Annotating antibiotic-resistance protein in PPI network using GNN

NGOC KHOI DANG – JUN CHEN

CS 6824 – VIRGINIA TECH – SPRING 2022

Problem

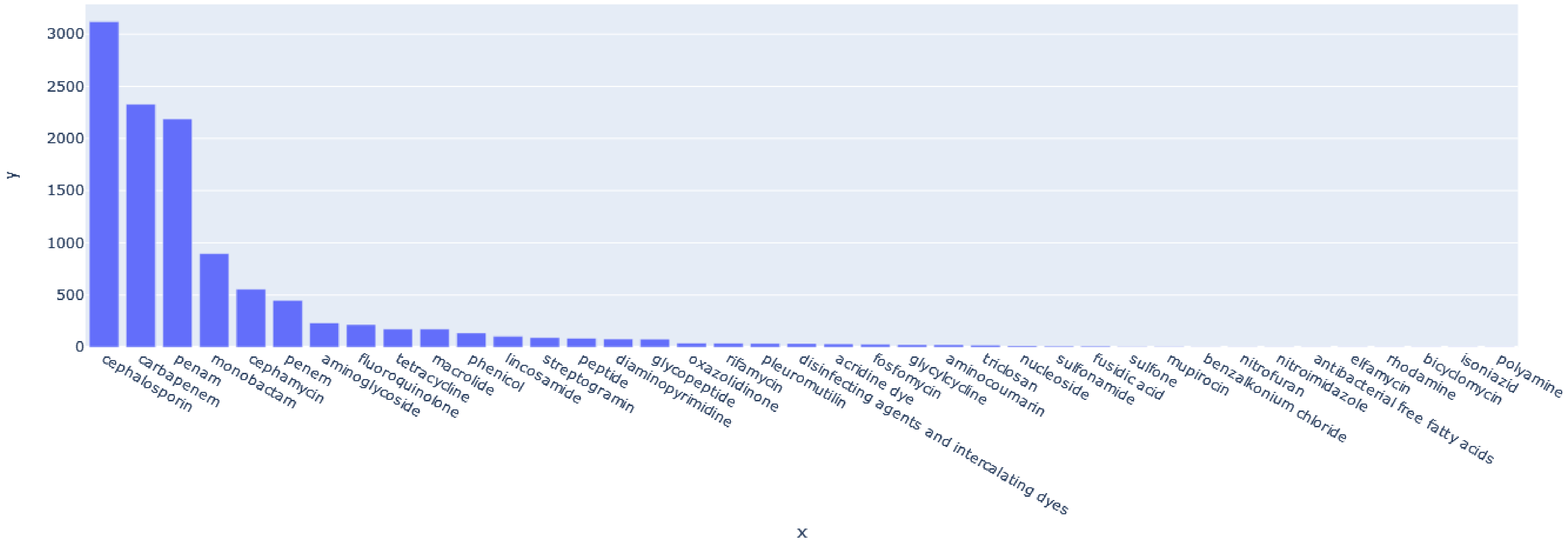
- Antibiotic resistance is currently a global threat spanning clinical, environmental, and geopolitical research domains. The environment is increasingly recognized as a key node in the spread of antibiotic resistance genes, which confer antibiotic resistance to bacteria.
- The protein-protein interaction (PPI) network of an organism serves as a skeleton for its signaling circuitry, which mediates cellular response to environmental and genetic cues. Understanding this circuitry could improve the prediction of gene function and cellular behavior in response to diverse signals.
- In this project, we develop a computational pipeline to annotate antibiotic-resistance proteins (ARPs) in the PPI network based on the known ARPs and graph neural network (GNN) model

Method

- Data:
  - PPI network: STRING v11.5. The STRING database currently covers 67M proteins from 14k organisms and 20B interactions.
  - ARPs database: CARD v3.2.1. The CARD is a rigorously curated collection of characterized, peer-reviewed resistance determinants and associated antibiotics. CARD has 4605 ARPs in 39 classes. Some ARPs are labeled in multiple classes.
  - GNN: Graph neural networks are a supervised learning algorithm that can extract important information from graphs and make useful predictions. Based on the edges information from PPI and nodes information from CARD, we expect to predict the AR categories of unlabeled nodes in the network

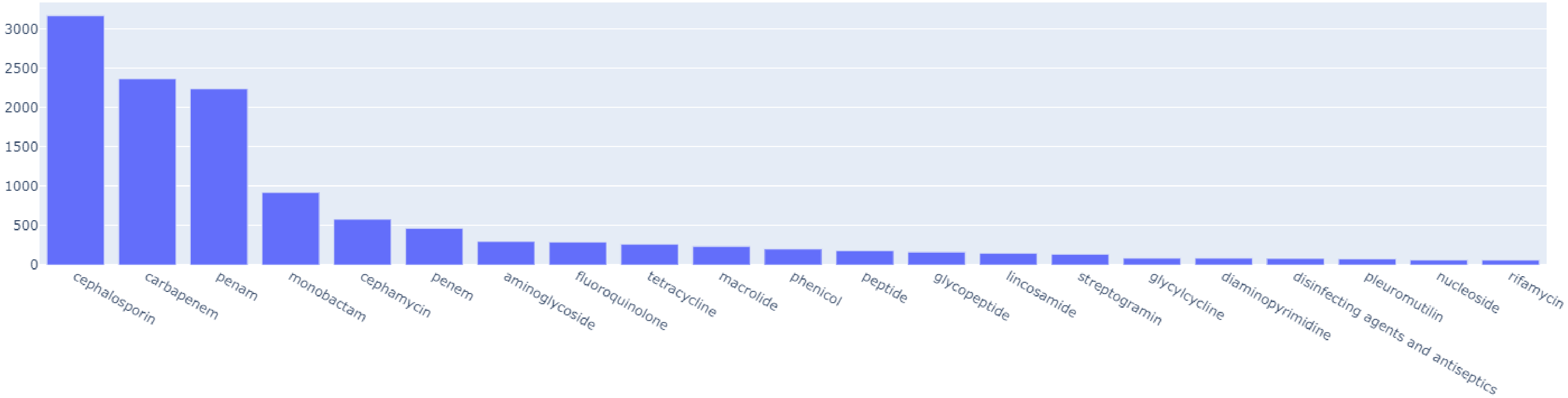
Preprocessing CARD

- CARD has 39 classes but the data is extremely imbalance



Preprocessing CARD

- After removing minorities and merging classes => We have 21 classes



Preprocessing network

1) Alignment: We use BLASTP to align proteins in PPI against CARD database (default settings)

2) Design the network:

   a) Nodes: proteins from STRING

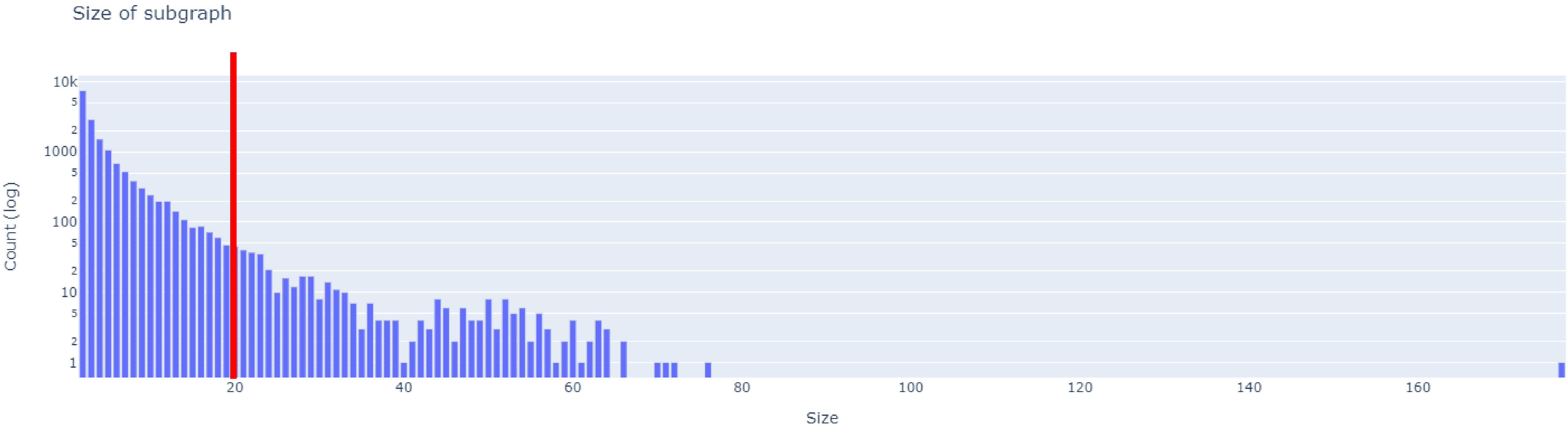
      - Label: defined by the alignment with > 75% identity
      - Features: 21 alignment bit-score features and 20 amino acid compositions features (values from 0-1)

   b) Edges: evidence score from STRING

      - Feature: 1 if it has the fusion, co-occurrence or neighborhood evidence, otherwise 0

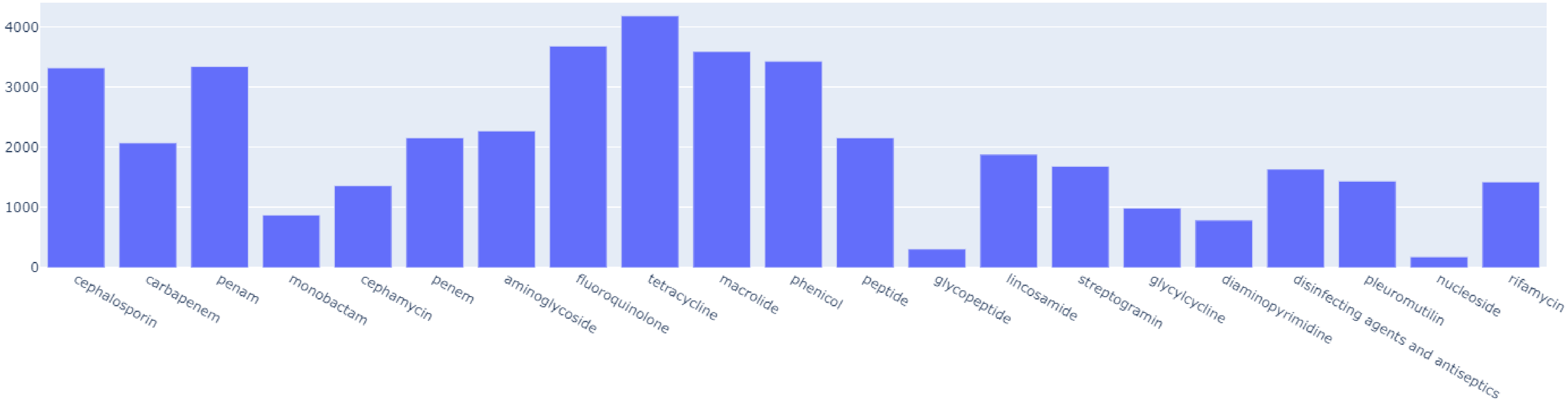
The network has many small disconnected subgraphs!

- Network summary: nodes: 78,531, edges: 116,196, average degree: 2.9592



Cleaning the network

- Remove the small subgraph (<20 nodes)
- Add pseudo-edges that connect subgraphs together (weight = 0.001)
- Network summary: nodes: 13,794, edges: 54,951 average degree: 7.9674

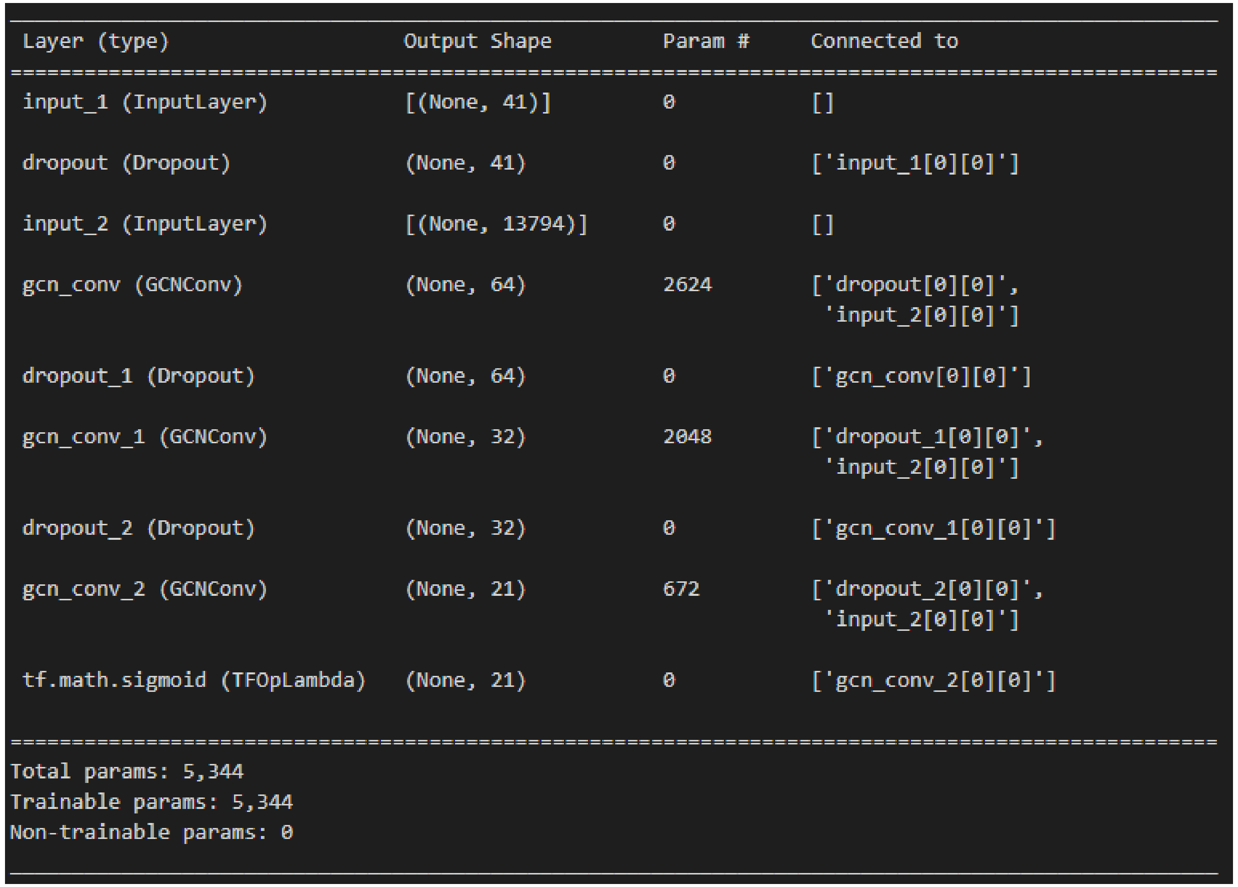


GNN model

- Input data is divided into 3 groups:

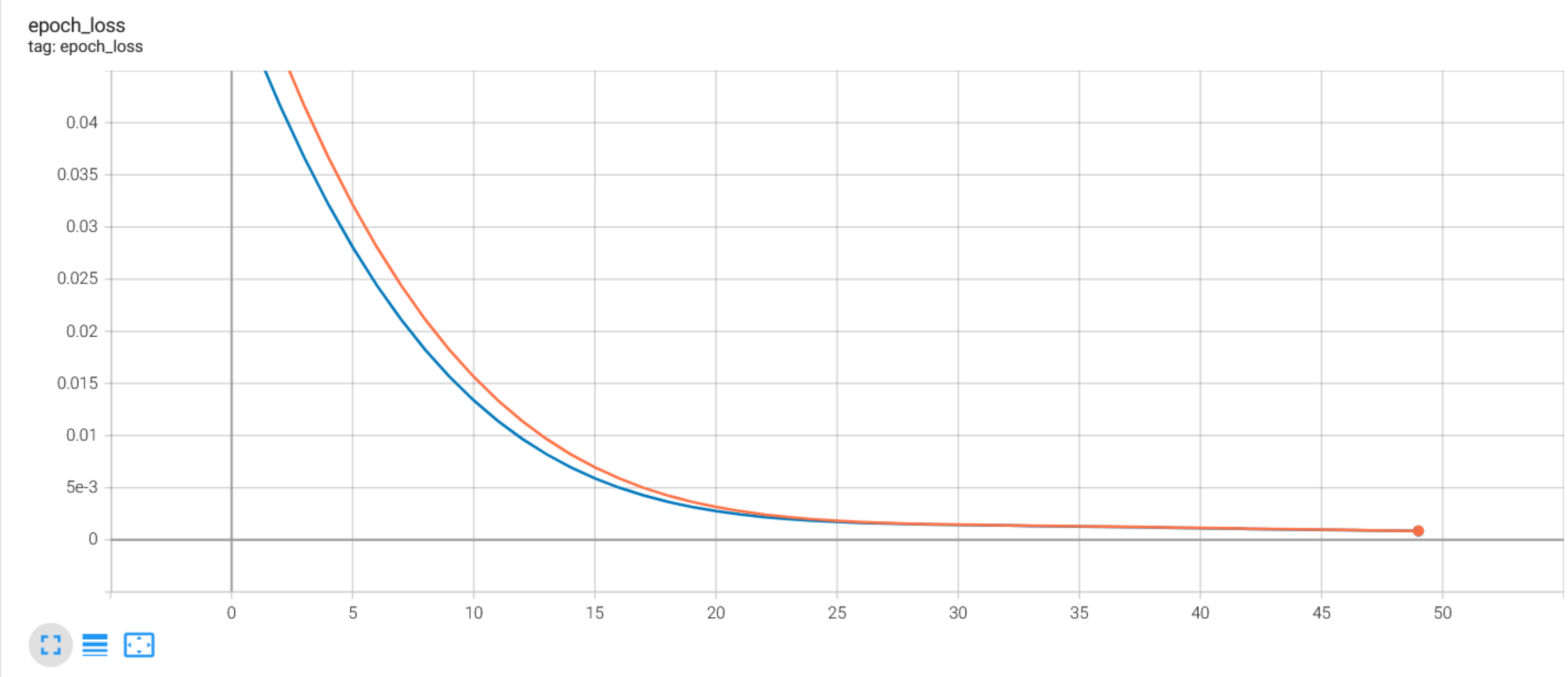
  training : validating : testing with ratio 7:2:1

- Model has three layers
- Parameters:
  - Number of channels in the first layer: 64
  - Number of channels in the second layer: 32
  - Dropout rate: 0.5
  - L2 – regularization: 5e-4
  - Learning rate: 1e-2

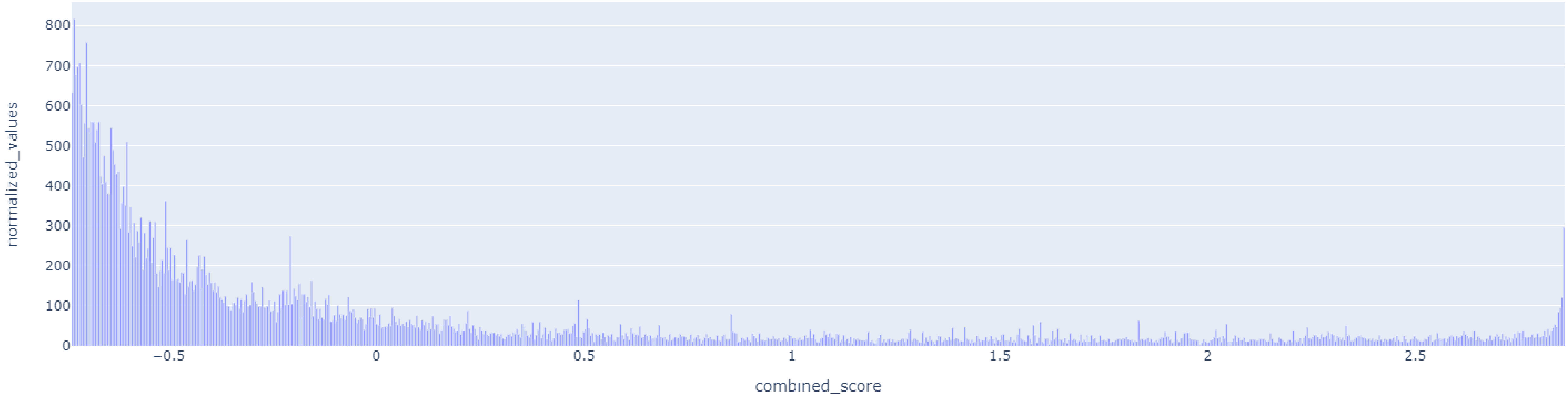


```
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 41)]         0           []

 dropout (Dropout)              (None, 41)           0           ['input_1[0][0]']

 input_2 (InputLayer)           [(None, 13794)]      0           []

 gcn_conv (GCNConv)             (None, 64)           2624        ['dropout[0][0]',
                                                                  'input_2[0][0]']

 dropout_1 (Dropout)            (None, 64)           0           ['gcn_conv[0][0]']

 gcn_conv_1 (GCNConv)           (None, 32)           2048        ['dropout_1[0][0]',
                                                                  'input_2[0][0]']

 dropout_2 (Dropout)            (None, 32)           0           ['gcn_conv_1[0][0]']

 gcn_conv_2 (GCNConv)           (None, 21)           672         ['dropout_2[0][0]',
                                                                  'input_2[0][0]']

 tf.math.sigmoid (TFOpLambda)   (None, 21)           0           ['gcn_conv_2[0][0]']

==================================================================================================
Total params: 5,344
Trainable params: 5,344
Non-trainable params: 0
__________________________________________________________________________________________________
```

Training


epoch_loss
tag: epoch_loss
0.04
0.035
0.03
0.025
0.02
0.015
0.01
5e-3
0
0
5
10
15
20
25
30
35
40
45
50

Edge-weight distribution



Edge feature:

- 1: 30,886
- 0: 23,537

Evaluating

Prediction