Project 3.2

Derek Caramella
Department of Data Science, University of Rochester
DSCC 475: Time Series Analysis and Forecasting in Data Science
Dr. Anand
12 December 2021

INSTRUCTIONS:

- You are welcome to work on this project individually or in teams (up to 2 members in each team max).
- If you plan to use PyTorch, a good resource is to review and modify the example code provided for the problem. We plan to review this example code in class as well.
- As outlined in the beginning of the code, you need to have the “arff2pandas” package to access the data files.

For the submission, please make sure to hand in the following:

- A document (PDF, Word etc) that captures your responses to the questions below *separately* from the code to facilitate grading.
- Your code files and output
- Both team members on team should please submit the work to Blackboard.

Overview

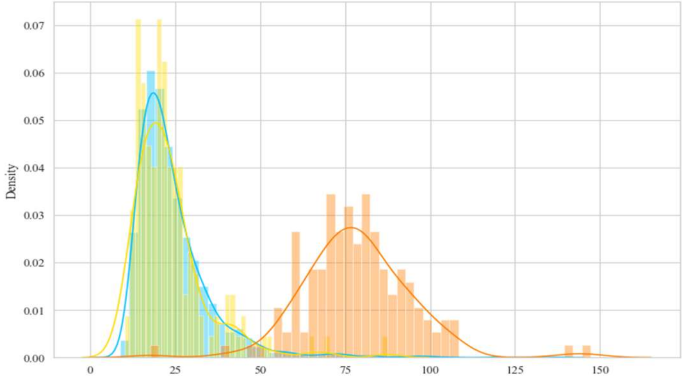
In this project, you will work with LSTM-based autoencoders to classify human heart beats for heart disease diagnosis. The dataset contains 5,000 Time Series examples with 140 timesteps. Each time-series is an ECG or EKG signal that corresponds to a single heartbeat from a single patient with congestive heart failure. An electrocardiogram (ECG or EKG) is a test that checks how your heart is functioning by measuring the electrical activity of the heart. With each heartbeat, an electrical impulse (or wave) travels through your heart. This wave causes the muscle to squeeze and pump blood from the heart. There are 5 types of hearbeats (classes) that can be classified: i) Normal (N); ii) R-on-T Premature Ventricular Contraction (R-on-T PVC); iii) Premature Ventricular Contraction (PVC); iv) Supra-ventricular Premature or Ectopic Beat (SP or EB); v) Unclassified Beat (UB). The shape of the time-series and the position of the impulses allows doctors to diagnose these different conditions. For the purposes of this project, we are interested in 2 classes: *Normal* and *Abnormal* (which includes class 2-5 above merged).

This is an example of an anomaly detection problem where class imbalance exists, i.e., number of each of the individual positive (abnormal) instances are smaller than the normal case. The autoencoder approach is suited well for such **applications of anomaly detection**. In anomaly detection, we learn the pattern of a normal process. Anything that does not follow this pattern is classified as an anomaly. For a binary classification of rare events, we can use a similar approach using autoencoders.

A sample code example (in Python) implementation of auto-encoder “AutoEncoders_anomaly_detection_ecg_SAMPLE.py” is provided. Review and run the code and answer the following questions:

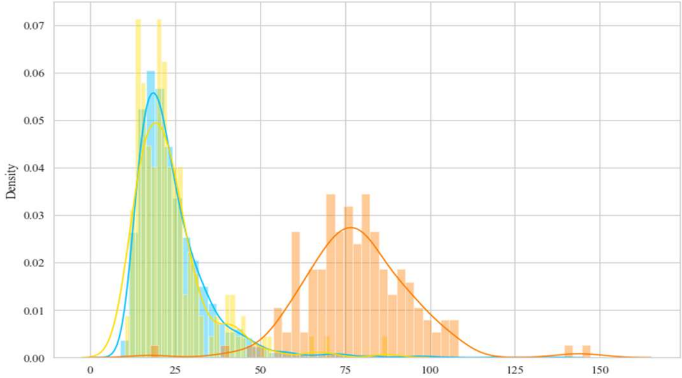
1. A critical hyper-parameter when using auto-encoders is the threshold applied to the reconstructed time-series to classify between normal and abnormal. The default threshold in the code is set to 45. Run the code for 50 epochs.
    a) For the normal and abnormal test set defined in the code as "test_normal_dataset" and "anomaly_dataset", vary the threshold value from 15 to 75 (both included) in increments of 10 and report (as a graph or a table) the proportion of normal and abnormal time-series that were correctly classified, i.e., recall.

Threshold 15



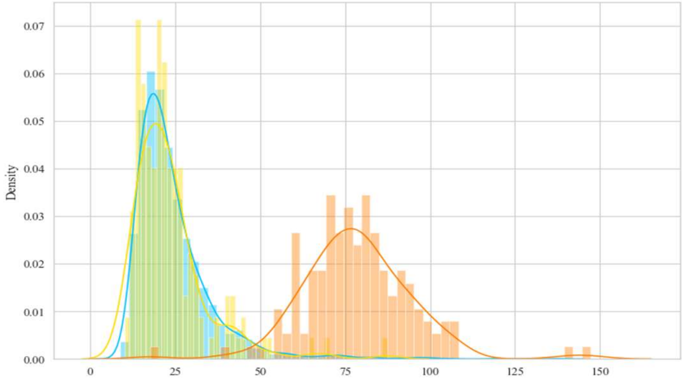
Normal Recall: 27/145 (18.62%)
Anomaly Recall: 145/145 (100.0%)

Threshold 25



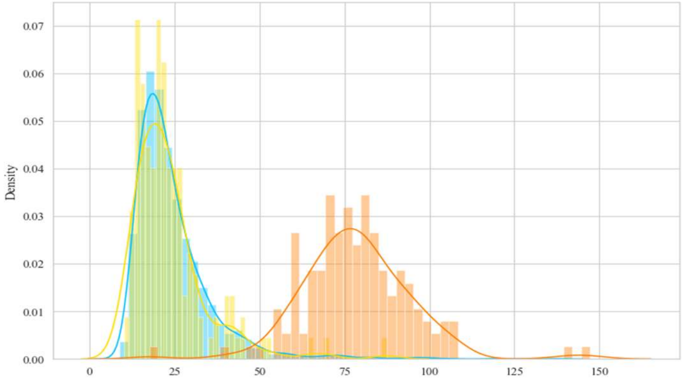
Normal Recall: 101/145 (69.66%)
Anomaly recall: 144/145 (99.31%)

Threshold 35



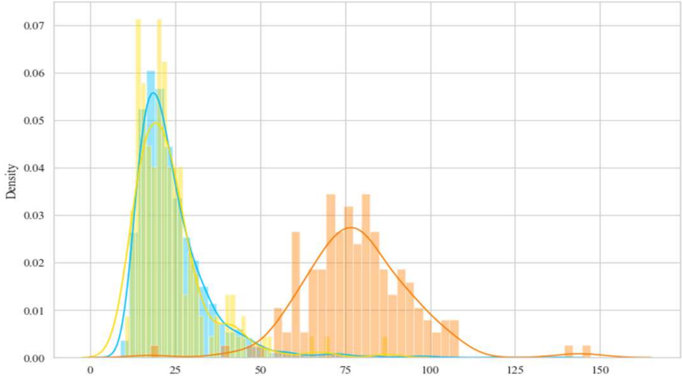
Normal Recall: 129/145 (88.97%)
Anomaly Recall: 144/145 (99.31%)

Threshold 45



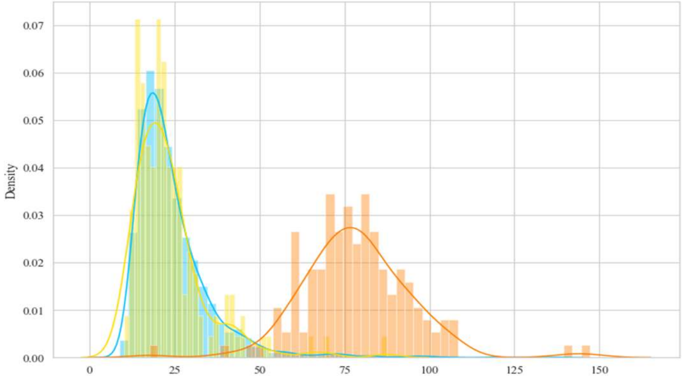
Normal Recall: 140/145 (96.55%)
Anomaly Recall: 143/145 (98.62%)

Threshold 55



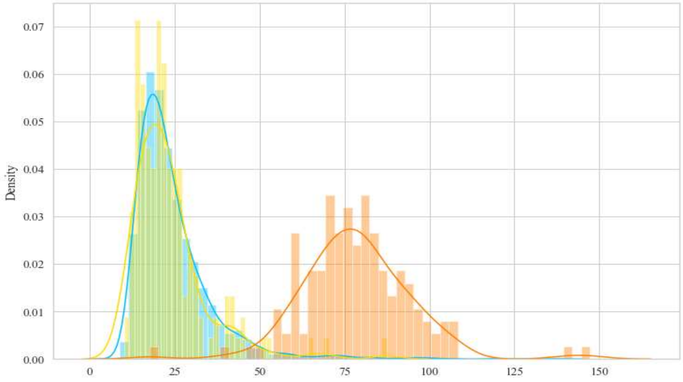
Normal Recall: 142/145 (97.93%)
Anomaly Recall: 140/145 (96.55%)

Threshold 65

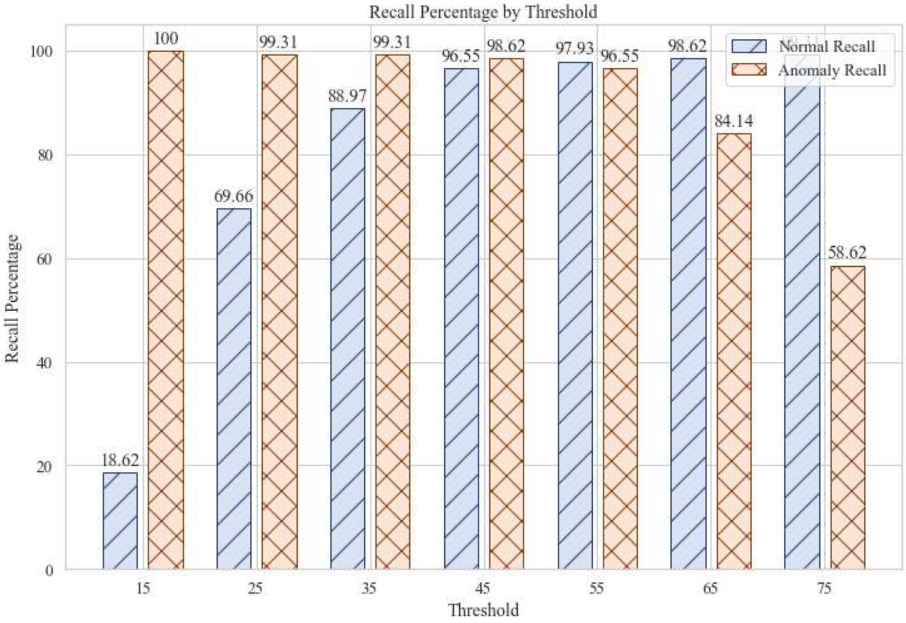


Normal Recall: 143/145 (98.62%)
Anomaly Recall: 122/145 (84.14%)

Threshold 75



Normal Recall: 144/145 (99.31%)
Anomaly Recall: 85/145 (58.62%)

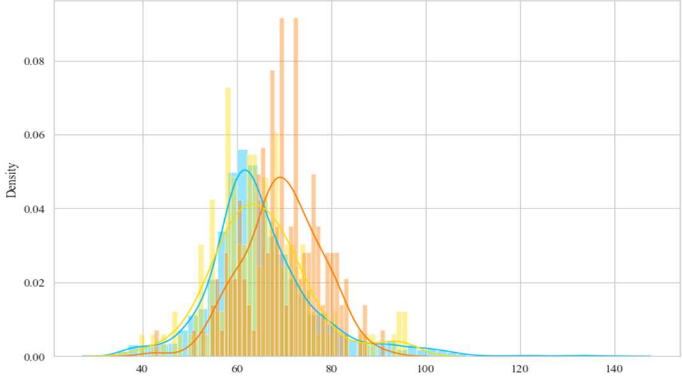


b) Briefly explain the trend you see in the Recall values as you increase the threshold.

When the embedding dimension is fixed, as the threshold is increased, normal recall increases. However, when the embedding dimension is fixed, as the threshold is increased, anomaly recall decreases. As the threshold is increased, more instances are classified as normal since the loss tolerance is greater relative to the previous threshold step.

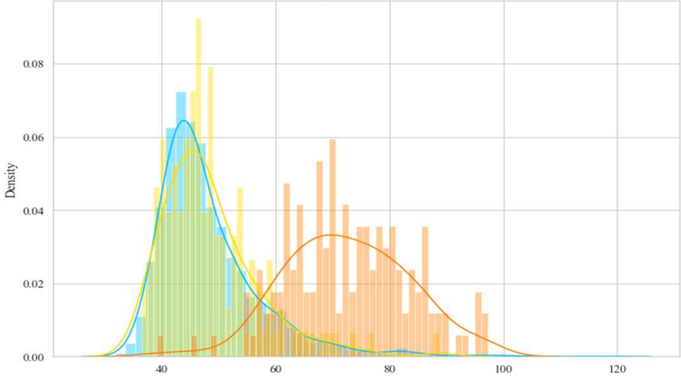
2. In the above example, the embedding dimension (i.e., output length of encoder and input length of decoder) was set constant at 8.
    a) Embedding dimension length is typically an important hyperparameter that can affect the performance of the technique. Vary the embedding dimension from 2 to 8 in increments of 2 and report the training and validation loss after 25 epochs.

Embedding Dimension 2



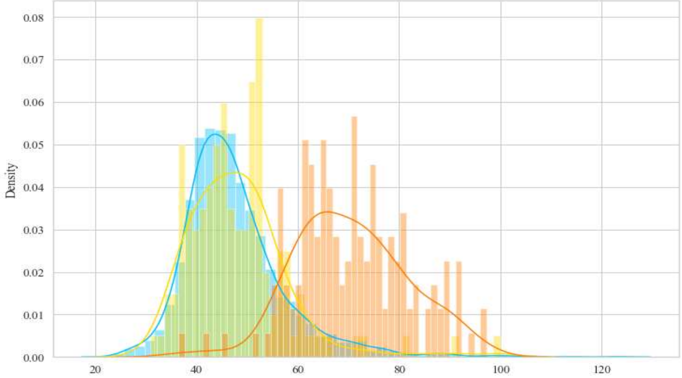
Normal Recall: 3/145 (2.07%)
Anomaly Recall: 144/145 (99.31%)

Embedding Dimension 4



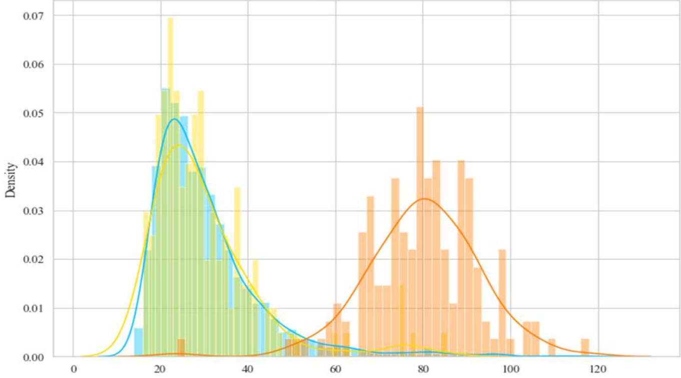
Normal Recall: 52/145 (35.86%)
Anomaly recall: 144/145 (99.31%)

Embedding Dimension 6



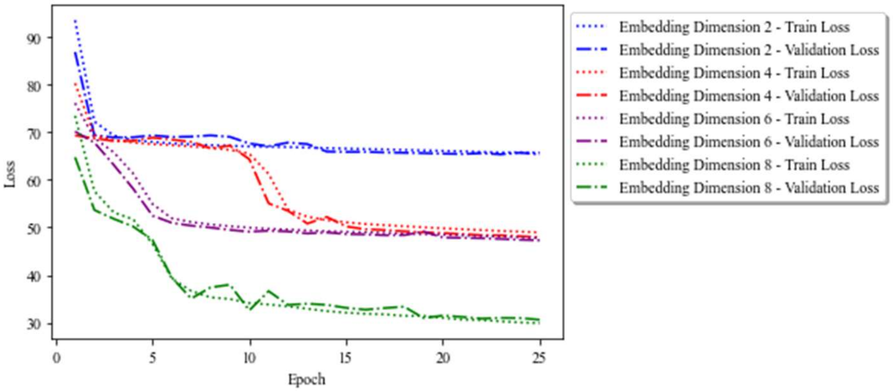
Normal Recall: 59/145 (40.69%)
Anomaly Recall: 143/145 (98.62%)

Embedding Dimension 8



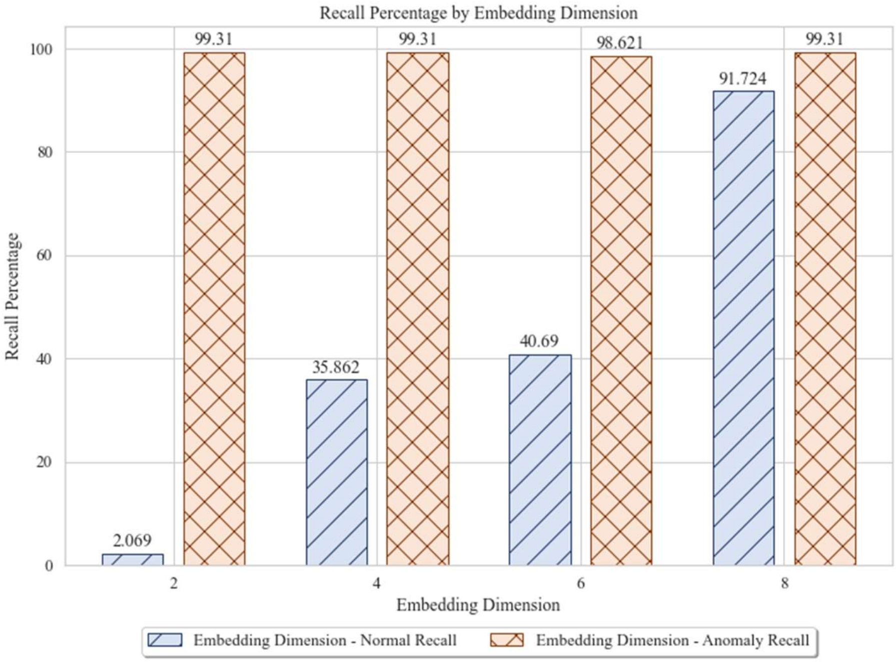
Normal Recall: 133/145 (91.72%)
Anomaly Recall: 144/145 (99.31%)

    b) Briefly explain the trend you see in the training and validation loss.



When the epoch level is fixed, as the embedding dimension is increased, the loss is significantly decreased. As the embedding dimension is increased, in the bottle neck structure, since the encoder does not reduce the array to a small dimension, the decoder does not produce significant loss.

c) Compute the proportion of normal and abnormal time-series correctly classified (i.e., Recall) for the same test set in Q.1 above for each of the embedding dimension values from (a). You can set the threshold to 45.



d) Briefly explain the trend you see in the Recall in part (c) above.

When the threshold is fixed, as the embedding dimension is increased, normal recall increases. As the embedding dimension is increased, loss decreases; therefore, records in the past step that surpassed the threshold, may not break the threshold in the current step since the loss is reduced.