds upon Lee et. al's theoretical underpinning for a fully nonparametric hazard estimator for time-varying covariates [18]. As a hazard estimator, BoXHED 2.0 aims to estimate the probability of an event ocurring in $[t, t + dt)$

$$\lambda(t, X(t))Y(t)dt \tag{8}$$

where $\lambda(t, X(t)$ is the hazard at time t with time-varying covariates $X(t)$, and $Y(t) \in 0, 1$ indicates whether the subject is at-risk during $[t, t + dt)$. For example, a patient currently undergoing invasive-ventilation is not at risk for another invasive-ventilation event.

In traditional survival analysis problems, such as mortality prediction, there can only be a single event occurrence. This principle of "You only die once" [14] does not apply to recurring events such as invasive-ventilation. Thus, a patient may have more than one event over the course of a single ICU stay. For the special case with recurring events, a counting variable $N(t)$ is introduced. $N(t)$ indicates the number of events that have occurred by time $t$. Thus, we define the i-th sample of a dataset with $n$ data samples as

$$X_i(t), T_i(t), N_i(t). \tag{9}$$

BoXHED 2.0 uses functional gradient boosting to minimize the negative log-likelihood functional (likelihood risk) for the log-hazard function, defined as

$$R_n(F) = \sum_{i=1}^{n} \left\{ \int Y_i(t) e^{F(t, X_i(t))} dt - \int F(t, X_i(t)) dN_i(t) \right\}. \tag{10}$$

A function $\hat{F}$ which minimizes the likelihood risk is a candidate for the log-hazard estimator. The estimate for hazard can be found with $\hat{\lambda} = e^{\hat{F}}$.

### 1.6 Related Works

To provide a baseline machine learning benchmark dataset from MIMIC III data, a preprocessing pipeline was created, and has become standard for many works utilizing MIMIC III [1]. Data for this work is drawn from the same pipeline, modified for MIMIC IV and invasive-ventilation. With recent improvements in natural language processing (NLP), clinical notes have become increasingly prevalent within the realm of clinical machine learning, being used in isolation as well as in conjunction with numerical electronic health records (EHR). Khadanga's foundational work established several methods of combining clinical notes with time-series data, using Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs) to predict in-hospital-mortality, decompensation, and length-of-stay [2]. Decompensation and length-of-stay prediction are both similar tasks to survival analysis. However, prediction models do not have the ability to capture right-censored data as a survival analysis model would. In more recent works, the Bidirectional Encoder Representations from Transformers (BERT) was used, increasing performance on MIMIC III benchmarks [3]. Khadanga and Deznabi's works both use an averaging approach for patients with multiple notes. A more recent work, Clinical XLNet, takes a different approach by training an LSTM to generate sequential embeddings, or embeddings generated from a sequence of embeddings [5]. Further, there has been much recent interest in multi-modal fusion, or different methods of joining clinical note embeddings with time-series data [9]. For simplicity, this work does not explore advanced fusion methods, but relies on simple concatenation.

While less explored, clinical notes have been used in combination with EHR measurements for survival analysis. In BERTSurv, note embeddings and tabular EHR data are used to train a simple feed-forward network for survival analysis [6]. BERTSurv concatenates all notes over the first 4 hours of a patient's stay, which results in note data becoming a static data point rather than a time-varying narrative. With a key advantage of survival analysis being the ability to model time-varying covariates, this method limits the informativeness of clinical notes. We posit that analyzing notes in real-time will be a more effective use of clinical notes.

# 2. METHODS

## 2.1 Overview

To augment the original BoXHED 2.0 dataset with note embeddings, we choose a language model, define a fine-tuning target, fine-tune the language model, and extract note embeddings. The considerations for choosing the language model were made based off of BoXHED 2.0's existing dataset. BoXHED 2.0 is trained on MIMIC IV EHR data. Therefore, we choose Clinical-T5, a model pre-trained on MIMIC III and MIMIC IV data. To obtain, extract, and preprocess data from MIMIC IV into usable survival data, we follow the methods from a similar work using BoXHED 2.0 monitoring invasive-ventilation [19]. The cohort and data were extracted from a MIMIC-III preprocessing pipeline modified for MIMIC-IV [1]. Binary and multi-class classification fine-tuning targets were chosen to preserve the notion of right-censoring while capturing the time-dynamic nature of hazard estimation. Using these fine-tuning targets, Clinical-T5 was fine-tuned for 10 epochs, and the checkpoint with the best F-score was chosen. Finally, the fine-tuned clinical language model was used to extract embeddings from radiology notes and augment times-series data.

## 2.2 Model Choice

The choice of language model is an important consideration when extracting note embeddings. The decision to use transformers was straightforward, as attention-based transformers have been widely adopted for their state-of-the-art performance on various benchmarks. Clinical language models are domain-specific language models specifically trained on healthcare data, enabling them to outperform general models on tasks involving clinical text, while achieving similar performance to larger, general models [7]. We focused our search on models pretrained on MIMIC III and/or MIMIC IV data, as pre-training of language models is a lengthy process and outside the scope of this work.

One model we considered was Clinical-Longformer, sourced from the Hugging Face library. Clinical Longformer's enhanced architecture includes a global attention mechanism, allowing it to efficiently process longer texts by considering information from distant parts of the input sequence [8]. However, upon fine-tuning on MIMIC-IV notes, it failed to converge. This was may be due to it having been pre-trained solely on MIMIC III data.

With MIMIC IV notes only being released recently, Clinical T5 was the only publicly available language model pre-trained on MIMIC IV data. Clinical T5 is a variant of the T5 (Text-To-Text Transfer Transformer) model that has been pre-trained on clinical text data from the MIMIC-III and MIMIC-IV corpora. It is suited for healthcare applications, enabling tasks such as summarization, question answering, and text generation using clinical narratives and medical records [24]. Thus, we chose Clinical T5 as our language model. This would allow implementation of the fine-tuning process without the need to dedicate additional time and GPUs to the lengthy process of pre-training on MIMIC IV data. While there are several versions of Clinical T5, we chose Clinical-T5-Base, a model initialized from T5-Base, then further trained on MIMIC-III and MIMIC-IV. While larger T5 models were available, we decided against them due to their greater resource requirements for fine-tuning. Clinical-T5-Base, having the same architecture as T5-Base, contains 220 million parameters. We download Clinical-T5-Base from physionet.org [16].

## 2.3 Implementation

### *2.3.1 Cohort Curation*

We begin cohort curation by creating a set of all patients and ICU stays within the original BoXHED 2.0 time-series data. Note data are drawn from the MIMIC IV note dataset. Note data entries are labelled with a note id, ICU stay id, and chart time, or the time at which a particular entry or observation is documented in a patient's medical chart. Using the $chart\ time$, $note\ id$, and $icustay\ id$, we merge each note with the time series data to obtain the most recent note. Figure A.2 shows an example of merging note data into the time series by chart time.

### *2.3.2 Model Architecure*

The model architecture consists of a fine-tuned Clinical-T5 language model for extracting note embeddings, a classification/ regression head for fine-tuning, and BoXHED 2.0, for performing survival analysis (figure 1).

The original Clinical-T5 model contains 220 million parameters. T5 models are sequence-to-sequence, meaning that they have an encoder and a decoder. For the purposes of generating embeddings, only the encoder was used. Thus, the encoder has 110 million parameters. The classification head contains 640 thousand parameters. An initial attempt was made to fine-tune with a frozen encoder, but this failed to converge during fine-tuning. Therefore, the full encoder with classification head was fine-tuned.

T5 models have a maximum input sequence length of 512 tokens. Therefore, we truncate the input sequences at 512 tokens before feeding them into the model. While T5 models are typically used in an encoder-decoder scenario, we make use of the encoder only. The output of the encoder is a 768 dimension representation of the note, which is fed into a classification head. The last hidden layer of our classification head is a 64 dimension tensor, which we output as our note embedding. We pick 64 dimensions as opposed to 768 dimensions due to computational constraints for downstream training on BoxHED 2.0.

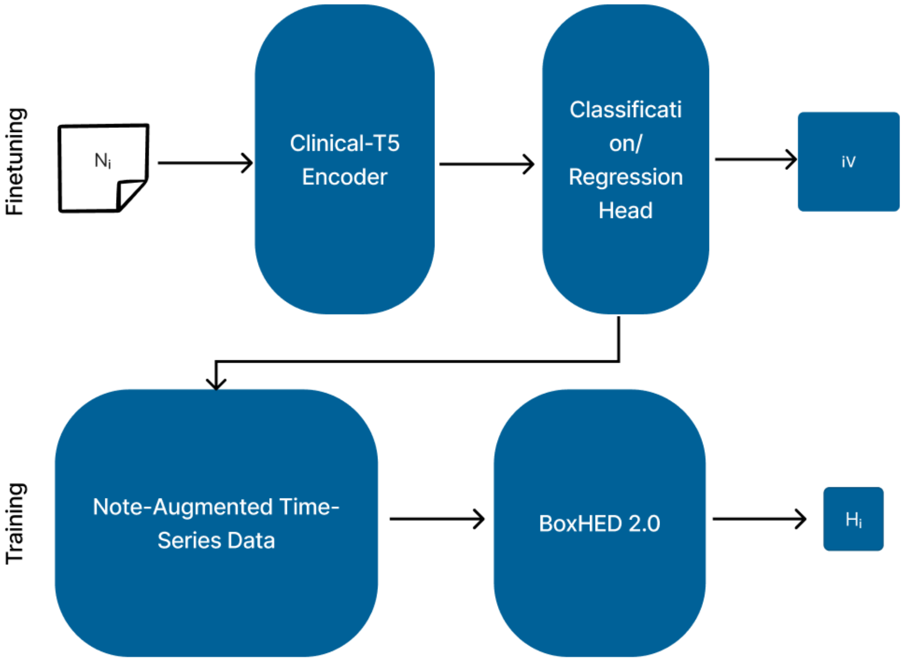
Finetuning
$N_i$
Clinical-T5 Encoder
Classificati on/ Regression Head
iV
Training
Note-Augmented Time-Series Data
BoxHED 2.0
$H_i$

**Figure 1:** Model Architecture for BoXHED Fuse. In the fine-tuning step, the Clinical-T5 Encoder is trained on individual notes from the training set with an invasive-ventilation (iV) target, which may be classification or regression. Once fine-tuning is completed, both training and test notes are fed to the encoder, and the last hidden state of the classification head is used to augment the time series data. This augmented data is then used to train and evaluate BoXHED 2.0.

### *2.3.3 Choosing the Fine-Tuning Target*

Next, we define training targets to fine-tune the language model. The targets are used to capture a notion of time-to-event. Unlike a hazard estimator, which estimates the instantaneous probability that an event will occur, we fine-tune the language model by predicting whether or not an event will occur within a defined time window. The fine-tuning data consists of all unique notes within the training cohort and the fine-tuning target.

A naive attempt was made for direct regression on time-to-event. However, this target lacked the ability to distinguish between censoring time and survival time, and classification targets were chosen instead.

The first classification target, $delta\_in\_2\_days$, defines a binary classification problem. The language model is fine-tuned to predict whether or not there will be an event occurring within 2 days of the current epoch. With binary classification, a patient with a positive label has at least 1 true event occurrence in the near future, indicated by $delta$. However, a patient with a negative label has the possibility of 0, 1, or more event occurrence in the future. This approach retains the time-dynamic nature of survival analysis and accounts for censoring.

The second classification target, $delta\_in\_1, 3, 10, 30, 100\_hours$, defines a multi-class classification problem. The language model is fine-tuned to predict whether or not there will be an event occurring within 6 buckets, where the first 5 buckets correspond to an event occurring within 1, 3, 10, 30, or 100 hours, respectively and the 6-th bucket corresponding to ICU stays with an event in > 100 hours or with no event at all (right-censored). We found that bucketing into 1, 3, 10, 30, and 100 hours provided a good distribution of time points within each bucket. This increased the specificity of each class without sacrificing the idea of censored ICU stays, as all censored ICU stays were bucketed together. Intuitively, notes which are more recent will have a greater impact on the patient's current risk. Thus, having a fine-grained sense of recency can create a more nuanced note embedding for survival analysis.

The reason for a categorical target rather than a regression target is the nature of right-censored data. In a time-dynamic prediction problem, there is no way of knowing whether a subject will be censored or have an event occurrence. Therefore, using a categorical time-window approach allows right-censored data to be grouped together with patients who will have a future event that does not fall into the time-window.

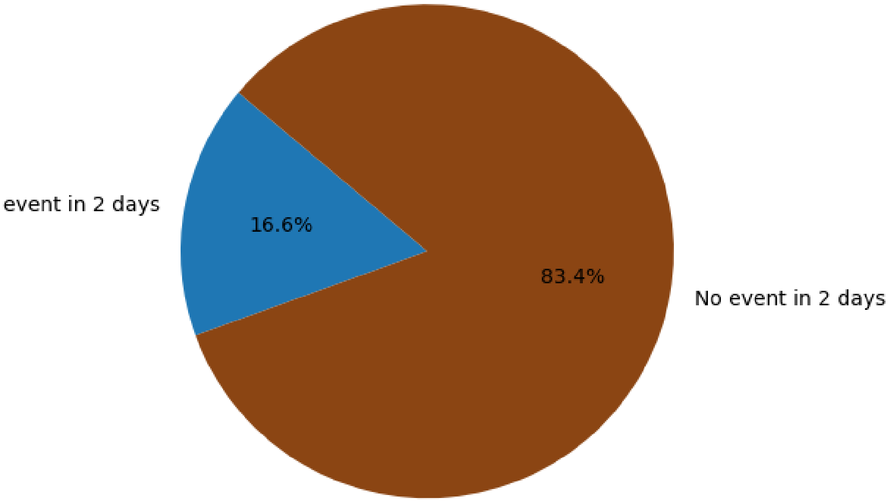

**Figure 2:** Binary Classification: Distribution of training cohort note data with/without an event in 2 days.

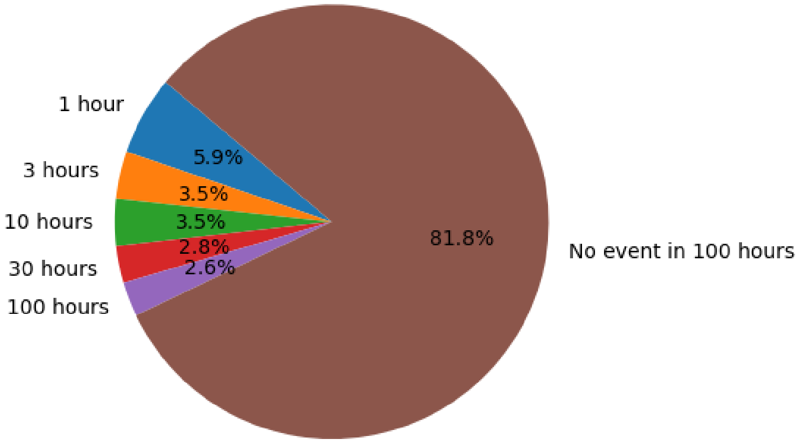

**Figure 3:** Multi-class Classification: Distribution of training cohort note data defined by 6 time buckets.

### *2.3.4 Fine-tuning and Embedding Extraction*

The fine-tuning of our language model was performed using the Hugging Face Trainer class, which is built on PyTorch and provides a simple solution for fine-tuning transformer models. The optimizer (AdamW) and linear scheduler with warmup are both chosen from the default Trainer implementation. AdamW is a well tested optimizer with strong convergence and stability during training. Initial efforts were made to perform k-fold training and cross validation. However, due to the resource and time costs of fine-tuning, a simple 80/20 train/validation split was chosen. Our loss function is CrossEntropyLoss for binary classification and BCEWithLogitsLoss for multi-class classification. The fine-tuning hyperparameters are shown in table 1.

We fine-tune on a single NVIDIA GeForce GTX 1080 Ti GPU for 10 epoch, for Clinical-T5. This took 6 hours per epoch. The batch size, 48, was calculated from 2 samples per device with 8 gradient accumulation steps and 3 devices. Due to the memory requirements of fine-tuning Clinical-T5, 2 notes per device was the maximum amount possible for our GPUs. Model checkpoints were saved at the end of each epoch, and the best model checkpoint was picked using F-score from the evaluation set. Figures 2 and 3 show that the majority of notes fall into the category of not having an event in the given time window. Due to the imbalanced fine-tuning datasets, the F-score is a useful metric for choosing an optimal fine-tuned model. For the binary classification target, the 5-th epoch was optimal, and for the multi-class classification target, the 6-th epoch was optimal.

Using the fine-tuned language model, we extract note embeddings, merging them with the tabular data according to their chart times (time the notes were charted). On the embedding extraction pass, each note from the train and test cohorts passes through the fine-tuned Clinical-T5 encoder and classification head, and the last hidden state of the classification head is extracted as a 64 dimension note embedding. In each row of the sequential tabular data, the most recent note embedding is added along with the $time\_since\_note$, the time elapsed since the subject's most recent note. Figure 4 illustrates the data augmentation, and table 2 provides a detailed view of what this would look like within the training dataset. Finally, the note-augmented dataset is used to train BoXHED 2.0 for invasive-ventilation hazard estimation.

**Table 1:** Fine-tuning Hyperparameters. The device batch size is 2. However, using 8 accumulation steps and 3 devices, the actual batch size is 48.

| | |
|---|---|
| Batch Size | 48 |
| Learning Rate | 2e-5 |
| Optimizer | AdamW |
| Scheduler | linear |
| Scheduler Warmup steps | 200 |

### 2.4 Data

Time-series and note data are sourced from MIMIC-IV, the Medical Medical Information Mart for Intensive Care, version IV. MIMIC-IV is a freely accessible electronic health record dataset, collected from the deidentified patient health records from Beth Israel Deaconess Medical Center. MIMIC-IV data is grouped into three sections – $hosp$, $icu$, and $note$. Our work makes use of $icu$ and $note$. Data for BoXHED 2.0 are primarily drawn from the $chartevents$ table from the $icu$ module, which contains routine vitals, ventilator settings, and the majority of information for a patient's stay. Chart events are entered manually by doctors and nurses, and automatically by medical devices.

To preprocess the data for survival analysis, we use Harutyunyan's preprocessing pipeline, modified for MIMIC IV and invasive-ventilation[1]. This is the same modified pipeline used for invasive-ventilation risk monitoring in another BoXHED 2.0 paper [19]. Starting risk estimation at 24 hours into the ICU stay, we obtain a cohort of 29,108 patients and 36,068 ICU stays. From this cohort, we obtain 94618 radiology notes. We add several additional features relevant to invasive-ventilation and respiration, namely, inspiratory/expiratory, oxygen flow and consumption, and respiratory rate. We impute missing values with the last observed value. To track invasive-ventilation recurrence, we also add the cumulative number of past invasive-ventilations as well as time since last invasive-ventilation (if any). Next, we randomly split the patient cohort into training and test sets resulting in a train set with 24,764 patients (30,716 ICU stays) and a test set with 4,344 patients (5,352 ICU stays). The train set contains 76148 relevant radiology notes, and the test set contains 18470.

Drawing data from MIMIC IV's chart events and some lab events, data is merged into a time-series. Each time-step, or "epoch", contains a start time, end time, ICU stay identifier, subject, delta, and 96 other covariates. Epochs have a median duration of 17 minutes and do not overlap. The event indicator is represented by a binary variable, $delta$. If invasive-ventilation occurs at the end of an epoch, it is indicated by a positive $delta$ value.

Invasive ventilation is a recurring event. Therefore, there may be multiple $delta$ values (event indicators) within a single ICU stay. Within the training cohort, 69.0% of ICU stays have 0 events, 28.9% have 1 event, and 2.1% of ICU stays contain 2 or more occurrences of invasive-ventilation.

While MIMIC III contains general and clinical notes, MIMIC IV only contains clinical notes, namely radiology and discharge notes. This means there is less frequency of note information per ICU stay in MIMIC IV compared to MIMIC III. MIMIC III contains 31.7 notes on average per ICU stay, while MIMIC IV contains 3.7 radiology notes and 1.0 discharge notes per ICU stay. Figure A.1 shows the distributions of notes between MIMIC III and MIMIC IV. We choose MIMIC IV notes because BoXHED 2.0 is trained on MIMIC IV time-series data. However, the reduced amount of notes per stay means we cannot directly compare with similar works utilizing MIMIC III notes.

Radiology notes describe a variety of imaging modalities such as x-ray, tomography, magnetic resonance imaging, and ultrasound. Discharge notes, on the other hand, contain discharge summaries for hospitalizations. They are long form narratives which describe a patient's reason for admission, their stay in the hospital, and their discharge instructions. Both note types contain relevant information for invasive-ventilation risk monitoring. However, the timing and length of discharge notes makes it less optimal for survival analysis.

Radiology notes are shorter and more frequent with a mean of 266 tokens and 2.8 radiology notes per ICU stay. On the other hand, discharge notes only occur at the end of ICU stays and contain 3368 tokens on average. Considering the time-dynamic nature of invasive-ventilation clinical decision support, radiology notes provide more information during the icu stay while the patient is still at risk. For reasons of note size, frequency, and timeliness, this work utilizes only radiology notes.

In the final augmented dataset, the time series data is concatenated with the note embeddings. Note embeddings are represented by 64 columns in the CSV. Each epoch contains the most recent note embedding for the patient, or NaN if the patient has no previous notes. Taking inspiration from the variable $t\_from\_last\_IV\_t\_start$, which indicates the time since the beginning of the last invasive-ventilation event, an additional variable $time\_since\_note$ was added. This variable adds a notion of note recency to the data.

Table 2 shows a sample dataset, which is stored as a CSV file to be consumed by BoXHED 2.0. Each row of the data frame defines a single, non-overlapping "epoch", which begins at $t\_start$ and ends at $t\_end$. $X0$ to $X95$ represent the 96 covariates sourced from Harutyunyan's data extraction pipeline [1] modified for MIMIC IV. $emb0$ to $emb63$ represent the 64 dimensional note embedding corresponding to the most recent note. If the subject has no notes at the start of an epoch, the embeddings have a NaN (not a number) value. There is a note introduced for a subject with $ICU\_Stay$ 1 at row 1, meaning that there was a note entered into the patient's EHR sometime within the epoch [0.0200, 0.0747). The event indicator $delta$ shows that there are 2 events occurring in $ICU\_Stay$ 1 and 1 invasive-ventilation event occurring in $ICU\_Stay$ 2. Row 4 shows a note that was recorded 10.5 hours before the beginning of the ICU stay, resulting in $time\_since\_note$ = 10.50, and row 7 shows a second note being added for $ICU\_Stay$ 2.

Figure 4 shows, graphically, how note embeddings are concatenated with time-series data before being fed into BoXHED 2.0.

**Table 2:** Example Augmented Dataset

| | ICU_Stay | t_start | t_end | X0 | ... | X95 | emb0 | ... | emb63 | time_since_note | delta |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0.011 | 0.020 | 0.266 | ... | 0.206 | NaN | ... | NaN | NaN | 0 |
| 1 | 1 | 0.020 | 0.075 | 0.325 | ... | 0.981 | 0.329 | ... | -0.617 | 0.000 | 1 |
| 2 | 1 | 0.075 | 0.107 | 0.783 | ... | 0.438 | 0.329 | ... | -0.617 | 0.055 | 0 |
| 3 | 1 | 0.107 | 0.153 | 0.757 | ... | 0.779 | 0.329 | ... | -0.617 | 0.087 | 1 |
| 4 | 2 | 0.207 | 0.235 | 0.962 | ... | 0.086 | 0.412 | ... | 0.542 | 10.500 | 0 |
| 5 | 2 | 0.235 | 0.272 | 0.359 | ... | 0.024 | 0.412 | ... | 0.542 | 10.528 | 1 |
| 6 | 2 | 0.272 | 0.312 | 0.639 | ... | 0.905 | 0.412 | ... | 0.542 | 10.565 | 0 |
| 7 | 2 | 0.312 | 0.351 | 0.435 | ... | 0.235 | -0.203 | ... | 0.511 | 0.000 | 0 |

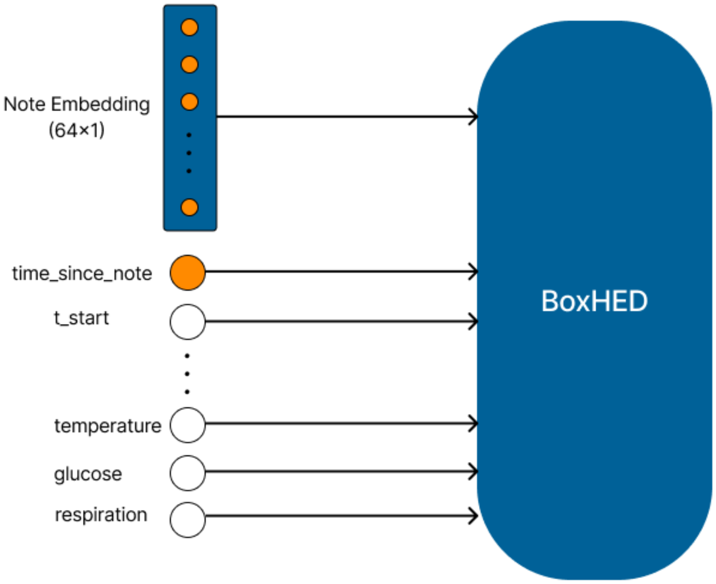

**Figure 4:** Note embedding augmentation by concatenation. Time series columns include start time, end time, ICU stay id, subject id, delta, and 96 covariates. 64 dimensional note embeddings are concatenated to time-series data, adding 64 columns to the CSV file.

### *2.4.1 Additional Experiments*

With many patients having multiple notes over the course of an ICU stay, a method of extracting note embeddings from sequences of clinical notes was attempted. This was inspired by Clinical XLNet [5], which had a two-phase approach of meta-fine-tuning using individual notes followed by fine-tuning using sequences of embeddings.

Mimicking this two-phase approach, we first extract embeddings from single notes following a similar binary classification fine-tuning procedure described in 2.4.1 and 2.4.2. Then, we pass a sequence of note embeddings through a Long Short-Term Memory network (LSTM) to extract sequential embeddings. Thus, for an ICU stay with 3 notes, a single sequential embedding can represent the accumulated note data for the patient. To train the LSTM the binary classification target $delta\_in\_2\_days$ for the LSTM was created in the same manner as the fine-tuning target for Clinical-T5, using sequences of embeddings as training data.

For hyperparameter testing, the LSTM was evaluated using bayes optimization over batch size, learning rate, and schedule/ no scheduler. Next, we used grid search to evaluate bidirectional/ unidirectional LSTM and the maximum number of notes in a sequence. The training process failed to converge after several epochs[1].

Huang's original approach aimed to predict mortality and prolonged-mechanical-ventilation (PMV). It also utilizes MIMIC III data and Clinical XLNet, whereas we use MIMIC IV data and Clinical T5. MIMIC III contains a greater frequency and number of notes per patient, which could be a possible reason for this failure to converge. With how different our approaches are, we fail to imitate the sequential embedding from Huang's approach, and do not continue with sequential note embedding extraction.

[1] This "epoch" refers to a single pass over the training set, not the "epoch" defined for the BoXHED 2.0 dataset

# 3. RESULTS

## 3.1 Performance

We evaluate BoXHED 2.0 performance using area-under-the-receiver-operating-characteristic curve (AUROC) and area-under-the precision-recall-curve (AUPRC).

As a hazard estimator, BoXHED's performance is measured by its ability to properly flag high-risk patients. A flag is considered a true positive if the patient experiences an invasive-ventilation event at some future point within the same ICU stay, and a false positive otherwise. The decision of whether or not to flag a patient is determined by a threshold, which can vary depending on the clinical setting. For example, it may be useful to flag patients with an estimated hazard of 0.30 in certain cases, whereas a higher threshold of 0.95 may be more useful in other cases. AUROC and AUPRC allow for evaluation across all thresholds.

The receiver-operating-characteristic (ROC) curve is the curve defined by the true positive rate and the false positive rate across all classification thresholds. AUROC measures the model's ability to discriminate between positive and negative classes. AUROC values range from 0 to 1. A purely random model will give AUROC = 0.50 on a large enough sample size, while a model with better ability to discriminate will have a value closer to 1.

The precision recall (PR) curve is the curve defined by the precision and recall across all classification thresholds. AUPRC measures the ability of the model to perform well on imbalanced datasets. Precision refers to the proportion of correctly predicted positive (flagged) instances among all instances predicted as positive, while recall represents the proportion of correctly predicted positive instances among all actual positive instances.

We found that both note-augmented datasets outperformed the baseline (time-series only) dataset by a small margin. Data with embeddings fine-tuned on binary classification outperform the baseline in both AUROC and AUPRC, while data with embeddings fine-tuned on multi-class classification outperform the baseline in AUPRC while maintaining AUROC. Figures B.1, B.2, B.3, and B.4 show the ROC and PR curves for each augmented dataset.

**Table 3:** Survival Analysis

| **Dataset** | **AUROC** | **AUPRC** |
|---|---|---|
| Baseline – Time series only | 0.85 | 0.34 |
| Note-Augmented – Fine-tuned on binary classification | **0.86** | **0.38** |
| Note-Augmented – Fine-tuned on multi-class classification | 0.85 | 0.36 |

### 3.2 Variable Importance

Assuming $K$ trees and $L$ internal nodes per tree, variable importance for the k-th variable $I_k$ is defined by

$$I_k = \sum_{m=0}^{M-1} I_k(g_m) = \sum_{m=0}^{M-1} \sum_{l=1}^{L} \Pi_{m,l} I(v(m,l) = k) \geq 0 \tag{11}$$

where $g_m$ is the m-th tree, and $\Pi_{m,l}$ is the split score at node $l$ of tree $m$. The split score is the measure of reduction in likelihood risk $R_n$ (10) from splitting at node $l$. $I(v(m,l) = k)$ indicates that we are only summing split scores for the k-th variable. By summing over all $L$ internal nodes and $M$ trees, $I_k$ represents the total reduction in likelihood risk that can be attributed to variable $k$. To obtain relative importances, variable importances are scaled linearly to be between 0 and 1.

As expected, the baseline dataset, containing only time series data, has similar variable importances with the augmented datasets, with Fraction inspired oxygen, pH, and Glascow coma scale total being the three most important numerical variables in all three datasets. Figures B.5, B.6, and B.7 show the top 20 feature importances for each of the three datasets.

The model trained on the dataset augmented with binary classification fine-tuned embeddings places a high importance on a single embedding feature, 'emb1', while the other embeddings show less importance (shown in figure B.8). The order of embedding features has no direct interpretation, as embedding features are learned and not defined. However, having a single embedding feature with a higher importance could indicate that the embedding could benefit from dimensionality reduction.

On the other hand, the model trained on the dataset augmented with multi-class classification fine-tuned embeddings has less skew of importances for different embedding features (shown in figure B.9). This may indicate a more nuanced understanding of note data. Even if there is a more nuanced note embedding from this dataset, it is not ultimately as useful for downstream hazard estimation.

Based off of AUROC and AUCPR, the augmented dataset with embeddings fine-tuned on binary classification performed better overall. We expect this to reflect in a greater total variable importance of embedding features. We define the embedding percentage contribution as

$$\frac{\sum_{e=1}^{E} I_e}{\sum_{k=1}^{K} I_k}, \tag{12}$$

the sum of variable importances for all $E$ embedding features divided by the sum of variable importances for all $K$ features, which includes embedding features. Summing total embedding feature importances, we find that binary classification embeddings have a 27.8% contribution compared to multiclass classification embeddings with 20.3% contribution. Thus, note embeddings fine-tuned on binary classification show better performance on the training set and greater feature importance overall.

Furthermore, the variable $time\_since\_note$ is among the top 4 most important variables in both augmented datasets, indicating that the notion of note recency or timeliness provides valuable information for note-augmented hazard estimation.

Skewed embedding feature importances suggest the potential for a richer or more compact note embedding. Nonetheless, the high relative variable importance of some embedding features along with $time\_since\_note$ suggest that note embeddings were useful for BoXHED 2.0's hazard estimation, corroborating the improvement in AUROC and AUPRC results.

### 3.3 Discussion of Results

Based on the marginal improvements in AUROC and AUPRC, there are three likely explanations, which leave room for future work and exploration.

First, MIMIC IV clinical note information could provide weak support for invasive ventilation. This is supported by the lower frequency and diversity of MIMIC IV notes compared to MIMIC III, and could be tested with further testing of BoXHED2.0 using solely note-information.

Second, BoXHED 2.0's time-series dataset already could have near-optimal hazard estimation. Considering that BoXHED 2.0 already outperforms similar survival models on invasive-ventilation risk monitoring by a large margin [19], adding additional features to a near-optimal model can only provide marginal improvements. This could be tested by augmenting data for weaker survival models, which may see greater improvements.

Third, the method of fine-tuning could be lacking in nuance and embedding richness. This is indicated by the highly skewed embedding feature importances shown in figure B.9 and B.9, where BoXHED appears to pick only a few useful embedding features to use. Experimenting with various fine-tuning methods and language models may lead to richer note embeddings.

Regardless, the marginal improvements in AUROC and AUPRC show that the addition of clinical note embeddings creates a richer dataset for BoXHED 2.0 to monitor invasive-ventilation risk.

# 4. CONCLUSION

## 4.1 Discussion

BoxHED 2.0 provides medical professionals with early detection and risk monitoring of invasive-ventilation. Through automatic assessment of patient risk of invasive-ventilation, BoX-HED 2.0 can assist medical professionals in risk monitoring and decision support for invasive-ventilation interventions. The introduction of clinical note embeddings to BoXHED 2.0 increased its performance, measured by AUROC and AUPRC. Following in the footsteps of other clinical models which have effectively used clinical notes in multi-modal approaches, this work shows that this multi-modal approach can be extended to the realm of survival analysis and recurrent survival analysis.

## 4.2 Limitations and Future Work

This work was a proof of concept, testing the added informativeness of clinical notes in survival analysis. By choosing Clinical-T5 as our model, we were limited by a 512 token input size, leaving some of the note data unused. To capture the full extent of the note's information, future work might implement a chunking strategy, where a single note is divided into chunks and evaluated together during fine tuning. An alternative, sliding window model, such as Long- former, could also be used to capture the entire note, as such a model would support longer input sequences.

One limitation of this work is it's data source. MIMIC IV note data is limited to clinical notes – radiology and discharge. Compared with MIMIC III note data which contains a greater variety of notes such as Nursing and Physician notes, MIMIC IV contains much less note data per ICU stay, as seen in figure A.1. Replicating this study on MIMIC III data would produce a greater amount of note embeddings per ICU stay, which could allow for more frequent and timely note embeddings in the final augmented time series dataset. This would also allow for comparative analysis among other papers utilizing MIMIC III note embeddings. Further, MIMIC IV is still under development and may add additional note types to the dataset in the future, making it comparable to MIMIC III notes.

Lastly, further experiments could investigate BoXHED 2.0's performance on a note-embedding-only dataset, note augmentation on other survival models, and fine-tuning other language models. These experiments would provide a better understanding of how rich the note embeddings themselves are, and how they could be improved.

# REFERENCES

[1] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Scientific Data*, vol. 6, p. 96, June 2019.

[2] S. Khadanga, K. Aggarwal, S. Joty, and J. Srivastava, "Using Clinical Notes with Time Series Data for ICU Management," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (Hong Kong, China), pp. 6431–6436, Association for Computational Linguistics, 2019.

[3] I. Deznabi, M. Iyyer, and M. Fiterau, "Predicting in-hospital mortality by combining clinical notes with time-series data," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (C. Zong, F. Xia, W. Li, and R. Navigli, eds.), (Online), pp. 4026–4031, Association for Computational Linguistics, Aug. 2021.

[4] A. Pakbin, X. Wang, B. J. Mortazavi, and D. K. K. Lee, "BoXHED2.0: Scalable boosting of dynamic survival analysis," Feb. 2023.

[5] K. Huang, A. Singh, S. Chen, E. T. Moseley, C.-y. Deng, N. George, and C. Lindvall, "Clinical XLNet: Modeling Sequential Clinical Notes and Predicting Prolonged Mechanical Ventilation," Dec. 2019.

[6] Y. Zhao, Q. Hong, X. Zhang, Y. Deng, Y. Wang, and L. Petzold, "BERTSurv: BERT-Based Survival Models for Predicting Outcomes of Trauma Patients," Mar. 2021.

[7] E. Lehman, E. Hernandez, D. Mahajan, J. Wulff, M. J. Smith, Z. Ziegler, D. Nadler, P. Szolovits, A. Johnson, and E. Alsentzer, "Do We Still Need Clinical Language Models?," Feb. 2023.

[8] Y. Li, R. M. Wehbe, F. S. Ahmad, H. Wang, and Y. Luo, "Clinical-Longformer and Clinical-BigBird: Transformers for long clinical sequences," Apr. 2022.

[9] B. Yang and L. Wu, "How to leverage the multimodal EHR data for better medical prediction?," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4029–4038, Association for Computational Linguistics, Nov. 2021.

[10] A. E. W. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow, L.-w. H. Lehman, L. A. Celi, and R. G. Mark, "MIMIC-IV, a freely accessible electronic health record dataset," *Scientific Data*, vol. 10, p. 1, Jan. 2023.

[11] J. M. Walter, T. C. Corbridge, and B. D. Singer, "Invasive Mechanical Ventilation," *Southern Medical Journal*, vol. 111, pp. 746–753, Dec. 2018.

[12] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark, "MIMIC-IV."

[13] X. Wang, A. Pakbin, B. Mortazavi, H. Zhao, and D. Lee, "BoXHED: Boosted eXact Hazard Estimator with Dynamic covariates," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 9973–9982, PMLR, July 2020.

[14] H. v. Houwelingen, H. Putter, and J. C. v. Houwelingen, *Dynamic prediction in clinical survival analysis*. No. 123 in Monographs on statistics and applied probability, Boca Raton, Fla.: CRC Press, Taylor & Francis, 2012.

[15] A. Johnson, T. Pollard, S. Horng, L. A. Celi, and R. Mark, "MIMIC-IV-Note: Deidentified free-text clinical notes."

[16] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. E215–20, June 2000. Publisher: Ovid Technologies (Wolters Kluwer Health).

[17] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, p. 160035, May 2016.

[18] D. K. K. Lee, N. Chen, and H. Ishwaran, "Boosted nonparametric hazards with time-dependent covariates," *The Annals of Statistics*, vol. 49, Aug. 2021.

[19] A. Pakbin, Z. Nowroozilarki, D. K. Lee, and B. J. Mortazavi, "Predicting Real-time, Recurrent Adverse Invasive Ventilation from Clinical Data Streams," in *2023 IEEE 19th International Conference on Body Sensor Networks (BSN)*, pp. 1–4, 2023.

[20] T. Pham, L. J. Brochard, and A. S. Slutsky, "Mechanical Ventilation: State of the Art," *Mayo Clinic Proceedings*, vol. 92, pp. 1382–1400, Sept. 2017.

[21] L. Yu, A. Halalau, B. Dalal, A. E. Abbas, F. Ivascu, M. Amin, and G. B. Nair, "Machine learning methods to predict mechanical ventilation and mortality in patients with COVID-19," *PLOS ONE*, vol. 16, p. e0249285, Apr. 2021.

[22] A. B. Mehta, S. N. Syeda, R. S. Wiener, and A. J. Walkey, "Epidemiological trends in invasive mechanical ventilation in the United States: A population-based study," *Journal of Critical Care*, vol. 30, pp. 1217–1221, Dec. 2015.

[23] Y. Li, R. M. Wehbe, F. S. Ahmad, H. Wang, and Y. Luo, "A comparative study of pretrained language models for long clinical text," *Journal of the American Medical Informatics Association*, vol. 30, no. 2, pp. 340–347, 2023. Publisher: Oxford University Press.

[24] E. Lehman and A. Johnson, "Clinical-T5: Large Language Models Built Using MIMIC Clinical Text."

# APPENDIX A: DATA

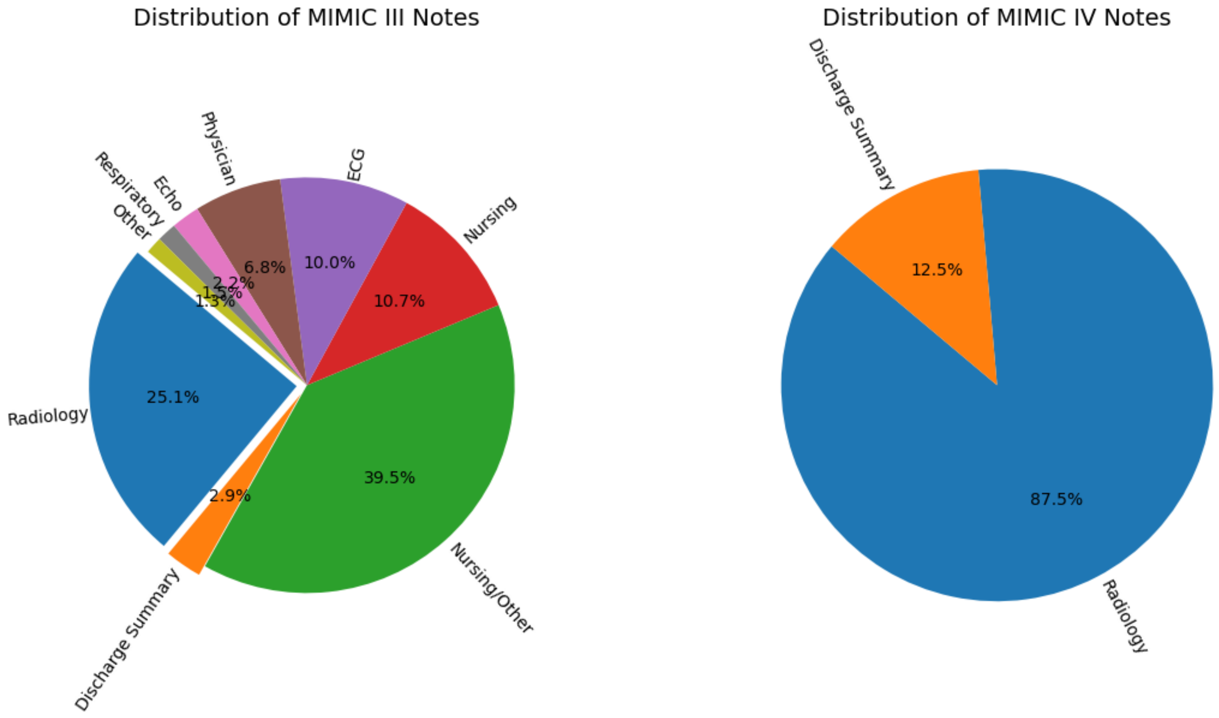

**Figure A.1:** MIMIC III vs MIMIC IV note distributions. While MIMIC IV has more notes in total, MIMIC III has a greater amount of notes per patient due to the greater variety of note types in the dataset. MIMIC IV only contains clinical notes, while MIMIC III contains a combination of clinical and general n

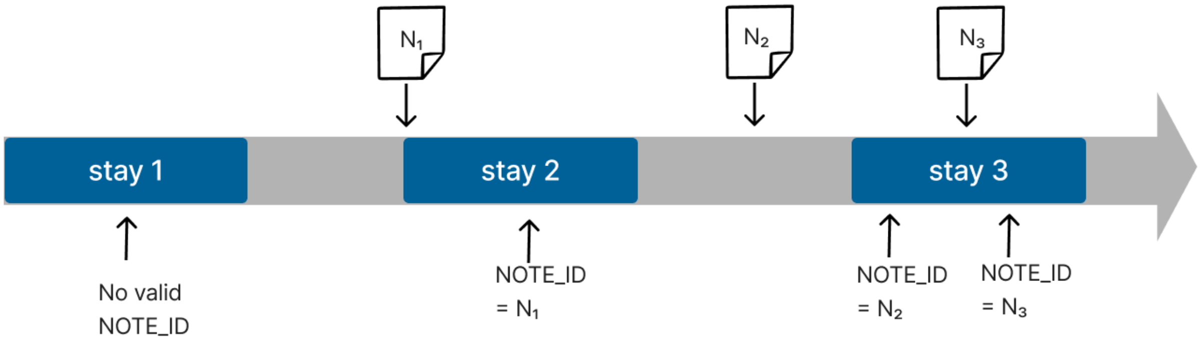

**Figure A.2:** Note data augmentation for a patient with three ICU stays.

# APPENDIX B: RESULTS

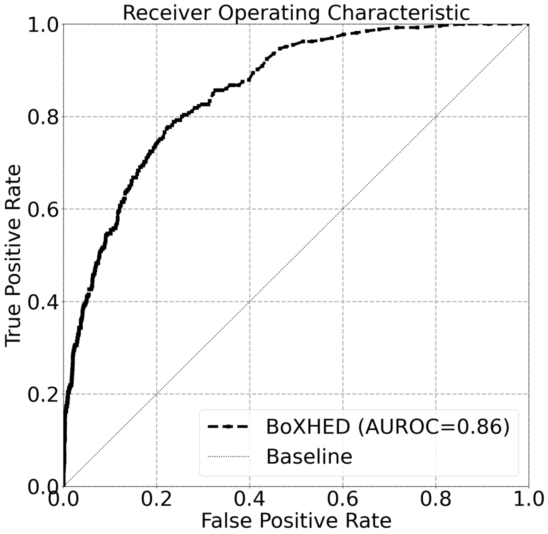

**Figure B.1:** Reciever operating characteristic curve from augmented dataset. Embeddings are drawn from binary classification fine-tuned model.

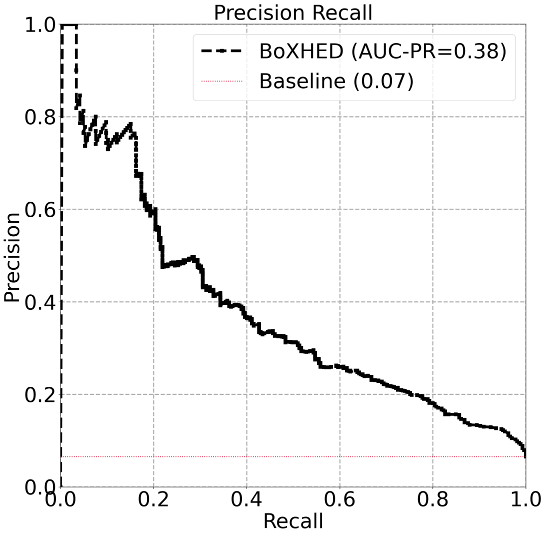

**Figure B.2:** Precision recall curve from augmented dataset. Embeddings are drawn from binary classification fine-tuned model.

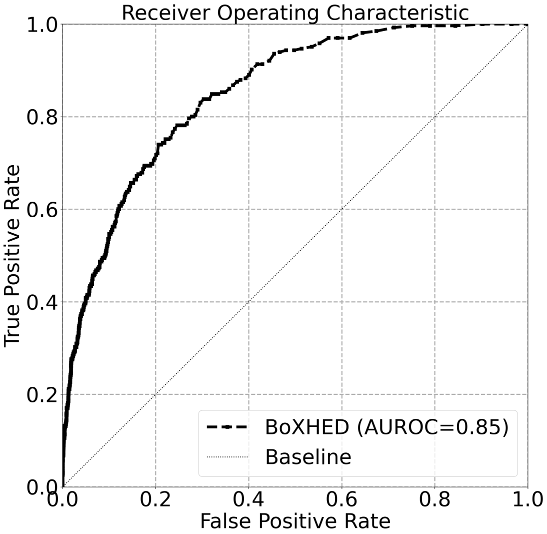

**Figure B.3:** Receiver operating characteristic curve from augmented dataset. Embeddings are drawn from multi-class classification fine-tuned model.

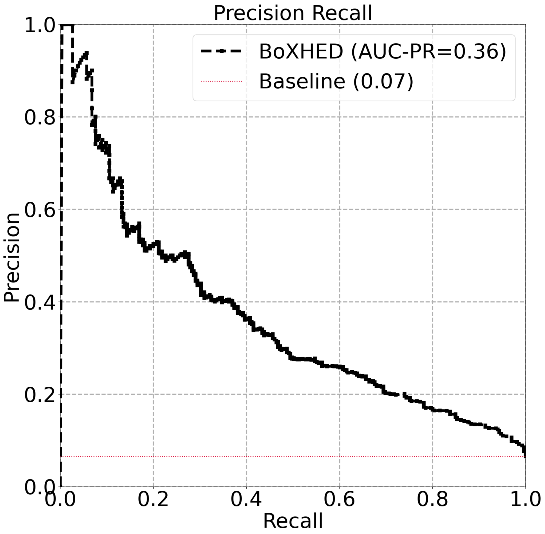

**Figure B.4:** Precision recall curve from augmented dataset. Embeddings are drawn from multi-class classification fine-tuned model.

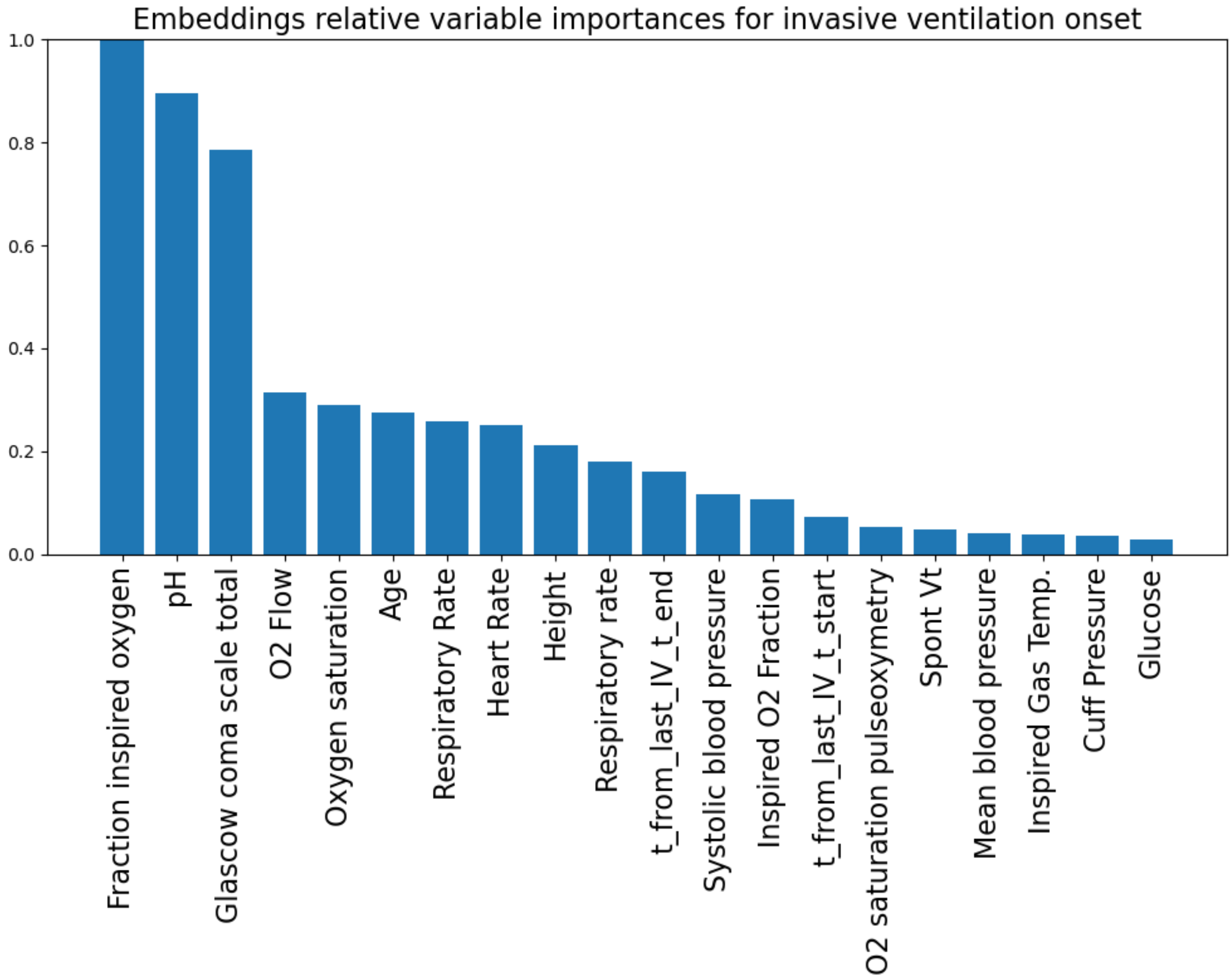

**Figure B.5:** Variable importances for baseline dataset.

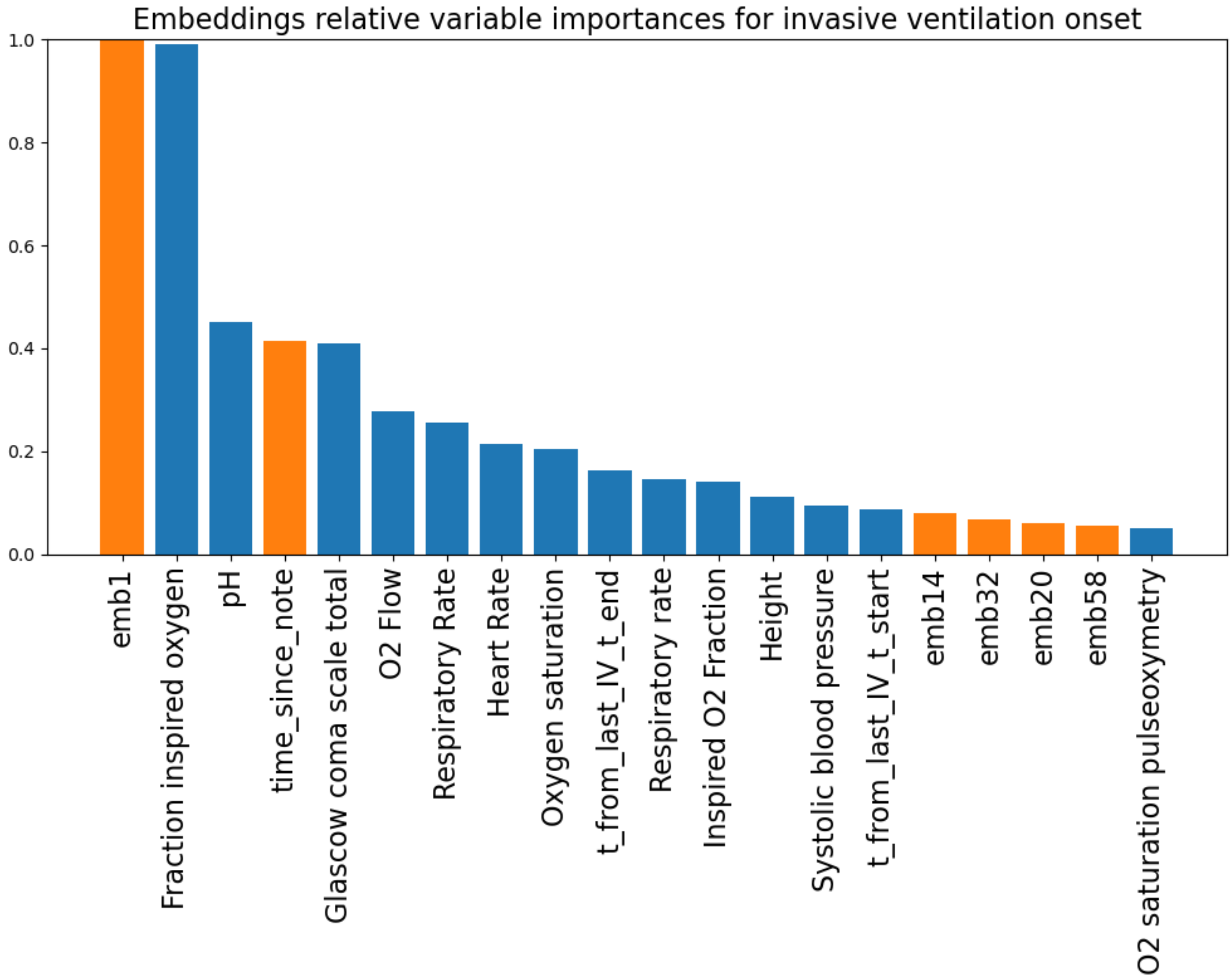

**Figure B.6:** Variable importances for augmented dataset. Embeddings are drawn from binary classification fine-tuned model.

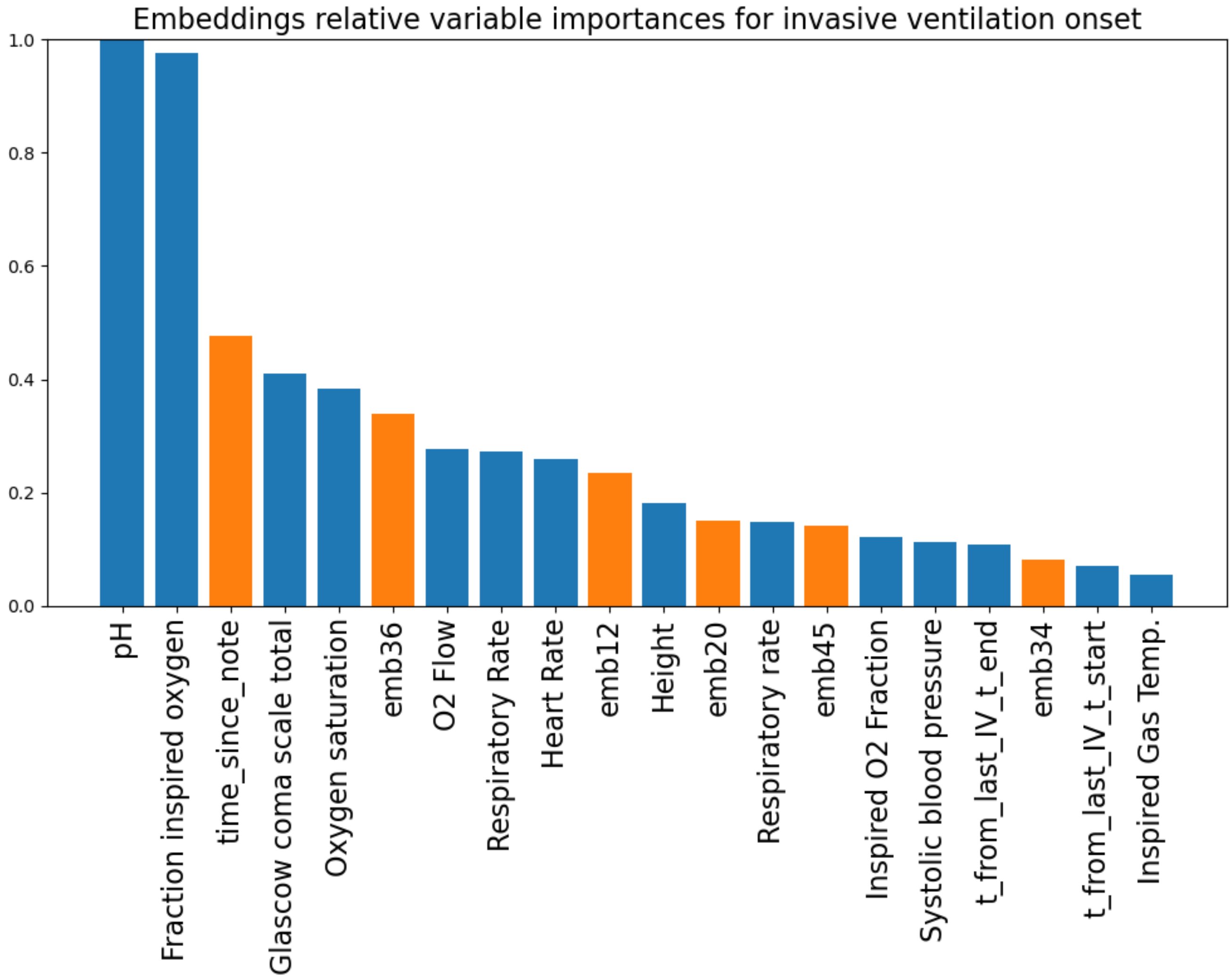

**Figure B.7:** Variable importances for augmented dataset. Embeddings are drawn from multi-class classification fine-tuned model.

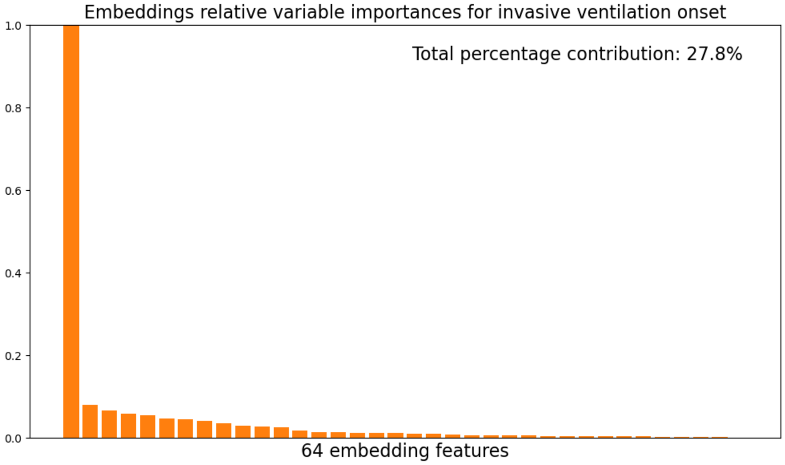

**Figure B.8:** Variable importances for embeddings drawn from binary classification fine-tuned model. Due to the arbitrary nature of feature order, embedding features are not labelled. 27.8% of total variable importance is attributed to embedding features.

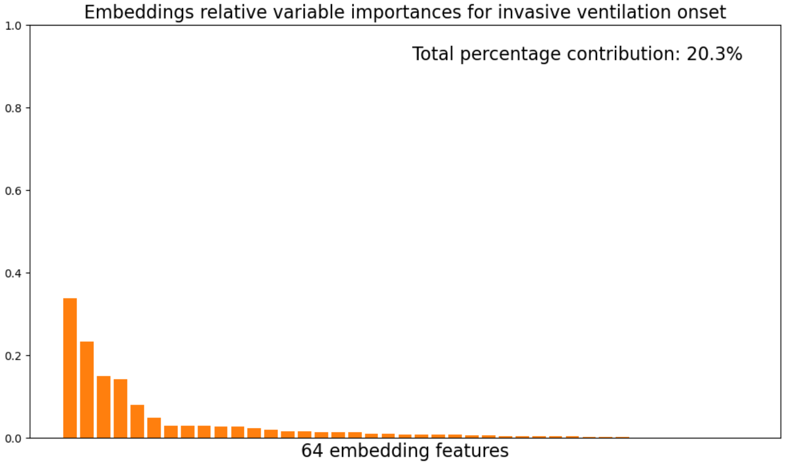

**Figure B.9:** Variable importances for embeddings drawn from multi-class classification fine-tuned model. Due to the arbitrary nature of feature order, embedding features are not labelled. 20.8% of total variable importance is attributed to embedding features