By using the method given in the paper `Stable Biomaker` on our data, it seems really inefficient. The method seems to be designed for data in small scale, as for our data, which is much larger than the data used in the paper, it will take roughly a week running on my own computer (with CPU i9-9980hk). The running time will grow exponentially longer as the size of data doubles. In order to evaluate the performance, each time I randomly select 40 data cases from the data , 20 control data and 20 tinnitus data and repeat the experiment for several times. The best accuracy (output as `mean_acc` in the code provided) is about 58%, while the accuracy rate of these groups vaaries from about 51% to about 58%, with an average of about 55%.

In my own model, I build a 3-layer neural network, while for each layer, the data pass through a nn.linear layer and then pass through a relu activation function. The detailed parameters that have the best performance can be seem in the code. Each time randomly select half of the data as training data and the rest as testing data. CrossEntropy loss is employed and we train the data in full batch since the data size is not larger enough to have requirements for minibatches. The accuracy is shown in the picture below, which is mostly more than 60%.

```
train_acc 0 0.5
test_acc 0 0.510752688172043
train_acc 10 0.5
test_acc 10 0.510752688172043
train_acc 20 0.5
test_acc 20 0.510752688172043
train_acc 30 0.9666666666666667
test_acc 30 0.5806451612903226
train_acc 40 0.9571428571428572
test_acc 40 0.6021505376344086
train_acc 50 0.9761904761904762
test_acc 50 0.6182795698924731
```

```
test_acc 0 0.489247311827957
test_acc 5 0.489247311827957
train_acc 10 0.5
test_acc 10 0.489247311827957
test_acc 15 0.489247311827957
train_acc 20 0.5523809523809524
test_acc 20 0.489247311827957
test_acc 25 0.6021505376344086
train_acc 30 0.9761904761904762
test_acc 30 0.6236559139784946
test_acc 35 0.6559139784946236
train_acc 40 0.9952380952380953
test_acc 40 0.6236559139784946
test_acc 45 0.6397849462365591
train_acc 50 0.9904761904761905
test_acc 50 0.6505376344086021
test_acc 55 0.6505376344086021
```

```
train_acc 0 0.5
test_acc 0 0.510752688172043
train_acc 5 0.5
test_acc 5 0.510752688172043
train_acc 10 0.5
test_acc 10 0.510752688172043
train_acc 15 0.5
test_acc 15 0.510752688172043
```

```
train_acc 20 0.5
test_acc 20 0.510752688172043
train_acc 25 0.5095238095238095
test_acc 25 0.5053763440860215
train_acc 30 0.9714285714285714
test_acc 30 0.5698924731182796
train_acc 35 0.9428571428571428
test_acc 35 0.5698924731182796
train_acc 40 0.9666666666666667
test_acc 40 0.5860215053763441
train_acc 45 0.9714285714285714
test_acc 45 0.5967741935483871
train_acc 50 0.9714285714285714
test_acc 50 0.5967741935483871
train_acc 55 0.9761904761904762
test_acc 55 0.6182795698924731
train_acc 60 0.9857142857142858
test_acc 60 0.6021505376344086
train_acc 65 0.9904761904761905
test_acc 65 0.6021505376344086
train_acc 70 1.0
test_acc 70 0.6129032258064516
train_acc 75 1.0
test_acc 75 0.5860215053763441
train_acc 80 1.0
```

```
test_acc 80 0.6021505376344086
train_acc 85 1.0
test_acc 85 0.6129032258064516
train_acc 90 1.0
test_acc 90 0.6129032258064516
train_acc 95 1.0
test_acc 95 0.6129032258064516
```

```
train_acc 0 0.5
test_acc 0 0.489247311827957
train_acc 5 0.5
test_acc 5 0.489247311827957
train_acc 10 0.5
test_acc 10 0.489247311827957
train_acc 15 0.6476190476190476
test_acc 15 0.5053763440860215
train_acc 20 0.8761904761904762
test_acc 20 0.5860215053763441
train_acc 25 0.9
test_acc 25 0.6021505376344086
train_acc 30 0.9190476190476190
test_acc 30 0.5967741935483871
train_acc 35 0.9428571428571428
test_acc 35 0.5806451612903226
train_acc 40 0.9428571428571428
test_acc 40 0.5860215053763441
train_acc 45 0.9619047619047619
test_acc 45 0.5806451612903226
train_acc 50 0.9904761904761905
test_acc 50 0.5967741935483871
train_acc 55 0.9952380952380953
test_acc 55 0.6182795698924731
train_acc 60 0.9952380952380953
test_acc 60 0.6505376344086021
train_acc 65 0.9952380952380953
test_acc 65 0.6397849462365591
train_acc 70 0.9952380952380953
test_acc 70 0.6344086021505376
train_acc 75 0.9952380952380953
test_acc 75 0.6236559139784946
train_acc 80 0.9952380952380953
test_acc 80 0.6236559139784946
```

```
train_acc 85 0.9952380952380953
test_acc 85 0.6075268817204301
train_acc 90 0.9952380952380953
test_acc 90 0.6021505376344086
train_acc 95 0.9952380952380953
test_acc 95 0.6129032258064516
train_acc 100 0.9952380952380953
test_acc 100 0.6182795698924731
train_acc 105 0.9952380952380953
test_acc 105 0.6236559139784946
train_acc 110 0.9952380952380953
test_acc 110 0.6182795698924731
train_acc 115 0.9952380952380953
```

```
train_acc 0 0.5
test_acc 0 0.510752688172043
train_acc 5 0.5
test_acc 5 0.510752688172043
train_acc 10 0.5047619047619047
test_acc 10 0.510752688172043
train_acc 15 0.8238095238095238
test_acc 15 0.5376344086021505
train_acc 20 0.9285714285714286
test_acc 20 0.5752688172043011
train_acc 25 0.9380952380952381
test_acc 25 0.5967741935483871
train_acc 30 0.9190476190476190
test_acc 30 0.5376344086021505
train_acc 35 0.9238095238095239
test_acc 35 0.5591397849462365
train_acc 40 0.9380952380952381
test_acc 40 0.5645161290322581
train_acc 45 0.9714285714285714
test_acc 45 0.553763440860215
train_acc 50 0.9952380952380953
test_acc 50 0.5698924731182796
train_acc 55 0.9952380952380953
test_acc 55 0.5913978494623656
train_acc 60 0.9952380952380953
test_acc 60 0.5806451612903226
train_acc 65 0.9952380952380953
test_acc 65 0.5806451612903226
train_acc 70 0.9952380952380953
test_acc 70 0.5752688172043011
train_acc 75 0.9952380952380953
test_acc 75 0.5860215053763441
```

```
train_acc 80 0.9952380952380953
test_acc 80 0.5913978494623656
train_acc 85 0.9952380952380953
test_acc 85 0.5806451612903226
train_acc 90 0.9952380952380953
test_acc 90 0.5860215053763441
train_acc 95 0.9952380952380953
test_acc 95 0.5860215053763441
train_acc 100 0.9952380952380953
test_acc 100 0.5860215053763441
train_acc 105 0.9952380952380953
test_acc 105 0.5860215053763441
train_acc 110 0.9952380952380953
test_acc 110 0.5860215053763441
train_acc 115 0.9952380952380953
```

```
train_acc 80 0.9619047619047619
test_acc 80 0.543010752688172
train_acc 85 0.9952380952380953
test_acc 85 0.5483870967741935
train_acc 90 0.9952380952380953
test_acc 90 0.521505376344086
train_acc 95 0.9952380952380953
test_acc 95 0.5806451612903226
train_acc 100 0.9952380952380953
test_acc 100 0.5645161290322581
train_acc 105 0.9952380952380953
test_acc 105 0.5752688172043011
train_acc 110 0.9952380952380953
test_acc 110 0.6397849462365591
train_acc 115 0.9952380952380953
test_acc 115 0.5645161290322581
train_acc 120 0.9952380952380953
```

```
test_acc 120 0.6559139784946236
train_acc 125 0.9952380952380953
test_acc 125 0.6559139784946236
train_acc 130 0.9952380952380953
test_acc 130 0.6021505376344086
train_acc 135 0.9952380952380953
test_acc 135 0.6182795698924731
train_acc 140 0.9952380952380953
test_acc 140 0.6290322580645161
train_acc 145 0.9952380952380953
test_acc 145 0.6559139784946236
train_acc 150 0.9952380952380953
test_acc 150 0.6666666666666666
train_acc 155 0.9952380952380953
test_acc 155 0.6559139784946236
train_acc 160 0.9952380952380953
test_acc 160 0.6505376344086021
train_acc 165 0.9952380952380953
test_acc 165 0.6505376344086021
train_acc 170 0.9952380952380953
test_acc 170 0.6505376344086021
train_acc 175 0.9952380952380953
test_acc 175 0.6559139784946236
train_acc 180 0.9952380952380953
test_acc 180 0.6505376344086021
```

```
train_acc 185 0.9952380952380953
test_acc 185 0.6666666666666666
train_acc 190 0.9952380952380953
test_acc 190 0.6666666666666666
train_acc 195 0.9952380952380953
test_acc 195 0.6666666666666666
```

```
train_acc 0 0.5
test_acc 0 0.489247311827957
train_acc 5 0.5
test_acc 5 0.489247311827957
train_acc 10 0.5
test_acc 10 0.489247311827957
train_acc 15 0.5047619047619047
test_acc 15 0.489247311827957
train_acc 20 0.8095238095238095
test_acc 20 0.5268817204301075
train_acc 25 0.9238095238095239
test_acc 25 0.5860215053763441
train_acc 30 0.9285714285714286
test_acc 30 0.5698924731182796
train_acc 35 0.9333333333333333
test_acc 35 0.5752688172043011
train_acc 40 0.9619047619047619
test_acc 40 0.5645161290322581
train_acc 45 0.9666666666666667
test_acc 45 0.5860215053763441
train_acc 50 0.9809523809523809
test_acc 50 0.5860215053763441
train_acc 55 0.9904761904761905
test_acc 55 0.5806451612903226
```

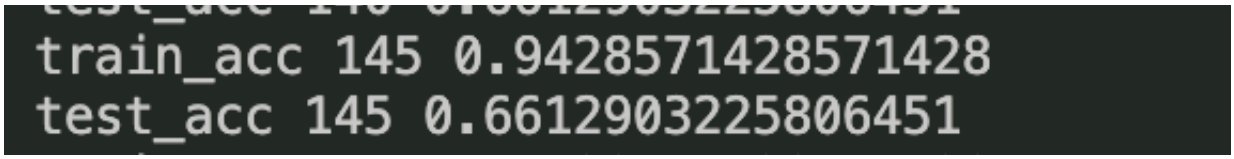
```
test_acc 90 0.6075268817204301
```

```
train_acc 95 1.0
test_acc 95 0.612903225806451 6
train_acc 100 1.0
test_acc 100 0.591397849462365 6
train_acc 105 1.0
test_acc 105 0.612903225806451 6
train_acc 110 1.0
test_acc 110 0.612903225806451 6
train_acc 115 1.0
test_acc 115 0.607526881720430 1
train_acc 120 1.0
test_acc 120 0.607526881720430 1
train_acc 125 1.0
test_acc 125 0.612903225806451 6
train_acc 130 1.0
test_acc 130 0.602150537634408 6
train_acc 135 1.0
test_acc 135 0.607526881720430 1
train_acc 140 1.0
test_acc 140 0.607526881720430 1
train_acc 145 1.0
test_acc 145 0.607526881720430 1
train_acc 150 1.0
test_acc 150 0.602150537634408 6
train_acc 155 1.0
test_acc 155 0.607526881720430 1
train_acc 160 1.0
test_acc 160 0.607526881720430 1
train_acc 165 1.0
test_acc 165 0.602150537634408 6
train_acc 170 1.0
test_acc 170 0.602150537634408 6
train_acc 175 1.0
test_acc 175 0.602150537634408 6
```

```
train_acc 180 1.0
test_acc 180 0.602150537634408 6
train_acc 185 1.0
test_acc 185 0.602150537634408 6
train_acc 190 1.0
test_acc 190 0.602150537634408 6
train_acc 195 1.0
test_acc 195 0.602150537634408 6
```

```
tensor(-0.0612)
train_acc 0 0.4761904761904761 6
test_acc 0 0.516129032258064 5
train_acc 5 0.5333333333333333
test_acc 5 0.553763440860215
train_acc 10 0.8142857142857143
test_acc 10 0.5698924731182796
train_acc 15 0.9285714285714286
test_acc 15 0.6182795698924731
train_acc 20 0.9380952380952381
test_acc 20 0.6612903225806451
train_acc 25 0.9428571428571428
test_acc 25 0.6666666666666666
train_acc 30 0.9428571428571428
test_acc 30 0.6774193548387096
train_acc 35 0.9428571428571428
test_acc 35 0.6827956989247311
train_acc 40 0.9380952380952381
test_acc 40 0.7043010752688172
train_acc 45 0.9428571428571428
test_acc 45 0.6720430107526881
train_acc 50 0.9428571428571428
test_acc 50 0.6720430107526881
train_acc 55 0.9428571428571428
test_acc 55 0.6666666666666666
```

```
train_acc 60 0.9428571428571428
test_acc 60 0.6720430107526881
train_acc 65 0.9428571428571428
test_acc 65 0.6666666666666666
train_acc 70 0.9428571428571428
test_acc 70 0.6666666666666666
train_acc 75 0.9428571428571428
test_acc 75 0.6720430107526881
train_acc 80 0.9428571428571428
test_acc 80 0.6720430107526881
train_acc 85 0.9428571428571428
test_acc 85 0.6612903225806451
train_acc 90 0.9428571428571428
test_acc 90 0.6559139784946236
train_acc 95 0.9428571428571428
test_acc 95 0.6559139784946236
train_acc 100 0.9428571428571428
test_acc 100 0.6559139784946236
train_acc 105 0.9428571428571428
test_acc 105 0.6666666666666666
train_acc 110 0.9428571428571428
test_acc 110 0.6612903225806451
train_acc 115 0.9428571428571428
test_acc 115 0.6666666666666666
train_acc 120 0.9428571428571428
test_acc 120 0.6666666666666666
train_acc 125 0.9428571428571428
test_acc 125 0.6666666666666666
train_acc 130 0.9428571428571428
test_acc 130 0.6612903225806451
train_acc 135 0.9428571428571428
test_acc 135 0.6612903225806451
train_acc 140 0.9428571428571428
test_acc 140 0.6612903225806451
```



Another significant advantage of the method is running time, it can speed up at least 100x by our model using CPU when comparing to the model provided in the paper. What's more, our model allows us to train and test using coda, which can achieve even higher performance.

Since 60% is still not a satisfying result in 2-classification tasks, maybe a better model is needed. The main difficulty is that for complex networks, the amount of data is far less than required. A possible solution may be adversarial networks, which contain a generator and a discriminator at the same time, maybe we can make use of the generator to produce some useful data.