Cracking the "Sepsis" Code: Assessing Time Series Nature of EHR Data, and Using Deep Learning for Early Sepsis Prediction

Soodabe Sarafrazi[1], Rohini S Choudhari[1], Chiral Mehta[1], Himanshi K Mehta[1], Omid K Japalaghi[1], Patricia A Francis-Lyon[1]

[1]University of San Francisco, San Francisco, CA, USA

Abstract

On a yearly basis, sepsis costs US hospitals more than any other health condition. A majority of patients who suffer from sepsis are not diagnosed at the time of admission. Early detection and antibiotic treatment of sepsis are vital to improve outcomes for these patients, as each hour of delayed treatment is associated with increased mortality. In this study our goal is to predict sepsis 12 hours before its diagnosis using vitals and blood tests routinely taken in the ICU. We have investigated the performance of several machine learning algorithms including XGBoost, CNN, CNN-LSTM and CNN-XGBoost. Contrary to our expectations, XGBoost outperforms all of the sequential models and yields the best hour-by-hour prediction, perhaps due to the way we imputed missing values, losing signal that relates to the time-series nature of the EHR data. We added feature engineering to detect change points in tests and vitals, resulting in 8% improvement in XGBoost. The change point features may be investigated with other models.

1. Introduction

The PhysioNet/Computing in Cardiology Challenge 2019 [1] provided opportunity for researchers to develop methods to computationally detect sepsis, a major cause of mortality, using hour-by-hour electronic health Record (EHR) data. Sepsis is an actionable condition, and early detection is key to improved outcomes.

1.1. Sepsis

Sepsis is a frequent cause of death throughout the world in people of all ages. Sepsis is defined as a "life-threatening organ dysfunction due to a dysregulated host response to infection and septic shock as persisting hypotension requiring vasopressors to maintain a mean arterial pressure (MAP) of 65 mmHg or more and having a serum lactate level of greater than 2 mmol/l despite adequate volume resuscitation" [2].

Sepsis is a clinical syndrome that may accompany infection. Rather than the typical release of chemicals that combat infection, in sepsis the immune response may trigger widespread inflammation, resulting in blood clots and leaky blood vessels. This may result in impaired blood flow to vital organs, depriving them of nutrients and oxygen, which can lead to multiple organ damage.

Clinicians and healthcare professionals often find it difficult to diagnose sepsis with certainty. Signs and symptoms of sepsis are usually nonspecific, varying by patient and type of infection, making diagnosis before complications arise difficult [3].

It is crucially important to identify and diagnose sepsis at its early developmental phase before organ damage begins. Although sepsis may start with an ordinary infection, high temperature may not present. There is a need to facilitate diagnosis that is reliable given frequently confounding clinical observations.

Machine learning (ML) has been employed to detect sepsis, from ER data. An ML algorithm based on gradient tree boosting detected sepsis and severe sepsis four hours before onset using only six vital signs and their changes over time [5], achieving AUROC of 0.96 and 0.85 respectively. Also, sepsis was predicted in advance by a Cox proportional hazards model, using diagnosis of sepsis as the time-to-event outcome. This model produced the TREWscore from features readily available to clinicians, enabling identification of patients having sepsis a median of 28.2 [interquartile range (IQR), 10.6 to 94.2] hours before diagnosis. We are unaware of hour-by-hour advance prediction of sepsis prior to this challenge.

1.2. Gradient Boosted Trees

Gradient boosted trees exist within the context of decision tree and ensemble tree algorithms. Single decision trees may be used to successfully classify datasets that are large, contain redundant and missing data, and are not linearly separable. However, they tend to be high-variance models prone to overfitting [7].

To reduce variance, bagging [7] was developed, where subsamples are used to grow an ensemble of trees, each fit to a different dataset drawn from the random

subsampling process. The random forest approach [8] further reduces variance by de-correlating bagged trees by randomly selecting a subset of variables for splitting.

Gradient boosted trees [9-12], differ in that trees are grown sequentially, fitting the residuals of the previous models, producing an additive model that learns from previous error. XGBoost is a fast, high performance implementation of this algorithm [13].

As applied to the sepsis problem, the XGBoost algorithm may be explained by figures 1 and 2. Tree 1 is an example of a tree that classifies patients with/without sepsis based on heart rate and body temperature. But is prone to false positives. Tree 2 is fit to the residuals of Tree 1, adding to the model the ability to distinguish, on the basis of systolic blood pressure, a patient with unfavorable platelet count and temperature, but who is not septic.

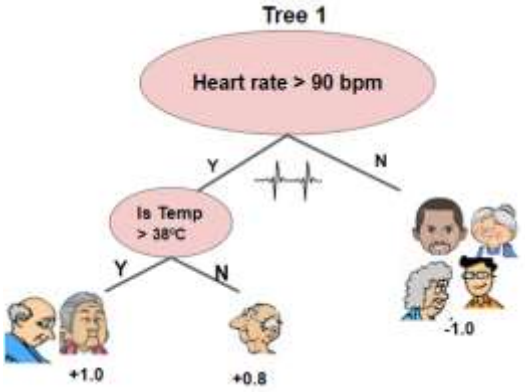

Figure 1. XGBoost: sepsis class by HR, temperature

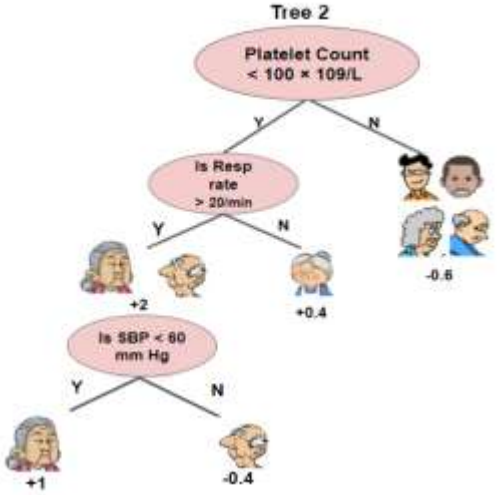

Figure 2. XGBoost: sepsis class based on platelet count, respiration and systolic blood pressure is added to Tree 1.

1.3. LSTM, CNN-LSTM, CNN

Long short-term memory (LSTM) is an artificial recurrent neural network, (RNN) architecture. It is well-suited to classifying and making predictions in time series data. A common LSTM unit is composed of a cell and three gates. The cell remembers values over arbitrary time intervals and gates regulate the flow of information [14].

Convolutional Neural Networks (CNNs) came to prominence for use in image classification. The CNN learns to recognize sub-images useful for classification. Similarly, CNN may be trained to extract useful features from non-image sequential data.

To add memory to a CNN, one or more LSTM layers may be added to the model (see figure 3). The CNN layer extracts features from the dataset [15]. Evaluation of multiple models showed that a simple CNN architecture outperforms canonical recurrent networks such as LSTMs across a diverse range of tasks [16]. Here we investigate performance of all above-mentioned architectures.

2. Data

The PhysioNet-CinC Challenge data was sourced from the ICU of three different hospitals. Each pipe-delimited text file contained a single patient's data, with hour-by-hour values in rows. There were 41 columns defining vital signs, laboratory values, demographics and outcome SepsisLabel, defined on the challenge website.

Of total 8 features in vital signs, 4 features (HR, O2Sat, SBP & MAP) have <15% missing values, 3 features (Temperature, DBP, Resp) have 15-90% missing data and remaining have >90%. Of total 26 features in lab values, Serum-Glucose is the only feature with <90% missing value, remaining have >90% missing data. Of total 6 features in demographics, 2 features (Unit1 & Unit2) have 15-90% missing values and remaining are fully populated (Refer to Table 1).

About 7.27% of patients had sepsis [2932/40,336]. Of all the sepsis patients 14.53 % [426/2932] patients reported sepsis in the first hour.

2.1. Utility Function Explained

Each PhysioNet-CinC 2019 challenge entry was graded for its binary classification performance using the function depicted in Figure 3 [6]. A score is calculated for each patient s at each hour in the ICU t.

The top panel depicts hourly scoring for patients who at some point have sepsis: True Positives (TP, red) or False Negatives (FN, blue). TPs are assigned a small penalty for each hour of detection prior to 12 hours before onset of sepsis. The red line shows points assigned for detection within the 15-hour window preceding onset, with maximum points at detection 6 hours before onset ($t_{optimal}$). Detection later than 3 hours after the onset of sepsis is deemed late; no points are rewarded for those hours. FNs accrue negative points after ($t_{optimal}$), with highest penalty for every hour after onset that sepsis goes

undetected.

The bottom panel depicts hourly scoring for patients who never develop sepsis: False Positives (FP, orange) or True Negatives (TN, green). TN accrue no points, while each hour of FP predictions accrues a small penalty.

The score for a classifier is computed by summing the hourly scores U(s,t), then normalizing based on maximum possible score of 1 (correctly predicting every hour of every patient) and minimum of zero [6].

3. Methods

Machine learning (ML) algorithms were implemented using open-source libraries [17-19].

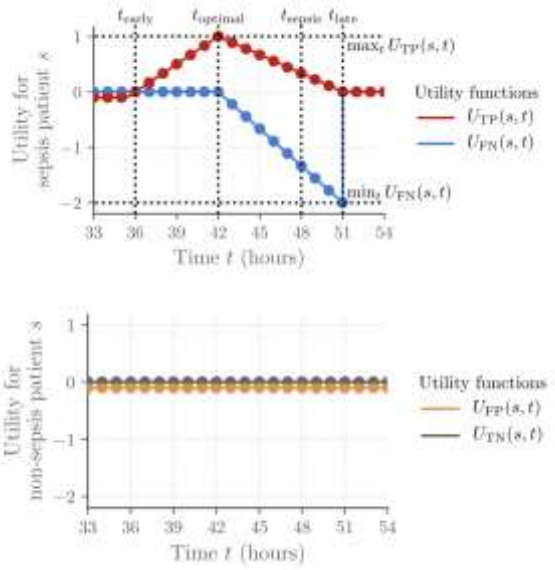

Figure 3. from the PhysioNet Challenge Organizers: utility function for sepsis patient (upper) with t_{sepsis} = 48 and a non-sepsis patient (lower) [6].

3.1. Data Preparation

As described in section 2, the data was sparse, with most variables missing values for >90% of the rows. For the 5 ML models reported in table 2, missing data was forward filled, imputing initial values with typical values for healthy people.

For further development of the most successful model feature engineering was employed. To capture signal relevant to doctors' suspicions of sepsis, variables were created to reflect change points in lab tests, with value 2 assigned for a newly ordered test, 1 for a non-expired test, and 0 for either an expired or never-ordered test.

3.2. Sequential Neural Networks

The common factor in all our sequential Neural Network (NN) models is that prediction at each point in time is not only a function of information at that time but also of a sequence of the preceding 5 hours. Therefore, information of 6 consecutive hours is used for each hour-by-hour prediction.

3.2.1. CNN-LSTM

In this approach we employed 2 layers of CNN before stacked layers of bidirectional LSTM (BDLSTM). These layers were followed by a dense layer of 2 nodes for classification. Here CNN layers of multiple kernels with sizes of 4 and 2 performed feature extraction. These extracted features were then fed into the LSTM models for further analysis. (Figure 4). The goal was to minimize the binary cross-entropy loss function summed over all outputs at the end.

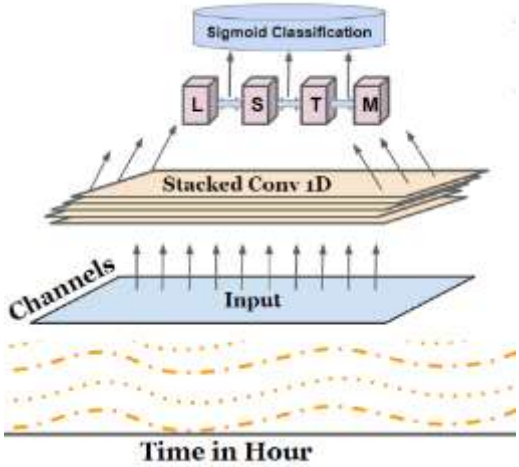

Figure 4. Stacked CNN-BDLSTM architecture.

3.2.2. CNN

LSTM is a powerful ML method, however it has some drawbacks [16]. The major one is the problem of overfitting we controlled by early stopping. Therefore, we investigated the performance of CNN layers alone. CNN followed by a dense layer of 2 can also take into account the sequential nature of the data.

3.2.3. NN-XGboost

Further improvements were made to NN models by replacing the sigmoid layer with an XGboost model for classification (Figure 5). NN models used were CNN-LSTM and CNN. XGBoost used the features that were extracted by NN to make a binary classification.

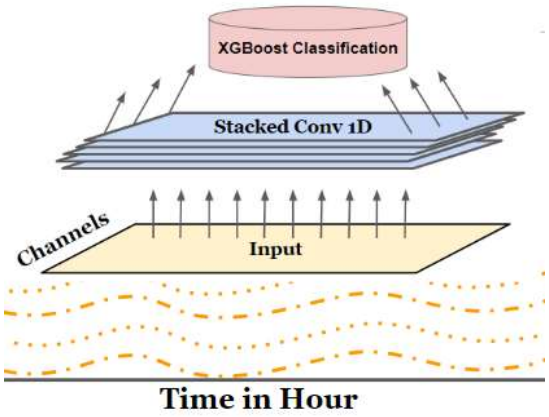

Figure 5. CNN-BDLSTM-XGBoost architecture.

3.3. XGBoost

As the performance of stacked NN-XGBoost model was promising, we decided to investigate the performance of a simple XGboost model by itself. Note that here the format of input is totally different from sequential NN models: sepsis status of the patients at each point of time is made using only the information of that time.

4. Results

We have modeled hourly EHR data utilizing a variety of approaches as we sought to unlock the times series nature of the Physionet-CinC data to output a sequence of hourly sepsis predictions from multiple input series, each the values of a lab test or a vital sign throughout the patient's ICU stay. Challenges to accurate prediction included sparse data (25 of 26 lab tests had >90% missing values), patients whose sequential data is of varying length (ranging from 8 hours to 230 hours) having non-comparable start times, and an imbalanced dataset. Results of different models are shown in Table 1. Here the normalized utility is from scoring the test set obtained by a stratified train-test split of the Physionet data (4% testing).

The poorest performer was CNN-LSTM, a recurrent network utilizing CNN for feature selection. CNN by itself outperformed the CNN-LSTM architecture. Although recurrent networks are often the first choice for modeling sequential data, LSTM is prone to overfitting, and there were only 2932 sepsis patients. Also, in forward filling so many values, many of which were set to typical values for healthy people, perhaps we lost the time-series nature of the data, and therefore the advantage that LSTM would have conferred.

CNN had more success, showing that it was able to extract useful features from the sparse EHR data. Performance improved, as it did for CNN-LSTM, when these extracted features were passed to an XGBoost model in place of the CNN and CNN-LSTM sigmoid output node. However, the best model using the input variables was the XGBoost model alone, outperforming all sequential NN models. Deep learning models would be expected to be disadvantaged by the relatively small number of observations in the dataset given the complexity of their models.

The best model was the featured engineered XGBoost model, where change point information on all lab tests was captured, adding .05 to the normalized utility (an 8% improvement) over XGBoost alone. This addressed the shortcomings of the way we imputed the many missing values, as the engineered features contained zero for any test that had never been ordered or was expired. Also, the point at which doctors' suspicions led to the ordering of a new test was captured in these columns.

5. Conclusion

In this study we investigated the performance of some sophisticated and powerful machine learning algorithms as applied to prediction of sepsis from hourly EHR data. As the nature of sepsis diagnosis is sequential and at each point of time it is beneficial to consider previous time step lab values and their rates of change, we expected that sequential models would outperform XGBoost. However, XGBoost, a much faster model, outperformed all of the sequential models. This may be due to the way we imputed missing values, which was to replace missing values by either forward-fill or typical values of the variables. This was addressed by adding added features engineered to detect change points in tests and vitals (when ordered, when expired or never-ordered) resulting in 8% improvement in XGBoost. Future work could include running sequential models with the engineered features.

Table 1. Performance of different models

Model	Normalized Utility
CNN-LSTM	0.32
CNN	0.38
CNN-LSTM-XGboost	0.39
CNN-XGboost	0.48
XGBoost	0.49
XGBoost with expire	0.54

References

[1] Early Prediction of Sepsis from Clinical Data: the PhysioNet/Computing in Cardiology Challenge 2019 [Internet]: https://physionet.org/challenge/2019.

[2] Singer M, Deutschman CS, Seymour CW, Shankar-Hari M,

Annane D, Bauer M, Bellomo R, Bernard GR, Chiche JD, Coopersmith CM, Hotchkiss RS, Levy MM, Marshall JC, Martin GS, Opal SM, Rubenfeld GD, van der Poll T, Vincent JL, Angus DC. The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). JAMA. 2016 Feb 23;315(8):801-10

[3] National Guideline Centre (UK). Sepsis: Recognition, Assessment and Early Management. London: National Institute for Health and Care Excellence (UK); 2016 Jul. (NICE Guideline, No. 51.)

[4] Kumar A, Roberts D, Wood KE, Light B, Parrillo JE, Sharma S, Suppes R, Feinstein D, Zanotti S, Taiberg L, Gurkha D, Kumar A, Cheang M, Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock. Crit Care Med. 2006 Jun;34(6):1589-96. PubMed PMID: 16625125

[5] Mao Q, Jay M, Hoffman JL, Calvert J, Barton C, Shimabukuro D, Shieh L, Chettipally U, Fletcher G, Kerem Y, Zhou Y. Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and ICU. BMJ open. 2018 Jan 1;8(1):e017833.

[6] KE Henry, DN Hager, PJ Pronovost, S Saria A targeted real-time early warning score (TREWScore) for septic shock. Science translational medicine. 2015 Aug 5;7(299).

[7] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. New York: Springer series in statistics; 2001.

[8] Breiman L. Random forests. Machine learning. 2001 Oct 1;45(1):5-32.

[9] Friedman JH. Greedy function approximation: a gradient boosting machine. Annals of statistics. 2001 Oct 1:1189-232.

[10] Friedman JH. Stochastic gradient boosting. Computational statistics & data analysis. 2002 Feb 28;38(4):367-78.

[11] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics. 2000;28(2):337-407.

[12] Johnson R, Zhang T. Learning nonlinear functions using regularized greedy forest. IEEE transactions on pattern analysis and machine intelligence. 2014 May;36(5):942-54.

[13] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining 2016 Aug 13 (pp. 785-794). ACM.

[14] Sepp Hochreiter and Jürgen Schmidhuber, Long Short-Term Memory, Neural Computation Volume 9, Issue 8, November 15, 1997, p.1735-1780.

[15] CNN-RNN: A Unified Framework for Multi-Label Image Classification, Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, Wei Xu; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2285-2294.

[16] An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling Shaojie Bai, J. Zico Kolter, Vladlen Koltun.

[17] https://www.tensorflow.org/.

[18] F.Cholletandothers. Keras. 2015.

[19] https://xgboost.readthedocs.io/en/latest/.

Address for correspondence:

Patricia Francis-Lyon
2130 Fulton Street, San Francisco, CA 94117-1080
pafrancislyon@usfca.edu