Time-Padded Random Forest Ensemble to Capture Changes in Physiology Leading to Sepsis Development

Ben Sweely[1], Austin Park[1], Lia Winter[1], Longjian Liu[2], Xiaopeng Zhao[1]

[1] Department of Mechanical, Aerospace, and Biomedical Engineering
University of Tennessee, Knoxville, TN, USA
[2] Department of Environmental and Occupational Health
Drexel University, Philadelphia, PA, USA

Abstract

Background: The goal of this project was to predict sepsis six hours before current clinical detection methods. Sepsis is the leading cause of in-hospital deaths, and it is one of the costliest complications to treat. Detection of sepsis is complicated and not yet efficient. Each hour of delay in treatment for a septic patient results in a 4-8% increase in chance of mortality.

Method: The dataset provided consists of files that contain hourly parameter measurements for over 40,000 unique patients. The challenge given is a complex problem with a lot of available data. Therefore, a model that would best capture the complexity of this problem was needed. A boosted random forest ensemble was chosen and developed in MATLAB in hopes of producing the best results for this challenge. The provided data was time padded for 8 additional hours' worth of data, 10-fold cross-validated, and imputated with previous data. Many ensemble methods were tested with Random Under-Sampling Boosting performing the best. For this model, the hyper-parameters were optimized via a grid search to find an optimal model.

Results: Using the optimized hyper-parameters along with the correct pre-processing techniques, a 10-fold average utility score of 0.421 was achieved on the training sets A and B combined. We participated in Physionet Challenge under the name SOS: Searching of Sepsis and the utility score on full test set is 0.314. Our official rank is #14.

1. Introduction

Sepsis is a life-threatening condition that occurs when a patient's body initiates a dysregulated response to infection that can cause tissue damage, organ failure, and even death [1-4]. Sepsis can be difficult to diagnose because the manifestation of symptoms varies from patient to patient, and many times the early signs of sepsis are similar to those of other conditions [2, 3].

Globally, it is estimated that nearly 30 million people suffer from sepsis each year and 6 million of these cases result in death [3]. The CDC estimates that sepsis claims the lives of 270,000 of the 1.7 million adults in the US who develop the condition each year [2]. This corresponds with a mortality rate of over 15%. For patients who develop septic shock, the associated mortality rate is higher than 40% [1]. A recent study found that sepsis is the costliest disease state in the US, with total expenditures of $24 billion [5].

It is a widely accepted notion that early detection of sepsis significantly improves patient prognosis and chance of survival [1, 3-5]. Each hour of delay in treatment, especially antibiotic administration, for a septic patient results in a 4-8% increase in mortality [6].

The 2019 PhysioNet problem challenges participants to use physiological data to detect sepsis six hours earlier than the clinical prediction of sepsis [7]. Sepsis is defined, according to the Sepsis-3 guidelines, as clinical suspicion of infection via ordering of blood cultures IV antibiotics and a two-point increase in SOFA score [1].

2. Data

The PhysioNet datasets were provided as a set of pipe separated value (.PSV) files. Each patient has a single PSV file, with there being a total of 40,336 patients. Each row contains hourly entries of the 40 given parameters. Missing parameter values are indicated with a NaN entry. The last column is the sepsis label, which is 0 for a negative sepsis diagnosis and 1 for a positive sepsis diagnosis. Sepsis patients make up about 10% of the total patient data given, and the timestamps that indicate sepsis make up around 3%. The data is highly unbalanced, which adds to the difficulty of finding a good model.

3. Methods

3.1. Pre-processing

Feature Selection has been shown to help solve the data dimensionality problem. First, we looked for features that had severely missing data. Then, we did a literature survey

on each parameter to determine its relevance (Tab. 1). Considering the availability and relevance of each parameter we initially narrowed the feature size from 40 down to 24. Later, we tested the reduced feature size against the original for each model. We found that in our final model, it performs significantly better with all 40 features. We plan on doing further analysis with the literature survey that was conducted.

Table 1. Relevant parameters for sepsis prediction.

Abbreviation	Definition	Measurement	Current Use in Sepsis Diagnosis	Source
Age	Years	100 for patients 90 or above	Mortality rate of 26% in patients 60–64, 38% in patients ≥85	Starr, 2014
SBP	Systolic blood pressure	mmHg	<100 mmHg (OR more than 40mmHg below normal), <65 mmHg = septic shock (qSOFA)	NICE Guideline, No. 51., 2016
Lactate	Lactic acid	mg/dL	>2.8	Muller, 2000
HR	Heart rate	beats per minute (bpm)	>130 bpm	NICE Guideline, No. 51., 2016

Imputation is an important step in pre-processing. We combed through each patient's data and replaced any missing data points with the nearest previous value. If all the data was missing for a feature, we tried replacing the remaining missing data with zero. This imputation method yielded the best results in terms of utility score.

After replacing all the missing data, the data had to be split for cross-validation purposes. We created 10 equal folds of data based on the patient size (4033 patients per fold). Nine of these folds were used to train the model while the remaining fold was used for testing. In order to accurately evaluate a model, we would train and test it 10 times with the test set being a different fold each time. When submitting for official scoring, the results were nearly identical to the 10-fold average we would calculate.

Due to the highly unbalanced nature of the dataset, we tried balancing the data samples. We concatenated all the patients' data together. Then, we separated the timestamps labeled sepsis from non-sepsis. We noticed the size of the non-sepsis dataset was 54 times larger than the sepsis dataset. We created 54 fully-balanced datasets and used them to train the models that we tested. This is similar to oversampling the data. For a few models this greatly improved the utility score, but for the final model it was not necessary. This will be discussed in more detail later in the paper.

One of the greatest breakthroughs we had with pre-processing was made by padding the data with the previous time-stamps' data. This was done by looking at X previous timestamps (rows) and appending those to the current row of data that was being fed into the model. If X=8, then the current timestamp would have (8+1)*40 features. In this example, the first 8 timestamps would not have enough previous data to add. Therefore in these cases, zeros were appended to fill in for the missing data. For example, the first row would have its original data plus 8*40 zeroes appended onto it. The third row would have the original, then the second, then the first rows followed by 6*40 zeroes. This method yielded a solid 2% increase in utility score.

3.2. Model Selection

Machine learning methods have been widely used in a variety of clinical and biomedical problems [8-17]. We tried a few different types of machine learning models. First, we tried using an Artificial Neural Network. This model gave us a Utility Score of 0.31, but we could not further improve it. Next, we tried using a single decision tree. With the optimal parameters, it, too gave us a maximum Utility Score of 0.31. Finally, we came across Random Forest Ensembles. There are many different methods to creating a random forest. We explored a few of those methods.

It should be noted that we created these models on Matlab version R2019a. The base function we used was "fitcensemble". From there we had to decided which method to use. We tried all available built-in ensemble methods and chose the best three based on the literature and documentation available to us: Bag, AdaBoostM1, and RUSBoost. These 3 methods were tested in great detail. Bag and AdaBoostM1 had to have the data be balanced to perform well, but RUSBoost has a built-in boosting algorithm that pseudo balances the data itself. We eventually had to make a decision on which model to go forward with, because we could not continue with all three due to time constraints. RUSBoost was ultimately decided upon due to it the nature of the algorithm and the good Utility Scores it gave.

3.3. Model Explanation

A random forest ensemble is essentially a group of simple decision trees that each take in the data and make a prediction. These decision trees are made with samples of the overall population. Each decision tree starts with a root node and is split based on some decision into 2 other nodes for binary trees. This process is continued until there is no more loss in entropy. The final layer consists of the predictions which are called leaves. The model takes all the predictions and does a majority vote to determine the final prediction. RUSBoost refers to Random Under-Sampling Boosting. Boosting takes the information from each decision tree and uses that to help create the next tree. Boosting attempts to represent every nuance in the data so it can have an accurate depiction of the bigger picture. Random Under-Sampling refers to taking random samples of the data to create each tree where each sample is balanced because the algorithm under-samples the majority class to match the size of the minority class. There

is one other state-of-the-art algorithm on the market that works in a similar way. This is Synthetic Minority Over-Sampling Technique (SMOTE). This algorithm does the opposite of RUS by over-sampling instead of under-sampling. Over-sampling makes more synthetic data of the minority class to try to balance the dataset. SMOTE was combined with AdaBoostM1 to create SMOTEBoost. The results using SMOTEBoost were compared to RUSBoost, and for this specific problem, RUSBoost performed slightly better.

3.4. Model Training

As previously stated, the model was trained with 9 folds of 36,297 patients' data. Using fitcensemble with RUSBoost, there were a few parameters that could be changed. Three were very important: the number of learners (simple decision trees), the max number of splits allowed, and the learning rate. Increasing the number of trees resulted in slower training time but usually correlated with a higher Utility Score. We noticed a plateau around 700 learners with the default 10 max number of splits. Raising the max number of splits allowed each simple learner to become more complex. We noticed that using a higher max split was beneficial when the number of trees was lower, and changing max splits at a higher number of learners did not affect the score significantly. The learning rate was adjusted a few times to gauge the effect on the score. Eventually, it was deemed that the safest place to leave it was at a value of 1.

3.5. Post-processing

The biggest post-processing done was parameter optimization. There is a slew of parameter optimization techniques available. We decided to do a grid search and try to find a local maximum. From there, we could make the grid finer and hopefully find the best score with the current model structure. Earlier it was stated that the utility score started to plateau around 700 learners. Unfortunately, when submitting a model of that size, it was never able to finish in the allotted time of 24 hours. Eventually, we discovered that similar results could be obtained with less learners if the max splits were increased. We set up the grid search to scan through 50 to 300 learner with a step of 25 as well as adjusting the max splits from 10 to 200 using a step of 10. The amount of time padding was shown to alter results, so it was stepped through the range of 6 to 14 with a step of 2. For each of these conditions, 10 folds were calculated and averaged to make sure we were getting accurate results.

Another post-processing method that has was tested is called forced conformity. This method looks at the first positive sepsis prediction and forces all the latter labels to conform to a positive label as well. This was implemented in hopes of not having a few accidental false negatives that would drastically reduce our utility score. This method was tested on several different iterations of our model but was ultimately left out due to it consistently lowering our Utility Score.

4. Results

As stated earlier, the results from RUSBoost were found to be the best. The best score that was achieved was using 700 learners at a max split of 10 with 8 padding. This gave us a Utility Score of about 0.42. Unfortunately, this model took too long to test and could not be used on this challenge. Using parameter search, we found a local maximum at 300 learners with 45 max splits and 10 pad that yielded a Utility Score of 0.4095. When submitting, this model could not finish on time either. The best model that we could submit was 150 learners with 50 max split and 8 pad. This model gave us a Utility Score of 0.399. This model yielded a prediction accuracy of 87.7% based on timestamps. A confusion matrix is shown (Tab. 2) to represent the accuracy of the model on predicting patients.

Table 2. Confusion table: the rows represent the actual number of sepsis patients and the columns represent the predicted number of sepsis patients. It can be seen that the overall accuracy (67.7%) is less than the timestamp accuracy (87.7%).

	Positive	Negative	PRED
Positive	239	64	78.9%
Negative	958	2772	74.3%
TRUE	20.0%	97.7%	67.7%

We participated in Physionet Challenge under the name SOS: Searching of Sepsis and the utility score on full test set is 0.314. Our official rank is #14.

5. Discussion

When analyzing the predictions, it was seen that the biggest penalty to our utility score was due to too many false negatives. For patients who do not develop sepsis until after 24 hours, the model captured the change to sepsis quite well. However, for patients who developed

sepsis early on in their ICU stay, the model would consistently classify them as non-septic. This problem is likely due to the fact that the model would rely too heavily on the ICU LOS parameter. Looking at the trees being created, the first split usually was based on this parameter and was splitting around the value of 25. In the future, we want to implement a 2 model approach to combat this issue. One model would not use ICU LOS to make its predictions and this model would be tested on the short LOS patients while a second model similar to the current one would predict on the rest of the patients.

Although the final Utility Score was not the highest in the competition, we feel its score still reflects the ability to capture changes in a patient's physiology that relates to sepsis occurrence. This is especially true with the patients who had longer hospitalizations.

Acknowledgments

We would like to thank Soheil Borhani, Garrett Dessinger, Will Clayton, and Trey DeLong III for useful discussions and input. This work was supported in part by the National Science Foundation under grant numbers 1661615 and 1659502.

References

[1.] Singer M, Deutschman CS, Seymour CW, Shankar-Hari M, Annane D, Bauer M, et al. The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). Jama-Journal of the American Medical Association. 2016, 315(8):801-10.

[2.] Centers for Disease Control and Prevention. What is Sepsis? [Internet]. 2019 Aug [cited 2019 Aug 31]; Available from: https://www.cdc.gov/sepsis/what-is-sepsis.html

[3.] World Health Organization. Sepsis. [Internet]. 2018 Apr [cited 2019 August 31]; Available from: https://www.who.int/news-room/fact-sheets/detail/sepsis

[4.] Mayo Clinic. Sepsis. [Internet]. 2018 Nov [cited 2019 Aug 31]; Available from: https://www.mayoclinic.org/diseases-conditions/sepsis/symptoms-causes/syc-20351214

[5.] Paoli CJ, Reynolds MA, Sinha M, Gitlin M, Crouser E. 2018 Aug [cited 2019 Aug 31]. Epidemiology and Costs of Sepsis in the United States-An Analysis Based on Timing of Diagnosis and Severity Level. Critical care medicine, 46(12), 1889–1897.

[6.] Kumar A, Roberts D, Wood KE, Light B, Parrillo JE, Sharma S, et al. Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock. Critical Care Medicine. 2006;34(6):1589-96.

[7.] Reyna MA, Josef C, Jeter R, Shashikumar SP, M. Brandon Westover MB, Nemati S, Clifford GD, Sharma A. Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019. Critical Care Medicine, (In Press).

[8.] Samaya Baljepally, Sarah Enani, Soheil Borhani, Tony Z Zhuang, Xiaopeng Zhao, Prediction of Mortality Associated with Early Onset Pneumonia in Acute Myocardial Infarction, Informatics in Medicine Unlocked, 100211, 2019

[9.] H. Xia, B.J. Daley, A. Petrie, and X. Zhao, A Neural Network Model for Mortality Prediction in ICU, Computing in Cardiology, September 9-12, 2012, Krakow, Poland.

[10.] H. Xia, N. Keeney, B. Daley, A. Petrie, and X. Zhao, Prediction of ICU in-hospital Mortality Using Artificial Neural Networks, 5th Annual Dynamic Systems and Control Conference, Palo Alto, CA, Sep 22 – 25, 2013; invited contribution.

[11.] D. Ryan, B. Daley, K. Wong, and X. Zhao, Prediction of ICU In-Hospital Mortality Using a Deep Boltzmann Machine and Dropout Neural Net, BSEC 2013 Conference: Collaborative Biomedical Innovations Program, May 21 – 23, 2013, Oak Ridge, TN

[12.] H. Xia, G. Garcia, J. Bains, D. Wortham, and X. Zhao, Matrix of Regularity for Improving the Quality of ECG, in Focus issues: Signal quality in cardiorespiratory monitoring, Physiological Measurement, 33:1535-1548, 2012

[13.] H. Xia, G. Garcia, and X. Zhao, Automatic Detection of Electrode Misplacement in ECG: A Tale of Two Algorithms, in Focus issues: Signal quality in cardiorespiratory monitoring, Physiological Measurement, 33:1549-1561, 2012

[14.] J. McBride, A. Sullivan, H. Xia, A. Petrie, and X. Zhao, "Reconstruction of physiological signals using iterative retraining and accumulated averaging of neural network models," Physiological Measurement, 32: 661-675, 2011

[15.] J. McBride, X. Zhao, T. Nichols, V. Vagnini, N. Munro, D. Berry, and Y. Jiang, Scalp EEG-Based Discrimination of Cognitive Deficits after Traumatic Brain Injury Using Event-Related Tsallis Entropy Analysis, IEEE Transaction on Biomedical Engineering, 60(1), 90-96, 2013

[16.] J. McBride, X. Zhao, N. Munro, C. Smith, G. Jicha, L. Hively, L. Broster, F.A. Schmitt, R.J. Kryscio, and Y. Jiang, Spectral and Complexity Analysis of Scalp EEG Characteristics for Mild Cognitive Impairment and Early Alzheimer's Disease, Computer Methods and Programs in Biomedicine, 114 (153-163), 2014

[17.] Joseph C. McBride, Xiaopeng Zhao, Nancy B. Munro, Gregory A. Jicha, Frederick A. Schmitt, Richard J. Kryscio, Charles D. Smith, and Yang Jiang, Sugihara Causality Analysis of Scalp EEG for Detection of Early Alzheimer's Disease, NeuroImage: Clinical, Volume 7, Pages 258–265, 2015

Address for correspondence:

Xiaopeng Zhao, Ph.D., Professor
Department of Mechanical, Aerospace, and Biomedical Engineering
University of Tennessee
Knoxville, TN 37996
Email: xzhao9@utk.edu