TRIDE: A Temporal, Robust, and Informative Data Augmentation Framework for Disease Progression Modeling

Anonymous authors
Paper under double-blind review

Abstract

Modeling the progression of a target disease using electronic health records (EHRs), especially early predicting the onset of a disease, is critical for timely and accurate clinical interventions. While numerous deep learning-based prediction models have shown great success in handling sequential multivariate data such as EHRs, they often ***lack temporal robustness***. This is problematic because they may not perform consistently well across different early prediction hours as training data become scarce upon targeting further future. Indeed, having even one weak point of time can significantly restrict the reliability of the models. In this work, we present **TRIDE**, a *temporal, robust and informative data augmentation* framework that can learn temporal representations of EHRs and use them to generate diverse and meaningful training samples by optimizing the level of data transformation. We validate TRIDE on modeling the progression of an extremely challenging disease, *septic shock*, by using real-world EHRs collected from two different medical systems. Our results show that TRIDE significantly outperforms strong baseline models across different prediction times and datasets, and thus enhances the temporal robustness. Further, we provide in-depth analyses of the generated samples and estimated model parameters to clarify the processes.

1 Introduction

A disease progression model (DPM) estimates disease's progression from historical data, such as multivariate time-series electronic health records (EHRs) (Mould, 2012). One important purpose of DPM is to *detect diseases early*, that is, to predict whether a patient will develop a target disease n hours later. Robust and accurate early prediction models can provide timely clinical interventions (Che et al., 2015; Marlin et al., 2012; Choi et al., 2016b;a; Zhou et al., 2013; Choi et al., 2016c; 2017a; Li et al., 2015) that will reduce the risk of mortality and the burden on patients and the healthcare system (Kumar et al., 2006; Wang et al., 2014). Many deep learning-based early prediction models have demonstrated great success in recent years by utilizing RNNs, CNNs, or Transformers to EHRs (Birkhead et al., 2015; Choi et al., 2016a;c; Esteban et al., 2016; Lipton et al., 2015; Zhou et al., 2011; Luo et al., 2020; Rasmy et al., 2021; Yang et al., 2021). However, one of the major shortcomings with them is that they do not perform consistently well across different early prediction hours, that is, ***the lack of temporal robustness***.

A major reason for the lack of temporal robustness is that as we increase n hours in our early prediction task, the observations are farther away from disease onset, and there are fewer labeled data (EHRs), which often leads to underperforming and sensitive models (Lin et al., 2019; Zhang et al., 2019; Khoshnevisan & Chi, 2021). Figure 1 shows a simulation result for an early prediction task of septic shock n hours (x-axis) later. The yellow bars indicate the numbers of training data available. As n increases (i.e., targeting more challenging early predictions), the size of the training data continuously decreases and so does the model's prediction performance (AUCs). Thus, securing sufficient amount of quality training data is the key to enhance temporal robustness of early prediction models.

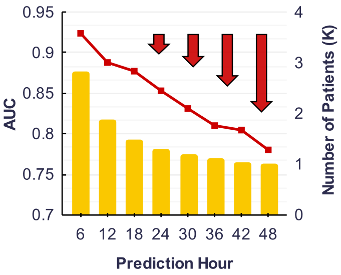


Figure 1: Temporal Robustness

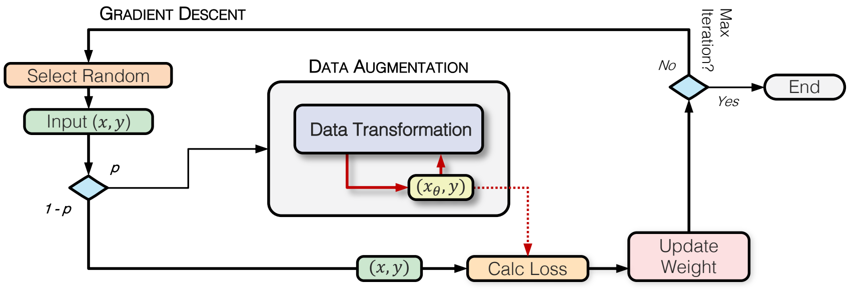


Figure 2: Overview of Proposed Framework (TRIDE)

Notes: The outer loop indicates gradient descent and the inner loop with red solid arrows represents data augmentation module. θ is the level of data transformation and x_θ is the generated synthetic sample using θ. p is the sampling probability that controls the size of data augmentation.

In this work, we propose **TRIDE**, a **t**emporal, **r**obust, and **i**nformative **d**ata augm**e**ntation framework to address the DPM's lack of temporal robustness, especially on early prediction tasks. As shown in Figure 2, TRIDE generates diverse yet informative samples for a target prediction model by integrating a ***data augmentation module*** into a model training process (gradient descent). Specifically, TRIDE optimizes the level of data transformation in a way that the generated samples can improve the model's robustness while preserving the original labels. The effectiveness of TRIDE is evaluated on the task of septic shock early prediction. Sepsis is a life-threatening condition caused by a dysregulated body response to infection (Singer et al., 2016). ***Septic shock*** is the most severe complication of sepsis, associated with high mortality rate and prolonged length of hospitalization (Singer et al., 2016). Timely treatment is particularly critical as every hour delay in antibiotic treatment leads to 8% increase in the chance of mortality. Early prediction of septic shock is yet challenging due to vague symptoms and subtle body responses (Kumar et al., 2006). Also, sepsis, like cancer, involves various disease etiologies that span a wide range of syndromes, and different patient groups may show vastly different symptoms (Tintinalli et al., 2011). Using real-world data from two different US medical systems, we show that TRIDE can optimize the level of data transformation which would significantly improve the temporal robustness of early prediction models.

Contributions There have been a few lines of research which aim to improve the robustness of prediction models. One is based on existing knowledge from large-scale pretrained models (Li et al., 2020; Rasmy et al., 2021; Pang et al., 2021) and the other is data augmentation approaches that can address the problem from the root by generating extra labeled data (Esteban et al., 2017; Che et al., 2017; Baowaly et al., 2019; Poulain et al., 2022). However, to the best of our knowledge, there is no prior work on optimizing the level of data transformation for augmenting training data in order to improve model performance or temporal robustness. To summarize, this work makes the following contributions: **(1)** To the best of our knowledge, TRIDE is the first data augmentation approach for multivariate time-series EHRs, which optimizes the data quality (i.e., the level of data transformations) to generate diverse, challenging yet informative synthetic samples; **(2)** our proposed framework outperforms the baselines on two real-world EHR datasets with various settings for an extremely challenging task, septic shock early prediction and we provide in-depth analyses on data transformation; **(3)** Our framework integrates the data augmentation component (i.e., EHR language model) into the regular gradient descent of early prediction models, thus capable of generating target-oriented synthetic samples.

2 TRIDE: Temporal, Robust, and Informative Data Augmentation

2.1 Problem Definition

Our dataset can be represented as $\mathcal{D}_{train} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ..., (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$, where N is the total number of patients or hospital visits. It is composed of multivariate irregular time series data and each visit $\boldsymbol{x}_k$ consists of a sequence of events: $\boldsymbol{x}_k = \{\boldsymbol{x}_k^1, ..., \boldsymbol{x}_k^{T_k}\}$, where $\boldsymbol{x}_k^t$ represents patient's records at timestamp t and T_k is the number of events in k-th visit, which varies across different visits. We have $\boldsymbol{x}_k^t \in \mathbb{R}^S$, where S is the number of discrete symptom or token generated from clinical measurements at each event. In addition, an output label $y_k = \{1, 0\}$ (i.e., shock or non-shock) is provided for each visit $\boldsymbol{x}_k$. The main goal of early prediction is to learn a prediction function $\mathcal{F}$ over $\mathcal{D}_{train}$ that can best approximate an unknown function $f : X \rightarrow Y$ where X is the true distribution

Algorithm 1: TRIDE

```
Input: Training data $\mathcal{D}_{train} = \{x_i, y_i\}_{i=1}^{n}$, pretrained EHR language model $\mathcal{G}$,
       prediction model $\mathcal{F}$, model weights $\mathcal{W}$, loss function $\mathcal{L}$, learning rate $\eta$
Parameters: Sampling probability $p$, $\theta$ upper bound $\mathcal{U}$, transformation space $\Theta$
Output: Prediction model $\mathcal{F}$ with trained weights $\mathcal{W}$
1  Randomly initialize model weights $\mathcal{W}$;
2  while termination criterion not met do
3      Select a training sample pair $(x, y)$ at random from $\mathcal{D}_{train}$;
4      With probability $p$:
5          Initialize $\theta_0 \leftarrow 0$;
6          for $i \in \{1, ..., N\}$ do
7              $\max_{\Delta\theta} \nabla\mathcal{L}(y, \mathscr{F}(\textsc{TransformSequence}(\mathcal{G}, x, \theta_i)))\Delta\theta$
8              Updating $\hat{\theta}_i \leftarrow \hat{\theta}_{i-1} + \Delta\theta$
9          end
10         $x_{\theta^*} \leftarrow \textsc{TransformSequence}(\mathcal{G}, x, \theta^*)$
11         $\mathcal{W} \leftarrow \mathcal{W} - \eta\nabla_{\mathcal{W}}\mathcal{L}(y, \mathcal{F}(x_{\theta^*}))$
12     With probability $1 - p$:
13         $\mathcal{W} \leftarrow \mathcal{W} - \eta\nabla_{\mathcal{W}}\mathcal{L}(y, \mathcal{F}(x))$
14 end
```

over entire population. However, when the size $\mathcal{D}_{train}$ is small or does not well represent X, it would be difficult to approximate f and lead to the lack of temporal robustness.

2.2 TRIDE, A Time-Series Data Augmentation Framework

Motivated by AnonymizedWork, we propose TRIDE, designed for augmenting multivariate time-series data such as EHRs to derive a temporally robust early prediction model. Specifically, TRIDE utilizes a contextualized EHR language model and an optimization function to transform visit-long sequences into challenging training samples which is well suited for early prediction models. TRIDE is composed of two key components (see Algorithm 1): **(1) data augmentation** (lines 5-11, inner loop), which generates optimally different synthetic sequences using constrained worst-case optimization; **(2) modified gradient descent** (lines 3-15, outer loop), which integrates the data augmentation module into model training and updates the weights interactively.

2.2.1 Input and Parameters

Given N training sample pairs $\mathcal{D}_{train} = \{x_i, y_i\}_{i=1}^{N}$ (visit sequence x_i and class labels y_i) as input, a predefined probability p selects a subset of samples and a transformation function $\mathcal{G}$ turns them into synthetic samples $\mathcal{D}_{syn}$ for augmentation. For data transformation, pretrained EHR language model $\mathcal{G}$ and transformation level θ (i.e., the distance between an original and transformed sequence) are utilized to generate synthetic sequence x_{θ_i}, in which θ lies in a predefined space $\Theta = \mathbb{R}^{[0,\mathcal{U}]}$. In this work, a pretrained EHR language model similar to CEHR-BERT (Pang et al., 2021) is used. However, since this approach is not transformation function ($\mathcal{G}$) specific, any language models that can generate a realistic time-series sequence can be utilize. The rest of the inputs is general components for a prediction task: prediction model $\mathcal{F}$, its weights $\mathcal{W}$, loss function $\mathcal{L}$, and learning rate η. TRIDE can work with any types of prediction models and matching loss functions that are differentiable and effective in evaluating loss values.

2.2.2 Data Augmentation: Optimizing Transformation Level θ

The vital part of the TRIDE algorithm is to estimate the optimal level of data transformation (θ) and this is performed by two functions: (1) constrained worst-case token (symptom) masking and (2) language model-based token prediction. Especially, the TransformSequence function in

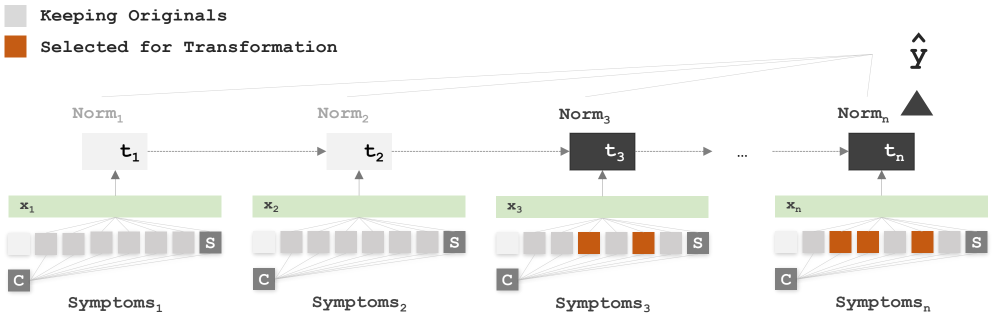


Figure 3: Mechanism of TRANSFORMSEQUENCE Function

Algorithm 1, which includes the aforementioned functions, introduces a new way of transforming multivariate time-series data (see Figure 3 for the visual description of data transformation).

First, data transformation starts with selecting target tokens for masking and we perform this task aligned with the worst-case optimization. As the objective of the optimization function is to maximize the loss values with regard to θ, we select tokens that most contribute to the loss or prediction of target class. The contribution level of input is calculated using gradient attribution (Simonyan et al., 2014) in two steps. The time-series data can be seen as document-level sequences where an event represents a sentence and a list of events within a visit represents a document. In addition, because we utilize RNN-based models such as LSTM, there exist multiple cells across time steps and each event (sentence) is fed to the corresponding time step. Due to this hierarchical structure (two levels, sentence to document) of prediction framework, it is not trivial to calculate contribution level ($l1$ norm of gradients) of output directly with respect to token-level input. Therefore, first, we determine the contribution level of each time step using the gradient attribution (the upper part of Figure 3). Within each time step, we calculate the self attention score for tokens with respect to a special token ([CLS]) and use it as a contribution score with time step. As [CLS] token contains every information collected from other tokens in a sentence and the token is often used as input for various classification tasks, we assume that attention scores with respect to [CLS] token would represent the contribution level of tokens.

Second, we estimate the amount of data transformation for controlling the data diversity. Transformation level θ determines both the number of time steps to transform and the number of tokens to mask. Specifically, we first select top θ numbers of time steps (i.e., $\theta * E$ where E is the number of events in a visit x) based on the contribution level of time steps, and then we distribute θ number of mask tokens among the selected time steps, based on the attention scores. Figure 3 illustrates the process where the amount and positions of data transformation are decided. Once target tokens within chosen time steps are replaced with [MASK] tokens, pretrained language model $\mathcal{G}$ predicts the tokens based on the context information to generate a realistic sequence of symptoms. At the end of the process, the transformed sequence with the estimated theta is added to the training dataset to update model weights.

2.2.3 DATA TRANSFORMATION $\mathcal{G}$: EHR LANGUAGE MODEL

Based on the similarity between our medical data abstracted from EHRs (sequence of symptoms) and textual data in natural language (sequence of words), we adopt the original BERT (Devlin et al., 2019) architecture with some modification. First, position embeddings are trained on the order numbers of tokens. In this work, a visit "sequence" represents a list of variable number of tokens where each token represents a single symptom, namely a measurement of each feature (e.g., temperature measured as normal) and it is ordered by the start and end time of each symptom. Second, motivated by previous works (Nguyen et al., 2017; Pang et al., 2021) and to fully incorporate temporal aspects of our sequences, we introduce an additional special token ([t-HR]) that represents time intervals between consecutive events. Features (e.g., vital signs) in EHRs are measured irregularly based on their needs and clinicians make such decisions to effectively diagnose patients' conditions and keep track of disease trajectory. That being said, by incorporating time intervals into model training, learned embeddings could embrace useful information regarding clinicians' practice of measuring patients' conditions. We expect that this would play an important role in understanding temporal relations between tokens (symptoms) within a visit and benefit the self-attention process inside the BERT framework. For a visual representation of our EHR language model, see Figure 7 in Appendix A.1.2. For pretraining a model, we inherit "masked language modeling" procedure from the original BERT paper in which the authors mask some percentage of the input tokens at random and then predict those masked tokens based on the encoder output.

2.2.4 Model Training: Modified Gradient Descent

This step is identical to the regular gradient descent of any differentiable prediction models (lines 11 and 13). Given either the original training samples or transformed samples and its corresponding class labels, TRIDE updates the model parameters $\mathcal{W}$ based on the gradient computed and the predefined learning rate η.

3 Experiment Setup

3.1 Datasets and Preprocessing

Two EHR Datasets 210,289 visits of adult patients (i.e., age $>$ 18) admitted to *Christiana Care Health System* (CCHS) in Newark, Delaware (07/2013-12/2015); 106,844 adult patient visits from *Mayo Clinic* (MAYO) in Rochester, Minnesota (07/2013-12/2015) are used. For consistency, we define our target population as *suspected of infection*, identified by administration of any anti-infectives, or a positive PCR test result. This definition and the following data pre-processing steps are determined by three leading clinicians with extensive experience.

Labeling We adopt the agreement between International Classification of Diseases, Ninth Revision (ICD-9) codes recorded in EHRs, and our expert-defined rules based on the Third International Consensus Definitions for Sepsis and Septic Shock Singer et al. (2016) to achieve the most reliable population across all datasets. Our clinicians identify septic shock at event-level as having received vasopressor(s) or persistent hypotension for more than 1 hour (systolic blood pressure (SBP)$<$90; or mean arterial pressure$<$65; or drop in SBP$>$40 in an 8-hour window).

Sampling Using the agreement criteria results in 2,963 positive cases in CCHS and 3,499 cases in MAYO. To balance the number of positive and negative cases, we perform a stratified random sampling by maintaining the same underlying age, gender, ethnicity, and length of stay distribution. In addition, 10,000 randomly sampled cases (i.e., patient visits) that do not fall under the agreement are used for pretraining EHR language model. Appendix A.2.4 describes the summary statistics.

Feature Selection A total 19 continuous features are used, as suggested by our experts; 1) vital signs (8): heart rate, temperature, systolicBP, etc., 2) lab test results (11): BUN, lactate, platelet, white blood cell count (WBC), etc. However, any other types of features can be used if the features could be properly discretized and represent clinical symptoms.

Discretization and Aggregation Motivated by AnonymizedWork, we utilized temporal abstraction (Shahar, 1997) to aggregate and discretize low-level numeric time-series data within 60 minutes interval into high-level concept sequences (i.e., a list of symptom tokens). This approach can greatly reduce the effect of irregular time intervals and missing values in EHRs.

3.2 Model Evaluation

Baseline and Proposed Models We evaluate our proposed framework, TRIDE, with five different EHR representation learning models, including BERT-based language models. In addition, all models described below employ a vanilla LSTM on their top layer for the prediction purposes. The following describes the baseline EHR representation learning models: **(1) EHR**, raw (not abstracted) EHRs with numerous enhancements: i) standardization with feature-wise mean and standard deviation, ii) aggregation of the events within an hour together with statistical features, and iii) expert rule-based imputation (Kim & Chi, 2018), where vital signs and laboratory results are carried forward with the last value for 8 hours and 24 hours, respectively.; **(2) Anon (AnonymizedWork)**, language modeling-based approach that learns representations from a sequence of abstract symptoms (tokens) same as our proposed model, but employ static language model (word2vec). **(3) BERT**, our proposed algorithm that employ BERT to learn contextualized EHR representations from visit-long sequences. The difference from ANON is the model architecture (static vs. contextualized); **(4) TIMEBERT**, a variant of the BERT model that incorporates additional time information in a form of special time interval tokens in its input; **(5) XFERBERT**, a BERT model with transfer learning. This is the same as BERT, but it is pretrained on dataset from another domain (i.e., hospital). For example, if evaluation is performed on CCHS data, a BERT model pretrained on Mayo data represents this model; Lastly, **(6) TRIDE**, our proposed time-series data augmentation algorithm, is compared to the five aforementioned models that do not incorporate data augmentation to investigate additional performance improvement from data augmentation.

Evaluation Metrics We measure the temporal robustness of models using two types of evaluation metrics aligned with our definition: (1) For raw prediction performance, we use the five metrics (accuracy, precision, recall, F1-score, and AUC) and focus on F1-score and AUC for comparison. (2) For lower-bound performance, we report the lowest performance among performance scores across all prediction hours and denote it as *F1-LB* (Lower Bound) or *AUC-LB*. The more temporally robust model will have larger lower-bound performance. In addition, we utilize 2 folds stratified cross validation and repeat an experiment 3 times for each fold. The performance of the models are reported with mean values and standard deviations from total 6 trials of experiments, and we provide significance test results.

Regular vs. Robust Two types of test set are employed for evaluation. First, "*Regular*" set, is the regular test set used for early prediction, in which the numbers of patients in training, validation, and test sets increase as models predict septic shock in nearer future. The other test set is denoted as "*Robust*" because in this setting, models are evaluated upon the same group of patients across varying prediction hours. In this setting, the only differences made by the variation of prediction hours is that the number of training and validation data. Therefore, we could evidently measure the robustness of proposed model, meaning that we could observe how model performances changes while the training data sizes vary (i.e., how sensitive each model is to training data size variances). Specifically, in *Robust* setting, every prediction tasks use the same test set, which is for the 48 hour early prediction task (see Table 3 in Appendix A.2.3 for more detail).

Hyperparameters The configuration of TRIDE hyperparameters is as follows: 1) the hyperparameter p that controls augmentation size ($\mathcal{S}$) ranges from 5% to 30% with a 5% increase; 2) the upper bound of theta ($\mathcal{U}$) that constraints θ values is set to be in the range of [0.1, 0.7] with 0.1 interval. Details of the other hyperparameter configurations are described in Appendix A.2.5.

4 RESULT

4.1 EVALUATION OF TEMPORAL ROBUSTNESS

We presents two prediction results for evaluating temporal robustness. 48 hours early prediction results reveal the general robustness (i.e., performance at specific time point) of models and 6-48 hours early prediction results uncover the temporal robustness (i.e., performance across times).

4.1.1 48 HOURS EARLY PREDICTION

Table 1 presents the experiment results (i.e., average performance from multiple runs and corresponding standard deviation) for 48 hours septic shock early prediction on two datasets, CCHS and MAYO, and six models. The first block (top three rows) on each dataset compare the baseline models and the result show that our BERT significantly outperforms ANON and EHR across two datasets, demonstrating that learning contextual information from EHRs could help building a robust model. Moreover, the performance increase from ANON to BERT is bigger in MAYO data than CCHS, assuming that this could be because of the MAYO's larger training samples used for pretraining BERT (see Table 4 in Appendix A.2.4). Prior works often show that BERT performs better when pretrained with larger training data. The second block (4-5 rows) on each dataset compare BERT-based models, XFERBERT and TIMEBERT, and shows different results between CCHS and MAYO. In CCHS, XFERBERT outperforms BERT, but BERT outperforms XFERBERT in MAYO. This suggests that training data size and quality could have a great impact on BERT performance. When comparing BERT with TIMEBERT, TIMEBERT outperforms BERT on both datasets, suggesting

Table 1: Model Performance for 48 Hours Septic Shock Early Prediction

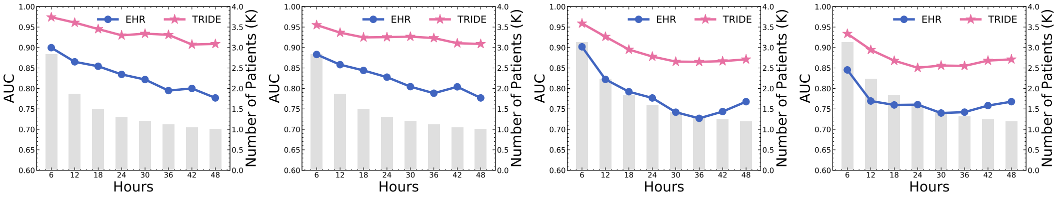
Dataset	Model	Accuracy	Precision	Recall	F1-Score	AUC
CCHS	EHR	0.706(± 0.027)	0.706(± 0.021)	0.705(± 0.044)	0.705(± 0.031)	0.777(± 0.044)
	ANON	0.748*(± 0.017)	0.739(± 0.026)	0.769(± 0.040)	0.753*(± 0.018)	0.832*(± 0.020)
	BERT	**0.789***†(± 0.030)	**0.771***(± 0.039)	**0.826***†(± 0.020)	**0.797***†(± 0.023)	**0.870***†(± 0.013)
	XFERBERT	0.814*†(± 0.016)	0.807*†(± 0.016)	**0.826***†(± 0.034)	0.816*†‡(± 0.018)	0.897*†‡(± 0.009)
	TIMEBERT	**0.821***†(± 0.021)	$\underline{\mathbf{0.825}}^{*\dagger\ddagger}$(± 0.017)	0.816*†(± 0.041)	**0.820***†(± 0.025)	**0.900***†‡(± 0.009)
	TRIDE	$\underline{\mathbf{0.829}}^{*\dagger\ddagger}$(± 0.011)	**0.813***†‡(± 0.030)	$\underline{\mathbf{0.856}}^{*\dagger}$(± 0.033)	$\underline{\mathbf{0.833}}^{*\dagger\ddagger}$(± 0.010)	$\underline{\mathbf{0.914}}^{*\dagger\ddagger\S(}$± 0.010)
MAYO	EHR	0.697(± 0.049)	0.682(± 0.043)	0.729(± 0.144)	0.697(± 0.079)	0.768(± 0.055)
	ANON	0.680(± 0.016)	0.648(± 0.028)	0.774(± 0.081)	0.703(± 0.024)	0.759(± 0.022)
	BERT	**0.771***†(± 0.017)	**0.744***†(± 0.026)	**0.820***(± 0.077)	**0.778***†(± 0.027)	**0.852***†(± 0.021)
	XFERBERT	0.750*†(± 0.021)	0.723*†(± 0.019)	0.800*(± 0.059)	0.758*†(± 0.028)	0.832*†(± 0.018)
	TIMEBERT	**0.789***†‡(± 0.015)	**0.769***†‡(± 0.028)	**0.820***(± 0.054)	**0.792***†(± 0.019)	**0.869***†‡(± 0.013)
	TRIDE	$\underline{\mathbf{0.801}}^{*\dagger\ddagger}$± 0.010)	$\underline{\mathbf{0.777}}^{*\dagger\ddagger}$(± 0.010)	$\underline{\mathbf{0.836}}$*(± 0.016)	$\underline{\mathbf{0.806}}^{*\dagger\ddagger\S(}$± 0.011)	$\underline{\mathbf{0.871}}^{*\dagger\ddagger}$(± 0.012)

Notes: The best model for each block is marked in **bold** and the best overall model is additionally underlined. *, †, ‡, and § indicate significant differences to EHR, ANON, BERT, and TIMEBERT, respectively ($p < .05$).

that incorporating time interval tokens into training was effective in learning temporal patterns of EHRs. In addition, the smaller performance improvement in MAYO data indicates that using TIMEBERT on EHRs with less uniformly distributed time intervals could be less effective. Lastly, TRIDE outperforms TIMEBERT on two datasets, suggesting that our proposed algorithm can work well on early prediction. Again, we assume that TRIDE outperforming slightly less in Mayo data could be because of the data quality and size. As shown in the XFERBERT case, Mayo may have richer and better data (see Appendix A.2.4). We conjecture that because MAYO already has good quality data, there could be less improvement for TRIDE to make.

4.1.2 6-48 HOURS EARLY PREDICTION

Figure 4 shows the experiment results for 6 to 48 hours early prediction on two models, EHR and TRIDE. We vary datasets and test sets to show the generalizable temporal robustness of TRIDE. All four plots exhibit similar patterns in which 1) TRIDE significantly outperforms EHR across all prediction hours and 2) the performance decrease of TRIDE from 6 to 48 hours is smaller than EHR (i.e., lower sensitivity to training data size decreases). Such patterns demonstrate that TRIDE can build temporally robust early prediction models.



(a) CCHS-Regular (b) CCHS-Robust (c) MAYO-Regular (d) MAYO-Robust

Figure 4: Model Performance (AUC) on 6-48 Hours Early Prediction

Table 2 presents the model performance from 6 to 48 hours early prediction. To measure model's temporal robustness, we utilizes two types of test set and two additional evaluation metrics for measuring lower-bound (LB) performance. In general, the table shows consistent results with the 48 hours early prediction. For all four settings, BERT outperformed others in all four metrics, showing the effectiveness of using BERT on EHRs and early prediction. And XFERBERT outperforming BERT in CCHS only, suggesting that data size and quality could be important in training BERT. Next, TIMEBERT outperforms BERT except for F1-LB on MAYO data. As mentioned before, we assume that less uniformly distributed time intervals in Mayo data could decrease the impact of TIMEBERT. Lastly, we compare TRIDE with TIMEBERT. The results show that TRIDE outperforms TIMEBERT in all settings including the robust and lower-bound metrics, which makes TRIDE the most temporally-robust model among the six models.

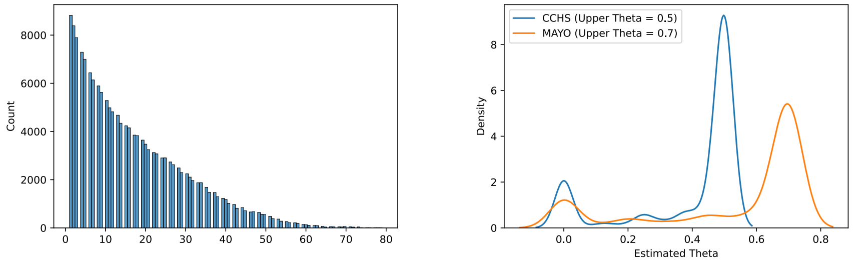
Table 2: Model Performance for 6-48 Hours Septic Shock Early Predictions (CCHS)

Dataset	Setting	Model	F1	AUC	F1-LB	AUC-LB
CCHS	**Regular**	EHR	0.762(± 0.016)	0.831(± 0.020)	0.705(± 0.031)	0.777(± 0.044)
		ANON	0.803*(± 0.008)	0.879*(± 0.006)	0.753*(± 0.018)	0.832*(± 0.020)
		BERT	**0.838**$*^{\dagger}$(± 0.006)	**0.909**$*^{\dagger}$(± 0.006)	**0.791**$*^{\dagger}$(± 0.020)	**0.868**$*^{\dagger}$(± 0.022)
		XFERBERT	0.848$*^{\dagger\ddagger}$(± 0.005)	0.920$*^{\dagger\ddagger}$(± 0.005)	0.809$*^{\dagger}$(± 0.024)	0.885$*^{\dagger\ddagger}$(± 0.017)
		TIMEBERT	**0.851**$*^{\dagger\ddagger}$(± 0.007)	**0.925**$*^{\dagger\ddagger}$(± 0.005)	**0.821**$*^{\dagger\ddagger}$(± 0.016)	**0.894**$*^{\dagger\ddagger}$(± 0.008)
		TRIDE	**0.867**$*^{\dagger\ddagger\S(}$± 0.005)	**0.936**$*^{\dagger\ddagger\S(}$± 0.004)	**0.833**$*^{\dagger\ddagger}$(± 0.018)	**0.907**$*^{\dagger\ddagger\S(}$± 0.007)
	Robust	EHR	0.744(± 0.020)	0.823(± 0.026)	0.705(± 0.031)	0.777(± 0.044)
		ANON	0.782*(± 0.014)	0.864*(± 0.008)	0.753*(± 0.018)	0.832*(± 0.020)
		BERT	**0.819**$*^{\dagger}$(± 0.009)	**0.898**$*^{\dagger}$(± 0.009)	**0.792**$*^{\dagger}$(± 0.021)	**0.873**$*^{\dagger}$(± 0.020)
		XFERBERT	0.827$*^{\dagger\ddagger}$(± 0.005)	0.909$*^{\dagger\ddagger}$(± 0.006)	0.813$*^{\dagger}$(± 0.027)	0.889$*^{\dagger\ddagger}$(± 0.016)
		TIMEBERT	**0.836**$*^{\dagger\ddagger}$(± 0.008)	**0.915**$*^{\dagger\ddagger}$(± 0.005)	**0.822**$*^{\dagger\ddagger}$(± 0.016)	**0.897**$*^{\dagger\ddagger}$(± 0.008)
		TRIDE	**0.848**$*^{\dagger\ddagger\S(}$± 0.008)	**0.926**$*^{\dagger\ddagger\S(}$± 0.004)	**0.834**$*^{\dagger\ddagger}$(± 0.021)	**0.911**$*^{\dagger\ddagger}$(± 0.006)
MAYO	**Regular**	EHR	0.736(± 0.023)	0.784(± 0.021)	0.697(± 0.023)	0.727(± 0.019)
		ANON	0.765*(± 0.004)	0.816*(± 0.005)	0.703(± 0.024)	0.759*(± 0.022)
		BERT	**0.810**$*^{\dagger}$(± 0.007)	**0.876**$*^{\dagger}$(± 0.003)	**0.779**$*^{\dagger}$(± 0.011)	**0.845**$*^{\dagger}$(± 0.010)
		XFERBERT	0.790$*^{\dagger}$(± 0.009)	0.856$*^{\dagger}$(± 0.004)	0.745$*^{\dagger}$(± 0.024)	0.811$*^{\dagger}$(± 0.029)
		TIMEBERT	**0.814**$*^{\dagger}$(± 0.005)	**0.883**$*^{\dagger\ddagger}$(± 0.004)	**0.779**$*^{\dagger}$(± 0.005)	**0.849**$*^{\dagger}$(± 0.013)
		TRIDE	**0.829**$*^{\dagger\ddagger}$(± 0.008)	**0.891**$*^{\dagger\ddagger\S(}$± 0.003)	**0.790**$*^{\dagger\ddagger}$(± 0.018)	**0.865**$*^{\dagger\ddagger\S(}$± 0.005)
	Robust	EHR	0.701(± 0.031)	0.768(± 0.028)	0.666 (± 0.045)	0.740 (± 0.030)
		ANON	0.736*(± 0.009)	0.801*(± 0.007)	0.703*(± 0.024)	0.759(± 0.022)
		BERT	**0.782**$*^{\dagger}$(± 0.013)	**0.860**$*^{\dagger}$(± 0.006)	**0.768**$*^{\dagger}$(± 0.011)	**0.838**$*^{\dagger}$(± 0.004)
		XFERBERT	0.761$*^{\dagger}$(± 0.016)	0.840$*^{\dagger}$(± 0.010)	0.735*(± 0.032)	0.809$*^{\dagger}$(± 0.036)
		TIMEBERT	**0.790**$*^{\dagger\ddagger}$(± 0.013)	**0.869**$*^{\dagger\ddagger}$(± 0.006)	**0.767**$*^{\dagger}$(± 0.011)	**0.849**$*^{\dagger\ddagger}$(± 0.007)
		TRIDE	**0.803**$*^{\dagger\ddagger\S(}$± 0.011)	**0.875**$*^{\dagger\ddagger\S(}$± 0.004)	**0.779**$*^{\dagger}$(± 0.020)	**0.851**$*^{\dagger\ddagger}$(± 0.006)

Notes: For each dataset, the best performing model for each block is marked in **bold**, and the best model across ALL is underlined and marked in **bold**. *, †, ‡, and § indicate statistically significant differences compared to EHR, ANON, BERT, and TIMEBERT, respectively ($p < .05$).

4.2 Analysis of Data Transformation

Figure 5 shows analysis results on the data transformation components of TRIDE, which essentially control the quality (diversity) of generated samples. The left sub-figure answers the where question – that is, which events or tokens were mostly transformed by TRIDE, while the right figure clarifies that how much tokens TRIDE transformed for augmentation.



(a) Distribution of Transformed Time Indices (b) Distribution of Estimated Thetas

Figure 5: Analysis Results for Data Transformations

Transformation, Where? By design, TRIDE transforms most contributing time steps within a visit. The left figure shows the distribution of the indices of transformed time steps on its x-axis, and we denote most recent time step as 0. As expected, the result show that more recent time steps (which are close to 0) were transformed more, thus indicating that they contributed more to predictions.

Transformation, How Much? In addition, we had an analysis on the transformation level (θ). The right figure shows the distributions of θs estimated during the entire training steps of TRIDE, using the best hyperparmeters, on two datasets. We can observe that two distributions are neither uniform nor skewed, showing that the TRIDE is capable of determining the optimal transformation levels with respect to target dataset and prediction models, as opposed to fixed or random levels.

5 Related Work

Temporal Robustness There exist many definitions of robustness, for example, test set performance, resistance to small and adversarial input perturbations or attacks, generalizability within or across domains (Hendrycks et al., 2019; Drenkow et al., 2021; Yi et al., 2021). However, in this work, given the nature of disease progression modeling, we focus on two components to define our temporal robustness: (1) Raw prediction performance on test sets. But, in our case, we focus on the performance across time; (2) The more important component is the worst-case performance across prediction hours. In real-world scenario, it is very difficult to know when a patient will develop a target disease. Thus, having any weak points or models throughout prediction hours would significantly degrade the model's reliability.

Language Models for EHRs Due to the importance of the ability to adjust vector representations corresponding to surroundings or contexts (Richardson et al., 2020; Tonelli et al., 2018), language models such as BERT (Devlin et al., 2019) have been widely used to assist the learning of contextualized embedding for medical concepts (e.g., diagnosis codes). Li et al. (2020) introduced BEHRT and aimed to develop a pretrained model that predicts the existence of specific medical codes in certain visits. Different from the original BERT, it incorporated patients' ages to imply temporal orders of the codes. Shang et al. (2019) proposed G-BERT that incorporated a graph neural network (GNN) to expand the context of each clinical code through ontologies. Rasmy et al. (2021) designed Med-BERT, which utilized a domain-specific pretraining task and large-scale dataset (20M patients). Pang et al. (2021) introduced CEHR-BERT which resembles our TIMEBERT. They trained a temporal BERT for EHRs by inserting time-indicative artificial tokens between visits and employed a new pretraining task, visit type prediction. Despite of the advances in learning context-aware language models for EHRs, (1) the previous works may not be suitable for capturing dynamics of rapid disease progression from multivariate time-series EHRs due to their discrete and coarse-grained medical codes (Lee et al., 2020) and more importantly, (2) no prior works have aimed to address temporal robustness in early prediction tasks.

Data Augmentation for EHRs Largely two groups of generative adversarial network (GAN)-based data augmentation approaches have been applied to time-series EHRs which contains either discrete medical codes or real-valued multivariate data. One focuses on generation solely, aiming to generate most realistic synthetic samples regardless of the samples' impact on model robustness (Choi et al., 2017b; Esteban et al., 2017; Baowaly et al., 2019; Li et al., 2021). The other group of works aimed to build detection or prediction models using a GAN architecture combined with semi-supervised learning (SSL) (Che et al., 2017; Yu et al., 2019; Cui et al., 2020; Poulain et al., 2022). This line of approaches showed improvements in predictions, but had no direct control over data transformation level, which is closely related to generating quality (i.e., sufficiently different and challenging) training data and can help improve model robustness.

6 Conclusion

Modeling patient disease progression is an essential task that supports clinical decision making. However, training data often become scarce and it leads to a significant problem, the lack of temporal robustness of early prediction models. To address the problem, we propose TRIDE, a temporal, robust, and informative data augmentation framework that generates sufficiently diverse yet informative training samples, thus deriving a temporally robust model. Toward robustness, TRIDE optimizes the diversity (i.e., the level of data transformation) of training samples by controlling the position and number of tokens to transform within time-series EHRs. The experimental results on two real-world EHR datasets demonstrate the temporal robustness of our framework compared to various baseline models. In addition, two analysis results on data transformation reveal that TRIDE's ability to accurately optimize the transformations towards improving robustness.

REFERENCES

Mrinal Kanti Baowaly, Chia-Ching Lin, Chao-Lin Liu, and Kuan-Ta Chen. Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, 26(3):228–241, 2019.

Guthrie S Birkhead, Michael Klompas, and Nirav R Shah. Uses of electronic health records for public health surveillance to advance public health. *Annu. Rev. Public Health*, 36:345–359, 2015.

Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 507–516, 2015.

Zhengping Che, Yu Cheng, Shuangfei Zhai, Zhaonan Sun, and Yan Liu. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 787–792. IEEE, 2017.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *MLHC*, pp. 301–318, 2016a.

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *SIGKDD*, pp. 1495–1504. ACM, 2016b.

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. In *Advances in Neural Information Processing Systems*, pp. 3504–3512, 2016c.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In *SIGKDD*, pp. 787–795. ACM, 2017a.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pp. 286–305. PMLR, 2017b.

Limeng Cui, Siddharth Biswal, Lucas M Glass, Greg Lever, Jimeng Sun, and Cao Xiao. Conan: complementary pattern augmentation for rare disease detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 614–621, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. Robustness in deep learning for computer vision: Mind the gap? *arXiv preprint arXiv:2112.00639*, 2021.

Cristóbal Esteban, Oliver Staeck, et al. Predicting clinical events by combining static and dynamic information using recurrent neural networks. In *ICHI*, pp. 93–101. IEEE, 2016.

Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. *Proceedings of the International Conference on Machine Learning*, 2019.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rygGQyrFvH`.

Farzaneh Khoshnevisan and Min Chi. Unifying domain adaptation and domain generalization for robust prediction across minority racial groups. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I*, pp. 521–537, Berlin, Heidelberg, 2021. Springer-Verlag. ISBN 978-3-030-86485-9. doi: 10.1007/978-3-030-86486-6_32. URL `https://doi.org/10.1007/978-3-030-86486-6_32`.

Yeo Jin Kim and Min Chi. Temporal belief memory: Imputing missing data during rnn training. In *IJCAI*, pp. 2326–2332, 2018.

Anand Kumar, Daniel Roberts, Kenneth E Wood, Bruce Light, Joseph E Parrillo, et al. Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock. *Crit. Care Med.*, 34(6):1589–1596, 2006.

Dongha Lee, Xiaoqian Jiang, and Hwanjo Yu. Harmonized representation learning on dynamic ehr graphs. *Journal of biomedical informatics*, 106:103426, 2020.

Hui Li, Xiaoyi Li, Xiaowei Jia, Murali Ramanathan, and Aidong Zhang. Bone disease prediction and phenotype discovery using feature representation over electronic health records. In *BCB*, pp. 212–221. ACM, 2015.

Jin Li, Benjamin J Cairns, Jingsong Li, and Tingting Zhu. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. *arXiv preprint arXiv:2112.12047*, 2021.

Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Sci. Rep.*, 10(1):1–12, 2020.

Chen Lin, Julie Ivy, and Min Chi. Multi-layer facial representation learning for early prediction of septic shock. In *Big Data*, pp. 840–849. IEEE, 2019.

Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 647–656, 2020.

Benjamin M Marlin, David C Kale, Robinder G Khemani, et al. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *IHI*, pp. 389–398. ACM, 2012.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pp. 3111–3119, 2013.

DR Mould. Models for disease progression: new approaches and uses. *Clin. Pharmacol. Ther.*, 92 (1):125–131, 2012.

Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. `Deepr`: A convolutional net for medical records. *IEEE Journal of Biomedical and Health Informatics*, 21(1):22–30, 2017. doi: 10.1109/JBHI.2016.2633963.

Chao Pang, Xinzhuo Jiang, Krishna S Kalluri, Matthew Spotnitz, RuiJun Chen, Adler Perotte, and Karthik Natarajan. Cehr-bert: Incorporating temporal information from structured ehr data to improve prediction tasks. In *Machine Learning for Health*, pp. 239–260. PMLR, 2021.

Raphael Poulain, Mehak Gupta, and Rahmatollah Beheshti. Few-shot learning with semi-supervised transformers for electronic health records. 2022.

Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):1–13, 2021.

Safiya Richardson, Jamie S. Hirsch, Mangala Narasimhan, James M. Crawford, Thomas McGinn, Karina W. Davidson, , and the Northwell COVID-19 Research Consortium. Presenting Characteristics, Comorbidities, and Outcomes Among 5700 Patients Hospitalized With COVID-19 in the New York City Area. *JAMA*, 323(20):2052–2059, 05 2020. ISSN 0098-7484. doi: 10.1001/jama.2020.6775. URL https://doi.org/10.1001/jama.2020.6775.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Yuval Shahar. A framework for knowledge-based temporal abstraction. *Artificial intelligence*, 90 (1):79–133, 1997.

Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 5953–5959. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/825. URL https://doi.org/10.24963/ijcai.2019/825.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. 2014.

Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315(8):801–810, 2016.

Judith Tintinalli, Stapczynski J, John Ma O, Cline D, Cydulka R, and Meckler G. *Tintinallis emergency medicine A comprehensive study guide*, chapter 146: Septic Shock, pp. 1003–1014. McGraw-Hill Education, 7 edition, 2011.

Marcello Tonelli, Natasha Wiebe, Braden J. Manns, Scott W. Klarenbach, Matthew T. James, Pietro Ravani, Neesh Pannu, Jonathan Himmelfarb, and Brenda R. Hemmelgarn. Comparison of the Complexity of Patients Seen by Different Medical Subspecialists in a Universal Health Care System. *JAMA Network Open*, 1(7):e184852–e184852, 11 2018. ISSN 2574-3805. doi: 10.1001/jamanetworkopen.2018.4852. URL https://doi.org/10.1001/jamanetworkopen.2018.4852.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 85–94, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623754. URL https://doi.org/10.1145/2623330.2623754.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Xi Yang, Yuan Zhang, and Min Chi. Multi-series time-aware sequence partitioning for disease progression modeling. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint*

Conference on Artificial Intelligence, IJCAI-21, pp. 3581–3587. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/493. URL `https://doi.org/10.24963/ijcai.2021/493`. Main Track.

Chenyu Yi, SIYUAN YANG, Haoliang Li, Yap peng Tan, and Alex Kot. Benchmarking the robustness of spatial-temporal models against corruptions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=MQlMIrm3Hv5`.

Kezi Yu, Yunlong Wang, Yong Cai, Cao Xiao, Emily Zhao, Lucas Glass, and Jimeng Sun. Rare disease detection by sequence modeling with generative adversarial networks. *arXiv preprint arXiv:1907.01022*, 2019.

Yuan Zhang, Xi Yang, Julie Ivy, and Min Chi. Attain: attention-based time-aware lstm networks for disease progression modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI*, pp. 10–16, 2019.

Jiayu Zhou, Lei Yuan, Jun Liu, and Jieping Ye. A multi-task learning formulation for predicting disease progression. In *SIGKDD*, pp. 814–822. ACM, 2011.

Jiayu Zhou, Jimeng Sun, Yashu Liu, Jianying Hu, and Jieping Ye. Patient risk prediction model via top-k stability selection. In *SDM*, pp. 55–63. SIAM, 2013.

A Appendix

A.1 Proposed Method: TRIDE

A.1.1 Data Augmentation

Figure 6 illustrates the detailed training procedure of TRIDE. First, *the outer loop* follows the regular training process, gradient descent, of any supervised learning models. Until we hit the max iteration, we select a random data point, calculate the loss, and update the weight of a classifier based on the loss. After the max iteration, the training process ends, and we test the model with a test set. The new component added to the regular training process is located in *the inner loop*, where we generate sufficiently different training samples. Here we use a predefined sampling probability p to select data points to transform. Our goal of data transformation here is to optimize the transformation level (θ) with respect to a classifier (loss) and dataset for building a robust model. Once optimization finishes, transformed data sample using the optimal theta is added to the outer loop and used to update the model weights. By actively incorporating data augmentation component into a regular training process, and providing challenging data samples to the classifier, TRIDE can help build a robust classifier or prediction model.

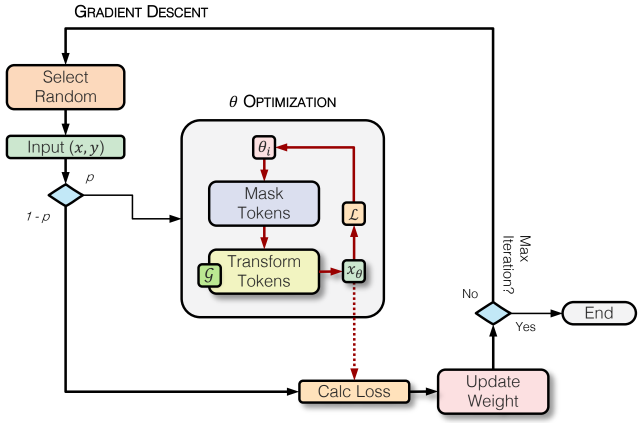


Figure 6: Flowchart of TRIDE

A.1.2 Pretraining EHR Language Models

Figure 7 illustrates the pretraining process of our timeBERT. We use the sequences of symptoms that are extracted from original real-valued time-series data using temporal abstraction as an input to BERT-based models. Input to the BERT-based model is composed of three embeddings similar to the original BERT – *token* embeddings, *segment* embeddings, and *position* embeddings, and all three embeddings utilize the sequences of symptoms resulted from temporal abstraction. Token embeddings are trained on individual tokens recognized from a tokenizer; segment embeddings are learned on indicators of events (e.g., a number that represents whether this sequence is originated from an event at $t1$ or $t2$) and play a role of differentiating sequences from different events; and position embeddings are trained on the order of tokens, where the order is determined by each token's start and end time.

In this work, "sequence" is defined as a list of variable number of tokens where each token represents a single symptom, namely a measurement of each feature (e.g., temperature measured as normal), and the sequence length is determined by the number of symptoms collected in one event (or time stamp). Depending on tasks, our model uses either a single event-size sequence or multiple sequences with a visit-size. In every case, we place a special token ([CLS]) at the first place of a sequence and another special token ([SEP]) at the end, similar to BERT. In case of preparing visit-size sequences, each sequence of tokens from an event is concatenated with another sequences from its neighboring event using a special token ([SEP]). In addition, motivated by previous works Nguyen et al. (2017); Pang et al. (2021) and to fully incorporate temporal aspects of our sequences, we introduce an additional special token ([t-HR]) that represents time intervals between consecutive events. The tokens are in brown color in Figure 7. The features (e.g., vital signs) in EHRs are measured irregularly based on their needs and clinicians make such decisions to effectively diagnose patients condition and keep track of disease trajectory. That being said, by incorporating time intervals into model training, learned embeddings could embrace useful information regarding clinicians' practice of measuring patients conditions. We expect that this would play a important role in understanding temporal relations between tokens (symptoms) within a visit and benefit the self-attention process inside the BERT framework.

Different from the original BERT, in which WordPiece tokenizer is utilized for recognizing tokens from a sentence, our model split a sentence with white spaces to identify individual tokens similar to the classic word2vec Mikolov et al. (2013) approach. The rationale behind this decision is: 1) we assume that feature and its value together should be considered as one token or word to fully

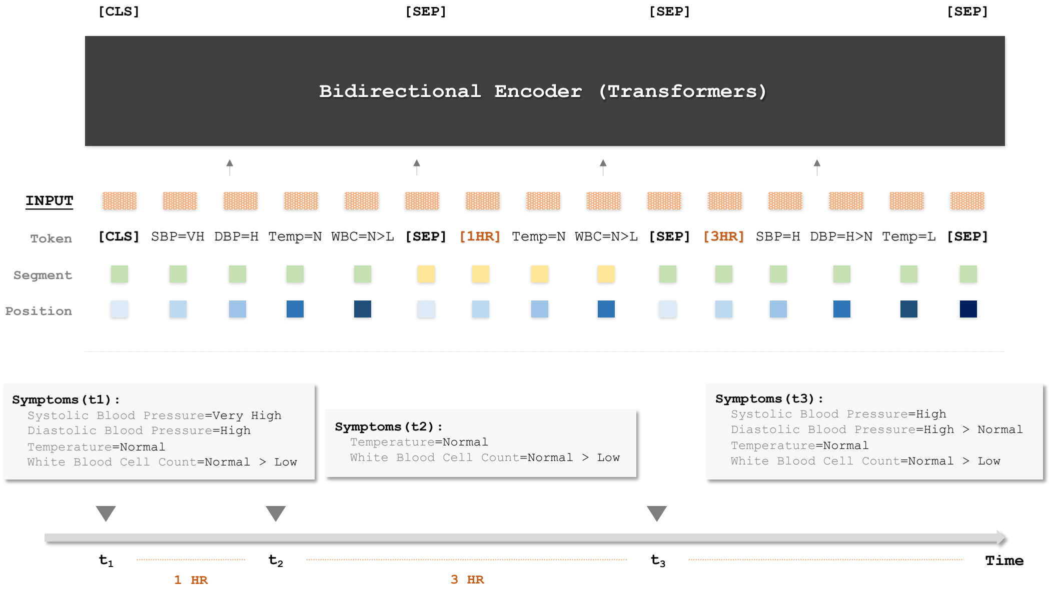

[CLS]
[SEP]
[SEP]
[SEP]
Bidirectional Encoder (Transformers)
INPUT
Token
[CLS] SBP=VH DBP=H Temp=N WBC=N>L [SEP] [1HR] Temp=N WBC=N>L [SEP] [3HR] SBP=H DBP=H>N Temp=L [SEP]
Segment
Position
Symptoms (t1):
Systolic Blood Pressure=Very High
Diastolic Blood Pressure=High
Temperature=Normal
White Blood Cell Count=Normal > Low
Symptoms (t2):
Temperature=Normal
White Blood Cell Count=Normal > Low
Symptoms (t3):
Systolic Blood Pressure=High
Diastolic Blood Pressure=High > Normal
Temperature=Normal
White Blood Cell Count=Normal > Low
t1
t2
t3
Time
1 HR
3 HR

Figure 7: BERT for Multivariate Time-Series EHRs

represent a patient's symptom and form a semantic unit; and 2) we understand the objective of using WordPiece tokenizer is not suitable to our setting as, in our setting, there exists a fixed number of vocabulary generated from a rule-based abstraction process, whereas the main purpose of using sub-word-based tokenizers such as WordPiece or byte pair encoding (BPE) is to better handle open vocabulary problems and to reduce sparsity (i.e., the number of vocabulary) in learning natural language Sennrich et al. (2015).

A.2 Experiments

A.2.1 Task Description: Early Prediction

Our task is to predict whether a patient will develop septic shock m hours later when provided with the last n hours of patient's records. In this work, m ranges from 6 to 48 hours as requested by our clinicians as described in the result section. For this setting, the shock visits are right-aligned by their first onset of septic shock and the non-shock visits by a truncated time point. In addition, the non-shock visits are trimmed so that they have the same distribution of length as shock visits and become balanced. Lastly, due to the rapid progression of sepsis, this work only focuses on the five days of patients' records.

A.2.2 Generating Time Tokens for Irregular Time Intervals

EHRs are measured irregularly, thus resulting in irregular time intervals between two consecutive events. To abstract and standardize such irregular numbers and to utilize them into embedding training, we categorize the time duration of intervals into 25 buckets based on the frequency of intervals. Specifically, the buckets starts with 1 hour and increment with 1 hour interval until 24 hours of time interval, then time intervals with more than 24 hours are grouped into one bucket (¿ 24 hours). Figure 8 shows the simplified distribution of bucketized time intervals. We can observe that the time intervals from CCHS are more uniformly distributed across different duration while the time intervals from Mayo are mostly centered around 1 hour interval. We can infer that, from this distribution, Mayo data are more regularly and frequently measured compared to CCHS data.

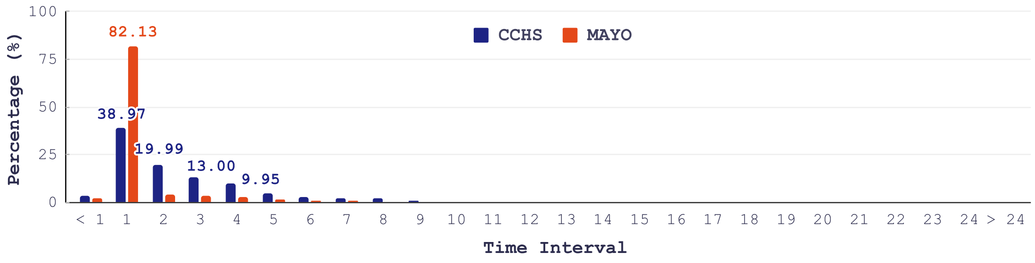


Figure 8: Distribution of Time Intervals

A.2.3 Data Split

We construct training, validation, and test sets for each prediction hour to include data samples (i.e., patients) from previous hours of prediction tasks. This division procedure enables us to more effectively monitor model performance when the number of patients increases or decreases. The numbers in Table 3 represent the number of patient visits included in each set, and the percentages indicate the proportion of current set size compared to corresponding total training, validation, test sets.

A.2.4 Pretraining Data for EHR Language Models

From our target population, we randomly sampled 10,000 patient visits with 927,131 events that are not used for model evaluation, for pretraining EHR language models. Specifically, 5,000 visits each are sampled from the two classes determined by the expert rule. For shock visits, all events up to the

Table 3: Data Split Summary Statistics

	CCHS				**MAYO**			
Hour	**Train**	**Validation**	**Test**	**Test (Robust)**	**Train**	**Validation**	**Test**	**Test (Robust)**
0	3,775	813	812	217	4,362	937	937	263
6	2,839	612	611	217	3,166	681	680	263
12	1,873	404	404	217	2,284	492	491	263
18	1,495	323	322	217	1,864	401	401	263
24	1,310	283	282	217	1,632	351	351	263
30	1,201	259	259	217	1,447	311	311	263
36	1,117	241	240	217	1,338	288	287	263
42	1,046	225	225	217	1,267	272	272	263
48	1,009	216	217‡	217	1,223	262	263‡	263

Notes: (‡) marks indicate the test sets used for the robust setting. In the robust setting, the marked test set will be used solely throughout entire early prediction tasks (6-48 hours).

Table 4: Pretraining Data Summary Statistics

Domain	**Model**	**# Vocab**	**# Visits**	**# Events**	**Events(Stat)**	**# Tokens**	**Tokens(Stat)**
CCHS	BERT	308	10,000	276,359	27.6(±50.5)	1,574,369	157.4(±266.3)
	timeBERT	333	10,000	276,359	27.6(±50.5)	1,840,728	184.1(±316.3)
MAYO	BERT	337	10,000	492,783	49.3(±105.2)	2,682,264	268.2(±532.5)
	timeBERT	362	10,000	492,783	49.3(±105.2)	3,165,047	316.5(±634.7)

Notes: Events(Stat) and Tokens(Stat) columns show the average values and corresponding standard deviations at visit level.

first onset of septic shock are selected while the events in non-shock visits are cut to have the same distribution of length as the shock visits.

Each sequence is composed of discretized symptoms (i.e., tokens) measured within a 60-minute interval. To train contextualized EHR representations based on BERT architecture, the event-level sequences are concatenated with a special token ([SEP]) and all tokens in a visit are used at the same time during training. As BERT restricts the number of tokens that can be included in a sentence (i.e., a chunk of tokens) to 512, in case of the number of tokens in our visit-long sequence exceeds the maximum, the sequence will be divided into several sequences with 512 symptom tokens when pretraining our BERT-based representations. Table 4 provides the summary statistics of data used for pretraining EHR representations. Two variations of representation learning model, BERT and timeBERT, utilize the same number of visits for pretraining, but have different number of vocabulary as timeBERT additionally incorporates special tokens that represents time intervals.

A.2.5 Hyperparameters and Implementation Details

For LSTM, a mini-batch Adam optimizer with a batch size of 16 and 72 hidden units are employed. The training epochs are determined by the validation loss via early stopping with patience of 3. For MuLan, a dimension size of 100 is used for output representations. To pretrain EHR language models from unlabeled EHRs, we employ the structure of a 'bert-based-uncased' model imported from Hugging Face's transformer package[1] Wolf et al. (2020). For pretraining BERT from scratch, a training epoch was set to 60 for CCHS and 50 for MAYO with batch size of 4 and model block size (i.e., maximum number of tokens to consider) of 512. We monitored training loss and stopped training when converged. For pretraining timeBERT, a training epoch was set to 100 for both datasets with the same size of batch and model block. For both BERT-based models, an Adam optimizer with weight decay rate (0.01) and learning rate (2e-05) was used to pretrain the models, and we mask 15% of tokens for this task. When pretrained BERT is used for constrained worst-case token masking and prediction, we use *top-k sampling* Holtzman et al. (2020) with the k value as 10 to predict masked symptoms given context. Further, to obtain optimal transformation level θ^* from the

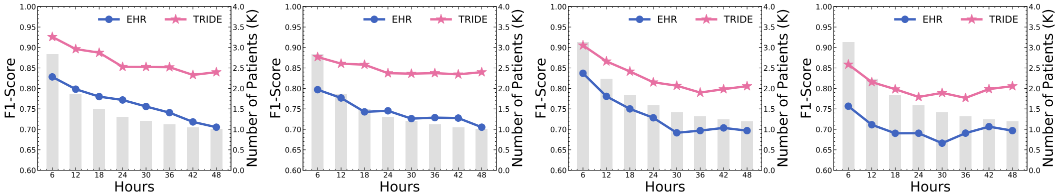
[1] https://huggingface.co/bert-base-uncased

constrained worst-case transformation, we utilize Scipy's *optimize* function Virtanen et al. (2020). The initial theta, θ_0 is set to 0.0 only during the first optimization and replaced with previously estimated theta $\hat{\theta}_{i-1}$ to expedite the process. The configuration of TRIDE hyperparameters is described in the main text.

A.3 Result

A.3.1 6-48 Hours Early Prediction

Figure 9 shows the overall performance (F1-score) for 6-48 hours early prediction on two different datasets and test set variations. As described in the main text, TRIDE significantly outperforms the baseline, thus demonstrating temporal robustness.



(a) CCHS-Regular (b) CCHS-Robust (c) MAYO-Regular (d) MAYO-Robust

Figure 9: Model Performance (F1) on 6-48 Hours Early Prediction