A Comparison of Machine Learning Tools for Early Prediction of Sepsis from ICU Data

Po-Ya Hsu[1], Chester Holtz[1]

[1] University of California, San Diego, La Jolla, USA

Abstract

We explore the efficacy of modern machine learning methods for the task of modeling sepsis progression. We applied a novel imputation and feature selection scheme based on signal processing technology and our medical expertise. We compared the performance of several approaches including neural networks, sparse quantile regression, and baseline classification algorithms such as random forest and SVMs. Among all the experimented methods, CNN-LSTM neural network performed the best with the full test utility score of the challenge being 0.076. We conclude that the application of neural network, random forest, sparse quantile regression, neighborhood algorithms, and naive Bayes classifiers yields superior performance with respect to accuracy, sensitivity, and specificity. [Team: Sepsis ReSepsion]

1. Introduction

Early detection of Sepsis is vital for effective treatment, and each hour left untreated increases the chance of death, especially in the ICU [1, 2]. The task of detecting sepsis early is typically formulated as a multi-channel temporal classification task. Clinical data is commonly sampled irregularly, thus often requiring a set of hand-crafted preprocessing steps, such as binning, carry-forward imputation, and rolling means prior to the application of a predictive model. However, such naive imputation schemes lead to a loss of data sparsity, which may carry crucial information in this context. In light of these challenges, we propose to leverage spline-based interpolation models to imputation of unobserved samples and normalize the timescale of the variables.

1.1. Prior Work

Effective models for early sepsis prediction are desired since sepsis is a life-threatening condition. According to [1, 2], each one-hour delay of treatment of a case leads to an increase of $\sim 7\%$ in the mortality rate. The 2019 PhysioNet Computing in Cardiology (CinC) challenge seeks to develop automated methods for early sepsis detection based on the ICU data.

There is a maturing body of prior work on data-driven sepsis detection methods applying machine learning algorithms [3, 4]. In [3] and [4], unique machine learning approaches have been developed to achieve outstanding sepsis prediction. [5] evaluates two techniques for early prediction of sepsis: a temporal convolutional network, and a KNN-based approach leveraging Global Alignment Kernels. Our work is also similar to that of [6] who adopts a neural network classifier augmented with a multi-task Gaussian process regression layer to interpolate vital signs.

1.2. Contribution

We evaluated several baseline machine learning classification algorithms and deep learning techniques in this study. To resolve the issue of missing and imbalanced data, we developed an imputation and feature selection scheme for ICU data. In summary, our contributions include

- We proposed a method to handle irregularly sampled data via a spline-based imputation algorithm.
- We evaluated the performance of a variety of statistical machine learning algorithms on engineered features
- We proposed a novel deep learning-based video classification framework for the task of sepsis prediction

2. Method

In this section we review the technical details of the algorithms we applied to this problem and detail our analysis and numerical results. We decompose the prediction problem into stages as in Fig 1. We evaluated several different classes of predictors on our selected and learned features. In this section, we will briefly summarize and discuss the relative performance of each algorithm we applied and address their advantages and disadvantages in the context of sepsis detection.

ISSN: 2325-887X DOI: 10.22489/CinC.2019.206

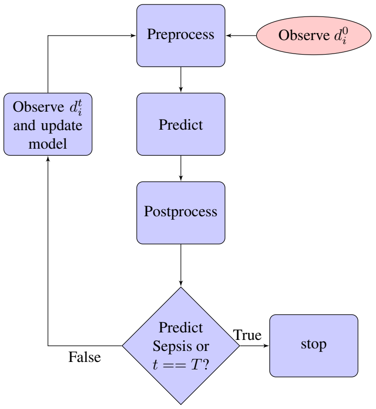


Figure 1: Sepsis prediction pipeline for patient i

2.1. Formulation

We frame the problem of early detection of sepsis as a multivariate time series classification problem. Given a new patient encounter, the goal is to continuously update the predicted probability that the encounter will result in sepsis, using all available information up until that time. We follow the general framework from [6]. Given a dataset $\mathcal{D} = \{d_i\}_{i=1}^n$ consisting of N independent patient encounters, each patient encounter d_i, is described by a set of covariates and covariate vectors. Baseline covariates $\mathbf{b_i} \in \mathbb{R}^{B\times 1}$ are available on hospital admission and consist of demographic information including gender and age. Lab covariates $\mathbf{l_i} \in \mathbb{R}^{L\times t}$ and vital-sign covariates $\mathbf{v_i} \in \mathbb{R}^{V\times t}$ are observed online for each time-step $0 \leq t \leq T$ where $t = 0$ denotes hospital admittance time. The goal is to predict, for each time t the likelihood that a particular patient d_i has or will contract sepsis.

It is important to note that the full duration time T may vary over patients in the dataset, and that $\mathbf{l_i}$ and $\mathbf{v_i}$ are only partially observed due to the irregular sampling procedure employed for each clinical variable. Additionally, each encounter in the training set is associated with a binary vector $\mathbf{o_i} \in \{0,1\}^T$ denoting whether or not the patient has acquired sepsis at each time step. Thus, the data for a single patient encounter can be summarized as a 4-tuple: $d_i = \{\mathbf{b_i}, \mathbf{l_i}, \mathbf{l_i}, \mathbf{o_i}\}$. For brevity, we adopt the notation $d_i^t = \{\mathbf{b_i}, \mathbf{l_i^t}, \mathbf{l_i^t}, \mathbf{o_i^t}\}$ to denote the observed sequence of variables at time t.

2.2. Data preprocessing and experimental setup

The training dataset for the Challenge consisted of $40,336$ subjects [7]. For each subject, the data included demographics, vital signs, laboratory values, onset time of sepsis, and sepsis label. We evaluated several approaches to preprocess the data. In particular, the prevalence of missing values and a tailed sequence-length distribution were the primary issues that affected the design of our preprocessing pipeline.

We train our method with k-fold cross validation: 80% of the full dataset, setting aside 10% as a validation set to select hyperparameters and a final 10% for testing.

2.3. Imputation and normalization

Prior to classification with our machine learning-based models, we logarithmically transformed continuous variables to reduce the influence of outliers and z-score-standardized each column.

To tackle the irregular feature sampling rate, we applied piecewise cubic Hermite interpolation polynomials [8]. After normalizing the timescale via interpolation, We filled remaining missing values - e.g. unobserved columns - with zeroes corresponding to the standardized empirical mean post z-score preprocessing.

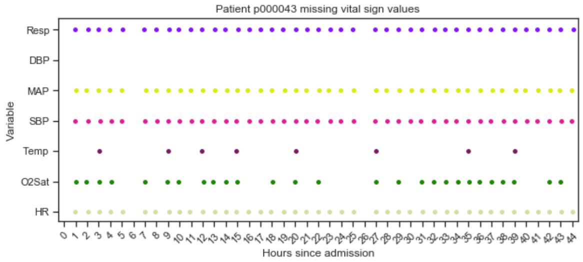


Figure 2: An example patient encounter that highlights the irregular sampling rates of vital sign variables

2.4. Feature Selection

In our machine learning based algorithms, statistical features are selected by deriving various statistical metrics that represent different local and global aspects of the underlying signal. We compute sliding-window features for a variety of different window sizes to capture local and global descriptors of the ICU data. In total, we compute 18 features including moment statistics about the waveform distribution (mean, variance, skewness, kurtosis) as well as quantile information.

In our neural network methods, different from the aforementioned machine learning methods, all 40 variables are exploited. The input is the processed data involving imputation and interpolation.

2.5. Classification

In this section, we summarize the details of a subset of methods we applied. We evaluated offline prediction via a variety of algorithms: Linear Least Squares (LLSE), Naive Bayes, Support Vector Regression/Machine (SVR/M), Hidden Markov Models (HMM), Random Forests, and LS-Gradient Boosting. We adopt LLSE as a baseline. Furthermore, we experimented with both online-sparse quantile regression (SQR) and online-Lasso - as online learners.

2.5.1. KNN-GAK

We present a simple, but effective algorithm based on k-nearest neighbors. Dynamic Time Warping (DTW) [9] is often applied in conjunction with KNN for time series classification. This method is known to exhibit highly competitive predictive performance on sequence classification tasks, and has been previously applied to the sepsis detection task [5]. As opposed to many other off-the-shelf time-sequence classifiers, KNN-DTW can intrinsically handle variable-length time series.

A major drawback of the DTW distance is that it is not rigorously a distance and is known not to be negative definite since it does not satisfy the triangle inequality, and as a result cannot be used to define a positive-definite kernel, contradicting most of the mathematical foundations of kernel methods. To resolve this issue, we leverage fast Triangular Global Alignment Kernels (GAKs) [10] which have been shown to be both faster and more efficient in classification tasks compared to other kernels based on DTW [10]. To the best of our knowledge, we are the first to propose the use of GAK-based KNN for sepsis detection.

We evaluate an extension of KNN-GAK for classification of multivariate time series, KNN-GAK-E. We address the multivariate nature of our setup by computing the GAK distance kernel (an $N \times N$ matrix containing the pairwise distances between all patients) for each time series channel separately. Each distance matrix is subsequently used for training a k-nearest neighbor classifier. A weighted ensemble is learned by combining all per-channel classifiers, a multi-channel classifier, and the baseline variables with a logistic regression predictor on a held out validation set. In all cases, KNN-GAK-E outperforms KNN-GAK and KNN-DTW on sepsis detection - achieving an f1 score of 0.58 and AURC of 0.236. We adopt the implementations of DTW and GAK from TSLearn [11] and classification algorithms from SKLearn [12]. The primary limitation of our neighbor-based methods is their poor scalability with respect to both memory and runtime, making their application for online-realtime prediction of sepsis limited.

2.5.2. Neural Network Algorithms

We also evaluated the performance of a deep learning-based algorithm. In particular, we constructed a composite comprised of a cnn-based encoder and lstm-based decoder. This dual-architecture has seen success in sequence processing tasks - e.g. video processing [13]. We provide an image of our architecture in Figure 3.

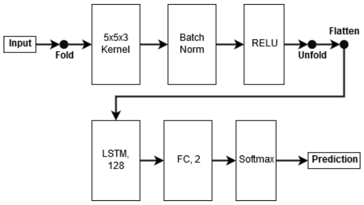
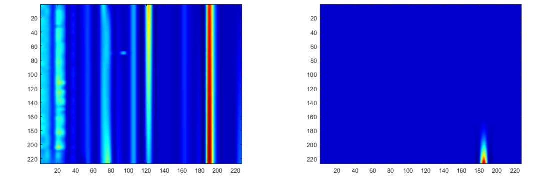


Figure 3: CNN-LSTM architecture for sepsis detection



(a) Patient with no sepsis (b) Patient diagnosed of sepsis

Figure 4: Demonstration of the input to our neural network

Generation of the input for neural networks is accomplished in two steps and requires a color map. Figure 4 illustrates the inputs of patients diagnosed of sepsis and not to the neural network model. The first step of the input generation is transformation $\phi(x_i) = \frac{x_i - x_{min}}{x_{max} - x_{min}}$ defined over the entire input. In the second step, the normalized data are reshaped to a squared color-coded image as $h(x_{norm}) = jet(\lfloor x_{norm} \times 99 \rfloor)$, in which x_{norm} is the scaled element and *jet* the color map defined as MATLAB command *jet(100)*. The size of the image is $N \times N \times 3$ for spatial dimension N.

A central capability of intelligent systems is the ability to continuously build upon previous related experiences to enhance the learning of new tasks. Pretraining is one paradigm that interprets transfer learning as the sharing of general information about the composition of natural images - e.g. the typical combinations of low-level visual primitives such as edges or curves. For trained neural networks, such information is typically encoded in the early layers. We pretrain our CNN encoder on the ImageNet dataset [14], and fine tune the LSTM and output layers.

The input to the CNN-LSTM network is a stack of images of dimension $80 \times 80 \times 3$, with each image corresponding to a 12-hour period of time. The output of the network is the probability of the occurrence of sepsis within this timeframe. Our encoder is a CNN with three $5 \times 5 \times 1$ filters for each color channel followed by the operations of batch normalization and ReLU. The output of our encoder is a single 19200×1 vector representation of a 12-hour period. Our decoder is composed of a single layer LSTM

Table 1: Performance of Sepsis Detection Classifiers

Classifier	Accuracy	Sensitivity	Specificity
Naive Bayes	84%	25%	90%
Random Forest	91%	9%	99%
SQR	60%	66%	58%
XGboost, SVM GP, SGD LR, HMM	$\approx 90\%$	$\leq 5\%$	$\approx 99\%$
KNN-DTW	46%	71%	29%
KNN-GAK	48%	74%	31%
KNN-GAK-E	69%	62%	71%
CNN-LSTM + Transfer	90%	35%	95%
CNN-LSTM	95%	40%	75%

containing 128 hidden units. Classification is performed with a single-layer MLP with softmax activation.

2.6. Evaluation

We evaluate the performance of our algorithms with a variety of metrics including accuracy, sensitivity, specificity of final and intermediate predictions. Furthermore, the challenge score and running time are considered.

3. Results and Analysis

We conclude that neural network, random forest, sparse quantile regression, naive Bayes, and the neighborhood methods offer superior performance with respect to accuracy, sensitivity, and specificity. Table 1 displays the classification performance of each classifier. Random forests offer deceptively strong performance on average in comparison to the other algorithms, however the sensitivity is quite low. In contrast, sparse quantile regression outperforms other algorithms for sepsis detection and is robust to over fitting. Naive Bayes demonstrates balanced performance. Other classifiers show reduced capability for sepsis detection with $\leq 5\%$ sensitivity.

3.1. Challenge Scores

Our best challenge score is 0.076, which is achieved with our LSTM-CNN neural network. [Team: Sepsis Re-Sepsion]

4. Conclusion and Future Work

Our neural network is effective at learning complex patterns implicit in clinical data and associating them with sepsis, outperforming our baselines by a significant margin. Our future work will include a more in-depth exploration of the developed techniques and integration of additional engineered features to this task.

References

[1] Ferrer, et al. Empiric antibiotic treatment reduces mortality in severe sepsis and septic shock from the first hour. Critical care medicine 04 2014;42.

[2] Kumar A, 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):1589–1596.

[3] Calvert JS, et al. A computational approach to early sepsis detection. Computers in biology and medicine 2016;74:69–73.

[4] Nemati S, et al. An interpretable machine learning model for accurate prediction of sepsis in the icu. Critical care medicine 2018;46(4):547–553.

[5] Moor M, et al. Temporal convolutional networks and dynamic time warping can drastically improve the early prediction of sepsis. CoRR 2019;abs/1902.01659. URL http://arxiv.org/abs/1902.01659.

[6] Futoma, et al. Learning to detect sepsis with a multitask gaussian process rnn classifier. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17. 2017; 1174–1182.

[7] Reyna M, et al. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. 2019; .

[8] Carlson R, Fritsch F. Monotone piecewise bicubic interpolation. SIAM journal on numerical analysis 1985; 22(2):386–400.

[9] Keogh E, Ratanamahatana CA. Exact indexing of dynamic time warping. Knowl Inf Syst March 2005;7(3):358–386. ISSN 0219-1377.

[10] Cuturi M. Fast global alignment kernels. In Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11. ISBN 978-1-4503-0619-5, 2011; 929–936.

[11] Tavenard R, et al. tslearn: A machine learning toolkit dedicated to time-series data, 2017. https://github.com/rtavenar/tslearn.

[12] Pedregosa F, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 2011; 12:2825–2830.

[13] Ullah A, et al. Action recognition in video sequences using deep bi-directional lstm with cnn features. IEEE Access 2018;6:1155–1166. ISSN 2169-3536.

[14] Deng J, et al. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09. 2009; .

Address for correspondence:

Po-Ya Hsu
p8hsu@eng.ucsd.edu
9500 Gilman Drive, La Jolla, CA 92093