Early Detection of Sepsis in Patients

Sharanya Senthil, Hrithik Sarda, Raghavi Dube, Rishab Jaiprakash Khuba, Deril Raju

1. Introduction

Sepsis represents a severe global health crisis, characterized by the body's overwhelming and life-threatening response to infection which can lead to tissue damage, organ failure, or death. As reported by the Centers for Disease Control and Prevention (CDC), sepsis is a major cause of mortality and morbidity in the United States, affecting nearly 1.7 million individuals annually and resulting in approximately 270,000 deaths.

Addressing this critical issue, the Sepsis Prediction Project is designed to develop a machine-learning model capable of predicting the onset of sepsis in patients based on clinical data. Early detection is crucial for timely medical intervention, which can significantly improve patient outcomes by enabling faster and more accurate treatment decisions.

The project leverages advanced technological frameworks to build a robust predictive model. The core infrastructure includes GCP Storage for secure and scalable data storage, MLFlow for managing the complete machine learning lifecycle, and Apache Airflow for orchestrating the data pipeline processes. Deployment utilizes Google Cloud Platform's Cloud Composer 2, providing a serverless environment that ensures efficient resource management and scalability.

By combining these state-of-the-art technologies, the project aims to establish a highly efficient pipeline for processing new patient files to detect the onset of sepsis promptly. This initiative not only aims to enhance healthcare responses but also to set new standards in predictive healthcare analytics, potentially reducing the global burden of sepsis.

2. Dataset Information

2.1 Dataset Introduction

The dataset includes clinical data collected from patients, such as vital signs and laboratory results, which are essential for predicting sepsis. This data is particularly significant due to the life-threatening nature of sepsis, where early prediction can be lifesaving. The information was originally gathered for an open source data in 2019, focused on advancing methods in early sepsis detection. The main goal of the dataset was for participants to predict the onset of sepsis 6 hours before it could be clinically identified, underscoring the critical impact of timely and accurate predictions in healthcare.

2.2 Data Card

The data is sourced from ICU patients in three separate hospital systems. There are over 40,000 unique patients. The data for each patient will be contained within a single pipe-delimited text file. Each file will have the same header and each row will represent a single hour's worth of data. Available patient covariates consist of Demographics, Vital Signs, and Laboratory values

The dataset consists of around 50 features and over 1M records of data. These features can be categorized into Vital signs, Laboratory values, and Demographic data. The format and description of these features are available here

The files are in Pipe-separated values (PSV) format from 2 hospitals consisting of over 40,000 patients' hourly data. We have different data types Numerical (Vital signs, Laboratory values) and Categorical (patient demographics)

2.2 Data Sources

The dataset for our Sepsis Prediction Project is sourced from PhysioNet, a publicly accessible research resource that offers large collections of physiological and clinical data. Specifically, the data was collected as part of the "Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology 2019". This dataset aims to encourage the development of algorithms capable of predicting sepsis before its clinical diagnosis, using multi-parameter time-series data.

For further details, the dataset can be accessed directly through PhysioNet's platform. Here is the URL to the specific dataset:

PhysioNet 2019: Early Prediction of Sepsis from Clinical Data

By utilizing this resource, we adhere to all the licensing and data usage policies provided by PhysioNet, ensuring that our project conforms to ethical and legal standards for data use.

2.2 Data Rights and Privacy

We are using a dataset licensed under the Creative Commons Attribution 4.0 International Public License (CC BY 4.0). Full license here This license allows us to share and modify the dataset as long as we provide proper attribution and indicate any changes made. To comply with data protection laws such as the General Data Protection Regulation (GDPR) within the European Union, we ensure that all personally identifiable information has been anonymized.

This is critical in maintaining patient confidentiality and adhering to privacy standards. Additionally, we are committed to implementing robust security measures to protect the data from unauthorized access or breaches.

We need to handle the dataset ethically, especially given its sensitive medical context. The CC BY 4.0 license requires that we do not impose any additional restrictions that could hinder the licensed rights. We take responsibility for ensuring that any modifications or uses of the data remain compliant with the original licensing terms. This includes respecting moral rights, which are not covered by the license, and understanding that patent and trademark rights are not included. Moreover, the use of this dataset does not imply any endorsement by the licensor.

3. Data Planning and Splits

For the Sepsis Prediction Project, our approach to data management involves several key steps in preprocessing, followed by strategic splitting of the dataset for training, validation, and testing. These processes are essential to ensure that our predictive models are both robust and generalizable.

Preprocessing Steps:

- Cleaning: Initially, we clean the dataset by removing or imputing missing values, which is critical given the clinical nature of the data. Missing values will be imputed based on the median or mode of the data column, depending on whether it is continuous or categorical.

- Normalization: Vital signs and laboratory results vary significantly in their ranges; therefore, we will normalize these features to ensure that no single feature dominates the model’s performance due to its scale.

- Feature Engineering: We will derive additional features from the existing data, such as the rate of change of vital signs, which may provide more insights into the patient's condition and improve the model’s predictive power.

- Labeling: The dataset will be labeled according to the time window for predicting sepsis (e.g., 6 hours before clinical diagnosis). This involves identifying the onset of sepsis and labeling preceding time frames appropriately.

Data Splitting Strategy:

- Training Set (70% of the dataset): The largest portion of the dataset will be used for training the model. This set helps the model learn the patterns associated with the early signs of sepsis.

- Validation Set (15% of the dataset): This subset of data will be used to tune the model parameters and make adjustments to the model architecture.

- Testing Set (15% of the dataset): The final evaluation of our model will be conducted on this unseen data. The testing set is vital for simulation on how it would predict new patient data in a clinical setting.

4. GitHub Repository

Link: https://github.com/Rishab-KH/IE7374-Sepsis-Classification

Folder Structure:

- README.md: Project overview, installation instructions, usage guidelines
- data/: Raw and processed data files
- notebooks/: Jupyter notebooks for data exploration and modeling
- src/: Source code for data processing, modeling, and evaluation, DAGs for Airflow
- models/: Saved models and related metadata
- scripts/: Deployment and monitoring scripts
- tests/: For unit tests of scripts and methods
- .github/: For Github actions to trigger workflows

5. Project Scope

5.1 Problems

Early Detection of Sepsis: The primary problem our project aims to address is the early detection of sepsis in hospital patients. Detecting sepsis early is crucial as it significantly increases the chances of survival and reduces the risk of complications. Currently, every hour of delay in recognizing and treating sepsis increases mortality rates substantially.

Resource Utilization: Incorrect or delayed diagnosis not only affects patient outcomes negatively but also increases hospital costs and resource utilization. Efficiently predicting sepsis can help in better resource management and reduce financial burdens on healthcare facilities.

False Positives and Alarm Fatigue: Existing systems often raise too many false alarms, which can lead to alarm fatigue among medical staff. This decreases the overall effectiveness of clinical alert systems in busy hospital settings.

5.2 Current Solutions

Clinical Criteria and Scores: The current standard involves using clinical scoring systems like the SOFA (Sequential Organ Failure Assessment) score or qSOFA (quick SOFA) to identify patients at risk of sepsis. These scores are based on various physiological parameters but often lack the sensitivity and specificity required for early detection.

Electronic Health Record (EHR) Systems: Many hospitals use algorithms integrated into EHR systems to detect sepsis. These algorithms typically rely on changes in vital signs, lab results, and other clinical indicators. However, they can be slow to update and may not incorporate real-time data effectively.

Manual Monitoring: In many settings, the detection of sepsis still heavily relies on manual monitoring by healthcare professionals, which can be prone to human error and is not scalable in under-resourced or overly busy hospitals.

5.2 Proposed Solutions

Machine Learning Model for Prediction: We propose to develop a machine-learning model capable of predicting sepsis 6 hours before it can be clinically identified. We focus on using and optimizing existing models such as logistic regression, decision trees, and random forests for simplicity. We plan to implement a hybrid deep learning approach that combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. The CNN layers will extract salient features from each time window of data, while the LSTM layers will analyze the temporal dependencies of these features to enhance the prediction accuracy of the onset of sepsis. This approach harnesses the strengths of both architectures to handle the complex, multivariate nature of clinical time-series data effectively.

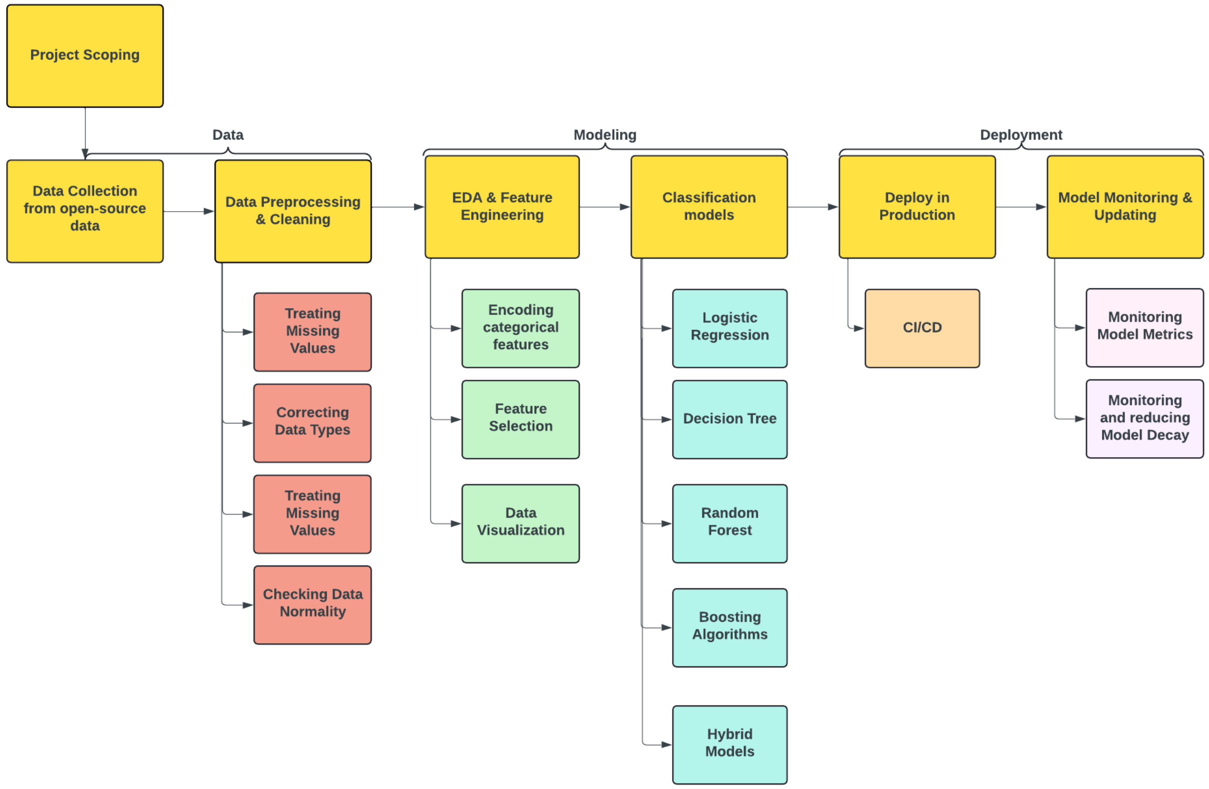
Integration with Real-Time Data Streams: By integrating our model directly with hospital monitoring systems, we can analyze patient data in real time, improving the timeliness and accuracy of sepsis prediction.

Custom Alert System: To address the issue of alarm fatigue, our system will include a tiered alert mechanism that differentiates warnings based on the risk levels, thus prioritizing high-risk cases and reducing unnecessary alerts.

Continuous Learning and Adaptation: Our solution will incorporate a feedback loop from clinical outcomes to continuously refine prediction algorithms, making our model adaptive to new patterns and changes in patient demographics or clinical practices.

6. Flow Chart and Bottleneck Detection

Flow Chart:



a. Data Collection
b. Data Preprocessing
c. Feature Engineering
d. Model Training
e. Model Validation
f. Model Deployment
g. Model Monitoring and Updating

Bottleneck Detection:

a. Data preprocessing and cleaning due to the high volume of missing and noisy data.
b. Real-time model inference and updating.

7. Metrics, Objectives, and Business Goals

7.1 Evaluation Metrics

- **Accuracy:** The percentage of total predictions that are correct, crucial for assessing the overall effectiveness of the model.

- **Sensitivity (Recall):** Ability of the model to correctly identify actual cases of sepsis. High sensitivity is critical as missing a sepsis diagnosis can be fatal.
- **Specificity:** Ability of the model to correctly identify non-sepsis cases, helping reduce false alarms and prevent alarm fatigue among clinical staff.
- **Precision:** Measures the accuracy of the sepsis predictions. High precision means that a high percentage of the positive predictions are correct.
- **F1 Score:** The balance between precision and recall. This is especially important in medical applications where both false negatives and false positives carry significant costs.
- **AUC-ROC Curve:** Area under the Receiver Operating Characteristic curve—a comprehensive representation of the model's ability at all classification thresholds.

7.2 Objectives

- **Develop a Reliable Prediction Model:** To create a machine learning model that can predict sepsis 6 hours before clinical symptoms manifest, thereby giving healthcare providers a crucial time advantage.
- **Integrate with Existing Clinical Workflows:** Ensure the model integrates seamlessly with existing hospital systems without disrupting current workflows.
- **Improve Patient Outcomes:** By predicting sepsis early, the model should significantly reduce the mortality and morbidity associated with sepsis.
- **Reduce Healthcare Costs:** Early sepsis detection and treatment can substantially decrease the length of hospital stays and use of resources, thus reducing costs.
- **Enhance Data Utilization:** Leverage existing clinical data more effectively, ensuring data collected in ICU settings are maximized for patient benefit.

7.3 Business Goals

- **Enhancing Patient Care:** Aligns with the healthcare industry’s overarching goal of improving patient outcomes through advanced technology and data-driven decision-making.
- **Operational Efficiency:** By reducing the incidence and severity of sepsis, the model helps lower the time and resources spent on sepsis patients, thus increasing hospital efficiency.
- **Cost Reduction:** Directly contributes to financial savings for healthcare facilities by lowering the costs associated with extended ICU stays and complex treatments.

- **Innovation and Leadership:** Positions the healthcare facility as a leader in innovative healthcare solutions, potentially attracting more funding and partnerships.
- **Regulatory Compliance and Safety:** Meets high standards of patient safety and regulatory compliance, reinforcing the hospital's commitment to quality care and innovation.

8. Failure Analysis

Biased Data Leading to Inaccurate Predictions

Risk: Biased data can occur if the training dataset is not representative of the general population or specific subgroups within the population. This can result from historical biases, sampling errors, or insufficient coverage of certain demographic groups.

Impact: Using biased data can lead the model to make inaccurate predictions, particularly for underrepresented groups. This could result in poorer health outcomes for these patients, exacerbating existing health disparities. Moreover, reliance on such a model could lead to mistrust in the system among clinicians and patients alike.

Mitigation:
Diverse Data Collection: Ensure the dataset includes a diverse range of patients, covering different demographics, medical histories, and treatment responses.
Continuous Monitoring: Regularly monitor the model's performance across different patient groups to identify any signs of bias and adjust the model accordingly.

Model Prone to Periodic Shifts

Risk: Models built on clinical data can become outdated due to periodic shifts in the underlying data distribution. These shifts could be due to changes in clinical practices, patient demographics, emerging health trends, or new pathogens.

Impact: If the model fails to adapt to these shifts, its accuracy and reliability can degrade over time, potentially leading to harmful outcomes if not identified swiftly. This can undermine the credibility of the predictive system and lead to reduced usage.

Mitigation:
Model Re-calibration: Regularly update the model to reflect new data and shifts in clinical practices. This could involve re-training the model periodically or continuously learning from new data.
Anomaly Detection Systems: Implement systems that can detect sudden changes in model performance or data patterns, triggering alerts for potential data shifts.

Alarm Fatigue

Risk: If the model generates too many false alarms, it can lead to alarm fatigue among healthcare providers, reducing the likelihood that the alerts will be acted upon.

Impact: Alarm fatigue can decrease the overall effectiveness of the sepsis prediction system and may result in negligence of critical alerts.

Mitigation: Optimize the balance between sensitivity and specificity to minimize false positives. Regularly review alert thresholds and incorporate clinician feedback.

9. Deployment Infrastructure

Our deployment infrastructure will make use of:

- AWS S3 buckets for reliable data storage, these buckets will be made available to client as well as a place to upload new patient data and generate predictions.
- We will be using Google cloud composer on GCP to host our AIrflow infrastructure, it will orchestrate the preprocessing of data fetched from S3.
- Model Training - Airflow schedules and manages the training process using Docker containers
- Prediction workflow - Airflow triggers predictions based on new data uploaded to the S3 bucket.
- Monitoring - AWS CloudWatch monitors the entire workflow, providing logs and alerts.
- GitHub Actions manage code deployments, ensuring continuous integration and deployment.
- Visualization - Streamlit provides interactive dashboards for real-time data visualization.

10. Monitoring Plan

What to monitor?

Model Performance Metrics:

Purpose: To ensure the predictive model maintains high accuracy, sensitivity, specificity, and precision over time.

Metrics: Track changes in accuracy, F1 score, ROC-AUC, precision, and recall. Monitoring should also include confusion matrix analysis to identify any shifts in false positives and false negatives.

Data Quality and Integrity:

Purpose: To ensure the data feeding into the model remains of high quality and free from corruption or loss during transfer and storage.

Metrics: Monitor the rate of missing data, data anomalies, outliers, and any signs of data corruption.

Operational Health:

Purpose: To ensure all components of the system function correctly and efficiently without disruptions.

Metrics: Monitor system uptime, response times, and resource utilization (CPU, memory usage). Check for any technical issues that could affect performance, such as slow response times or downtime.

- We could use ELK Stack (Elasticsearch, Logstash and Kibana) for real-time monitoring analysis.
- MLflow to log model parameters, code versions, metrics, and output files. This facilitates a detailed analysis of model performance over time and comparison across different model versions.
- Comprehensive Monitoring and Alerting: GCP Cloud Monitoring collects metrics, logs, and events from Google Cloud and external sources, providing robust visibility into the operational health of applications. It supports configurable alerts to notify about critical conditions, ensuring proactive management of system performance and reliability. Cloud Logging enables efficient storage, search, and analysis of logs from Google Cloud and AWS, aiding in troubleshooting and maintaining security compliance.
- If we notice major data quality issues, we could use Apache Griffin, an open source Data Quality Service platform that supports both batch and streaming data quality measures, providing a unified process for measuring data quality from different aspects.

11. Success and Acceptance Criteria

- The application should be accessible through a user-friendly front-end interface, allowing users to easily submit patient files and get predictions. Users should receive timely and accurate responses to their queries.
- The CI pipeline must include automated testing procedures that validate the codebase and functionalities after every update, ensuring that changes do not break existing features.
- The CD pipeline should support automated deployment of the chatbot application, enabling quick and efficient rollouts of updates and new features. In case of deployment issues, the CD pipeline must include mechanisms for quick rollback to previous stable versions, ensuring service continuity.
- Continuous monitoring of the chatbot's performance against predefined metrics (e.g., response time, accuracy) to ensure it meets the expected standards.

12. Timeline Planning

- Week 1-2: Data collection and preprocessing
- Week 3-4: Feature engineering and initial model training

- Week 5-6: Model validation and hyperparameter tuning
- Week 7-8: Deployment setup and testing
- Week 9-10: Monitoring implementation and final adjustments

13. Additional Information

- If we plan to adopt using deep learning models then we might also include Tensorboard into the architecture. This tool provides visualization and tooling needed for machine learning experimentation, tracking metrics such as loss and accuracy, and visualizing model graphs.