Mortality Prediction in ICU with Recurrent Neural Network

Jiawei Zhu, Baolin Wang, Shimiao Zhang

Georgia Institute of Technology, Atlanta, GA

Abstract

Developing a solid approach for the prediction of patients' mortality in intensive care units can provide multiple benefits in clinical application, e.g. for timely clinical resource allocation. Past research work has shown compelling results when using predictive modeling methods for such purpose. In the current study, Recurrent Neural Network is used to predict the in-hospital mortality of ICU patients. The learning task is performed by leveraging an openly available critical care database (MIMIC-III), and multiple combinations of features constructed from different datasets are used to examine the model prediction capability. It is found that the usage of a single feature related to diagnostic leads to the second optimal model, while including all the features as proposed in our work can eventually result in the best prediction outcome.

1. Introduction and Background

The annual waste cost in healthcare system in United States is very high. The source of waste cost mainly comes from unnecessary services, excessive administrative costs, and inefficiently delivered services. Also, there are a great number of mortality cases which can be prevented every year [1]. The wide applications of machine learning and the era of big data make it possible to use artificial intelligence to predict health condition of individuals. Such possibility can be a key to solve the current issues in healthcare system. For example, a recent paper has studied the use of generic predictive model to recommend diagnosis and medication for a subsequent visit of a patient based on the historical data from the electronic health record [2]. With the help of this model, patients can directly obtain the subsequent diagnosis or medication advice without visiting doctors, which can tremendously reduce the healthcare cost.

Mortality risk prediction for ICU patients has received wide attention in the past decades [3-4]. In this domain, current research work either exploits a so-called score-based method, or is conducted from a broader modeling perspective [4]. A well-known scoring system is Acute Physiology and Chronic Health Evaluation II (APACHE II), introduced by Knaus et al. in 1985. This model is based on 12 physiologic criteria, age and former health condition of the patients. Giving an integer score ranging from 0 to 71, higher scores correspond to more severe disease and a higher risk of death [5]. Score-based methods have shown various advantages compared to many other computational approaches in health care area, since they can be easily understood by converting the severity of a disease into a score, and are more straightforward when compared to other machine learning models. Significant advances have been achieved in score-based methods in past years, however, these models have suffered from multiple limitations such as unevaluated validity in different clinical conditions [6].

There is also a myriad of works to apply machine learning algorithm such as logistic regression (LR), Bayesian network (BN), artificial neural network (ANN) and so on, thanks to the richness and potential of available data and powerful computational capability of modern computers. However, most machine learning methods have to deal with challenges such as physiologic feature integration and model interpretability [4]. Model scalability is also a difficult issue because of the tremendous efforts required to clean and preprocess datasets, as well as the existence of a large amount of potential predictor variables in the electronic health record (EHR) [7].

Despite the difficulties, a number of research work has been conducted to resolve these issues. Luo et al. [4] introduced Subgraph Augmented Nonnegative Matrix Factorization (SANMF) on ICU physiologic time series, and tried to improve both of model accuracy and interpretability. They first used SANMF to convert time series into a graph representation and applied subgraph mining to automatically extract temporal trends. Non-negative matrix factorization was then applied to group trends in a way that approximates patient pathophysiologic states. These trend groups are eventually used as features in training a LR model for mortality risk prediction.

Deep learning and artificial neural networks may overcome aforementioned challenges by learning representations of the key factors and interactions from the data itself. Rajkomar et al. [7] proposed three deep learning neural network model: long short-term memory (LSTM), attention based time-aware neural network (TANN), and a neural network with boosted time-based decision stumps. These architectures are applied to data with varying length and data point density in nature. Their models were shown to be able to achieve a high accuracy for in-hospital mortality prediction (area under the receiver operator curve (AUROC) 0.93 - 0.94). Hrayr et al. [8] proposed four standardized clinical prediction benchmarks using the Medical Information Mart for Intensive Care (MIMIC-III) database. To do this, they developed multiple models including linear regression models and neural architectures, and performed experiments with standard LSTM and its modification, channel-wise LSTM. Several strong baselines for their benchmarks were also described in their work. Such benchmarks will also be used in our project to evaluate the model outcome.

2. Problem Formulation

The ultimate goal of this project is to predict in-hospital mortality of ICU patients from a combination of different factors using Recurrent Neural Network (RNN) model. The MIMIC-III database has patients' clinical records where we will be able to extract and construct multiple features including both of static and time-evolving information. Specifically, we are interested in studying the relationship between in-hospital mortality and trajectory of patients' interactions with the clinical systems. Such interaction will be evaluated from different aspects such as the frequency of a patient visiting ICU, stay duration in ICU, and how often a patient receives health care over time.

3. Approach and Implementation

3.1 Dataset Statistics and Preprocessing

All the data utilized in this project comes from MIMIC-III, a public dataset with clinical care data [9]. Currently, the data related to diagnosis, ICU stays, patients, admissions, and lab results is taken into consideration. This data can be found from the MIMIC-III csv files consisting of ADMISSIONS, ICUSTAYS, PATIENTS, DIAGNOSES_ICD, INPUTEVENTS_CV, INPUTEVENTS_MV, PRESCRIPTIONS and LABEVENTS. INPUTEVENTS_CV and INPUTEVENTS_MV tables have event-related information of patients. These two tables are from CareVue inputs and Metavision inputs respectively. Because the observations they have are not duplicated, these two tables should be unioned. PATIENTS table contains information about patient ID and whether the given patient is dead or not, and thus can be used to join other tables to identify the information about deceased patients and alive patients separately. ICUSTAYS table includes ICU check-in time and check-out time, and thus can be used to present how long patients stay in ICU. DIAGNOSES_ICD has information about what diagnoses the patients have corresponding to ICD9 code. PRESCRIPTIONS table provides patients' medication information. LABEVENTS mainly consists of laboratory test and test results. Based on these tables above, the prediction on the in-hospital mortality of patients can be made by taking consideration of what diagnoses, medicines, lab tests, ICU stay duration patients have. Summary statistics of the used tables are provided in **Table 1**.

The above datasets are pre-processed before they are fed into the feature construction process. We partition the datasets into three components based on the *subject_id* field, with a ratio (subject to change) of 65/20/15, and use them for the training, testing and validation phases, respectively. **Table 2** shows the specific fields for each dataset that we are using to construct the features.

Table 1. Descriptive statistics of tables used in the current project

Descriptive Statistics	Deceased Patients	Alive Patients
Event Count 1. Average Event Count 2. Max Event Count 3. Min Event Count	 185.438 5889 0	 117.408 4377 0
Encounter Count 1. Average Encounter Count 2. Max Encounter Count 3. Min Encounter Count	 1.541 41 1	 1.238 35 1
ICU Stay Duration 1. Average Stay Duration 2. Max Stay Duration 3. Min Stay Duration	 5.036 173.072 0	 5.264 171.623 0
Common Diagnosis	1. Hypertension NOS 2. CHF NOS 3. Atrial fibrillation 4. Acute kidney failure NOS 5. Crnry athrscl natve vssl	1. Hypertension NOS 2. Crnry athrscl natve vssl 3. Hyperlipidemia NEC/NOS 4. Atrial fibrillation 5. CHF NOS

1. Event Count: Number of events of a patient
2. Encounter Count: Number of unique dates when a patient visited the ICU
3. ICU Stay Duration: how long a patient stays in ICU (in days)
4. Common Diagnosis: top 5 frequently occurring disease in ICD 9 code

Table 2. Fields of the datasets used for feature construction

Datasets	Fields
Admissions	subject_id: String, hadm_id: String, admittime: String
Daignose_icd	subject_id: String, hadm_id: String, icd9_code: String
Icustays	subject_id: String, hadm_id: String, icustay_id: String, los: Double
Labevents	subject_id: String, hadm_id: String, itemid: String, valuenum[1]: Double
Patients	subject_id: String, is_dead[2]: Boolean
Prescription	subject_id: String, hadm_id: String, drug: String

1. valuenum: Normalized (i.e. divided by the max value) the original *valuenum* field in the labevents dataset
2. is_dead: Derived from the *dod_hosp* field from the patients dataset

3.2 Software Stack and Hardware Environment

For this project, we primarily use Apache Spark (2.4.0), PyTorch (1.0) and Python (3.7) for multiple purposes including data pre-processing, model development and evaluation, and data visualizations. Specifically, PySpark is used for computations of statistics of the datasets and extraction / construction of the features used for our models. We mainly rely on the Python and PyTorch library to build and evaluate our RNN models and write utility functions as necessary. The results are most likely to be visualized by multiple plots using Matplotlib library, however, depending on the actual needs, we may also utilize D3 tools for interactive visualizations.

Considering the size (~6 GB) of datasets used in this project is still manageable by local machine, we will perform experiments in a local Linux system (Ubuntu 18.10) with above software installed. The machine is equipped with Intel Core i7 CPU and 8 GB RAM.

3.3 Feature Construction

The construction of features is based on the tables of DIAGNOSES_ICD9, PRESCRIPTIONS, LABEVENTS, and ICUSTAYS. More specifically, ICD 9 code in DIAGNOSES_ICD9, drug in PRESCRIPTIONS, the item ID and value number in LABEVENTS, and length of stay in ICUSTAYS are considered as features that have impact on the mortality prediction of a patient in ICU units. ICD 9 code is a standardized code which corresponds to diagnoses and procedures recorded in the hospital. The first 3 or 4 digits prior to the decimal point of each ICD 9 code are extracted since these digits represent the procedure codes. Drug stands for the medicine prescribed to the patients. The item ID refers to the lab measurement type and value number refers to the result of the corresponding lab measurement. Length of stay is the time duration of a patient staying in the ICU units.

Creating feature map is a crucial part in feature construction. Each of ICD 9 codes after extraction, drug and item ID are mapped to a distinct number which is the feature ID. In addition, one more feature is added to the map which is the length of stay in ICU units. The total number of features is 4856; the number of diagnostic features is 1044; the number of diagnostic features plus length of ICU stay feature is 1045; the number of diagnostic features plus length of ICU stay feature plus lab result features is 1516. The tables in training dataset are utilized to construct the feature map. And any features that could be found in the tables in validation dataset or testing dataset but are not included in the feature map are discarded because the model does not learn these features in the training phase. In the second place, features in the dataset are converted to the feature ID on the basis of the feature map.

Since the input data that LSTM model requires has to be in sequence, all the features in the dataset for each patient needs to be sorted by the admit time that can be found in the ADMISSIONS table. Each table related to feature construction is joined with the ADMISSIONS table on both SUBJECT ID and HADM ID. Each HADM ID represents a distinct admission of a patient to the hospital. In the sequence input data, each feature that a patient has at a distinct date is represented as a tuple. The first element in the tuple is the feature id, and the second one is the feature value. All the feature values of the non-numeric features including ICD 9 code and drug are set to be 1 which indicates this patient has this feature. On the other hand, the feature values of the numeric features such as item ID and length of stay are the lab measurement value and time duration of the ICU stay.

3.4 Model Development and Evaluation

In this project, we are using RNN as the predictive model to take advantage of the well-defined time series data. Traditional neural networks can only accept a fixed-sized vector as input and produce a fixed-sized vector as output and use a fixed number of computational steps. There is no way to let the previous information persist, while the recurrent neural networks can address this issue. They are networks with loops in them, allowing us to operate over sequences of vectors and the information to persist.

More specifically, we are using LSTM since a lot of work has demonstrated its efficiency. At current stage, the architecture of the model is shown in **Figure 1**. The First layer is a fully-connected layer consisting of several hidden units that actually work as an embedding layer to map the high-dimensional inputs to low-dimensional space. This layer is followed by a LSTM layer with different hidden units. The third layer is a fully-connected layer consisting of 2 hidden units that correspond to two class (i.e. alive and dead). Sigmoid activity function is used after the first layer and no activity function is applied to the third layer.

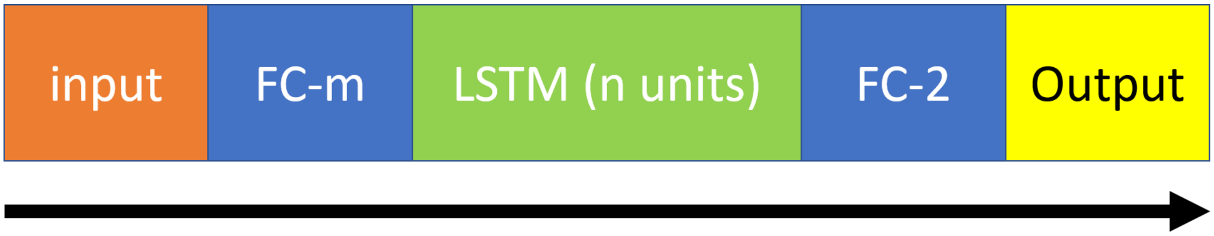


Figure 1. Schematic illustration of the model architecture.

To train the model, we use cross entropy loss function and Adam algorithm with a learning rate and L2 penalty to be tuned. We use AUROC as evaluation metrics for our model, since it is commonly used in binary classification problems. It suits our project well because the mortality prediction is a binary classification problem, even though the length of stay prediction can be treated as binary classification aimed at identifying patients at risk for long stays. Sigmoid activity function is used to compute the probability of mortality.

4. Results and Discussion

Several investigations on the impact of features on the mortality prediction result have been conducted to obtain better model performance. Different combinations of features are fed into the LSTM model with 2 epochs, and the produced AUC values and confusion matrices (**Figure 2**) are utilized to evaluate the impact

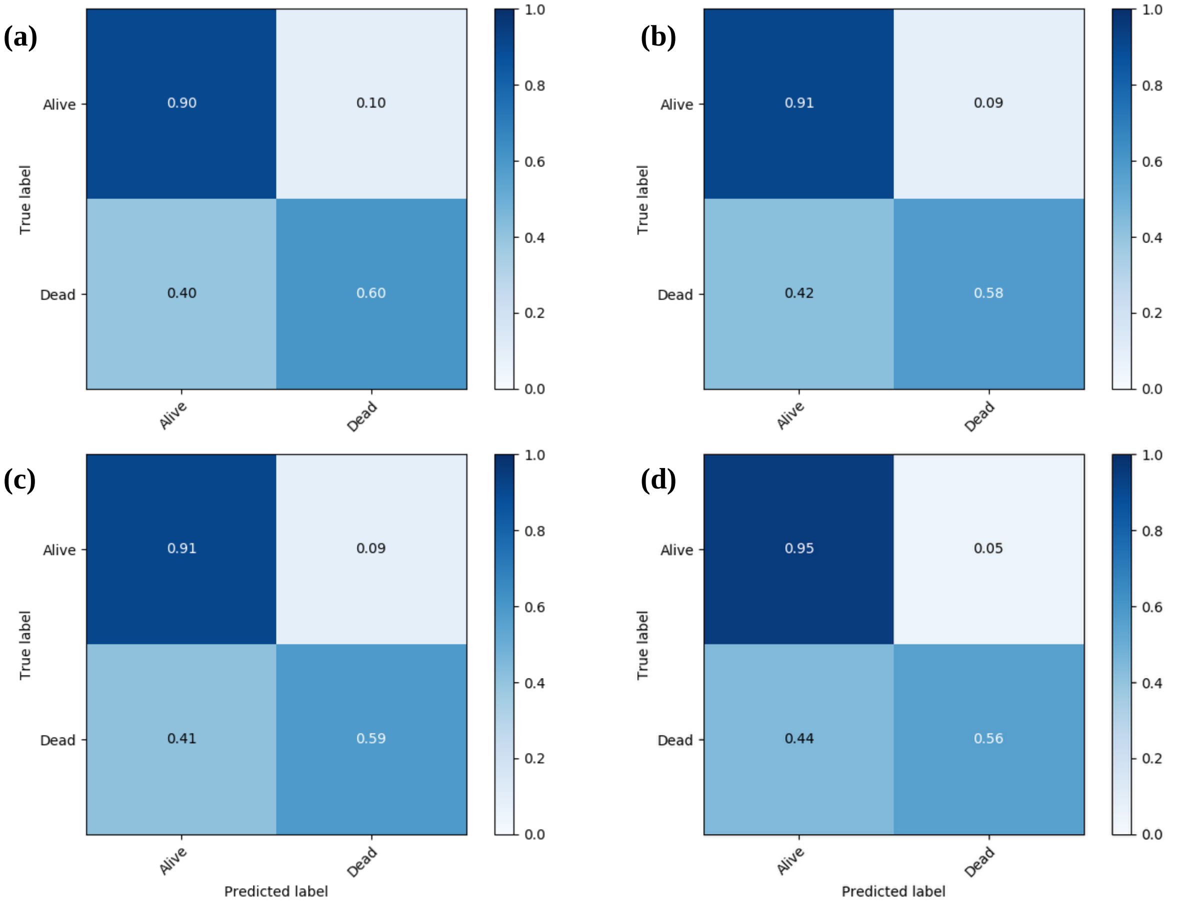
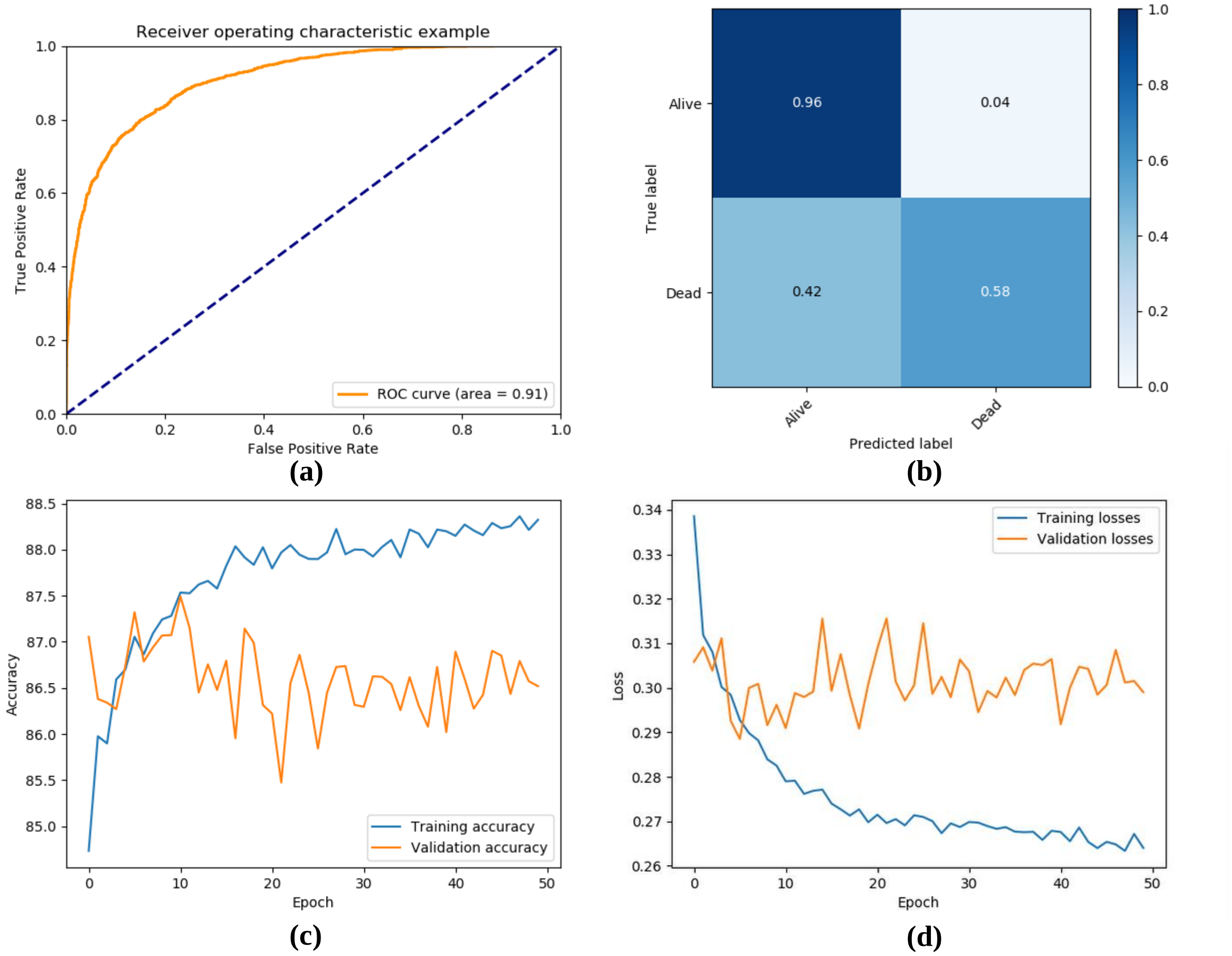


Figure 2. Confusion matrices when different combinations of features are used. (a) diagnostics only; (b) diagnostics + length of ICU stay; (c) diagnostics + length of ICU stay + lab events; (d) all features.

of the certain combination of features. False positive rate indicates that dead patients are predicted as alive and false negative rate means that alive patients are predicted as dead. High AUC score and low false positive and false negative rate indicates better model prediction. When only the diagnostic features (**Figure 2a**) are included in feature construction, the AUC score is 0.88, the rate of false positive is 0.40, and the rate of false negative is 0.10. As for the model with only diagnostic features and length of ICU stay feature (**Figure 2b**), the AUC, false positive rate, and false negative rate are 0.87, 0.42, 0.09. As for the model with diagnostic features, length of ICU stay feature, and lab result features (**Figure 2c**), the AUC, false positive rate, false negative rate are 0.87, 0.41, 0.09. When all the features consisting of diagnostics, medications, lab results, and length of ICU stay are taken into consideration (**Figure 2d**), the AUC, false positive rate, false negative rate are 0.90, 0.44, 0.05 respectively. It can be observed that the combination of all features outperforms the other combinations in terms of AUC and false negative rate. Accordingly, the combination of all features is utilized in our final LSTM model.

The results of our best model are primarily shown by four plots, i.e. curves of accuracy and loss verse epoch, ROC curve and confusion matrix (**Figure 3**). This model was developed after extensive parameter tunings using the combination of all features as discussed above. Specifically, it is a LSTM model that consists of the first fully-connected layer contains 64 hidden layer and the LSTM layer with 16 hidden units. The learning rate and L2 penalty for Adam method are 0.007 and 0.0004, respectively when we set the batch



(a) **(b)**

(c) **(d)**

Figure 3. Results for the best model developed in this work. (a) ROC curve; (b) confusion matrix; (c) curve of accuracy verse epoch; (d) curve of loss verse epoch.

size to be 64 and train the model with 50 epochs. The corresponding best accuracy of validation dataset is 0.875. The parameters settings are summarized in **Table 3**.

Table 3. Parameters used for the best model.

Parameters	**Values**
Hidden Layers in the First FC Layer	64
Hidden Units in the LSTM Layer	16
Hidden Layers in the Second FC Layer	2
Learning Rate	0.007
L2 Penalty	0.0004
Batch Size	64
Training Epochs	50

5. Conclusion

In this work, LSTM model is utilized to predict the in-hospital mortality of ICU patients. The public critical care database, MIMIC-III, is used for the model development and evaluation purposes. We considered the datasets that include patients' information on diagnosis, prescriptions, ICU stays and lab events. In addition to parameter tunings, multiple combinations of features were constructed and examined for the best prediction results. We started with a relatively naive method, which only included the features related to diagnostics. Despite its simplicity, the developed model was able to achieve a high AUC score of 0.88. More information, i.e. length of stays in ICU, medications and lab events, were then considered in our feature construction process. It was observed the addition of these features resulted in slight improvement in the model prediction performance, where the AUC was increased to 0.90. The model could be further improved by performing a parameter tuning step, which eventually led to the highest AUC score, i.e., 0.91, that we have observed throughout our model development process. As a result, we conclude that the usage of all features discussed above and the parameters described in the **Table 3** can yield the best prediction outcome.

References

[1] https://hcldr.wordpress.com/2018/07/15/wasted-healthcare-spending-a-750-billion-opportunity/, accessed on March 3, 2019.

[2] Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., and Sun, J., **2016**. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pp.301 - 318.

[3] Ge, W., Huh, J.W., Park, Y.R., Lee, J.H., Kim, Y.H. and Turchin, A., **2018**. An Interpretable ICU Mortality Prediction Model Based on Logistic Regression and Recurrent Neural Networks with LSTM units. In *AMIA Annual Symposium Proceedings*, Vol. 2018, p. 460, American Medical Informatics Association.

[4] Luo, Y., Xin, Y., Joshi, R., Celi, L. and Szolovits, P., **2016** February. Predicting ICU mortality risk by grouping temporal trends from a multivariate panel of physiologic measurements. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[5] Knaus, W.A., Draper, E.A., Wagner, D.P. and Zimmerman, J.E., **1985**. APACHE II: a severity of disease classification system. *Critical care medicine*, *13*(10), pp.818-829.

[6] Aminiahidashti, H., Bozorgi, F., Montazer, S.H., Baboli, M. and Firouzian, A., **2017**. Comparison of APACHE II and SAPS II scoring systems in prediction of critically ill patients' outcome. *Emergency*, *5*(1).

[7] Rajkomar, A., Oren, E., Chen, K., Dai, A.M., Hajaj, N., Hardt, M., Liu, P.J., Liu, X., Marcus, J., Sun, M. and Sundberg, P., **2018**. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, *1*(1), pp.18.

[8] Harutyunyan, H., Khachatrian, H., Kale, D.C. and Galstyan, A., **2017**. Multitask learning and benchmarking with clinical time series data. *arXiv preprint arXiv:1703.07771*.

[9] Johnson, A.E., Pollard, T.J., Shen, L., Li-wei, H.L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A. and Mark, R.G., **2016**. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3, p.160035.

Challenge:

There are some challenges encountered in the feature construction stage. The feature construction program mainly has two parts including feature mapping and producing sequence data. Since the size of the data after preprocessing is still very large, spark is utilized for feature mapping. In the beginning, we would like to use spark for both feature mapping and producing sequence data. However, it is very hard to output the sequence data using mutable arrays from spark with the correct format that the LSTM model program can easily read. And thus python is picked for producing the sequence data. In addition, because of the space in front of the words and letter case of the string features, the feature map is not constructed correctly in the beginning, which results in bad model performance. Almost all the patients are predicted as alive, and the ROC score is around 0.50. After modifying the string features, the model performance is enhanced tremendously, and the ROC score becomes 0.90.

Table of effort contributions

Member	**Contributions**
Shimiao Zhang	Mainly responsible for data pre-processing. Participate in proposal / draft / final report writeup, and final presentation
Jiawei Zhu	Mainly responsible for feature construction. Participate in proposal / draft / final report writeup, and proposal / final presentations
Baolin Wang	Mainly responsible for model development and results visualizations. Participate in proposal / draft / final report writeup, and proposal / final presentations