Early Prediction of Sepsis from Clinical Data

Elena Wang, Master in Statistical Science;

Dr. Ricardo Henao; Dr. Eric Laber

Duke University

Abstract

Sepsis is a syndromic response to infection and is always life-threatening and costly to treat. Because of the complexity of the early syndrome of sepsis, it is hard to find the early identification of this disease, so it attaches great importance to this early recognition. This report would investigate automated algorithms from traditional to sophisticated methods in terms of dealing with over 70% incomplete observations and 2:98 imbalance common problems in reality, which would contribute to the application of Sepsis prognosis and treatment. Especially, two major techniques will be introduced: Logistic Regression without Missing Values and Gradient Importance Learning within Missing Values.

1. Introduction

Sepsis is a life-threatening condition that occurs when the body's response to infection causes tissue damage, organ failure, or death (Singer et al., 2016). According to WHO and CDC's data, nearly 270k+ people die from Sepsis, and 24 billion would be spent on the treatment of Sepsis in U.S hospitals each year. It could be lifesaving and cost-saving if the Sepsis condition of a patient could be diagnosed in advance by using several vital signs. Our goal is to use Electronic Medical Record (EMR) data which is sourced from ICU patients from 3 separate hospitals, to develop the best machine-learning algorithm within or without imputing missing values in order to predict each patient's sepsis status by using selected vital signs as precise as possible, which could be helpful for the common problems of large incomplete observations and imbalance in reality.

Our data is provided by the Physionet database, which is a repository of freely available medical research data. Since this widely used database is managed by MIT lab and sponsored by NIH, we could consider that it is an authentic and practical real-world medical record and could be used to represent the information of potential Sepsis patients. It is reported that K-nearest-Neighbour (KNN) has satisfied Sepsis predictions (Perng et al., 2019) while GB-boost and Random Forest may have better performance (Ke li et al.). Therefore, there are many mature machine learning algorithms and a range of programming software in use for machine learning algorithms in the diagnosis of Sepsis status. However, these traditional machine learning algorithms are based on no missing values in the datasets and imputing missing values could not achieve 100% accuracy for real-world analysis, which stimulates an advanced algorithm "Gradient Importance Learning" to predict Sepsis status without imputation.

2. Method1: Traditional Logistic Regression

2.1 Data Description and Preprocessing

This Sepsis dataset, which is labeled by Sepsis-3 clinical criteria, contains Sepsis conditions, 8 vital physical signs, 26 laboratory values, and 6 demographic features with Gender as categorical and others as numeric for each hour of 40,336 patients' ICU stay in Emory University Hospital and Beth Israel Deaconess Medical Center from 2009 to 2019. The original dataset was separated by Physionet into training A (20,336 subjects, 41 variables) and testing B (20,000 subjects, 41 variables), in which each subject records time-series information to detect Sepsis binary outcome. Some of the variables are described in **Table1**, and more details about training A and training B can be referred in **Table2**. The response variable **SepsisLabel** is described following:

SepsisLabel, 1 if t>=t_sepsis - 6 and 0 if t<t_sepsis-6, while for non-sepsis patients is 0.

Heart rate(beats/min)	Gender(0 for Female)	Temperature(° C)	systolic BP(mm Hg)	Mean arterial pressure(mm Hg)	Diastolic BP(mm Hg)	Respiration rate(breaths/min)
ICU length of stay	Bicarbonate(mmol/L)	Fraction of inspired oxygen(%)	Unit1/Unit2(Administrative identifier for ICU)	carbon dioxide pressure from blood(mm Hg)	Oxygen saturation(blood, %)	Time between hospital and ICU admission

Table 1. Some Example Variables

	Patients #	Sepsis Patients	Covariates #	Missingness Rate	Average Time Points/Patient
Training A	20,336	1790	36	73%	39
Testing B	20,000	1142	36	74%	38

Table 2. Summary of Dataset

Missing Imputation: Firstly, Unit1, Unit 2, ICU length of stay, and Time between hospital and ICU admission variables are dropped since these variables did not convey information that could be used to prognose the Sepsis situation. Second, the End-tidal carbon dioxide variable has every observation missing (MCAR) and was dropped from both the train and test set. There are more than 50,000 missing values in each column except for Age, Gender, and Sepsis Label, and each value is based on neighbors for each subject (MNAR) in the biomedical view; therefore, I decided to impute the missing values in each column by using the last or next records of the missing values for each patient. If both the previous and next values are also missing, we would replace the mean for continuous variables and the mode for categorical variables.

For the **outliers**, several observations have values larger or smaller than the mean plus or minus 1.5 Interquartile Range, respectively. However, I considered that it is normal for patients in ICU to have extreme values for vital signs, so we did not remove outliers to ensure all observations are valid. I also checked for extreme values, like 999, and did not find this value.

2.2 Dealing with Imbalanced Outcome and Various Rearrangements

To avoid the situation that traditional machine learning model always classified the outcome into the majority group, it is necessary to make sure that our outcome has a balanced Sepsis situation. The original training and testing set had a very imbalanced Sepsis situation (2:98). Here I tried several different ways to rearrange datasets:

i. Considering the great benefit of early detection of Sepsis, we extracted the initial situation of patients, i.e. using the laboratory results and demographic features of the first time when patients visited hospitals. This gave us 20,336 records in the training set

and 20,000 data in test set. Notice that the final datasets are still imbalanced: percentages of 0 (do not develop sepsis) in the **SepsisLabel** in datasets A and B are around 91% and 94%, respectively. Afterwards, I kept different ways to deal with this problem:

a. Extracted 10% of the majority group for the Sepsis condition in each dataset and combined it with the minority group. The result before and after this process is shown in **Table3**. Afterward, the outcome of each training and testing is approximately balanced for further modeling.

	Train Set A			Test set B		
	Sepsis Label = 0	Sepsis Label = 1	Total	Sepsis Label = 0	Sepsis Label = 1	Total
Before	18,546(91%)	1,790(9%)	20,336	18,858(94%)	1,142(6%)	20,000
After	1,670(46%)	1,975(54%)	3,645	1,779(59%)	1,249(41%)	3,028

Table 3. Result of Data Pre-processing

b. Applied SMOTE method

ii. After modelling the original dataset after imputation, the AUC is showing around 0.78 in Gridsearch Cross Validation for Logistic Regression, which means that the dataset can be good for modelling. The next thing is to find which time point has the best AUC performance (should be increase as the time going through). In this case, from the last of all records to see the AUC, to avoid the missing time points, we reorder the dataset by counter wise time points and if the records in one time is less than 10% of the number of the dataset, then we wouldn't keep this time. **Figure1** visualization would make more sense.

TimePoint	1	2	3	4	5	6	7	8	9	10	...	327	328	329	330	331	332	333	334	335	336
SubjectID																					
p100001	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p100002	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p100003	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p100004	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p100005	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
p119996	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p119997	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p119998	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p119999	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
p120000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

20000 rows × 336 columns

Figure 1: Dataset Rearrangement

2.3 Above Datasets Model Results

All rearranged datasets above would be applied to the same traditional logistic regression and important variables' selections as well as compared with the AUC scores. The first two balanced datasets give around 0.69, and the third method for counter wise could achieve 0.65 (**Figure2**).

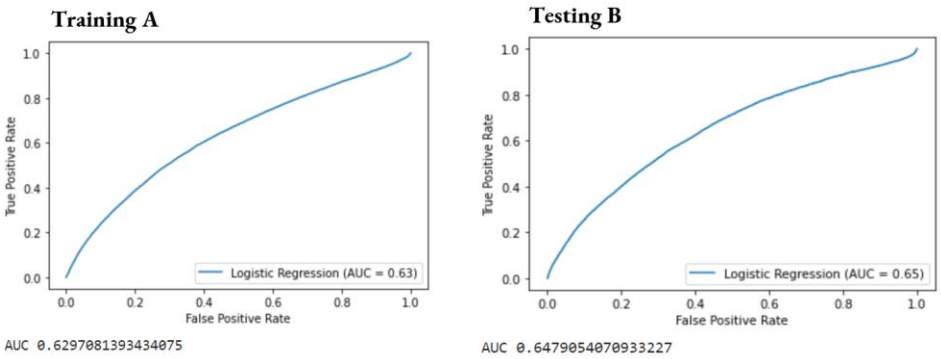

Figure2: Logistic Regression AUC

3. Exploratory Data Analysis

This part will dive into data performance to provide brief views and assumptions for future modeling after balancing data in the 1st method. First, standardization is a helpful transformation to be applied as mandatory for some models to work correctly, which can center the data to a mean of 0 and scale to a standard deviation of 1. Below distribution plots after standardization in **Figure3** indicates that all numerical predictors are in the same scale. Some of them, such as Hct, Hgb, HR, etc., look approximately normalized, which means that models requiring strict assumptions could be used later, for example, Logistic Regression with a penalty, KNN, Support Vector Machines (SVM), etc. **Figure4** displays a good separation between a categorical predictor Gender and objective SepsisLabel, which shows that sepsis/non-sepsis patients are evenly distributed in each gender. Logistic regression requires basic assumptions such as the independence of errors, absence of multicollinearity, etc. This needs us to check the correlation between the continuous variables. After checking for the correlation, it can be identified that besides MAP and SBP, BaseExcess and HCO3, Hgb, and HcT, most predictors have a low correlation with each other, which satisfies the requirement (**Figure5**).

To observe the performance of response variables, we executed some unsupervised machine learning techniques. Principal Component Analysis (PCA) method will reduce the dimension of the features down to 2 and observe whether there is any evidence of clustering for sepsis and non-sepsis patients. **Figure6** observes that there is no clear separation between the patients, and the first PC only explains 8.91% of the variance, which gives intuitive non-linear algorithms such as kernel SVM. K-means visualization (**Figure7**) also can approve non-linear findings with k=2, and there are 1759 non-sepsis vs. 2114 sepsis patients in cluster1 while there are 1675 non-sepsis vs. 1125 sepsis patients in cluster2, which means that cluster1 could be labeled as sepsis and cluster2 could be labeled as non-sepsis.

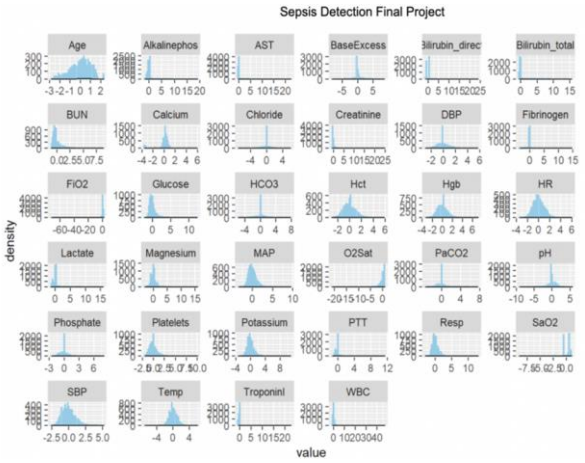

Figure3: Distribution of Covariates After Standardized

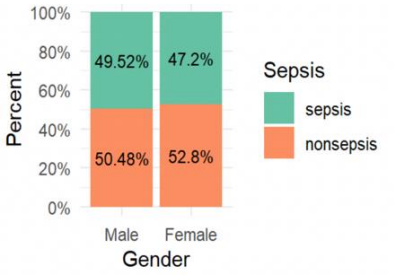

Figure4: Gender vs. Sepsis

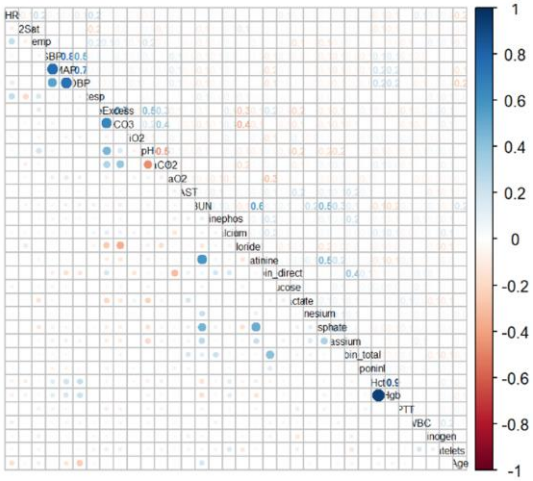

Figure5: Correlation Plot

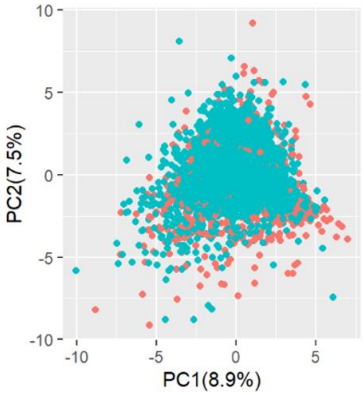

Figure6: PC1 vs PC2

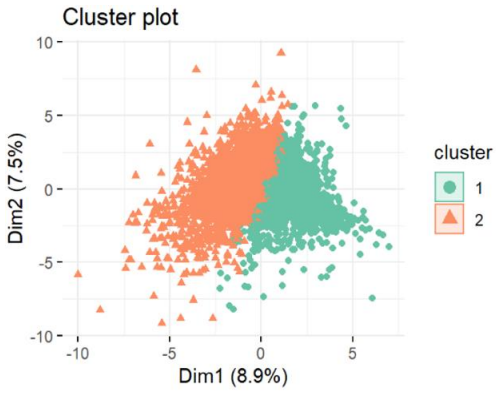

Figure7 : K-means Clustering

4. Method2: Logistic Regression by Gradient Descent Update

Here I implemented original imputed and standardized datasets to build basic Logistic Regression with Gradient Descent Update by TensorFlow Core APIs in order to decrease log loss in each iteration and compute a weighted cross entropy to trade off recall and precision by up- or down- weighting the cost a positive error relative to a negative error.

Parameters for better performance:

positive weight=50, batch size=12800, epochs=300, and learning rate=0.01

Below **Figure8** demonstrates fast convergent best results, which satisfies the requirements, and the Lowest Log Loss could achieve around 1.3, relatively highest AUC score around 0.65:

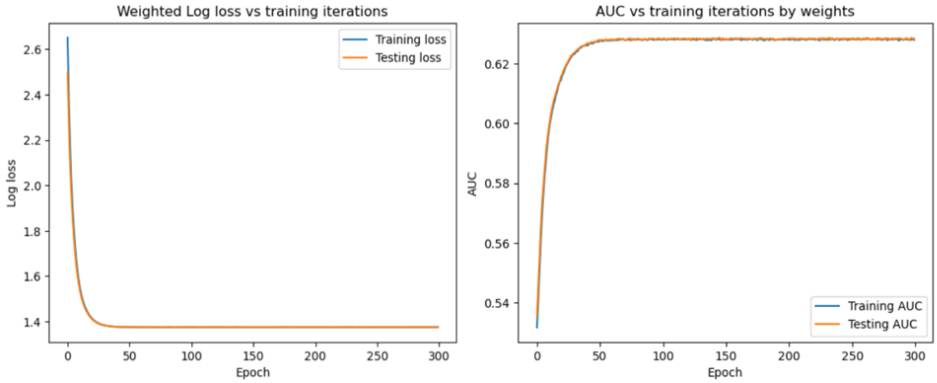

Figure8: Weighted Log Loss & AUC Curve

5. Method3: Gradient Importance Learning for Incomplete Observations

Some assumptions for missing value imputation do not satisfy the real-world application and could suffer from poor performance in the subsequent tasks, especially for datasets with large missingness rate or a small sample size, which also could constrain the capabilities of the prediction model. Gradient Importance Learning (GIL) method trains multilayer perceptrons (MLPs) and long short-term memories (LSTMs) to directly perform inference from inputs containing missing values, specifically, we employ reinforcement learning (RL) to adjust the gradients used to train these models via back-propagation. This allows the model to exploit the underlying information behind missingness patterns referred the paper "*Gradient Importance Learning for Incomplete Observations*" (Gao et al., 2022). Tabular analysis is designated here: training and testing sets separated by normal (all non-sepsis) and abnormal (all sepsis). However, this complicated algorithm is easily causing overfitting problems and several ways are tried to deal with such as, reducing the weights, increasing the weights & biases regularizers, adding drop-off layers, removing layers, adding positive weights, etc. Unfortunately, this method is kind of hard to get it work. The algorithms are showing below:

Algorithm 1 Gradient Importance Learning (GIL).

Input: $\mathcal{X}, \mathcal{Y}, \mathcal{M}, \mathbf{W}_{enc}, \mathbf{W}_{inf}, \pi_\theta, Q_\nu, \alpha_\theta, \alpha_\nu, \alpha, E$

Begin:

Initialize $\mathbf{W}_{enc}$ and $\mathbf{W}_{inf}$, actor π_θ and critic Q_ν

Sample $\mathbf{x}$ from $\mathcal{X}$ and obtain the corresponding label $\mathbf{y}$ from $\mathcal{Y}$

Obtain the feature $\zeta \leftarrow f_{enc}(\mathbf{x}|\mathbf{W}_{enc})$ and prediction $\hat{\mathbf{y}} = f_{inf}(\zeta|\mathbf{W}_{inf})$ from the encoding and inference layers, respectively

$\mathbf{s} \leftarrow (\mathbf{x}, \mathbf{m}, \zeta, \hat{\mathbf{y}})$

for $iter$ in $1 : max\_iter$ **do**

  Obtain importance from a behavioral policy $\mathbf{a} = \beta(\mathbf{s}|\pi_\theta)$

  Train the encoding layer following $\mathbf{W}'_{enc} \leftarrow \mathbf{W}_{enc} - \alpha\boldsymbol{\Delta} \cdot (\mathbf{x}^\top \odot \mathbf{a}^\top)$ as in (4)

  Train the inference layers following regular gradient descent, *i.e.*, $\mathbf{W}'_{inf} \leftarrow \mathbf{W}_{inf} - \alpha(\partial E/\partial \mathbf{W}_{inf})_{SGD}$

  Obtain the prediction following the updated weights $\hat{\mathbf{y}} \leftarrow f(\mathbf{x}|\mathbf{W}'_{enc}, \mathbf{W}'_{inf})$

  Obtain the reward $r \leftarrow R(\mathbf{s}, \mathbf{a})$

  Get a new sample $\mathbf{x}'$ from $\mathcal{X}$ and obtain the corresponding label $\mathbf{y}'$ from $\mathcal{Y}$

  Obtain the feature $\zeta' \leftarrow f_{enc}(\mathbf{x}'|\mathbf{W}'_{enc})$ and prediction $\hat{\mathbf{y}}' = f_{inf}(\zeta'|\mathbf{W}'_{inf})$ from the encoding and inference layers, respectively

  $\mathbf{s}' \leftarrow (\mathbf{x}', \mathbf{m}', \zeta', \hat{\mathbf{y}}')$

  Update the actor π_θ and critic Q_ν using $(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')$ following (5)

  $\mathbf{s} \leftarrow \mathbf{s}', \mathbf{W}_{enc} \leftarrow \mathbf{W}'_{enc}, \mathbf{W}_{inf} \leftarrow \mathbf{W}'_{inf}$

end for

The algorithm takes as input the training dataset X, the training targets Y, the missing indicators M, the weights of the encoding layers W_{enc}, the weights for the inference layers W_{inf} , the actor π_θ, the critic Q_ν, learning rates $\{\alpha, \alpha_\theta, \alpha_\nu\}$ and training loss function E. Our approach starts by initializing all the parameters W_{enc}, W_{inf}, π_θ, Q_ν, sampling $x \in X$ with the corresponding $m \in M$ that will be used for training in the first iteration, obtaining the feature ξ and the prediction $\hat{y}$, which constitute the initial state as $s = (\ x, m, \xi, \hat{y})$. In each iteration, first the importance is generated from a behavioral policy β that is conditioned on the target policy π_θ, such as the noisy exploration policy proposed in Lillicrap et al. (2015). Then the encoding layer is trained following (4), while the inference layers are trained following the regular gradient descent. After training, the new prediction is obtained following the updated weights and its value is assigned to $\hat{y}$, which is then used to generate the reward following the reward function R. Then the training sample $x^{'}$ for the next iteration is sampled and the corresponding $m^{'}, \xi^{'}, \hat{y}^{'}$ are obtained to constitute the next state $s^{'}$. Finally, the actor π_θ and critic Q_ν are updated following (5). We refer to Silver et al. (2014); Lillicrap et al. (2015) for more details on actor-critic RL.

6. Conclusions

This study made a good fitness of assumptions and comprehensive algorithms for logistic regression and also tried sophisticated way to make the forecast more precise. Missing entries leads poor performance by simply generation of estimation as the assumptions may do not satisfy the real-world applications, such as, patient time-series characteristics and a huge proportion of missingness rate or a small sample size and imputation errors would limit the model capabilities. However, GIL model is hard to get it work due to big overfitting drawbacks. Future improvements would be focused on simplifying and tuning GIL model for the future applications.

Reference

[1] Gaoqitong. (n.d.). Gaoqitong/gradient-importance-learning: Gradient importance learning for incomplete observations, Gao et al., ICLR 2022. GitHub. Retrieved December 1, 2022, from https://github.com/gaoqitong/gradient-importance-learning

[2] Reyna, M., Josef, C., Jeter, R., Shashikumar, S., Moody, B., Westover, M. B., Sharma, A., Nemati, S., & Clifford, G. D. (2019, August 5). Early prediction of sepsis from clinical data: The PHYSIONET/computing in cardiology challenge 2019. Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019 v1.0.0. Retrieved December 1, 2022, from https://physionet.org/content/challenge-2019/1.0.0/

[3] Singer, M., Deutschman, et.al,.(2016). The third international consensus definitions for sepsis and septic shock (sepsis-3). JAMA, 315(8), 801. https://doi.org/10.1001/jama.2016.0287

[4] Deng, H.-F., et.al., (2022). Evaluating machine learning models for sepsis prediction: A systematic review of methodologies. IScience, 25(1), 103651. https://doi.org/10.1016/j.isci.2021.103651

[5] Centers for Disease Control and Prevention. (2022, August 9). What is sepsis? Centers for Disease Control and Prevention. Retrieved November 28, 2022, from https://www.cdc.gov/sepsis/what-is-sepsis.html

[6] World Health Organization. (n.d.). Sepsis. World Health Organization. Retrieved November 28, 2022, from https://www.who.int/news-room/fact-sheets/detail/sepsis

[7] Zhao, X., Shen, W., & Wang, G. (2021). Early prediction of sepsis based on machine learning algorithm. Computational Intelligence and Neuroscience, 2021, 1–13. https://doi.org/10.1155/2021/6522633

[8] Goh, K. H., Wang, L., Yeow, A. Y., Poh, H., Li, K., Yeow, J. J., & Tan, G. Y. (2021). Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare. Nature Communications, 12(1). https://doi.org/10.1038/s41467-021-20910-4

[9] Kim, H. I., & Park, S. (2019). Sepsis: Early recognition and optimized treatment. Tuberculosis and Respiratory Diseases, 82(1), 6. https://doi.org/10.4046/trd.2018.0041

[10] Nemati, S., Holder, A., Razmi, F., Stanley, M. D., Clifford, G. D., & Buchman, T. G. (2018). An interpretable machine learning model for accurate prediction of sepsis in the ICU. Critical Care Medicine, 46(4), 547–553. https://doi.org/10.1097/ccm.0000000000002936

Acknowledgements: Health Data Science at Duke is supported by the National Center For Advancing Translational Sciences of the National Institutes of Health under Award Number UL1TR002553. The Duke Protected Analytics Environment (PACE) program is supported by Duke's Clinical and Translational Science Award (CTSA) grant (UL1TR001117), and by Duke University Health System. The CTSA initiative is led by the National Center for Advancing Translational Sciences (NCATS) at the National Institutes of Health.