Improving the Performance of a Neural Network for Early Prediction of Sepsis

ByeongTak Lee, KyungJae Cho, Oyeon Kwon and Yeha Lee

VUNO, Seoul, South Korea

Abstract

Early prediction of sepsis is a clinically important, yet remains challenging. As machine learning develops, there have been many approaches for prediction of sepsis using neural network-based models. In this work, We propose various methods including feature engineering, regularization technique, and train data sampling methods, which can boost the performance of the model. Our approach consist of three-component: a feature engineering, an auxiliary loss, and a manipulation of training distribution. In feature engineering, we employed a novel input imputation method that combines input decay, masking, and duration of missing and input transformation. As for regularization, we used the reconstruction error as the auxiliary loss. Meanwhile, we manipulated the distribution of training sample using normal point re-sampling and population-based sampling. On the validation set, our approach improved the performance of LSTM as AUROC/AUPRC of 0.045/0.017, and the performance of transformer is enhanced AUROC/AUPRC of 0.034/0.024. Finally, we submitted our transformer trained with proposed method on the official test set and obtained the utility score of 0.291 (Team name:vn, Rank:23).

1. Introduction

Sepsis is a life-threatening disease caused by an uncontrolled response to infection. In worldwide, an estimated 30 million people develop sepsis, and 20 percent of them die from it every year [1]. Missing golden time for appropriate treatment is considered as the main reason for mortality [2]. For this reason, early identification is critical for improving sepsis outcomes, yet remains challenging [3].

As machine learning technologies develop, there have been many studies applying a statistical model to predict sepsis [4]. In the meantime, a deep neural network emerges as they show superior performance to existing statistical models [5, 6]. Most works, however, concentrated on building architecture, and training techniques have not been explored thoroughly.

In this work, we introduce several techniques that improve the performance of various neural network for the early prediction of sepsis. After exploring various techniques for improving the neural network, we suggest following three methods in this study:

· *Input imputation and transformation*
· *Regularization via auxiliary loss*
· *Manipulation of training data distribution*

We applied these methods on widely used neural network models for sequential data; Long Short Term Memory(LSTM) and transformer; and studied the effect of each methods.

2. Methods

In this chapter, we describe the database employed in developing the model, the structure of the default neural network, methods to improve the neural network, and evaluation method.

2.1. Data

The subset of the entire database is utilized as the model development, and the rest of the data is used as the official test from Physionet Challenge 2019. For the convenience of a term, we call the data employed in the model development as the internal data, and official test data as the external data. The internal database is collected from two hospitals consisted of 20,336 and 20,000 patients, respectively, of which 1,790 and 1,142 patients underwent sepsis in each hospital. Forty variables, including vital, laboratory, and demographics, were used as predictor variables as the inputs. Twelve-hour prior to onset of sepsis was used as the label. Detail information about the database can refer to [7].

2.2. Default Model

LSTM & Transformer The architectures adopted in this work are LSTM and transformer those which capture underlying characteristics of time series data. First, three-layer of LSTM with the residual connection is employed as the first default model. Each layer contains a unit of 200. In the meantime, a three-layer of the transformer layer is

 ISSN: 2325-887X DOI: 10.22489/CinC.2019.162

implemented as the second default model. The structure in detail of the transformer model is identical to [8].

Hyperparameters for the model The learning rate of 1e-3 with Adam optimizer is applied for the training. The batch size for training is 32. Dropout is employed at an input of each layer of LSTM, multihead-attention layers of LSTM, and a classifier of both architecture as 0.1, 0.1, and 0.5, respectively. L2 regularization with a coefficient of 1-3 is applied. Every variable is normalized with z-score based on the training population.

2.3. Proposed methods

Performance improvement was studied by applying the following methods without any changes to the model. At first, feature engineering method is presented. Subsequently, Auxiliary loss for regularization and manipulation of training sample are introduced.

2.3.1. Feature engineering

EMR data unavoidably contains missing observation induced by medical events, abnormalities, and inconvenience. There are several methods to handle missing values of EMR data, including forward-imputation, mean-imputation, and utilization of masking [9, 10]. Forward-imputation assumes missing values as same as its last measurement. Masking is used to distinguish the true values from imputed values, which is often applied with a duration of missing. Recently, [11] suggested a novel missing value imputation model that decays the missing value to the default value as the difference between its last observation and current time increases. Meanwhile, variables of EMR tend to be non-stationary because they reflects the status of patients, which is more challenging to learn the attributes of the sequence [12]. To handle the problem, we computed difference between adjacent time step of each variable, which removes the temporal trends. In this work, we combined various approaches including decay to default imputation (4), masking (1), duration of missing (2), and adjacent difference value. The entire set of input is described in (5).

$$m_t^d = \begin{cases} 1, & \text{if } x_t^d \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$\delta_t^d = \begin{cases} 1 + \delta_{t-1}^d, & \text{if } t > 1, m_{t-1}^d = 0 \\ 1, & \text{if } t > 1, m_{t-1}^d = 1 \\ 0, & \text{if } t = 1 \end{cases} \tag{2}$$

$$\gamma_t = exp(-max(0, W_\gamma \delta_t + b_\gamma)) \tag{3}$$

$$\hat{x}_t^d \leftarrow m_t^d x_t^d + (1 - m_t^d)\gamma_{x^t}^d x_t^d \tag{4}$$

$$X_t = [\hat{x}_t; m_t; \gamma_t; \hat{x}_t - \hat{x}_{t-1}] \tag{5}$$

2.3.2. Auxiliary loss for regularization

The primary loss function employed in training is the cross-entropy function. Additionally, we adopted reconstruction loss to prevent the model from overfitting on a training dataset. The reconstruction error is computed as the L2 distance between input and linearly transformed output of the model (6).

$$Loss = || \, [\hat{x}; \hat{x}_t - \hat{x}_{t-1}] - \text{reconstructed values} \, ||^2 \tag{6}$$

2.3.3. Method of training data sampling

Data resampling The model trained on the skewed distribution of class tends to have difficulty in learning the properties of a minority class and is biased to the majority class [13]. In the internal data, the number of the sepsis label is only 1 of 30 of the normal label, which probably leads to the difficulty in training described above. To handle this problem, we oversampled minority class data so that balanced the class ratio in the training sample. Furthermore, we additionally manipulated the distribution of normal data point. The normal data point is composed of normal data point from sepsis patients and normal data point from non-sepsis patients, and the ratio between them is 1:8. We tested several ratios between the normal point from sepsis patients and non-sepsis patients and acquired the biggest increase in performance from ratio 1:4.

Population-based sampling We found that the performance of the model decreases whenever a specific portion of the dataset flow into the training process. It means that some part of data deteriorate training. We parallelly trained the model on five subsets of randomly selected samples from the bootstrapped data pool, and choose the model with the highest performance among them to handle the problem. At every epoch, the parallel training begins from the best model of the previous epoch.

2.4. Evaluation

The internal database is divided into three-component: training, validation, and intermediate data set. The intermediate data set is hold out during the model training and tuning and used for the comparison of the various techniques in the internal database. Each of set compose of 60, 20, and 20 percent of the internal database. The ratio of hospital A and B are equal in each set. As for performance metrics, we used Area Under Receiver Operator Curve (AUROC), Area Under Precision-Recall Curve (AUPRC), and the score function provided by Physionet

			Regularization	Training data sampling		
	AUROC/AUPRC SCORE	Base	Reconstruction loss (A)	Resampling (B)	Population training (C)	(A) + (B) + (C)
	Base (Forward-imputation)	0.773/0.127 0.287	0.771/0.128 0.286	0.768/0.120 0.289	0.776/0.129 0.292	0.783/0.122 0.301
Feature Engineering	Forward-imputation, masking, duration of missing	0.803/0.138 0.349	0.816/0.141 0.356	0.813/0.142 0.354	0.813/0.141 0.360	0.816/0.145 0.363
	Propose method	0.812/0.136 0.355	0.812/0.134 0.357	0.816/0.147 0.363	0.812/0.138 0.361	0.818/0.144 0.367

Table 1. The intermediate result of utilization of proposed methods on LSTM. The red-colored section indicates three lowest-score and the blue-colored section indicated three highest-score.

			Regularization	Training data sampling		
	AUROC/AUPRC SCORE	Base	Reconstruction loss (A)	Resampling (B)	Population training (C)	(A) + (B) + (C)
	Base (Forward-imputation)	0.785/0.120 0.297	0.789/0.127 0.294	0.785/0.124 0.293	0.788/0.123 0.308	0.789/0.120 0.309
Feature Engineering	Forward-imputation, masking, duration of missing	0.808/0.138 0.347	0.814/0.137 0.366	0.811/0.138 0.341	0.806/0.141 0.356	0.816/0.140 0.368
	Propose method	0.810/0.138 0.359	0.818/0.143 0.366	0.813/0.140 0.365	0.811/0.142 0.360	0.819/0.144 0.376

Table 2. The intermediate result of utilization of proposed methods on transformer. The red-colored section indicates three lowest-score and the blue-colored section indicated three highest-score.

Challenge 2019. We evaluated the performance using a sum of AUROC and AUPRC during model development and the utility score for the intermediate result. Lastly, one of the model from the intermediate result is selected and tested on the external test set.

3. Results

Intermediate result We trained LSTM and transformer on various combination of feature engineering, regularization and training data sampling. Table 1 is the result of a combination of suggested methods on the intermediate set. Blue-colored sections mean the three highest methods based on score, and the red-colored sections mean the three lowest methods. It is found that the performance has increased when the proposed methods are applied. In the intermediate set, LSTM with proposed methods achieved the performance with the AUROC/AUPRC of 0.773/0.127 and the score of 0.287. Comparing to the baseline, the performance gain was AUROC of 0.045, AUPRC of 0.017, and score of 0.080 (Table1). Similary, Transformer with proposed methods achieved the performance with AUROC/AUPRC of 0.819/0.144 and the score of 0.376. The performance gain is AUROC of 0.034, AUPRC of 0.024, and the score of 0.079, respectively (Table2).

Test result We submitted three-layer transformer which is trained with proposed techniques for the result from the external database. Our model achieved AUROC/AUPRC of 0.793/0.092, 0.812/0.083, and 0.771/0.038 from external test set A, B, and C. Also, we acquired the scores of 0.387, 0.351, -0.251 from set A, B, and C. The score on the entire test is 0.291. The detail of the result on the test set is described on table 3.

4. Discussion

Other methods Besides the methods described in the paper, we also explored various techniques, including the box-cox transformation of input variable , heteroscedastic uncertainty, and manifold mix-up. However, those do not show the improvement of the performance in the inter-

	Set A	Set B	Set C	Full set
Utility score	0.387	0.351	-0.251	0.291
AUROC	0.793	0.812	0.771	
AUPRC	0.092	0.083	0.038	
Accuracy	0.825	0.889	0.722	
F-measure	0.129	0.123	0.039	

Table 3. The result of entire test set from 3-layer transformer with proposed methods.

mediate set. It seems that many popular methods are not effective in sepsis prediction task.

Limitation & Future work Experiment for hyperparameter tuning has not been conducted thoroughly in this work. We expect that hyperparameter optimization such as learning rate, weight decay rate, optimization method, and reconstruction loss ratio, could further enhance the performance. Various architectures of neural network for early sepsis prediction have been suggested in Physionet Challenge 2019. We expect that the performance of those model can be further improved with our approaches.

5. Conclusions

In this work, we present various feature engineering and training techniques for sepsis prediction from clinical data. Utilizing proposed methods, we demonstrated the performance improvement on LSTM and transformer. In the intermediate set, our approach improved the performance of LSTM as AUROC/AUPRC of 0.045/0.017 and the score of 0.080. Also, the performance of transformer is increased AUROC/AUPRC of 0.034/0.024 and the score of 0.079. Finally, we submitted three-layer transformer trained with proposed methods and obtained the final score of 0.291.

References

[1] Singer M, Deutschman CS, Seymour CW, Shankar-Hari M, Annane D, Bauer M, Bellomo R, Bernard GR, Chiche JD, Coopersmith CM, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). JAMA 2016;315(8):801–810.

[2] Seymour CW, Gesten F, Prescott HC, Friedrich ME, Iwashyna TJ, Phillips GS, Lemeshow S, Osborn T, Terry KM, Levy MM. Time to treatment and mortality during mandated emergency care for sepsis. New England Journal of Medicine 2017;376(23):2235–2244.

[3] Paoli CJ, Reynolds MA, Sinha M, Gitlin M, Crouser E. Epidemiology and costs of sepsis in the united states: An analysis based on timing of diagnosis and severity level. Critical Care Medicine 2018;46(12):1889.

[4] 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.

[5] Kam HJ, Kim HY. Learning representations for the early detection of sepsis with deep neural networks. Computers in Biology and Medicine 2017;89:248–255.

[6] Moor M, Horn M, Rieck B, Roqueiro D, Borgwardt KM. Temporal convolutional networks and dynamic time warping can drastically improve the early prediction of sepsis. CoRR 2019;abs/1902.01659.

[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 2019;In press.

[8] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In Advances in Neural Information Processing Systems. 2017; 5998–6008.

[9] Lipton ZC, Kale D, Wetzel R. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In Machine Learning for Healthcare Conference. 2016; 253–270.

[10] Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. Doctor ai: Predicting clinical events via recurrent neural networks. In Machine Learning for Healthcare Conference. 2016; 301–318.

[11] Che Z, Purushotham S, Cho K, Sontag D, Liu Y. Recurrent neural networks for multivariate time series with missing values. Scientific Reports 2018;8(1):6085.

[12] Jung K, Shah NH. Implications of non-stationarity on predictive modeling using ehrs. Journal of Biomedical Informatics 2015;58:168–174.

[13] Johnson JM, Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data 2019;6(1):27.

Address for correspondence:

Yeha Lee
6th Floor, 507 Gangnamdae-ro, Seocho-gu, Seoul, South Korea
yehalee@vuno.co