EHR data analysis using machine learning models

Cao Cong Luan Tran

Georgia State University

August 1, 2024

Abstract

Researchers have been using electronic health record (EHR) data and machine learning (ML) techniques to build models for predicting various adverse outcomes in healthcare. However, preprocessing EHR data for ML tasks is complex, due to its messy nature, high dimensionality, irregular sampling, and missing values. It also requires more substantial effort and can be labor-intensive. Systematic and repeatable preprocessing methods for EHR data are becoming more and more important as machine learning (ML) becomes more and more integrated into healthcare. Nevertheless, the hierarchical structure of electronic health records (EHR) data is not well captured by the pretrained models now in use in the medical sector, which limits the generalization power of these models when applied to a variety of downstream tasks. Therefore, we have referenced some open-source frameworks created especially for hierarchically multimodal EHR data that simplifies the preparation of data collected from the EHR. Experiments conducted on eight downstream tasks at three different levels show how successful the framework is. The framework methodically converts organized EHR data into feature vectors, reducing the amount of choices a user has to make while implementing recommended practices. We carried out an experiment on the publicly accessible EHR data sets gathered from critical care units which is MIMIC-III. This allowed us to show the tool's usefulness and adaptability. Three clinically significant outcomes—shock, acute respiratory failure, and in-hospital mortality—were trained into separate machine learning models. The area under the receiver operating characteristics curve (AUROC) was used to evaluate the models. These frameworks are generalizable across prediction times, ML algorithms, and data sets, and robust to user-defined arguments settings, promoting clinically useful ML tools. Its effectiveness is demonstrated through multimodal evaluation and its applicability in low-resource clinical settings. However, they have limitations, such as computational inefficiency.

Contents

1 Introduction

Pre-trained models in machine learning improve performance and reduce training need, saving computational resources. However, they can be complex, inconsistent, and benefit from regularization and model strength. Additionally, evaluating uncertainty becomes more difficult with increased complexity and nonlinearity, making continuous learning more challenging.[1].

In order to create patient risk stratification models for a variety of adverse outcomes, such as infections linked to healthcare, sepsis and septic shock, acute respiratory distress syndrome, acute kidney injury, and others, researchers have so far been successful in utilizing data from electronic health records (EHRs) and machine learning (ML) tools. Even though these works make use of machine learning techniques, a significant amount of work needs to be done in preprocessing before ML is applied. EHR data are unstructured and frequently comprise high-dimensional, irregularly sampled time series with several different data types and missing values. Many decisions must be made in order to transform EHR data into feature vectors that are appropriate for machine learning algorithms. These decisions include how to manage missing data, what input variables to include, and how to resample longitudinal data.

The paper focuses on enhancing machine learning models for clinical prediction tasks by leveraging the hierarchical structure inherent in electronic health records (EHRs). The authors propose a novel pretraining strategy that incorporates hierarchical relationships within EHR data, leading to improved model performance and interpretability.

EHR data is rich with hierarchical information, such as patient visits, medical events, and clinical notes. Traditional pretraining methods often ignore these hierarchies, potentially missing critical relationships within the data. The motivation behind this research is to exploit these hierarchical structures to better capture the complexity and interdependencies within EHR data, thus enhancing the performance of downstream clinical prediction models.

This reference can speed up ML research using EHR data, even if it is not a one-size-fits-all preprocessing solution and further work is required to evaluate the boundaries of its generalizability. Framework streamlines the process by cutting down on the time and effort required for labor-intensive data pretreatment stages. Additionally, it gives researchers a rapid and realistic starting point by offering a baseline that is simple to share and replicate.

```
Data columns (total 4 columns):
 #   Column          Dtype
---  ------          -----
 0   ID              int64
 1   t               float64
 2   variable_name   object
 3   variable_value  object
dtypes: float64(1), int64(1), object(2)
```
[1]

Figure 1: This is a data analysis tool that uses tabular data with four columns: ID, t, variable_name, and variable_value. ID is a unique identifier for each example, and t is the time of recording. The pipeline assumes a time-invariant value recorded once when t is null. Each variable_name uniquely encodes the name of a variable, and the variable_value column may contain numbers or strings. Each variable_name can be automatically classified as numerical or categorical based on the variable_value type. Users can override the type of a variable_name to be numerical, categorical, or hierarchical. It does not assume the completeness of the data, as not every ID will have a value associated with every variable_name.

2 Methods

[1] We implement 3 processing steps: pre-filter, transform and post-filter.

2.1 Processing steps

2.1.1 Pre-filter

Rows with timestamps outside observation period are eliminated, and rarely occurring variables are removed, speeding up downstream analyses but potentially causing loss of useful information.

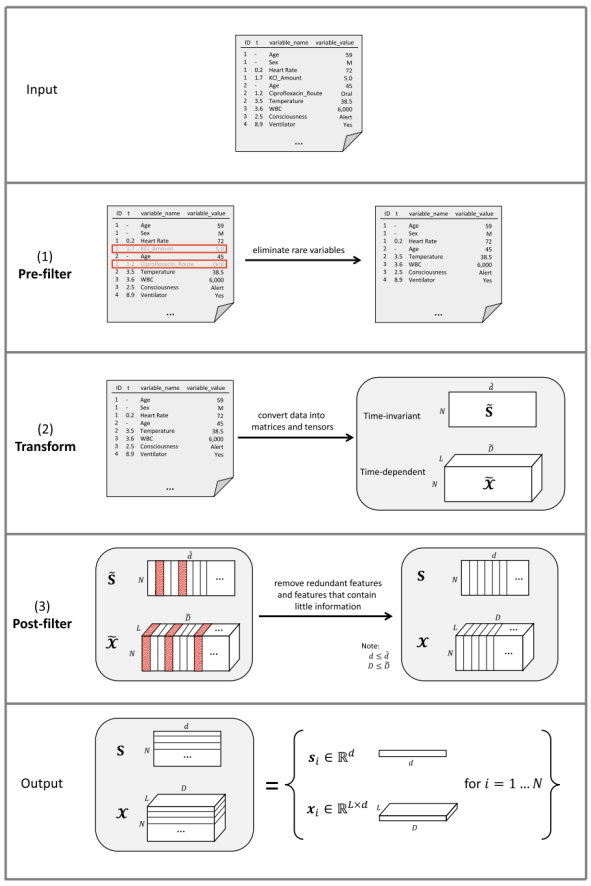
2.1.2 Transform

Depending on the types of timestamps, processing continues. Data corresponding to a variable name are processed as "time-invariant" if the timestamp t for that variable name is null; otherwise, the data are processed as "time-dependent" in order to capture longitudinal trends and dynamics.

2.1.3 Post-filter

The data is transformed by removing features equal to 1 or 0 in 100% of examples, combining duplicated features into a single feature, and converting them into a matrix $S \in R^{N*d}$ and a tensor $X \in R^{N*L*D}$. This data representation is used as input for machine learning algorithms. It can transform format-

ted EHR data into feature vectors, providing examples for both time-invariant and time-dependent data.



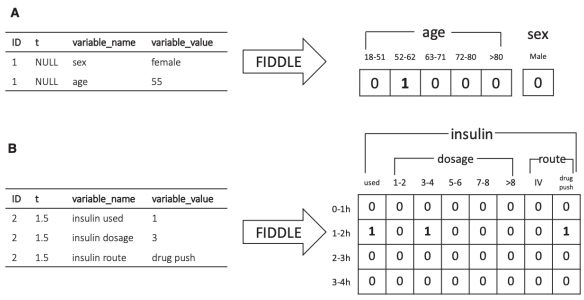
[1]

Figure 2: Overview: Pre-filtering, transforming, and post-filtering are the three steps of data processing. You can record timestamps at any level of detail (e.g., seconds, minutes, hours, days, visits, etc.) in the t column. We take into account time in hours in this sample input file. A patient with ID ¼ 1 and a heart rate ¼ 72 bpm measured at t ¼ 0.2 h is represented by a row with [1, 0.2, Heart Rate, 72]. It reduces unusual variables in the (1) pre-filter. In the second transform, time-invariant and time-dependent features are contained in tensors created by it from the data. It eliminates unnecessary and sometimes uninformative characteristics in the (3) post-filter. Binary vectors si and xi, which represent the features for each ID, make up the result. bpm: beats per minute; it: Flexible Data-Driven Pipeline: ID: unique identifier; KCl: potassium chloride; WBC: white blood cell.

2.2 Experiments and Applying to MIMIC-III

Five prediction tasks were established, each with a unique study group. We omitted infants and young children (under 18) from all analyses due to the differences in their physiology and risk variables from those of adults. Since it is challenging to accurately identify patients with treatment limits (e.g., those who may be placed on comfort measures) across data sets, we did not attempt to exclude them. On the other hand, this lets us contrast with earlier research. In the end, it might simplify the prediction tasks and reduce the trained models' clinical value. We utilized T ¼ 48 hours for in-hospital mortality in order to forecast whether the result will occur after T in accordance with previous research.11 We utilized both T ¼ 4 hours and T ¼ 12 hours for ARF and shock.

We directly fed the it-generated features into four different classification techniques, modifying them based on the kind of model: penalized logistic regression (LR), random forest (RF), 1-dimensional convolutional neural networks (CNN), and long short-term memory networks (LSTM). We created a feature vector with the shape R^{LD+d} by flattening the time-dependent features (x_i) and concatenating them with the time-invariant features (s_i) for the models that anticipate flat input (LR and RF). We



[1]

Figure 3: Examples of it input and output for time-invariant and time-dependent data

Table 1. Challenges in preprocessing EHR data and FIDDLE's solution

Challenges	Example	Solutions in FIDDLE
Some data have associated timestamps, while others do not	• Sex is recorded once at the time of admission and typically does not have a timestamp; • Administration of medications is time-stamped.	Handle time-invariant and time-dependent data separately.[2,16,17]

Data have heterogeneous types
• Categorical
• Numerical
• Hierarchical
	• Drug route is categorical: oral, IV
	• Heart rate is numerical: 70 bpm
	• ICD-9 code is hierarchical
		Different representations for each value type
		• Categorical: one-hot encoding[18]
		• Numerical: 3 options[19,20]
		  • Kept as continuous;
		  • Binned into quintiles and one-hot encoding; or
		  • Binned into quintiles and ordinal encoding.
		• Hierarchical: user specifies which level(s) of the hierarchy to encode; values are converted internally to categorical values.[21,22]

Data are sparse and irregularly sampled, and different variables can have different frequencies of recording	• Vital signs, such as temperature or heart rate, may be measured multiple times per day at different intervals; and • Laboratory tests are run infrequently (eg, once/twice every day).	• Irregular sampling: resample data into time bins, defined by the user input (dt, temporal granularity);[23] and • Different recording frequencies: handle "frequent" and "non-frequent" variables differently (determined by a user-defined threshold θ_{freq}), capturing richer information for "frequent" variables (see below).

After resampling the data according to some temporal granularity (dt) we might have:
• Multiple recordings within a time bin; and
• Not every time bin will have a recording (missing values)
	• Multiple (potentially different) heart rate values within an hour; and
	• Temperature measurements were interrupted when a patient is transferred between ICU wards.
		• Multiple recordings per time bin: use the most recent recording.
		  • Calculate summary statistics for "frequent" variables.[1]
		• Missing values:
		  • Imputation with carry-forward;[24–26] and
		  • Keep track of "presence mask" and "delta time" (how long the value has been imputed).[27,28]
		• Feature selection, filter out potentially uninformative features;[29–33] and
		• Combine duplicate features into a single feature, renaming the features where appropriate.[34]

High-dimensional feature space • Some features are rarely recorded or nearly constant; and • Some features are correlated or duplicated.	Data extracted from the EHR typically contain hundreds, if not thousands, of variables, including medications, labs, CPT codes, etc.	

Note: bpm: beats per minute; CPT: current procedure terminology; EHR: electronic health record; FIDDLE: Flexible Data-Driven Pipeline; ICD-9: International Classification of Diseases, Ninth Edition; ICU: intensive care unit; IV: intravenous.

[1]

Figure 4: Challenges in preprocessing EHR data and solution

replicated the time-invariant features (s_i) at each time-step of xi for the models (LSTM and CNN) that anticipate sequential input. This produced a feature matrix of the shape $R^{L*(d+D)}$.

We divided each study cohort into train and test sets (including ICU stays) based on the random assignment of each patient to the train or test partition. Using the training/validation data and a random search40 with a budget of 50, hyperparameters were chosen so as to maximize the average area under the receiver operating characteristics curve (AUROC).

MIMIC-III

Table name	Description	Example variables
PATIENTS	Information on unique patients	Age, Sex
ADMISSIONS	Information on unique hospitalizations	Admission type Admission location
ICUSTAYS	Information on unique ICU stays	Care unit Ward ID Admission-to-ICU time
CHARTEVENTS	Charted data, including vital signs, and other information relevant to patients' care	Heart rate Pain location Daily weight
LABEVENTS	Laboratory test results from the hospital database	Lactate WBC
INPUTEVENTS_MV	Fluid intake administered, including dosage and route (eg, oral or intravenous)	NaCl 0.45% Whole blood
OUTPUTEVENTS	Fluid output during the ICU stay	OR urine Stool
PROCEDUREEVENT_MV	Patients' procedures during the ICU stay	CT scan X-ray
MICROBIOLOGYEVENTS	Microbiology specimen from hospital database	Sputum
DATETIMEEVENTS	Documentation of dates and times of certain events	Last dialysis Pregnancy due

Note: We used all structured tables that pertain to patient health.
CT: computed tomography; ICU: intensive care unit; ID: unique identifier; OR: operating room; WBC: white blood cell.

[1]

Figure 5: Summary of MIMIC-III tables used in our analysis

2.3 Hierarchical Structure

[2] Medical images are the primary input for the majority of multimodal pretraining models in the medical domain, with other modalities like text and tabular data. Pretraining on multimodal EHR data without the use of medical images is the subject of very few studies. The drawback of the pretrained models is that they are unable to handle a variety of downstream tasks at different levels because all of the current pretrained work on EHR data adheres to the standard NLP pretraining procedure while ignoring the hierarchical nature of EHRs.

The "bottom-to-up" methodology is used and provide level-specific activities for self-supervised learning. We suggest rebuilding the numerical time-ordered clinical features at the stay level. Two pretraining approaches are developed for the entrance level. In the first, a collection of masked medication and ICD codes are predicted in order to represent intra-modality relations. In the second, modality-level contrastive learning is used to model inter-modality relations. We use a two-stage training technique from stay to admission levels to train the entire model.

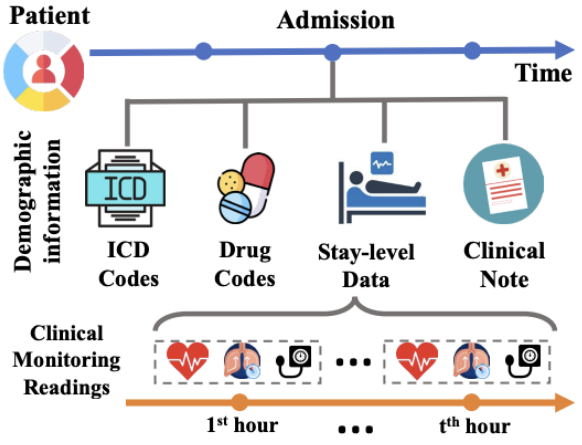
Using information extracted from the MIMIC-III dataset, we forecast the patient's likelihood of experiencing acute respiratory failure (ARF), shock,

Task		In-hospital mortality, 48 h $n=1264$		ARF, 4 h $n=2338$		ARF, 12 h $n=2093$		Shock, 4 h $n=2867$		Shock, 12 h $n=2612$	
Method		AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
MIMIC-Extract	*LR*	**0.859** (0.830–0.887)	0.445 (0.358–0.540)	0.777 (0.752–0.803)	0.604 (0.561–0.648)	0.723 (0.683–0.759)	0.250 (0.200–0.313)	0.796 (0.771–0.821)	0.505 (0.454–0.557)	0.748 (0.712–0.784)	0.242 (0.193–0.310)
	RF	0.852 (0.821–0.882)	**0.448** (0.359–0.537)	**0.821** (0.799–0.843)	**0.660** (0.617–0.698)	**0.747** (0.709–0.782)	**0.289** (0.235–0.356)	**0.824** (0.801–0.845)	**0.541** (0.488–0.588)	**0.778** (0.742–0.812)	**0.307** (0.248–0.369)
	CNN	0.851 (0.820–0.879)	0.439 (0.353–0.529)	0.788 (0.763–0.814)	0.633 (0.591–0.672)	0.722 (0.684–0.758)	0.258 (0.207–0.320)	0.798 (0.773–0.824)	0.520 (0.471–0.572)	0.741 (0.704–0.778)	0.247 (0.198–0.317)
	LSTM	0.837 (0.803–0.867)	0.441 (0.358–0.523)	0.796 (0.770–0.822)	0.634 (0.590–0.675)	0.700 (0.661–0.736)	0.229 (0.184–0.286)	0.801 (0.778–0.825)	0.513 (0.463–0.562)	0.753 (0.717–0.791)	0.248 (0.199–0.313)
FIDDLE	*LR*	0.856 (0.821–0.888)	0.444 (0.357–0.545)	0.817 (0.792–0.839)	0.657 (0.614–0.696)	0.757 (0.720–0.789)	0.291 (0.236–0.354)	0.825 (0.803–0.846)	**0.548** (0.501–0.595)	**0.792** (0.758–0.824)	0.274 (0.227–0.338)
	RF	0.814 (0.780–0.847)	0.357 (0.279–0.448)	0.817 (0.795–0.839)	0.652 (0.608–0.690)	0.760 (0.726–0.793)	0.317 (0.255–0.382)	0.809(0.786–0.833)	0.516 (0.467–0.566)	0.773 (0.740–0.806)	0.288 (0.231–0.355)
	CNN	**0.886** (0.854–0.916)	**0.531** (0.434–0.629)	**0.827** (0.803–0.848)	**0.666** (0.626–0.705)	0.768 (0.733–0.800)	0.294 (0.238–0.361)	**0.831** (0.811–0.851)	0.541 (0.493–0.589)	0.791 (0.758–0.823)	0.295 (0.239–0.361)
	LSTM	0.868 (0.835–0.897)	0.510 (0.411–0.597)	**0.827** (0.801–0.846)	0.664 (0.623–0.703)	**0.771** (0.737–0.802)	**0.326** (0.267–0.397)	0.824 (0.803–0.845)	0.541 (0.497–0.587)	**0.792** (0.759–0.823)	**0.314** (0.251–0.386)

Note: Reported as AUROC and AUPR with 95% CIs in parentheses on the respective held-out test set for the 5 prediction tasks. For each task (column), the bolded results are the best-performing model for either MIMIC-Extract or FIDDLE.
ARF: acute respiratory failure; AUROC: area under the receiver operating characteristics curve; AUPR: area under the precision-recall curve; CI: confidence interval; CNN: convolutional neural networks; FIDDLE: Flexible Data-Driven Pipeline; LR: logistic regression; LSTM: long short-term memory networks; RF: random forest.

[1]

Figure 6: Summary of performance on MIMIC-III for all it-based models, compared to MIMIC-Extract.



[2]

Figure 7: An illustration of EHR hierarchy..

or fatality within 48 hours for the stay-level evaluation. We use the same procedure to extract data from the MIMIC-III dataset in order to estimate the 30-day readmission rate for the admission-level evaluation. We extract the heart failure data from MIMIC-III and use it to perform four health risk prediction tasks for the patient-level evaluation.

Multiple time_ordered hospital admissions make up each patients data, as shown by P = $[A_1, A_2, \ldots, A_N]$, where A_i $(i \in [1, n])$ represents the i_th admission and N is the total number of admissions. Be aware that N may vary depending on the patient. A collection of D designated demographic characteristics is also present in each patient. A_i = S_i, C_i, L_i, G_i represents the components of each admission, which include numerous time_ordered stay_level data designated as S_i, a set of ICD codes indicated as C_i, a section of clinical notes marked as L_i, and a set of medication codes. The data at the stay_level S_i is made up of a series of hourly recorded monitoring stays, such as S_i = $[S_i^1, S_i^2, , S_i^M{}_i]$.

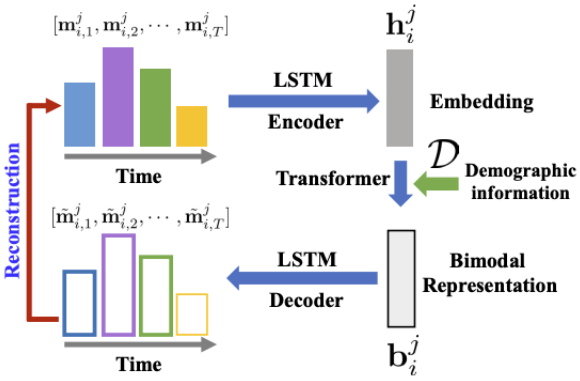


Figure 8: Stay-level self-supervised pretraining.

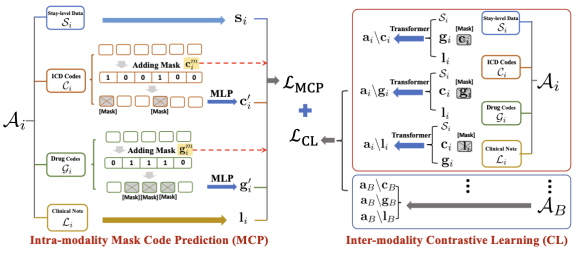


Figure 9: Admission-level self-supervised pretraining.

3 Results

Conduct experiments comparing models pretrained with hierarchical structures and those using standard pretraining methods. Apply the pipeline to multiple datasets and evaluate its performance in generating feature vectors. Present the combined impact of hierarchical pretraining and preprocessing on clinical prediction tasks. The cohorts for the MIMIC-III trial ranged in size from 8,577 to 19,342 cases. Up to 320 million rows were present in the prepared input tables. It took between 30 and 150 minutes to extract feature vectors for each cohort of MIMIC-III, depending on the quantity of the input data and the argument values. On the other hand, MIMIC-Extract required 8 hours in total (including database operations, etc.), while only 2 hours were needed for the feature processing step.

4 Discussion

This study preferences an open-source tool designed to streamline the preprocessing of EHR data for machine learning (ML) applications. The tool has shown promising results in predictive performance across various outcomes, prediction times, and classification algorithms, with AUROCs comparable to those of MIMICExtract applied to MIMIC-III, allowing users to tailor the pipeline to their cohort/task and facilitating reproducibility and sharing of preprocessing code.

The framework addresses many limitations in existing work by considering nearly all available structured data in MIMIC-III, producing features that capture a rich representation of a patient's physiological state and longitudinal history. It also allows ML models to leverage potentially high-dimensional patterns in the data.

[2] However, it has limitations, such as processing all numerical variables identically, considering only the structured contents in the EHR, and not harmonizing data across institutions. Carefully reviewing the learned model and validating it in ways that mimic the clinical use case remains necessary.

Despite these limitations, it can help speed up ML analyses but does not eliminate the critical need for model checking. In experiments, it generated high-dimensional feature vectors, which can leverage the entire structured contents of the EHR, po-

[2] tentially taking advantage of variables specific to a particular hospital or unintended short-cuts in the data. Researchers may consider tuning the filtering threshold in the framework to be more aggressive or applying downstream feature selection approaches to address this trade-off between improvements in performance and the technical debt associated with including more variables in deployed ML models.

5 Conclusions

As a whole, the framework can assist ML researchers in preprocessing data that has been taken out of the EHR. It can encourage development of more quickly and uniformly performed preprocessing stages, which will aid in the creation of ML tools that will be therapeutically beneficial.

References

[1] Shengpu Tang, Parmida Davarmanesh, Yanmeng Song, Danai Koutra, Michael W Sjoding, and Jenna Wiens. Democratizing EHR analyses with FIDDLE: a flexible data-driven preprocessing pipeline for structured clinical data. *Journal of the American Medical Informatics Association*, 27(12):1921–1934, 10 2020. ISSN 1527-974X. doi: 10.1093/jamia/ocaa139. URL https://doi.org/10.1093/jamia/ocaa139.

[2] Xiaochen Wang, Junyu Luo, Jiaqi Wang, Ziyi Yin, Suhan Cui, Yuan Zhong, Yaqing Wang, and Fenglong Ma. Hierarchical pretraining on multimodal electronic health records. pages 2839–2852, December 2023. doi: 10.18653/v1/2023.emnlp-main.171. URL https://aclanthology.org/2023.emnlp-main.171.

A Appendix: derivation