Machine Learning Algorithmic and System Level Considerations for Early Prediction of Sepsis

Lakshman Narayanaswamy*, Devendra Garg*, Bhargavi Narra*, Ramkumar Narayanswamy**

*Bangalore, India
**Teaq Group, Boulder, CO, USA

Abstract

This study presents a machine learning (ML) model that predicts onset of sepsis earlier in time than what is possible using common severity scoring systems. Our study's focus is on building solutions that maximizes sepsis prediction, is real-world implementable and usable by care providers particularly in developing countries like India. We have selected features based on the observation that patient vitals are available on an hourly basis, whereas lab results if available are less frequent. To capture the time series nature of the data, we trained the model using long short term memory (LSTM), a version of recurrent neural network (RNN) architecture. To capture locale specific pathology baseline, we have engineered features using two methods. We define a minimum & maximum value for vitals and lab tests and normalize the incoming data against this min-max value. Secondly, to leverage sparsely available lab data that signal increased sepsis risk, we define a synthetic "risk" feature. This risk feature is assigned a higher score when certain lab values are available and exceed a threshold. Our solution achieved an official utility score of 0.179 on the full test under the team name LDBR. Finally, we present practical considerations we discovered from our interactions with local hospitals and health-care providers.

1 Background

Sepsis is a life-threatening condition that occurs when the body's response to infection causes tissue damage, organ failure, or death. Sepsis is caused by an inflammatory immune response triggered by an infection which could be bacterial, fungal, or viral. In the U.S., nearly 1.7 million people develop sepsis and 270,000 people die from sepsis each year, many of the death being in hospitals. Internationally, an estimated 30 million people develop sepsis and 6 million people die from sepsis each year. (Reyna, 2019). An estimated 4.2 million newborns and children are affected by sepsis.

Sepsis is an enigmatic condition with heterogeneity in the host's response to the condition. Current techniques detect sepsis using scoring systems. However, these techniques do not leverage the inherent information available in-patient history, previous learnings and patterns inherent about sepsis manifestation. Currently, by the time sepsis is detected and the cause is identified, organ damage has already progressed to a potentially irreversible stage. Given the high incidence levels of sepsis and potential long-term damage to organs and/or loss of life, early detection of sepsis is gaining considerable attention in the machine learning community (Moor, 2019) (LSTM, n.d.). It is possible that the state-of-the-art machine learning techniques trained on large and diverse datasets can extract subtle, but highly discriminating leading-indicators for the onset of sepsis.

Machine learning is a subset of artificial intelligence, which builds sophisticated mathematical models of a phenomenon using multivariate measured data. The model is typically used to make predictions and support decision-making. Recently neural networks, particularly deep neural networks, have shown significant capabilities to model complex phenomena. In particular, Recurrent Neural Network (RNN) and LSTM within RNN has proved very effective in applications such as natural language modelling, translating languages, speech recognition, handwriting recognition, time-series prediction and anomaly detection (LSTM, n.d.).

2 Our Approach

In this work, we have taken a pragmatic approach on how to add value to the sepsis prediction problem. Detecting sepsis in developing countries implies working in a not-so-data-rich environment; both in terms of features and temporal sampling. The prediction model needs to be minimally complex, so that it is easily tune and adapted to local needs. We set out to find a solution which is incrementally better than the current situation in developing countries, rather than a 100% solution.

Given this minimally-viable-algorithm thinking, our first approach was to use a random forest decision tree. Random forest provides the ability to understand the influence of each feature in the prediction and also gives the care provider the ability to understand and interpret the prediction better than a black box algorithm that provides a number. While this approach gave us good results with the initial small dataset, it did not have the discriminating power when applied to the sepsis challenge dataset.

Next, we used a logistic-regression model using selected raw-features and synthetic compounded features. While the training was quick, it exhibited good learning as shown by low overall cost in both the training and validation data, the area under the receiver-operating-curve (ROC) stayed below 0.6.

Furthermore, given that the input data is temporally-sequential in nature and the final goal is to predict sepsis six hours ahead of the current best-practice, we choose to use a LSTM (Long Short-Term Memory) network, a type of recurrent neural network where the network has memory over extended time-periods and thus can learn temporal-signatures in addition to feature-signatures.

Figure 1: Major Phases in a Machine Learning Application Development

The process of training any machine learning algorithm can be represented using the block diagram shown in Figure 1.

The first block represents the act of capturing data about the patient, their symptoms and their health history. It could include physiological samples such as temperature, pressure, heart-rate captured at regular intervals in a hospital setting. It could also include laboratory test results as an when they are made available. Work by Futoma et al. (Futoma, 2017) is an example of a rich sepsis data set.

Typically, the **raw data** is captured across a multiplicity of instruments, laboratory reports, and patient questionnaires. Transcribing the data and consolidating them into a single source can be time consuming and error prone. Outlier detection and strategies for dealing with missing data are important at this stage. Patient privacy, data security, data-hosting, data-transport and data-format are all important considerations in the process of making the system operational.

Feature engineering is the art and science of determining which part of the data to use and how to use it efficiently. Typically, this step also involves annotating the data with labels which are needed for supervised learning applications. Collating data, conditioning it and selecting features for training are often the most time-consuming aspects of machine learning. Currently, there are companies dedicated to annotate large datasets including medical images, video, audio, hospital, clinic and laboratory data-sets.

The **processing-architecture** involves determining the right algorithm class that should be used given the problem statement. This decision should consider data-characteristics, available computational capacity and the application's real-time/non-real-time requirements. Lately other considerations have emerged - such as the requirement to cast the model's decision rational and explainable to a human. There are also emerging requirements for continual learning as external problem conditions change.

The **training phase** involves using the feature-engineered datasets, the network architecture and developing a mathematical model of the phenomena. A trained model is typically used to predict a future state for temporal data (example: tomorrow's weather) or classifying an object input to the model (example: cat picture).

The **operational** phase is about how the trained model is packaged, computationally executed and used for decision-making. Our current focus of this phase is on generalization. Going forward it will involve everything from end-use platform capabilities, explain-ability, decision-liability, customization, and regulatory compliance.

3 Data Source & Feature Engineering

Data used for this study is provided as part of the PhysioNet challenge[(Reyna, 2019)]. This data, made available to challenge participants, is from ICU patients from two separate hospitals. Each patient has a csv file where the column headers list the Vitals, Laboratory tests and Demographic values, while each row consists of hourly values for these parameters when available. A total of 40,366 patient files are available as part of this challenge.

3.1 Feature Description

We have noted that the hourly-sampled feature data varies widely across patients.

3.2 Sequential features

Given the clinical importance of vitals and that vital samples are available over 90% of the inputs, we use vital values of HR, O2Sat, SBP, DBP, MAP, Temp, Resp as our primary features. We use min-max normalization for each of the features. To capture locale specific pathology baseline, we have engineered features using two concepts

which we refer to as a "local baseline" and a "synthetic risk feature". We define a "local baseline" as minimum and maximum value for vitals and a threshold value for lab data. We normalize the incoming data against this baseline value. Clinically, certain lab test values such Lactate, pH and/or WBC are used to determine sepsis. However, these values are not routinely monitored and thus are missing for over 90% of the input. In order to deal with this situation, we defined a synthetic "risk" feature, where we check if the lab values for Lactate, pH and/or WBC are available for the selected time step (time step is a LSTM parameter). If the lab value is available and exceeds a threshold we set the risk feature as high. This technique helps include important signals that are available non-periodically from lab tests and other clinical events.

Table 1 lists the feature columns and their frequency distribution. As noted in the table a majority of the labels are not sampled or updated on an hourly basis.

Data availability	**Column(Feature) labels**
Columns values available 100%	Age, Gender, ICULOS, HospitalAdminTime, SepsisLabel
Column values available more then 80%	HR, O2Sat, SBP, MAP, Resp
Columns values available 50% to 70%	DBP, Temp
Columns values available 10% to 20%	Glucose
Column values available less than 10%	EtCo2, BaseExcess, HCO3, FiO2, pH, PaCO2, SaO2, AST, BUN, ALkalinephos, Calcium, Chloride, Creatinine, Bilirubin, Lactate, Magnesium, Phosphate, Potassium, Bilirubin, Troponin, Hct, Hgb, PT, WBC, Fibrinogen, Platelets

Table 1: Features in the sepsis challenge database and their availability

3.3 Non-sequential features

The following demographic features are available for every patient: Age, Gender, Hospital Admission Time, and ICU length of stay. Sepsis Label is given for each row to indicate if the patient has been diagnosed with Sepsis at that time instant. We use Gender, Age & ICU length of stay as features after binning.

4 Architecture and Training

The sepsis challenge is to detect sepsis six hours ahead of when it is currently detected in hospitals. The onset of sepsis can lead to irreversible damage to human organs and hence it should be detected and addressed as soon as it sets in. Clinically, care givers look for vitals and lab trends over time to detect the onset of sepsis. In order to learn temporal trends in a ML model, we choose Long Short Term Memory (LSTM), that uses feedback and memory-units to glean information from time sequences data. Figure 2 is a representation of the data as fed to the LSTM network. At each timestep t_n , $t_{(n+1)}$, $t_{(n+2)}$ and so on a set of features is input in to the LSTM and a sepsis/non-sepsis classification is expected as output.

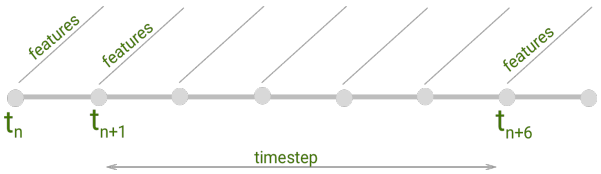

Figure 2: Temporal nature of data fed into the LSTM

Our implementation used Tensorflow v1.14 (tf.keras) and schematically shown in Figure 3. The first layer in our LSTM had 64 units; this was followed by a dropout layer, then a dense layer with 64 nodes and "relu" activation and finally a single node dense layer with sigmoid activation. The LSTM layer has a kernel and bias regularization of 0.2 and activity regularization of 0.1. The optimizer is "Adam" and loss function is "mse". Adam optimizer is based on adaptive moment estimation and "MSE" is mean squared error.

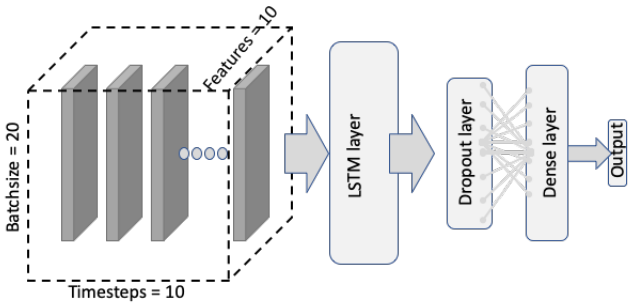

Figure 3 LSTM architecture for Sepsis detection

The training and testing data were split 75:25 and number of epochs set to 50. Given the unbalanced nature of the target, we used class weights for not-sepsis set to 0.01 and for sepsis set to 1000. We used early stopping and started with a learning rate of 0.0001 for up to ten epochs and increased the learning rate after ten epochs.

5 Results

Figure 4 graphs Receiver Operating Characteristic (ROC) curve, which shows the performance of the model. The output layer in our model is a layer with sigmoid activation that returns the probability of sepsis In order to classify the output as Sepsis or Not-Sepsis we need to choose an operating threshold. Figure 5 plots the precision and recall curve for various threshold values

that could be used. Given that an untreated case of sepsis could lead to organ-failure or death, we argue for maximizing True Positives at the expense of False Positives. In other words, we prioritized increasing Recall ever it meant lower Precision.

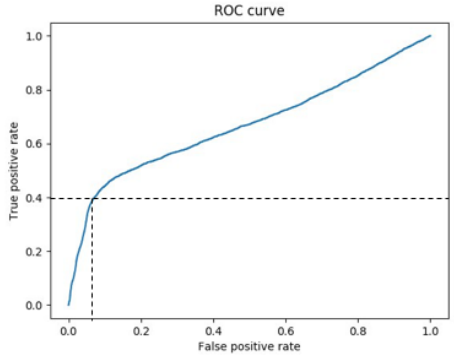

Figure 4: ROC of the sepsis prediction model with our operating point shown on the curve

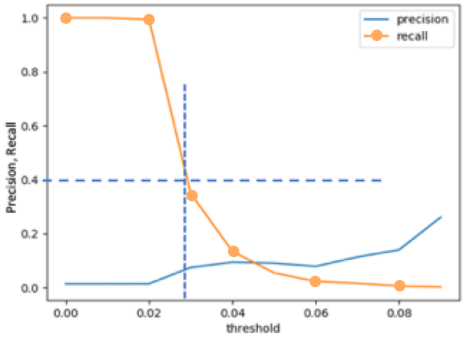

Figure 5: Precision and Recall as a function of the threshold at the output node

We chose the threshold value of 0.028 to maximize the number of True Positives (patients with Sepsis). At this threshold our recall value is 0.4 and the precision value is 0.08 as shown in Figure 5. The algorithm is graded for its binary classification performance using the organizer's utility function, which rewards(penalizes) classifiers for early (late) predictions of sepsis (Reyna, 2019). Our solution utility score on the full test set is 0.179. The challenge winners utility score is 0.36.

6 Deployment Considerations

In hospital setting, the system has to be robust and adaptable. In light of this, feature selection should acknowledge the cost and timeliness of acquiring specific features and the incremental information they add to the decision making process. It is important to have a well-thought-out fill in strategy when data for that feature is not available. This study is a case of using offline training of the model and using the model for dynamic inferences for each patient for a given time step. The offline approach is powerful in its simplicity during the training process, however an operational model will require the capability and process to incrementally update the model to reflect incoming data over time.

7 Conclusions and Future steps

This study developed a machine learning LSTM based model to predict sepsis six hours ahead of the typical sepsis detection today. The sepsis challenge organizers use an official "utility score" which rewards classifiers for early predictions of sepsis and penalizes them for late/missed predictions and for predictions of sepsis in non-sepsis patients. Our LSTM model scored a utility-score of 0.199, while the leader at the time of submission was at 0.433. Given this we realize that our model can be improved and we intend to continue refining it by improved how we selected the features, handled missing and noisy features, by defining a more sophisticated risk-factor synthetic feature and by training on more datasets.

We had significant learning in our discussions with local health-care providers and hospitals which we share below. The ML model's prediction should be explainable to get adoption by care providers. As a first step it would be helpful to provide weights of the used features for a given score. The model will also need a way to incorporate locale/regional influences, e.g. baseline Glucose readings which can vary based on ethnicity/region. The risk scoring system needs to fed into an alerting system that is easy to use by care providers and explainable. As mentioned earlier, ongoing training of the model, continuous evaluation of the score, subsequent actions by the care providers must be incorporated in the model training.

8 Acknowledgements

We like to thanks Dr. Arvind Kasaragod for his clinical inputs on the protocol for sepsis detection.

9 Bibliography

Futoma, J. E. (2017). An Improved Multi-Output Gaussian Process RNN with Real-Time Validation for Early Sepsis Detection. *Proceedings of Machine Learning for Healthcare 2017*, (p. JMLR W&C Track Volume 68).

LSTM. (n.d.). Retrieved from Wikipedia - Long short-term memory : https://en.wikipedia.org/wiki/Long_short-term_memory

Moor, M. H. (2019). Early Recognition of Sepsis with Gaussian Process Temporal Convolutional Networks and Dynamic Time Warping. *Proceedings of Machine Learning Research XX:1–IX,*.

Reyna, M. (2019). *Reyna MA, Josef C, Jeter R, Sashikumar SP, Westover MB, Nemita S, Clifford GD, Sharma A,* Early prediction of sepsic from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019. Critical care Medicine 2019; In press/

Address for correspondence:
Lakshman Narayanaswamy: nswamy.lakshman@gmail.com