DSC 275/475: Time Series Analysis and Forecasting

(Fall 2022)

Project 3.2 – LSTM-based Auto-encoders

Total points: 50

Submitted by: Aradhya Mathur and Lakshmi Nikhil Goduguluri

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 heartbeats (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:

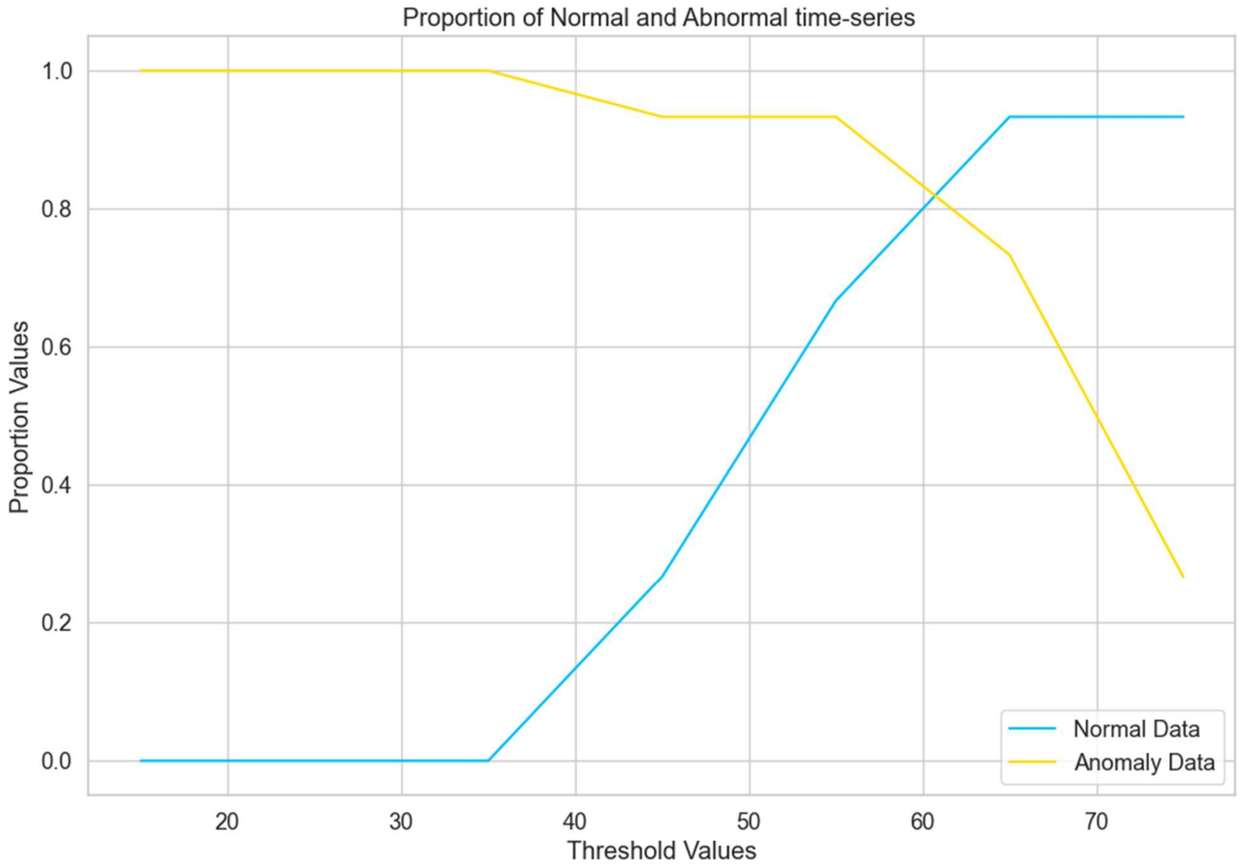
1. ***A critical hyper-parameter when using auto-encoders is the threshold applied to the reconstructed time-series to classify between normal and abnormal. The default threshold in the code is set to 45. Run the code for 50 epochs.***

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. ***(10 points)***

Answer) Table and Graph containing proportion of Normal and Anomaly for Embedding dimension 8:

Threshold Values	Proportion of Normal	Proportion of Abnormal
15	0.0000	1.0000
25	0.0000	1.0000
35	0.0000	1.0000

45	**0.2667**	**0.9333**
55	**0.6667**	**0.9333**
65	**0.9333**	**0.7333**
75	**0.9333**	**0.2667**



b) Briefly explain the trend you see in the recall values as you increase the threshold.
(5 points)

Answer) *It is quite evident that the recall values show increasing trend in case of correct proportion for Normal Data, while it shows decreasing trend in case of correct proportion for Anomaly Data.*

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. (15 points)***

   Answer) Table containing training and validation loss after 25 epochs, threshold 45**:**

Embedding Dimension	Training Loss	Validation Loss
2	70.06598649486419	68.05556606424265
4	68.23039791660923	67.23227454876077
6	64.09545456978583	62.55447374541184
8	53.69145231862222	52.80211494708883

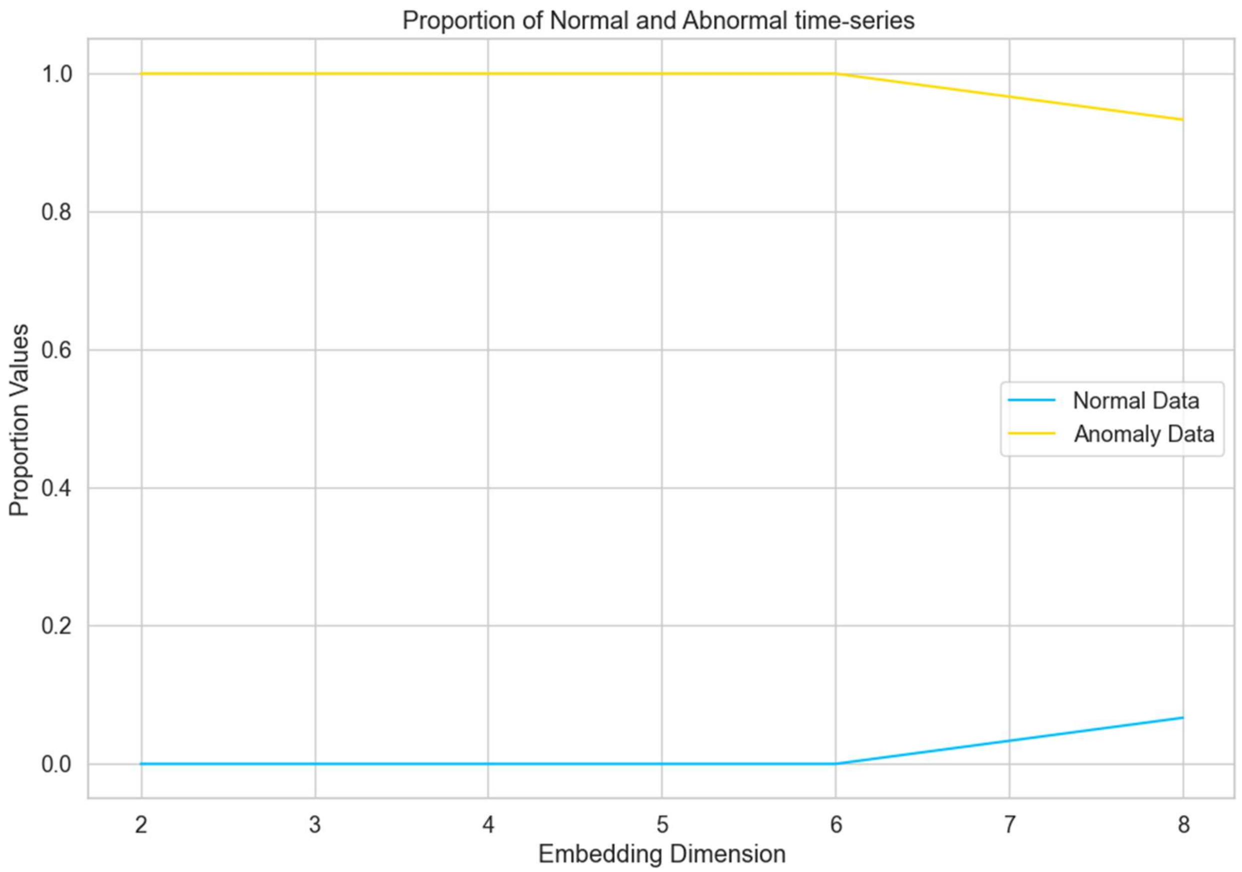
   b) ***Briefly explain the trend you see in the training and validation loss (5 points)***

   Answer) *It can be evidently observed that training and validation loss show decreasing trend with respect to increase in embedding dimension from 2 to 8. Although there is not much of a difference for dimension 2 and dimension 4 but still there is a decrease.*

   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. (10 points)***

   Answer) Table and Graph containing proportion of Normal and Anomaly for threshold 45:

Embedding Dimension	Proportion of Normal	Proportion of Abnormal
2	0.000000	1.000000
4	0.000000	1.000000
6	0.000000	1.000000
8	0.066667	0.933333



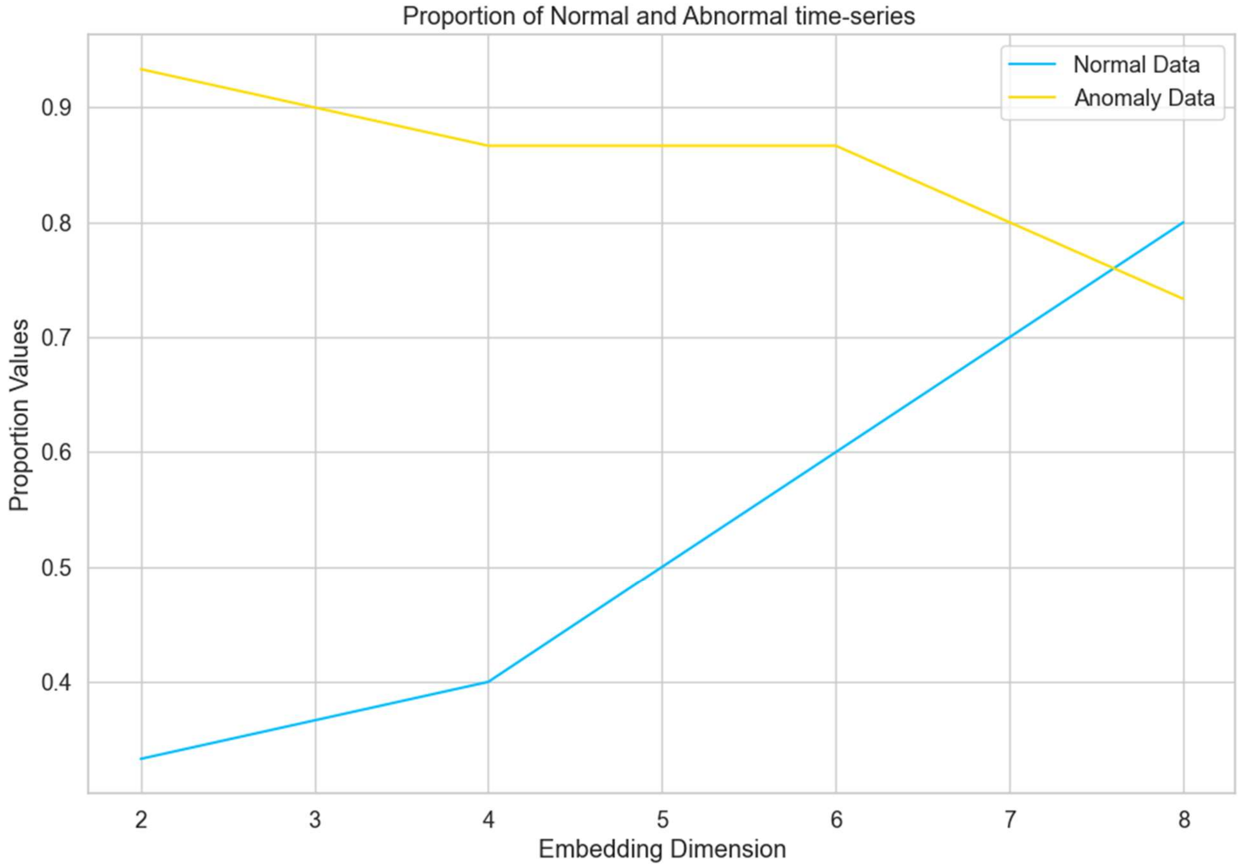
d) ***Briefly explain the trend you see in the Recall in part (c) above*** **(5 points)**

Answer) *It is evident from the above graph that for constant threshold of 45, proportion of normal and anomaly remains constant 0 and 1 respectively till embedding dimension of 6. For embedding dimension 8, proportion of Normal increases to 0.06667 and proportion of Anomaly decreases to 0.93333. Ideally, normal value should increase and anomaly value should decrease with respect to increase in embedding layer. In my opinion, threshold 45 is little less to compare for embedding layer 2,4 and 6. It was clearly observed that normal increases and anomaly decreases with increase in embedding layer from 2 to 8 for threshold 65 (additional file submitted).*

Supporting information:

Table and Graph containing proportion of Normal and Anomaly for threshold 65:

Embedding Dimension	Proportion of Normal	Proportion of Abnormal
2	0.333333	0.933333
4	0.400000	0.866667
6	0.600000	0.866667
8	0.800000	0.733333



It is evident that proportion of Normal increases and proportion of Anomaly decreases with respect to increase in embedding dimension from 2 to 8 for Threshold = 65