CSC 296S-01 Deep Learning (Spring 2026)

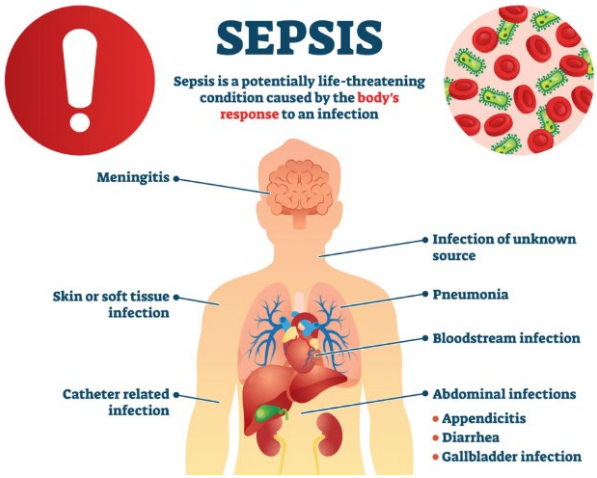
Project 1: Early Prediction of Sepsis using Deep Neural Networks

Due at 9:30 am, Wednesday, February 25, 2026

Demo: class time, Wednesday, February 25, 2026

1. Problem Formulation

Sepsis is defined as life-threatening organ dysfunction caused by infection. In 2017, Sepsis globally affected 49 million people with approximately 11 million deaths and is the third leading cause of death in United States hospitals. While vigorous research has been made on sepsis, no single treatment has been discovered for curing sepsis. Early prediction of sepsis is crucial to providing appropriate medical treatment for patients.



In this project, we explore (fully-connected) neural networks to make a binary (positive or negative) prediction of the onset of sepsis for each patient.

Design and implement your model using TensorFlow. Compare the recall, precision and F1-score for each label (1 = sepsis, 0 = non-sepsis). PLOT the confusion matrix and ROC curve of your best model.

2. Dataset

Data: https://drive.google.com/file/d/11sFmO9OiOc8agX6H5-iernbfUjk5mL7D/view?usp=sharing

We will use the PhysioNet/Computing in Cardiology Challenge 2019 dataset: https://physionet.org/content/challenge-2019/1.0.0/. The data are from two hospital systems. The entire training data (42 MB) consists of two parts: training set A (20,336 subjects) and B (20,000 subjects). No need to load entire dataset, i.e., use the set A, or B, or even a subset of them.

Each patient is a single pipe-delimited text file. Each row represents a single hour's data. Each file provides measurements over time. Each column provides a sequence of measurements over time (e.g., heart rate over several hours), where the header of the column describes the measurement. Each row of the table provides a collection of measurements at the same time (e.g., heart rate and oxygen level at the same time). The table is formatted in the following way:

```
HR |O2Sat|Temp|...|HospAdmTime|ICULOS|SepsisLabel
NaN|  NaN| NaN|...|        -50|     1|          0
 86|   98| NaN|...|        -50|     2|          0
 75|  NaN| NaN|...|        -50|     3|          1
 99|  100|35.5|...|        -50|     4|          1
```

There are 40 time-varying variables such as *HR, O2Sat, Temp ..., HospAdmTime*. **The final column, *SepsisLabel*, indicates the onset of sepsis, where 1 indicates sepsis and 0 indicates no sepsis.** *NaN* indicates that there was no recorded measurement of a variable at the time interval. **Please go over Table 1 on the data webpage for the meaning of each variable.**

You may drop the following 7 variables since they bare no contribution to sepsis or have too many missing values.

```
# Columns to drop
COLUMNS_TO_DROP = ['Bilirubin_direct', 'Bilirubin_total', 'TroponinI', 'Fibrinogen', 'Unit1', 'Unit2', 'EtCO2']
```

3. Requirements

- Check and handle missing values appropriately, e.g., replacing missing values using the last available values in the corresponding columns or simply use the mean (for numeric) or the mode (for categorical).

- Apply train and test split. Use training data to train your models and evaluate the model using test data

- Encode categorical features (if any) and normalize numeric features.

- Use EarlyStopping and ModelCheckpoint.
- Tune the following hyperparameters to record how they affect performance in your report. Tabulate your findings. Show at least three different configurations.
  - Neuron counts
  - # of layers
  - Optimizer: adam and sgd

4. Grading Breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

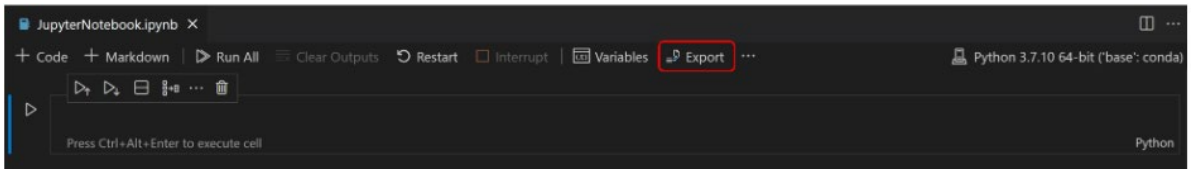
Use the evaluation form on Canvas as a checklist to make sure your work meets all the requirements.

5. Teaming

Students must work in teams of 2 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserves the right to assign different grades to team members depending on their contributions.

6. Deliverables

(1) The HTML version of your notebook that includes all your source code. In VS Code, you can export a Jupyter Notebook as an HTML file. To export, select the Export action on the main toolbar. You'll then be presented with a dropdown of file format options.



(2) Your report in PDF format, with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

- Problem Statement

- Methodology
- Experimental Results and Analysis
- Task Division and Project Reflection
- Additional Features

In the section "Task Division and Project Reflection", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

In the section "Additional Features", you describe and claim credit for additional features.

To submit your notebook and report, go to Canvas "Assignments" and use "Project 1".

All the deliverables must be submitted **by team leader** on Canvas before

9:30 am, Wednesday, February 25, 2026

NO late submissions will be accepted.

7. Possible Additional Features (5 pts per feature, 10 pts at most)

(1) Can you create **a more balanced dataset** by using oversampling or undersampling to train your model so that you model will not be biased to the more frequent classes?

Hint: See below for a potential oversampling technique:

https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

(2) Can you identify the most important features (this is called **feature importance analysis**) and train models only on those most important features, e.g., top-10 most important features?

Hint: One option is to use logistic regression to find the most important/influential features.

https://machinelearningmastery.com/calculate-feature-importance-with-python/

(3) By default, TensorFlow assigns the same weight to all training samples, which may bias your model to the more frequent classes when training on an unbalanced dataset. If the less frequent classes are more important, you may set different class weights in fit(). See if setting a higher class weight for sepsis versus non-sepsis will improve your original model.

Hint: Check the examples below.

https://hackernoon.com/how-to-use-class-and-sample-weights-in-keras-training

(4) What would be the optimal window size to use (how many hours we should look back) to achieve the best prediction results? Plot the F1 score for the sepsis (label 1) against the different window sizes you chose.

(5) Can you apply scaling factors for each feature to mimic clinician behaviors by weighting recent measurements more heavily? For example, heart rate measured 2 hours ago is given more weight than heart rate measured 5 hours ago.

Hint: use the Multiply layer to perform element-wise multiplication of two tensors.

https://keras.io/api/layers/merging_layers/multiply/

The similar idea is used in COMPOSER, one of the today's best sepsis prediction methods.

https://github.com/NematiLab/COMPOSER

8. In-class Presentation.

On the due day, each team has 10 minutes to present your work in the class. Explain your solutions by referring to your notebook. You do not have to prepare the PowerPoint slides for your presentation.