

Convolutional and Recurrent Neural Networks for Early Detection of Sepsis Using Hourly Physiological Data from Patients in Intensive Care Unit

Xin Li^{1,2}, G André Ng^{2,3,4}, Fernando S Schlindwein^{1,4}

¹Department of Engineering, University of Leicester, Leicester, UK

²Department of Cardiovascular Sciences, University of Leicester, Leicester, UK

³University Hospitals of Leicester NHS Trust, UK

⁴National Institute for Health Research Leicester Cardiovascular Biomedical Research Centre, UK

Abstract

PhysioNet Challenge 2019 aims to develop novel solutions for early detection the clinical onset of sepsis. We used training data provided and develop machine learning classifiers to predict clinical sepsis 6 - 12 hours ahead of the clinical onset.

Neural networks possess abilities to uncover insights from complex datasets. We have trained an ensemble classifier with a convolutional neural network (CNN) and a recurrent neural network (RNN) for early detection of sepsis. The classifiers were implemented in Python using Keras with Tensorflow as back-end. Both networks were combined using bagging to achieve better performance.

The database appeared imbalanced, and the class (positive) with small number of data entries was oversampled proportionally before training. 90 % of the augmented and oversampled data were used for training, with 10 % for testing. Our team "Leicester Fox" achieved accuracy of 90.6 % , AUROC of 0.745 and AUPRC of 0.079 in the test set A. The entry for the official phase of the PhysioNet/CinC 2019 competition received a normalized utility score of 0.237 on full data. Machine learning and neural networks approaches showed potential application for better prediction of sepsis, using real-world database with random missing data and imbalanced classes.

1. Introduction

Sepsis is a vital life-threatening condition caused by infection and overreaction by the immune system, which may cause critical subsequence including tissue damage, organ failure, or death [1]. Early detection and appropriate antibiotic treatment of sepsis have shown to effectively treat sepsis [1]. However, excessive antibiotic uses on false positive patients can develop antibiotic resistant bacterial strains and can waste resources. On the other hand, each hour of delayed treatment has been shown to

increase the risk of resulting in death of by 4-8 % [2]. Accurate early prediction remains an unsolved problem due to error-ridden and incomplete electronic medical records and the level of complexity of sepsis.

Convolutional neural networks (CNN) and Recurrent neural networks (RNN) are increasingly applicable to health care data providing highly accurate predictions. The 20th PhysioNet/Computing in Cardiology Challenge 2019 utilises 40 hourly-collected physiological vital signs and laboratory test results from patients admitted to intensive care units (ICUs). In this paper, we aim to develop and test machine learning techniques, especially neural networks, to predict sepsis up to 12 hours before the clinical onset. We have successfully created a data pipeline to process and clean data, identified important predictive features using both CNN and RNN networks.

2. Materials and Methods

2.1. Database and imbalanced classes

Hourly physiological data of 40 variables from 40,336 patients admitted to intensive care unit (ICU) provided by PhysioNet Challenge 2019 were used [3]. The database appeared imbalanced, and the class (positive) with small number of data entries was oversampled proportionally before training using python library imbalanced-learn [4]. Let χ be an imbalanced dataset with χ_{\min} and χ_{\max} being the subset of samples belonging to the minority and majority class, respectively. The balancing ratio of the dataset χ is defined as (eq. 1):

$$r_{\chi} = \frac{|\chi_{\min}|}{|\chi_{\max}|} \quad (1)$$

The balancing process is equivalent to resampling χ into a new dataset χ_{res} such that $r_{\chi} > r_{\chi_{\text{res}}}$. Data balancing can be performed by oversampling such that new samples are generated in χ_{\min} to reach the balancing ratio $r_{\chi_{\text{res}}}$ [4].

Missing values, frequently found from laboratory data, were replaced by 0 in CNN and -1 in RNN before processing.

2.2. Training Data Labelling

For Sepsis patients, data were positively labelled 12 hours before clinical onset, aiming to achieve all possible rewarding scores according to the scales of score awarding from utility function.

2.3. Classification Model Training

Training and classification was implemented in Python environment using Keras [5] running on top of TensorFlow [6]. A CNN and RNN classifiers were trained on 90 % of data as a training set, and the accuracy was evaluated on 10 % of data as a testing set.

2.4. Convolutional Neural Network

2.4.1. Feature selection

To remove the effect of missing data and speed up training, a subset of 11 features from the 40 variables were selected based on importance ranking from training a subset using random forest (Figure 1). The selected features were: Heart rate, Pulse oximetry, Temperature, Systolic BP, Mean arterial pressure, Diastolic BP, Respiration rate, End tidal carbon dioxide, Serum glucose, Gender and hours since ICU admission. Hospital admission time was not selected on purpose to achieve more generic and robust features when dealing with different centres.

2.4.2. Architecture

Data were augmented using past information up to 5 hours before the current hour depending on data availability and the differences of the consecutive hours were computed. The resulting 11×11 matrix, resulting 121 vector for each hour were fed in a CNN for training. As illustrated in Figure 2, the CNN includes 3 convolutional layers, 2 pooling layers and a fully connected layer. Batch normalization and dropout layers [7] were also added in between to avoid over fitting.

2.5. Recurrent Neural Network

2.5.1. Data Normalisation

Data normalization was performed using overall global maximum and minimum. The scaled feature x_{scaled} is de-

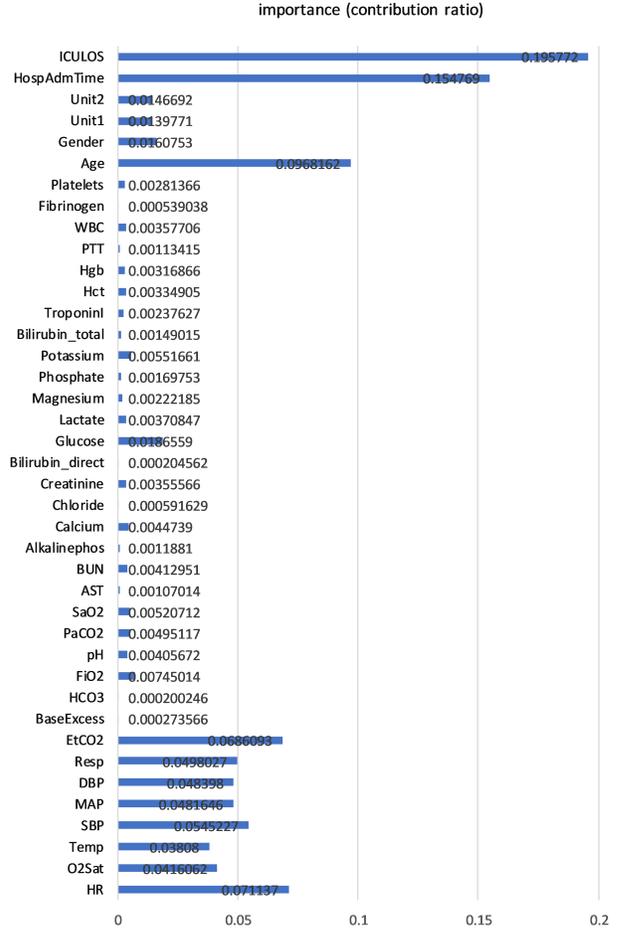


Figure 1. Importance ranking of the 40 features.

defined as (eq. 2):

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

where, x_{min} and x_{max} were calculated as minimum and maximum values from all available data.

2.5.2. Architecture

The 40 scaled features with past information up to 11 hours before the current hour (including) depending on data availability were used as input matrix for the prediction for the current hour, where no future information was used. The resulting 12×40 matrix was fed in a Long short-term memory (LSTM) network as input layer. Dropout was applied to input layer aiming to act as optimised feature selection. As illustrated in Figure 2, the RNN includes two stacked LSTM layers with 128 neurons. Dropout layers were added in between to avoid over-fitting. A dense layer of 32 nodes was followed before the last layer.

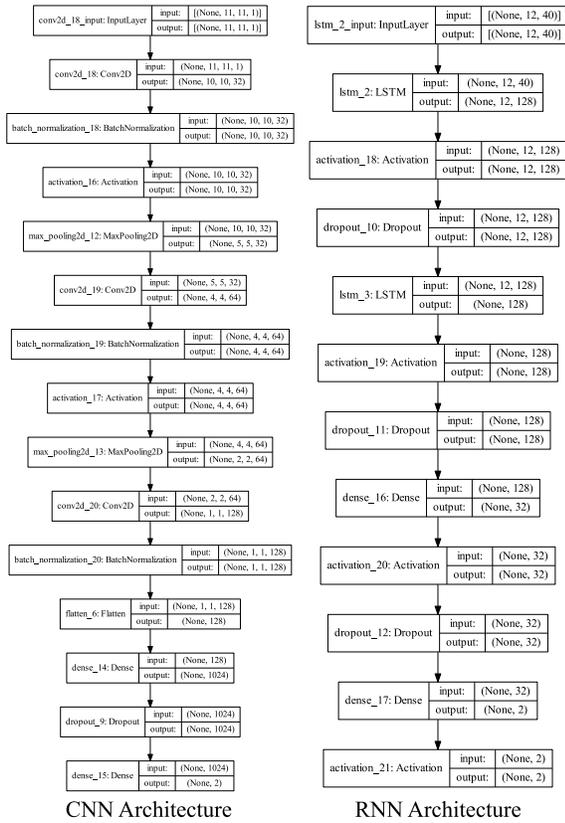


Figure 2. Architecture of CNN and RNN

2.6. Models Training

The last layer of both networks contains fully connected layers with the last layer with 1 neuron suitable for the binary classification problem, where the activation function is Sigmoid (eq. 3):

$$S(t) = \frac{1}{1 + e^{-x}} \quad (3)$$

Both neural networks were trained for 40 epochs with mini-batch size of 64 samples, where each epoch was shuffled. The neural networks apply the Adam optimization method with learning rate set to 0.00001 with a decay of 0.000001. Lost function of sparse categorical cross-entropy (eq. 4) was used with accuracy as evaluation metric. The cross entropy function was the objective function to be optimised during the model training process as follows:

$$L(X, r) = -\frac{1}{m} \sum_{i=1}^m \log p(R = r_i | X) \quad (4)$$

where x denotes to input data of the training sample, r is the marking sequence of the training sample, m is the

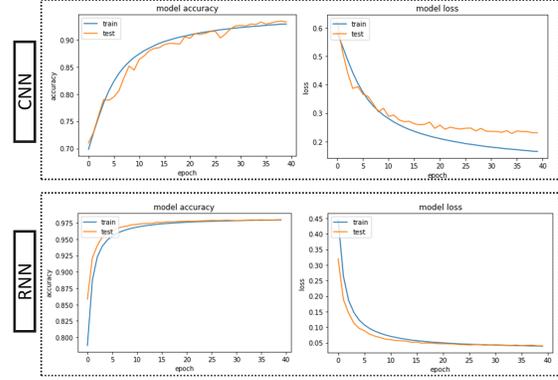


Figure 3. Training Loss and Accuracy per epoch

length of the marking sequence in the training sample, and p is the probability of the i^{th} output flagged as r_i .

2.7. Ensemble - bagging

Ensemble methods combine several classifiers to produce better predictive performance than a single decision tree classifier. The main principle behind the ensemble model is that a group of learners come together to form a strong learner, thus increasing the accuracy of the model. As illustrated in Figure 4, RNN and CNN were combined using ensemble modelling of simple bagging averaging of the possibilities output of both classifiers.

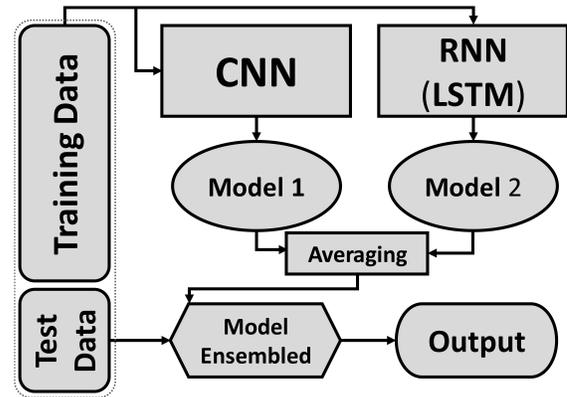


Figure 4. Ensemble-Bagging of two neural networks

3. Results and Discussions

A total of 1,552,210 input matrices from 40,336 patients were used, 1,241,768 were used as training set and 310,442 used as validation. CNN and RNN were trained with the final utility score on full data of 0.237.

As illustrated in Table 1, using a subset of the released training data, CNN alone achieved utility score of 0.236, while RNN achieved 0.279. The Ensemble-bagging of

both networks achieved a better utility score (0.288) as expected.

Table 1. Evaluation of CNN RNN and Ensemble using using a subset of the released training data

	Local					Remote
	AUROC	AUPRC	Accuracy	F-measure	Utility	Utility
CNN	0.919	0.374	0.894	0.226	0.742	0.236
RNN	0.927	0.284	0.875	0.184	0.626	0.279
Ensemble	0.964	0.383	0.927	0.294	0.786	0.288

Usually deep neural networks (CNNs and RNNs) are incredibly good at with images, sound, language and other 'natural' data, while tree-boosting frameworks could have performed better performance on given good hand-crafted features or measurements, such as the features provided in this Challenge. XGBoost [8] and LightGBM [9] are novel gradient boosting frameworks that use tree based learning algorithms, and they fit models to error terms and averages results within a generalised linear modelling framework using base-learner (weak) models at each iteration. In the current work, we have followed a similar idea of ensemble neural networks and demonstrated the possibility of achieving comparable results with gradient boosting methods dealing with ready features.

4. Conclusions

In this paper, we have successfully developed and deployed two neural networks including a CNN and a RNN, to be able to predict sepsis up to 12 hours before the clinical onset with relative high accuracy. More detailed hyperparameter optimisation of both networks and combining more ensemble neural networks could provide better performance. Bio-inspired systems such as neural networks approaches showed potential application for better prediction of sepsis, using real-world database with random missing data and imbalanced classes. With this data processing pipeline, it is possible to process and clean data, identified important predictive features and offer prediction scores in real time, which could be very useful for clinical practice to improve sepsis treatment and outcome.

Acknowledgements

This work was supported by the NIHR Leicester Biomedical Research Centre. XL received funding from the British Heart Foundation (PG/18/33/33780).

References

[1] 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) Consensus Definitions for Sepsis and Septic Shock. *JAMA* 2016;315(8):801–810. ISSN 0098-7484.

- [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. URL <https://doi.org/10.1056/NEJMoa1703058>. PMID: 28528569.
- [3] Reyna MA, Josef C, Jeter R, Shashikumar SP, 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).
- [4] Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research* 2017;18(17):1–5. URL <http://jmlr.org/papers/v18/16-365.html>.
- [5] Chollet F, et al. Keras. <https://keras.io>, 2015.
- [6] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <http://tensorflow.org/>. Software available from tensorflow.org.
- [7] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 2014;15:1929–1958. URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- [8] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. *CoRR* 2016;abs/1603.02754. URL <http://arxiv.org/abs/1603.02754>.
- [9] Ke G, Meng Q, Finley T, Wang T, Chen WJ, Ma W, Ye Q, Liu TM. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*. 2017; .

Address for correspondence:

Xin Li
 Department of Engineering
 Department of Cardiovascular Sciences
 University of Leicester, UK
 xl251@le.ac.uk