explICU: A Web-based Visualization and Predictive Modeling Toolkit for Mortality in Intensive Care Patients

Robert Chen, Member, IEEE, Vikas Kumar, Natalie Fitch, Jitesh Jagadish, Lifan Zhang, William Dunn, and Duenn Horng Chau, Member, IEEE

Abstract— Preventing mortality in intensive care units (ICUs) has been a top priority in American hospitals. Predictive modeling has been shown to be effective in prediction of mortality based upon data from patients’ past medical histories from electronic health records (EHRs). Furthermore, visualization of timeline events is imperative in the ICU setting in order to quickly identify trends in patient histories that may lead to mortality. With the increasing adoption of EHRs, a wealth of medical data is becoming increasingly available for secondary uses such as data exploration and predictive modeling. While data exploration and predictive modeling are useful for finding risk factors in ICU patients, the process is time consuming and requires a high level of computer programming ability. We propose explICU, a web service that hosts EHR data, displays timelines of patient events based upon user-specified preferences, performs predictive modeling in the back end, and displays results to the user via intuitive, interactive visualizations.

I. INTRODUCTION

There are four million visits per year to intensive care units (ICUs) [1]. The average ICU mortality rate is 8-19%, or 500,000 deaths annually. Prevention of mortality has garnered widespread interest. Due to the rise of electronic health record (EHR) systems, visualization of data and predictive modeling for mortality have become challenging tasks. The predictive modeling process is often lengthy and cumbersome, involving several steps including data formatting, model construction and model evaluation. Researchers usually construct and compare many models with their data, using a variety of classifiers.

While predictive modeling can help identify risk factors for mortality, researchers often desire to visualize data on the patient level. Visualization of time series processes is of particular interest in the ICU setting, since sudden acute events such as heart failure or renal failure can often lead to severe and sometimes deadly conditions. However, due to the increased volume and dimensionality of data with modern EHRs, it is often difficult to explore samples of data quickly.

We designed explICU, an ICU-focused, interactive web-based machine learning toolbox that contains two main modules: a patient level visualization module, and a predictive modeling module. EHR data is first processed and added to a database by the user. Currently, explICU is equipped with the Multiparameter Intelligent Monitoring in Intensive Care II ICU dataset (MIMIC-II); however, any EHR data following our data model may be added. We ported the MIMIC-II data into a JSON-like form where each separate event was formatted as a key with values for timestamp and event name. The data were stored in a MongoDB database where all events were stored under one collection. After adding their EHR to the database, researchers use the visualization module to select patients by various characteristics such as demographic features and clinical events, and examine time series visualizations of their disease progression. Finally, researchers can use the predictive modeling module to apply various classification algorithms to their uploaded patient data, compare the performance of various classifiers, and visualize significant risk factors via interactive charts.

To date, very little work has been done in creating such an interactive tool geared towards mortality prediction; yet the need is substantial. To demonstrate its utility, we apply our tool to the MIMIC-II ICU dataset, which contains de-identified, high temporal resolution data including lab results, electronic documentation, and bedside monitor trends and waveforms, for about 31,855 patients. Below we describe the background, as well as our implementation of the machine learning toolbox for mortality-related predictive modeling and visualization.

II. RELATED WORK

To establish the need and use cases for explICU we cover related work in three areas: (1) visualizing raw ICU time series data, (2) predictive modeling for ICU mortality, and (3) attempts to address a combination of these problems using an integrated application.

A. Visualizing Time Series Data

A number of studies have attempted to visualize raw time series data in the ICU setting. In a review paper by Rind et al. [2], 14 different tools for interactively visualizing clinical data are summarized. For example, Lifelines [3] is an interactive visualization tool showing a patient’s temporal history and integrates data similar to that found in the MIMIC-II database. However, such tools mainly serve as visualizations for data exploration purposes and do not integrate machine learning results. Further, such tools are usually demonstrated on small sets of patients. One of the strengths of using the MIMIC-II database is that it allows us to build a retrospective, exploratory visualization revealing patterns from over 30,000 patient histories that clinical researchers can leverage. Many existing tools use stacks of

1 The current implementation may be found at www.explicu.com.

2 https://mimic.physionet.org/
univariate plots over time [3]. Ordonez et al. [4] attempts to improve upon this paradigm by using a novel “spider-plot" visualization. However, users of his tool mentioned that it could more accurately and confidently identify patient diagnoses.

B. Machine Learning for the ICU

Many studies have been performed on clinical data using machine learning methods [5]. More specifically, some studies have used machine learning to predict mortality and ventilator failure in ICU patients [6,7]. Ghassemi et al. [8] used the MIMIC-II database for predictive modeling of ICU patient mortality. Most notably, these models are augmented with clinical notes, which have been shown to increase the model’s predictive power. However, no interactive or visual display of the results was produced to supplement the models. Gotz et al. [9] used select medication and diagnostic features to explore clustering, but did not use a large dataset such as MIMIC-II.

C. Interactive tools

There exist visualizations for other applications of medical data such as personal medical histories [3], lung transplant home monitoring [10], and glucose levels [11]. Mirador [2] is a newer tool that allows for interactive exploration and visualization of large data sets, but is not specific to clinical data. The concept of an interactive clustering visualization for non-clinical data has been addressed by Desjardins et al. [12]. The only visualization tool for ICU data is the Medical Information Visualization Assistant [13,14]. However, this tool only displays raw lab data, without any machine learning analytics or visualization.

III. METHODS

In this section, we detail our approach and implementation of our software.

A. General Outline

We import EHR data into a MongoDB database, and subsequently process data in two modules, the visualization module and predictive modeling module (Fig 1).

B. Data

Our current implementation includes a cohort of 30,000 patients from the MIMIC-II ICU dataset. We isolated events for diagnoses, medications and procedures. We also computed features for 31 high-risk comorbid conditions (known as Elixhauser Comorbidity Measures) [15], which were based upon aggregations of diagnostic codes (ICD-9 codes). These include comorbidities such as congestive heart failure, coagulopathies, neurological disorders, and others. A binary representation was used to indicate presence or absence of a comorbid condition.

C. Visualization Module

The visualization module allows the user to explore patient data in the ICU setting because often times certain adverse events trigger others. For example, sepsis may lead to multiple organ failure and eventually death. A timeline-style visualization allows clinical researchers to visualize important clusters of events spanning large blocks of time in an efficient manner.

Patient Filtering: To allow for easy viewing of subgroups of patients, we implemented an interactive filtering tool to visualize aspects of user-defined subgroups of patients. Fig 2 shows a screenshot of the visualization with three active data filters on. The user can create multiple filters based on the feature set to view a specific subset of the data and then can click on a data item’s “Show Events” button in the table to view a timeline of the patient underneath the table. A “Show Medical History” button allows users to see occurrences of events commonly associated with ICU mortality (Fig 2B).

Patient Timelines: First, select patient data was aggregated using SQLite query results to acquire a set of patients displaying a range of medical events. This acquired dataset was then imported as CSV data files to build the visualization. The vis.js visualization library was used to build the time lines for the patient data.

In order to maintain anonymity, the timestamps in the dataset are sequence numbers for each medication, diagnosis, and procedure for the patients. In order to effectively show these individual events, the sequence numbers were normalized and the differences between pairs of subsequent event timestamps were used to determine the duration of each event. Start and End time were used for those events that were repeated.

Fig 2C shows a visualization of all the diagnostic, medication and procedure events for one example patient. Each event category is displayed on horizontal lines and the count of the particular event is used to change the opacity of the rectangular block for the timeline event.

The patient timeline depicts diagnoses, medications and procedures performed on the patients in an easily interpretable visualization. Mortality cases are highlighted in a red box with which a user can see if selected patients have a fatal mortality event in their particular timeline.

5 http://www.visjs.org

---

3 See https://www.mongodb.org for a description of the MongoDB database features.
4 http://physionet.org/mimic2/
Each event associated with a given patient is represented by a color-coded box. Medications are green, procedures are yellow and diagnoses are light blue. This allows the user to see the similarity between the events of selected patients.

The individual timelines are scrollable and pannable so that a large number of events can be visualized.

D. Predictive Modeling Module

The predictive modeling module runs machine learning classifiers in the back end, and displays the results in the form of interactive visualizations. For each classifier, the medication, diagnosis and procedure features can be used as predictive features, while the occurrences of mortality events for each patient were used as the target labels. Currently, logistic regression is implemented. However, any classification algorithm may be implemented for this module.

To aid users in obtaining information quickly regarding risk factors, we developed an interactive visualization to show logistic regression results. We use interactive bar charts to compare the significance of all features as potential risk factors for mortality. The visualization is implemented using extensions of the D3.js\(^6\) and Morris.js\(^7\) packages.

IV. RESULTS

In this section we describe the application of our web service for the prediction of mortality in the ICU.

A. Prediction Task Setup

We made feature matrices from all 31,855 patients in the MIMIC-II database. All distinct comorbidity events were aggregated as binary features. That is, for each particular feature, a patient was assigned a value of 1 for that feature if it occurred at least once, and a value of 0 for that feature if it did not occur. The target value was constructed in the following manner:

1: mortality occurred after the last ICU visit
0: otherwise

We used logistic regression to classify each patient into either of these two classes.

B. Logistic Regression Implementation

The logistic regression classifier was evaluated with 10-fold cross validation, which is a standard model evaluation technique that divides the data into 10 equal parts, using 9 parts to train the model and the remaining 1 part to test it. We calculated the area under the ROC curve (AUC), sensitivity, and specificity across all 10 folds.

C. Logistic Regression Results

Our multivariate logistic regression model used all 31 comorbidity features. The AUC for the 31,855-patient model was 0.764, respectively. At the optimal cutoff, sensitivity and specificity for the model were 75% and 69%, respectively.

---

\(^6\) http://www.d3js.org
\(^7\) http://morrisjs.github.io/morris.js/
we plan to validate the tool's utility to increase understanding of a patient's state by generating and incorporating user feedback.

VI. CONCLUSION

We deployed a proof of concept for a web service that allows the user to upload data, explore data via interactive visualization of patient events, and see results of predictive modeling tasks for mortality. We showed scalability by integrating a database of 31,855 ICU patients.

REFERENCES