Sepsis Onset Prediction Applying a Stacked Combination of a Recurrent Neural Network and a Gradient Boosted Machine

Matthieu Scherpf[1], Miriam Goldammer[1], Hagen Malberg[1], Felix Gräßer[1]

[1] Institute of Biomedical Engineering, TU Dresden, Germany

Abstract

Early detection and treatment of sepsis is of utmost importance concerning sepsis outcome and costs. However, revealing patterns in vital signs and laboratory measurements which facilitate reliable prediction of sepsis onset remains challenging. Especially exploiting the time series characteristic of those measurements is expected to play a major role concerning successful sepsis prediction. Within this work, we propose a stacked combination of a recurrent neuronal network (RNN) and a light gradient boosted machine (LGBM) to target the objective of sepsis onset prediction. Here, 8 vital signs, 26 laboratory measurements and 3 demographic parameters are included as input to our classification model. Our last running model achieved a utility score on full test set of 0.114 (TU Dresden - IBMT).

1. Introduction

This work addresses Early Prediction of Sepsis from Clinical Data – the PhysioNet Computing in Cardiology Challenge 2019. For detailed information on the challenge, please read and cite the upcoming publication [1]. The authors of this paper form the team TU Dresden - IBMT as referenced in the challenge ranking.

To this day, sepsis exhibits a high mortality rate if not treated appropriately in time [2]. As a consequence, reliable sepsis onset prediction is an active field of research and various approaches have been proposed in the literature [3, 4]. Also the Physionet Challenge 2019 targets development and evaluation of a prediction algorithm based on a comprehensive dataset. Within the challenge, a predefined utility score is to be optimized which rewards true positive and penalizes both false positive and false negative predictions [1].

The provided dataset consists of 40336 electronic health records with 2932 records that exhibit sepsis at a given point in time. Hereafter, we use the terms *sepsis record* and *non-sepsis record* to distinguish between the two groups. Each record comprises routine vital signs, laboratory measurements and demographic data of one in-patient stay acquired from an intensive care unit. Concerning the gold standard, the latest sepsis definition from 2016 [2] is used.

Within this contribution we propose a stacked composition of two machine learning algorithms and additional feature reinjection as detailed in the following.

2. Methodology

We propose a stacked combination of a recurrent neural network (RNN) and an ensemble of decision trees trained with the light gradient boosted machine (LGBM) algorithm [5]. The complete pipeline from raw data input to the final classification output is shown in figure 1. Here, *raw data* denotes the data as it was provided for the challenge. This raw data is characterized by a large proportion of missing values, in particular for laboratory measurements.

The combination of an RNN and a decision tree ensemble is uncommon and rarely seen. We decided to realize such an approach for several reasons. First of all, we wanted to exploit the RNN's well-known capability of discovering time dependent patterns. Hence, we only use dynamic parameters for training the RNN. We assume these patterns to be of utmost importance for the early detection of an upcoming sepsis manifestation. Secondly, we observed that the RNN has the greatest issues to distinguish the classes when sepsis onset occurs early in the record (see sec. 2.1.3 for details). Therefore, we decided that a machine learning algorithm stacked onto the RNN potentially improves the classification performance. This is because on the one hand it reconsiders short-term developments but on the other hand it also takes the long-term developments encoded in the RNN's output into account. Additionally, we reinject chosen parameters as we expect this to further improve discovering short-term developments. We chose an LGBM as we expect it to be a more light weight classifier, i.e. less data is necessary for training than compared to, for example, a multilayer perceptron to reach similar performance.

We used 4-fold stratified cross validation for the evaluation of our model. Additionally during evaluation of each

 ISSN: 2325-887X DOI: 10.22489/CinC.2019.089

fold, one fourth of the training partition was again held out for validation, resulting in $\frac{9}{16}$, $\frac{3}{16}$ and $\frac{4}{16}$ of the original data for training, validation and testing, respectively.

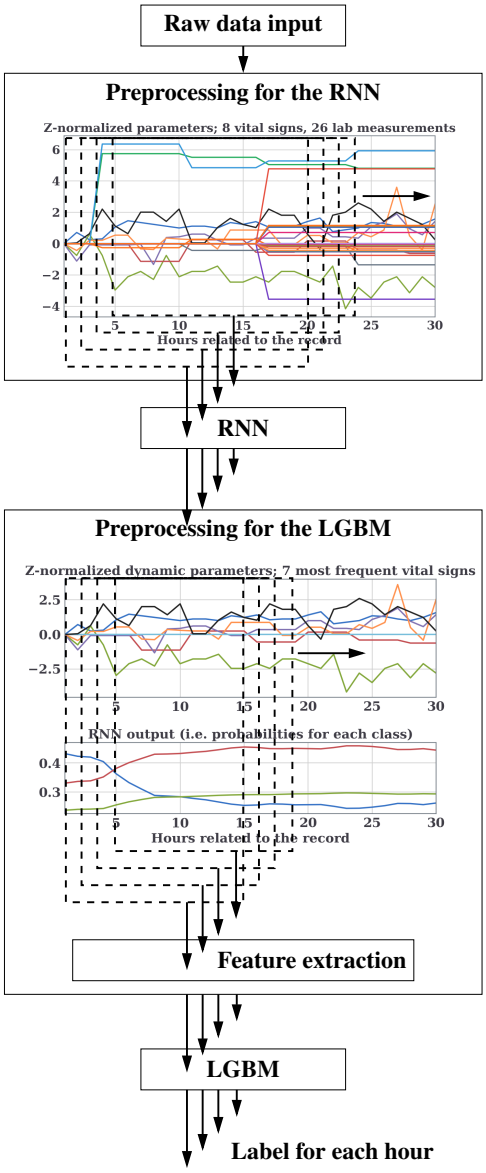


Figure 1: Processing pipeline of the proposed stacked combination of classifiers.

2.1. First model part: the recurrent neural network (RNN)

Two classes are defined for the calculation of the utility score for performance assessment: *sepsis hours (class 2)* and *non-sepsis hours*. As we expect the model to perform better according to a more distinct separation of the original data, we split the non-sepsis hours into *non-sepsis hours of non-sepsis records (class 0)* and *non-sepsis hours of sepsis records (class 1)*. Hence, we use three classes for training the RNN.

2.1.1. Preprocessing steps for the RNN

According to the 3 defined classes, we separated each sepsis record into the non-sepsis - if available, as some records only contain sepsis hours - and the sepsis part. After the occurrence of the first sepsis label, there are no more than a maximum of 10 sepsis hours until the end of the record. To prevent splitting the record potentially in between a pattern indicating sepsis manifestation, we shifted the originally defined labels backwards by a maximum of 18 hours - depending on data availability - before the first sepsis label occurred.

As we use only dynamic parameters for training the RNN, the input consists of 34 parameters, i.e. 8 vital signs and 26 laboratory measurements.

We perform a z-normalization for all input parameters of the network. Missing values are imputed according to a last observation carried forward (LOCF) strategy. Samples containing missing values which precede the first valid value are padded with zeros.

A sliding window with 20 hours of measurements is shifted over the input time series, aiming at revealing patterns in this multivariate signal sequence. Based on the patterns found, the network is intended to classify the most recent hour of this window as being one of the three defined classes.

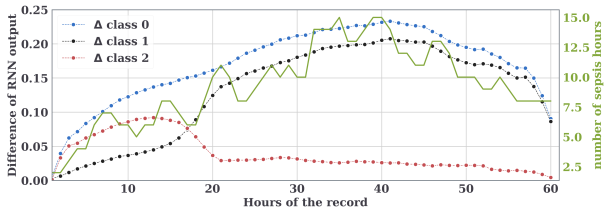
2.1.2. Setup of the RNN

Our RNN consists of 4 hidden layers. The first and second use the gated recurrent unit architecture (GRU) [6] with 64 and 32 units, respectively. The third and fourth are fully connected layers with 32 and 16 units, respectively. We performed a hyperparameter optimization using the validation set for the learning rate, the number of hidden layers and the number of neurons. Remaining network parameters such as batch size etc. were chosen based on our experience from preceding work [4].

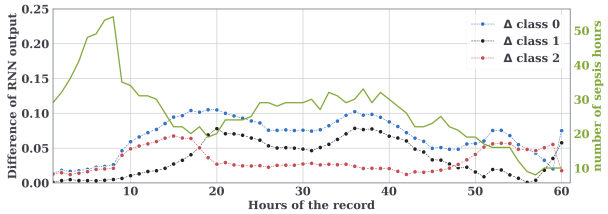
2.1.3. Analyzing the RNN output

Figure 2 shows the mean of the differences between sepsis records and non-sepsis records of the network's output for the first 60 hours for the validation data. We split the records into 3 groups according to a threshold for the mean of the RNN output for class 2. The smaller the differences, the more difficult it is to differentiate between the classes.

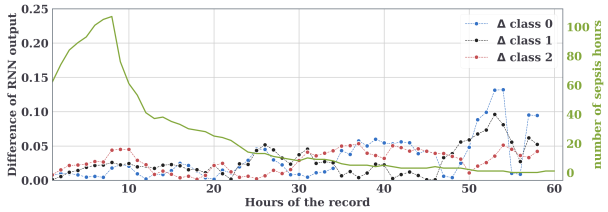
We made two major observations. Firstly, the differences of the network output for short recorded sequences, i.e. sepsis appears quite early in the record, is small (see



(a) $0 \leq mean(class\,2) < 0.15$; based on 185 sepsis records and 3908 non sepsis records



(b) $0.15 \leq mean(class\ 2) < 0.3$; based on 208 sepsis records and 2582 non sepsis records



(c) $0.3 \leq mean(class\ 2) \leq 1$; based on 157 sepsis records and 523 non sepsis records

Figure 2: Differences between sepsis and non sepsis records of the RNN output for each class for one validation set; See section 2.1 for the definition of $class0$, $class1$ and $class\ 2$.

Sequence	*Feature*
t_n	Parameter values = Pv
$[t_{n-1}, t_n]$	Mean(Pv), First derivative = Fd
$[t_{n-2}, t_n]$	Mean(Pv), Mean(Fd), Second derivative = Sd, Max(Pv), Min(Pv)
$[t_{n-3}, t_n]$	Mean(Pv), Mean(Fd), Mean(Sd), Max(Pv), Min(Pv)
$[t_{n-4}, t_n]$ $[t_{n-9}, t_n]$ $[t_{n-14}, t_n]$	Mean(Pv), Mean(Fd), Mean(Sd), Max(Pv), Min(Pv), Standard deviation(Pv)

Table 1: Handcrafted features which represent the input of the LGBM; n denotes one hour of the record. Thus, the interval $[t_{n-14}, t_n]$ represents one 15 hour window. We extracted the features for each of the most frequent vital signs and the RNN output.

figure 2 (b) and (c)). This seems plausible as the network is trained on discovering patterns in longer sequences because we use windows with a length of 20 hours. Conversely, the network seems to find patterns to distinguish between the classes with increasing length of the record which results in increasing differences between sepsis and non sepsis records. Secondly, the shape of the curves show differences. Based on these observation, we assumed that a classifier stacked onto the RNN output potentially leads to a performance improvement. Such a classifier would then be trained on the network's output. In that case, the output can be interpreted as high level features.

2.2. Second model part: the gradient boosted machine (LGBM)

We used a gradient boosted machine consisting of decision trees as the base estimators. As most of the data was already used for training the RNN, we decided to implement a light weight classifier which tends to better perform on less data compared to, for example, an additional neural network, e.g. a multilayer perceptron.

2.2.1. Preprocessing steps for the LGBM

We used 3 demographic parameters and derived handcrafted features (see table 1) from the RNN output and chosen parameters of the last 15 hours. The parameters include the 7 most frequent available in the data set, i.e. heart rate, oxygen saturation, temperature, systolic/mean and diastolic blood pressure and the respiration rate. This step can be interpreted as a reinjection which is a known technique regarding the implementation of neural networks [7]. Overall, this results in 313 input features for the LGBM. Non-available features were imputed by the median of that feature's values. Finally, the LGBM was trained using the training and validation set together and validated on the remaining test set. In this training process, we used the two original classes predefined in the data set.

2.2.2. Setup of the LGBM

We implemented the LGBM with an early stopping condition which is tied to the utility score of the test set. If there was no more improvement after 60 iterations we stopped the training process. We used the mean of the model output, i.e. the probabilities, for sepsis hours on the test set as the classification threshold. We therefore adjusted the classes' weights to counter the unbalanced proportions of the two occurring classes. The maximum number of leaves was fixed to 25 whereas the maximum tree depth was not fixed. We used a learning rate of 0.01. Remaining parameters were left as default. Within the cross validation the LGBM used 519 iterations on average.

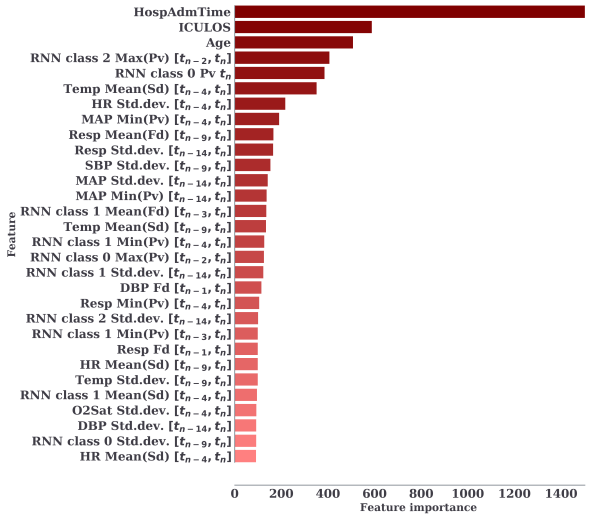


Figure 3: 30 most important features for the classification with the LGBM.

3. Results and discussion

Our stacked model achieved a utility score of 0.39 with a standard deviation of 0.013 in the 4-fold stratified cross validation. Additionally, we present the feature importance extracted from the LGBM in figure 3 averaged over the 4 folds. Unfortunately, we were not able to score our model on the held back challenge test data due to code issues in the final stage of the challenge. This is why we can only present cross validation results of our final model on the publicly available data set. Our last working submission which also represents our first entry achieved a utility score on full test set of 0.114. Therefore, we achieved rank 60.

When analyzing the feature importance extracted from the LGBM model, one can easily identify the value of the features derived from the RNN output. As expected, those features seem to be highly relevant for the final classification into sepsis and non-sepsis hours. However, especially the 3 demographic features seem to have the highest importance for the classification.

We assume that the model performance can be further improved by deriving features for the LGBM from more than one RNN. Another approach that could potentially improve the model performance is the combination of an RNN trained on longer sequences - as presented in this work - and a second RNN trained on shorter sequences. From those two models, features could than be derived for the LGBM.

4. Conclusion

In this contribution, we show that the combination of a recurrent neural network for extracting time dependent patterns related to sepsis development and a gradient boosted machine with the objective of learning from the network's output is a valid approach. We successfully combined these two different machine learning approaches and obtain remarkable results regarding sepsis onset prediction on the given dataset.

References

[1] 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);.

[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; 315(8):801–810. ISSN 1538-3598.

[3] Calvert JS, Price DA, Chettipally UK, Barton CW, Feldman MD, Hoffman JL, Jay M, Das R. A computational approach to early sepsis detection. Computers in biology and medicine 2016;74:69–73. ISSN 1879-0534.

[4] Scherpf M, Gräßer F, Malberg H, Zaunseder S. Predicting sepsis with a recurrent neural network using the mimic iii database. Computers in biology and medicine 2019;103395. ISSN 1879-0534.

[5] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds.), Advances in Neural Information Processing Systems 30. Curran Associates, Inc, 2017; 3146–3154.

[6] Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. URL http://arxiv.org/pdf/1406.1078v3.

[7] Chollet F. Deep learning with Python. Safari Tech Books Online. Shelter Island, NY: Manning, 2018. ISBN 9781617294433.

Address for correspondence:

Matthieu Scherpf
Institute of Biomedical Engineering
Fetscherstrasse 29, 01307 Dresden, Germany
Matthieu.Scherpf@tu-dresden.com