Predictive Class-modeling of Sepsis Analysis Model

Abstract

Sepsis is a life-threatening syndrome caused by severe infections.It triggers an overactive immune response, leading to symptoms such as high heart rate, fever, and rapid breathing. Severe cases of sepsis can result in organ damage, while septic shock, the most critical form, involves dangerously low blood pressure and multi-organ failure. Mortality rates are high, with 30% of severe sepsis patients and 50% of those with septic shock dying in the hospital.

Septic shock is a critical condition that requires swift medical intervention, and early detection is vital to improving survival rates. The variability in how sepsis manifests due to different organ failures makes the progression toward septic shock difficult to model with traditional approaches. However, viewing septic shock as an acute deviation from patient "stability" rather than as a specific target class provides a novel framework for prediction.

To implement such a protocol, a model could continuously monitor vitals such as blood pressure, heart rate, respiratory rate, and organ function markers, flagging significant and acute changes. This approach would allow medical teams to recognize patterns indicative of septic shock earlier, potentially improving patient outcomes through timely interventions.

In this study, it proposes a combination model to predict the onset of sepsis and its progression to septic shock. The goal is to develop a predictive algorithm by analyzing variable length time-series data without including the sepsis label as a training feature. Instead, the sepsis label will serve as the target classification at each time interval. For each patient, the last observation will be used as the final classification target, based on N hours of data observation, with septic shock prediction as a subset of overall sepsis prediction. The model leverages LSTM for a classification approach, aiming to intervene before the patient experiences critical organ damage caused by septic shock.

1. Introduction

1.1 Problem Statement:

Sepsis is a leading cause of hospital mortality, responsible for 1 in 5 deaths in the U.S.

It occurs when the body's immune response to infection becomes extreme, resulting in severe tissue damage, multi-organ failure, and potentially death. Early detection of sepsis is crucial, as timely intervention with antibiotics can lead to survival rates above 70%. However, once sepsis progresses to septic shock—a critical condition marked by unresponsive hypotension and systemic shutdown—the survival rate drops significantly.

The goal of this study is to facilitate the early detection of sepsis and sepsis shock prediction by using clinical data, machine learning algorithms, and k-dimensional vectors. Sepsis is defined by the Sepsis-3 guidelines as a two-point change in the Sequential Organ Failure Assessment (SOFA) score and clinical suspicion of infection, identified through blood culture orders or intravenous antibiotics.

1.2 Importance of the Study:

This study is critical for several reasons:

- Advancement in Predictive Analytics: By integrating cutting-edge machine learning techniques, this research contributes to the advancement of predictive models in clinical analytics, a field that directly impacts medicine and financial stability.
- Technological Innovation: The application of LSTM networks combined with the HD computing framework represents a novel approach in clinical modeling, potentially setting a new standard for accuracy and efficiency in predictions.

2. Methods

This project utilizes an enhanced variable length Long Short-Term Memory (LSTM) network for sepsis prediction, augmented by the HD computing algorithm for sepsis shock.

2.1 Data Cleaning

The first step in preparing the dataset involves categorizing the features into two types: time-series and static features. Time-series features are continuous variables that rarely contain NULL values, while static features often have many missing entries. To fill in missing data in both time-series and static features, we apply interpolation techniques. This approach helps estimate missing values based on surrounding data points, providing a more comprehensive dataset. After filling in missing values, it is important to standardize the variables. This can be done using either Gaussian distribution or min-max scaling to normalize the data, allowing different features to contribute equally to model training.

Next, we divide the dataset into three subsets: training (70%), test (20%), and validation (10%). This ensures that the model is trained on a portion of the data while leaving out a separate portion for unbiased testing and validation. Outliers in the data are also identified and assessed for inclusion based on their relevance. If an extreme value represents an edge case that the model should account for, it may be included; otherwise, it may be excluded to avoid skewing the results.

Feature selection is an essential part of the data cleaning process. Initially, we use Recursive Feature Elimination with Cross-Validation (RFECV), a wrapper technique that iteratively eliminates less relevant features to improve model performance. However, due to the high computational cost associated with large datasets, we opt for SelectKBest, a filter-based method that selects features based on their chi-square score. This method is computationally efficient and helps identify the most relevant features for the predictive model.

2.2 Data Windowing and Preprocessing

For time-series data, we employ variable-length Long Short-Term Memory (LSTM) networks, which are well-suited for sequential data. In this case, each sequence corresponds to an individual patient, where the sequence length represents the number of hours of recorded data, and the input size is the number of features per patient. Since each patient's records span different time periods, the sequence length varies. To handle this, techniques like pad_sequence(), pack_padded_sequence(), and pad_packed_sequence() are used to manage variable-length sequences. This ensures that the LSTM network can process patients with different observation periods efficiently.

For static variables, neural networks can serve as a powerful tool to handle non-sequential data, transforming them into feature vectors that can be combined with time series vectors to create a comprehensive feature model. This hybrid approach allows us to leverage the distinct advantages of both static and dynamic data, improving the model's predictive capacity.

Before feeding the data into the model, it may need to be reshaped into a dataloader format. This format allows us to group patient records by time while preserving the relevant features. LSTM is used to predict the next state of a patient, including the likelihood of sepsis, by analyzing both the patient's time-series data and static features. The sepsis label is set as the target classification label for each time interval. The LSTM's architecture enables the model to account for dependencies across time and predict future patient states more accurately. Note that the labels of sepsis patients themselves are not recorded as feature vectors in the parameter

learning process.

2.3 Sepsis Prediction

For predicting sepsis, we first use feature selection methods to identify the most relevant features in the dataset. Covariance calculations help detect extreme cases and relationships between features, allowing the model to focus on the most informative variables. For non-sepsis patients, the label remains 0 across all time intervals.

To build the prediction model, a hybrid approach is used: the LSTM model is applied to time-series features, while a standard Neural Network is used to handle static features. The outputs from both models are combined, with weights assigned to each to produce a final prediction. Additionally, ARIMA (AutoRegressive Integrated Moving Average) models may be used in conjunction with LSTM or as an alternative for evaluating sepsis predictions, especially when dealing with time-series forecasting.

LSTM Formulation :

The core operations within an LSTM cell can be mathematically described as follows :

$$h_t = f(W_{ih} \cdot X_t + b_{ih} + W_{hh} \cdot h_{(t-1)} + b_{hh})$$

h_t represents the hidden state at time t, a function of :

- $\rightarrow X_t$: the input at time t
- $\rightarrow W_{ih}, W_{hh}$: weight matrices for inputs and previous hidden states respectively.
- $\rightarrow b_{ih}, b_{hh}$: bias terms
- $\rightarrow f$: a nonlinear activation function, typically a sigmoid or tanh function, that helps model complex patterns in data.

Figure 1: LSTM Formulation

2.4 Sepsis Shock Prediction

Septic shock is considered when a patient's Mean Arterial Pressure (MAP) drops below 65 mmHg or lactate levels exceed 2 mmol/L. To predict this condition, we use HD computing transforms the data into a new space by extending the dimensionality of variables while retaining the most important ones. This allows the model to focus on the features that contribute to septic shock.

A logistic regression model or one class classification method can then be applied

to predict septic shock based on these reduced dimensions.

TorchHD, a framework for hyperdimensional computing, is utilized to further improve the accuracy of predictions. This technique can enhance the model's ability to handle large, high-dimensional datasets by encoding features in a way that simplifies computation while maintaining predictive performance.

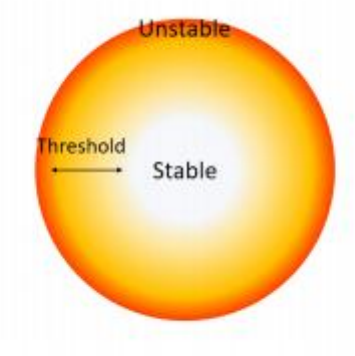


Figure 2: one hot classification

3. Experimental Results

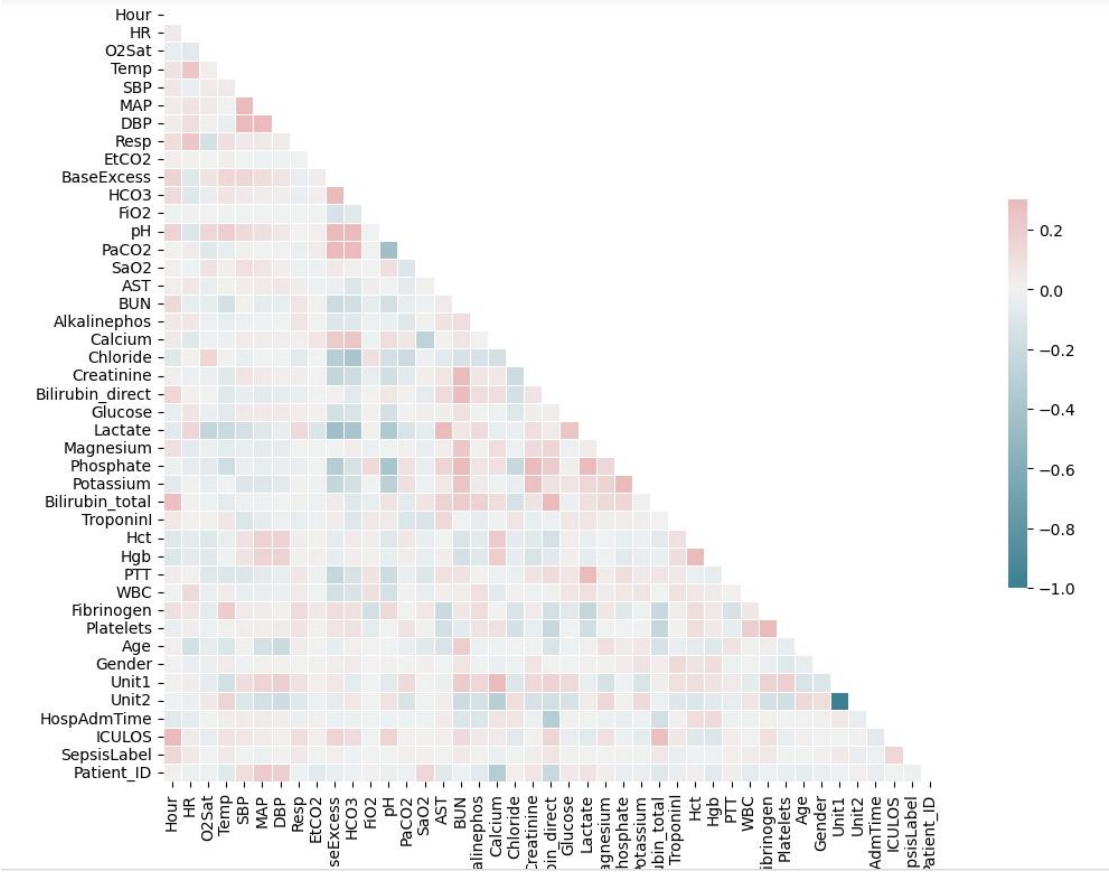
3.1 Feature Selection and Covariance Analysis

Given the high-dimensional nature of the dataset, with over 40 types of variable features, the initial step in our experiment was to reduce dimensionality using feature selection techniques. We applied SelectKBest in conjunction with the chi-square test to rank and select the most relevant features, which significantly improved the model's efficiency and performance. This approach allowed us to filter out features with low correlation to the target variable, thereby optimizing the feature set while maintaining essential predictive variables.

These were selected based on their chi-square scores, representing their statistical significance in predicting sepsis or septic shock. Examples of the top features include vital signs such as heart rate, respiratory rate, and lactate levels, all of which are clinically significant in tracking sepsis progression. The chi-square test provided a robust metric for quantifying the relationship between individual features and the target labels, particularly for categorical features, which was crucial in filtering out less relevant data.

```
chi-square is: [7.03726090e+05 8.36406204e+03 5.30548569e+00 3.16237034e+01
 3.82664224e+02 7.44902862e+02 2.01799451e+02 3.63323884e+03
 6.54918475e-01]
p-value is: [0.00000000e+000 0.00000000e+000 2.12583718e-002 1.87131472e-008
 3.27282754e-085 5.14882211e-164 8.45590397e-046 0.00000000e+000
 4.18359666e-001]
```

To further refine the feature set, we conducted a covariance analysis. This analysis revealed some multicollinearity between features like systolic and diastolic blood pressure, which we addressed by eliminating redundant variables. This step further reduced the dimensionality of the dataset, allowing for more efficient model training without sacrificing predictive power.



3.2 Data Normalization and Standardization

After feature selection, we normalized the data using Min-Max Scaling to ensure that all features were within the same range (0 to 1). This was essential because the dataset contained features with vastly different units, such as heart rate (measured in beats per minute) and lactate concentration (measured in mmol/L). Normalizing the data helped avoid bias during model training, where features with larger scales might disproportionately influence the outcome.

While both Min-Max Scaling improved the model's convergence during training,

normalization methods yielded slightly better results, particularly for algorithms like LSTM, which are sensitive to the magnitude of the input features.

3.3 Data Processing and Batch Handling

We tackled the challenge of managing a vast dataset where variables are time-continuous and each patient's data forms its own unique time series. To efficiently process this high-dimensional, variable-length time series data, we utilized dataloader and pad_sequence functions, ensuring that the data was encapsulated and batched appropriately for input into the LSTM model. This approach allowed us to handle varying sequence lengths for different patients without losing valuable temporal information. These preprocessing steps were crucial in handling the complexities of time-series medical data, which is inherently noisy and inconsistent in length.

3.4 Training Performance: Epochs vs. Loss Trend

During training, we observed that the loss function did not exhibit a sharp downward trend across epochs. Instead, the loss reduction was more gradual, especially in the early stages of training. This behavior can be attributed to the nature of the variable-length batches and the incremental updates to the LSTM's parameters as each batch of patient data was processed. Since the LSTM model continuously updated its weights with each batch, it adapted incrementally, which smoothed the loss curve and prevented abrupt decreases in the loss metric.

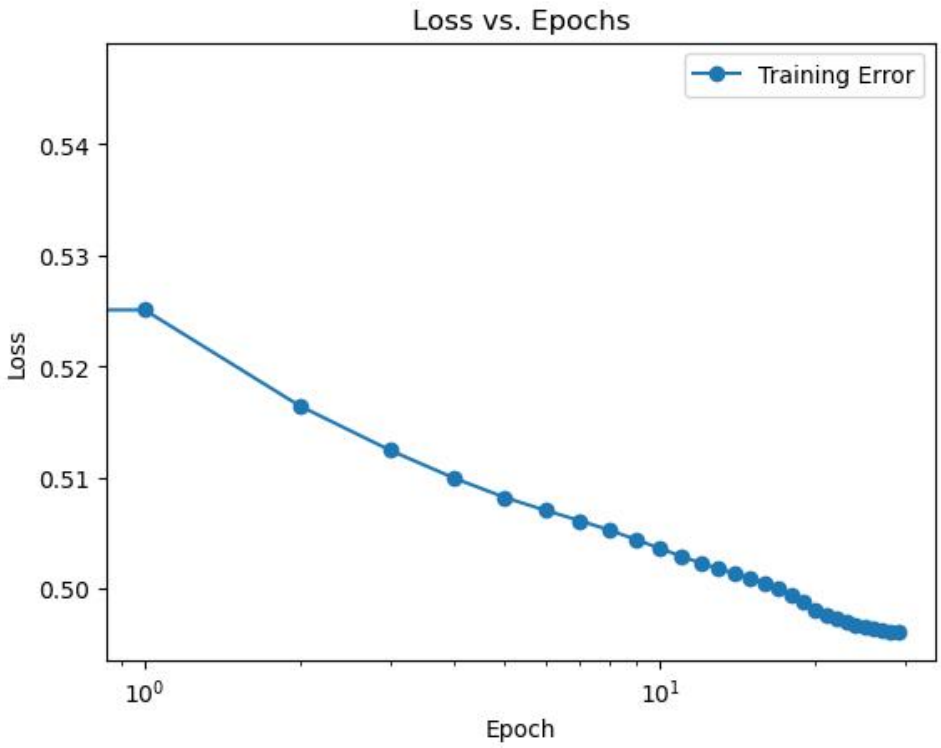
We found that the model's performance improved steadily with each epoch, although noticeable improvements only began to appear after around 30 epochs. We tested various configurations of hyperparameters, such as batch size, learning rate, and the number of LSTM layers, and determined that 30 to 50 epochs was the optimal update cycle for balancing operational efficiency and model performance. This epoch range allowed the model to stabilize and converge without overfitting or requiring excessive computational resources.

```
Epoch [1/30], Loss: 0.7214
Epoch [2/30], Loss: 0.6146
Epoch [3/30], Loss: 0.5568
Epoch [4/30], Loss: 0.5282
Epoch [5/30], Loss: 0.5169
Epoch [6/30], Loss: 0.5123
Epoch [7/30], Loss: 0.5092
Epoch [8/30], Loss: 0.5068
Epoch [9/30], Loss: 0.5050
Epoch [10/30], Loss: 0.5037
Epoch [11/30], Loss: 0.5027
Epoch [12/30], Loss: 0.5019
Epoch [13/30], Loss: 0.5011
Epoch [14/30], Loss: 0.5005
Epoch [15/30], Loss: 0.4998
Epoch [16/30], Loss: 0.4991
Epoch [17/30], Loss: 0.4985
Epoch [18/30], Loss: 0.4978
Epoch [19/30], Loss: 0.4972
Epoch [20/30], Loss: 0.4967
Epoch [21/30], Loss: 0.4961
Epoch [22/30], Loss: 0.4958
Epoch [23/30], Loss: 0.4956
Epoch [24/30], Loss: 0.4954
Epoch [25/30], Loss: 0.4953
Epoch [26/30], Loss: 0.4951
Epoch [27/30], Loss: 0.4950
Epoch [28/30], Loss: 0.4947
Epoch [29/30], Loss: 0.4946
Epoch [30/30], Loss: 0.4944
```

3.5 Loss vs. Epoch Curve Analysis

Early in training, the loss decreased marginally, reflecting the model's initial learning phase. As training progressed, the loss reduction became more consistent, demonstrating that the LSTM was capturing more relevant time-dependent patterns from the input sequences. However, due to the constant updating of parameters after each batch of patient data, the loss curve appeared smoother than typical machine learning models that use fixed-length batches and see more drastic loss reductions after each epoch.

By the time the model reached around 30 epochs, the loss had plateaued, indicating that the model had effectively learned from the data and further training would likely yield diminishing returns. This plateau suggested that the LSTM had reached an optimal state where it could accurately predict the onset of sepsis using the patients' time-series data.



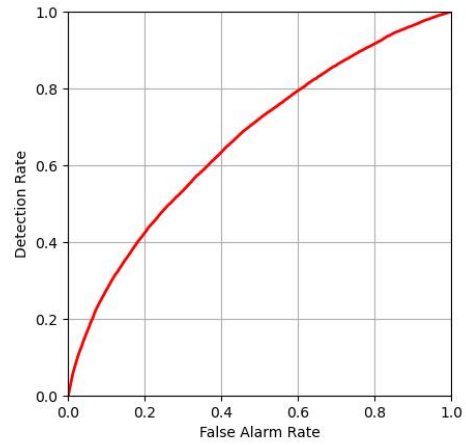
3.6 Model Evaluation and Future Improvements

Despite this relatively smooth convergence, the model's final performance was measured using the Area Under the Curve (AUC) metric. The AUC for the training dataset was 0.66, and for the testing dataset, it was 0.67. These results indicate that the model was able to distinguish between sepsis and non-sepsis patients with a moderate level of accuracy, although there is room for improvement. The closeness between the training and testing AUC scores suggests that the model generalizes well and is not overfitting to the training data.

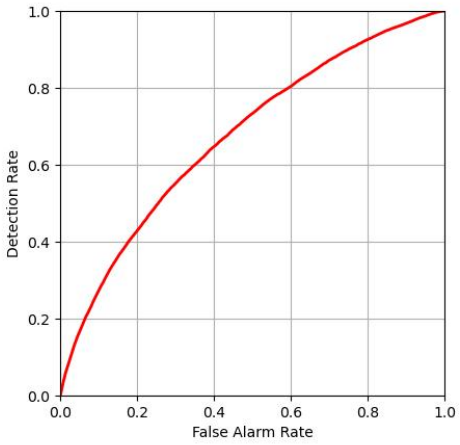
While the AUC scores of 0.66 for the training set and 0.67 for the test set are within an acceptable range for this type of complex time-series data, they highlight that further optimization is needed to improve the model's predictive accuracy. One potential area of improvement could be in refining the feature selection process or adjusting the architecture of the LSTM model. The relatively modest performance may be linked to the complexity of the sepsis prediction task, which involves many subtle and interrelated variables.

Future experiments may focus on incorporating advanced techniques such as attention mechanisms or bi-directional LSTMs to better capture long-term dependencies in patient time-series data. Additionally, further fine-tuning of hyperparameters, such as learning rate and batch size, could lead to improved model accuracy. Alternative models, such as Transformer-based approaches or hybrid models

that combine LSTM with traditional statistical methods like ARIMA, might also yield better performance.



0.6662911148422006



AUC 0.674339599026514

4. Discussion

A key factor observed in the experiment is the imbalanced number of classification labels within the dataset. For sepsis patients, the positive sepsis label typically appears only in the latter stages of the patient's data, while the majority of the time-series shows a negative label. This imbalance in the binary classification problem can skew the machine learning process, leading to lower predictive accuracy.

To address the imbalance issue, several optimization techniques can be applied in experiments:

- SMOTE (Synthetic Minority Over-sampling Technique) could help by generating synthetic samples for the minority class, thereby balancing the dataset and allowing the model to learn patterns from both classes more effectively.
- A simpler approach might be to adjust the weights of the loss function, assigning greater weight to misclassifications involving the minority class (sepsis-positive cases). This would incentivize the model to pay more attention to these instances.
- Another potential solution could involve splitting the dataset into smaller, balanced subsets, with each part containing an approximately equal representation of both positive and negative labels. Training separate models on these subsets could help mitigate the class imbalance.

Additionally, in the context of feature selection, the presence of over 40 variable

features presents a significant computational challenge. We used medical theory and data analysis to prioritize features, but more advanced methods such as Principal Component Analysis (PCA) for dimensionality reduction could be explored in future experiments. By reducing the number of variables while retaining the most informative ones, PCA could enhance model performance and reduce computational overhead. Setting up a validation dataset for testing different dimensionality reduction methods would be an effective way to compare the performance and efficiency of different models.

Another key factor lies in handling variable-length LSTM models. While padding approach helps maintain input consistency, further refinements may be needed to ensure that the model accurately captures temporal dependencies across patients with widely varying data sequences.

Although the current model leveraged selected features based on existing medical theories, further exploration of additional features such as Complications and Medications could reveal hidden patterns that significantly impact patient outcomes. These factors, which are not directly included in the current model, could provide critical insights into the severity of sepsis and its progression to septic shock. Incorporating such variables might improve the predictive power of the model, potentially leading to better early detection of sepsis.

Reference:

[1] Stock, M. (2022, October). *Tutorial on Hyperdimensional Computing*.

[2] Agrawal, R. (2024, March 18). *Interpolation Techniques Guide & Benefits | Data Analysis (Updated 2024)*.

[3] Reyna, M. A., Josef, C. S., Jeter, R., Shashikumar, S. P., Westover, M. B., Nemati, S., Clifford, G. D., & Sharma, A. (2019). Early Prediction of Sepsis From Clinical Data: The PhysioNet/Computing in Cardiology Challenge. *Critical Care Medicine, 48*(2), 210-217. https://doi.org/10.1097/CCM.0000000000004145

[4] Watkinson, N., Givargis, T., Joe, V., Nicolau, A., & Veidenbaum, A. (2021). Class-Modeling of Septic Shock With Hyperdimensional Computing. In *Proceedings of the 2021 IEEE Engineering in Medicine & Biology Society (EMBC)*, 1653-1659. https://doi.org/10.1109/EMBC46164.2021.9630353