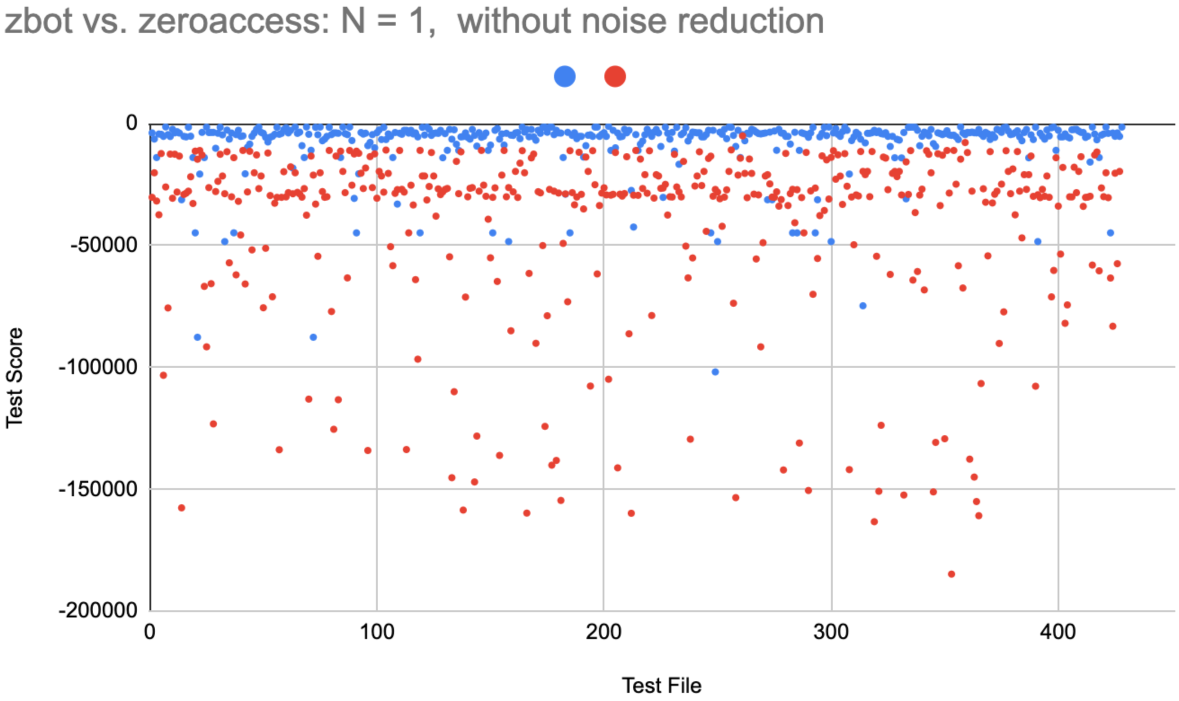
In this project, I implement my own HMM model by using Java language. First, I chose the first 80% of data from zbot folder for HMM model training, then I chose the rest 20% data from zbot, and the same amount of data from zeroaccess for testing. After finishing the data pre processing, I found out that the datasets we applied have 420 types of symbol(opcode), which means M = 420. After training and testing, I obtained the corresponding score of two malware families. Here are several score results under different conditions.

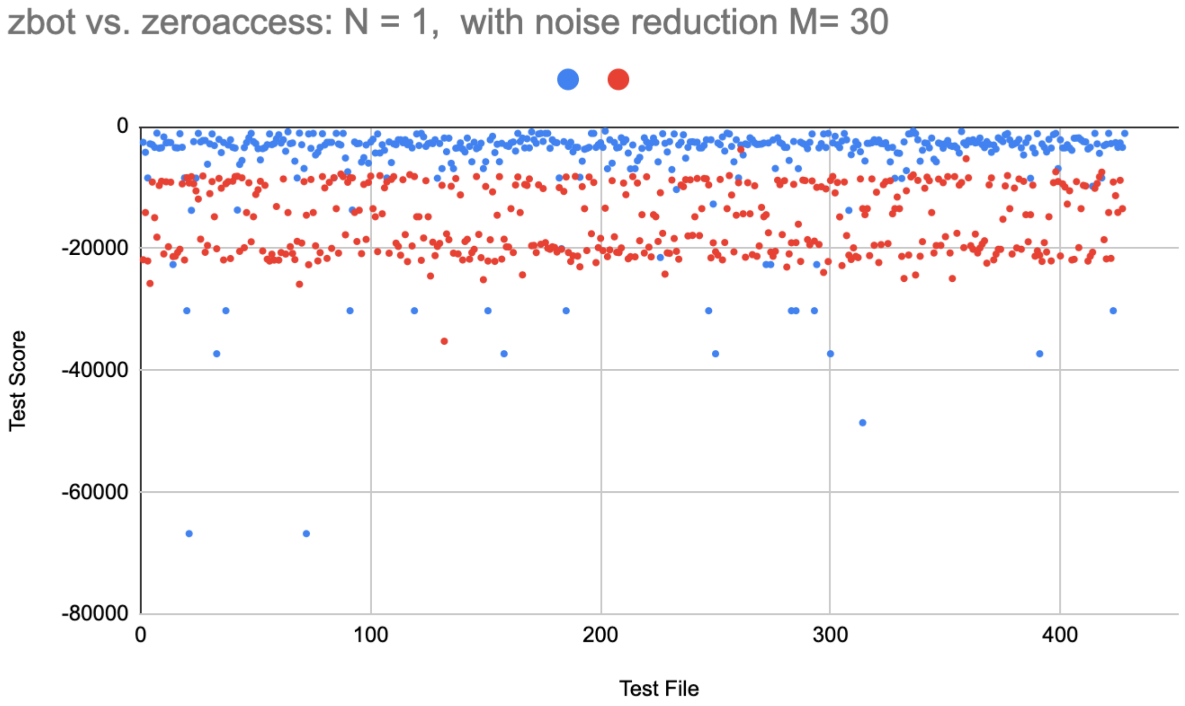
Two distinct malware families:
Blue points: zbot
Red points: zeroaccess

For one hidden state N = 1:
Score result which I applied M without noise reduction is shown as below graph:



From the graph, it shows that there is a threshold line almost separates two groups(distinguished by color blue and red), which implies the HMM model classifies the data from two malware families successfully.

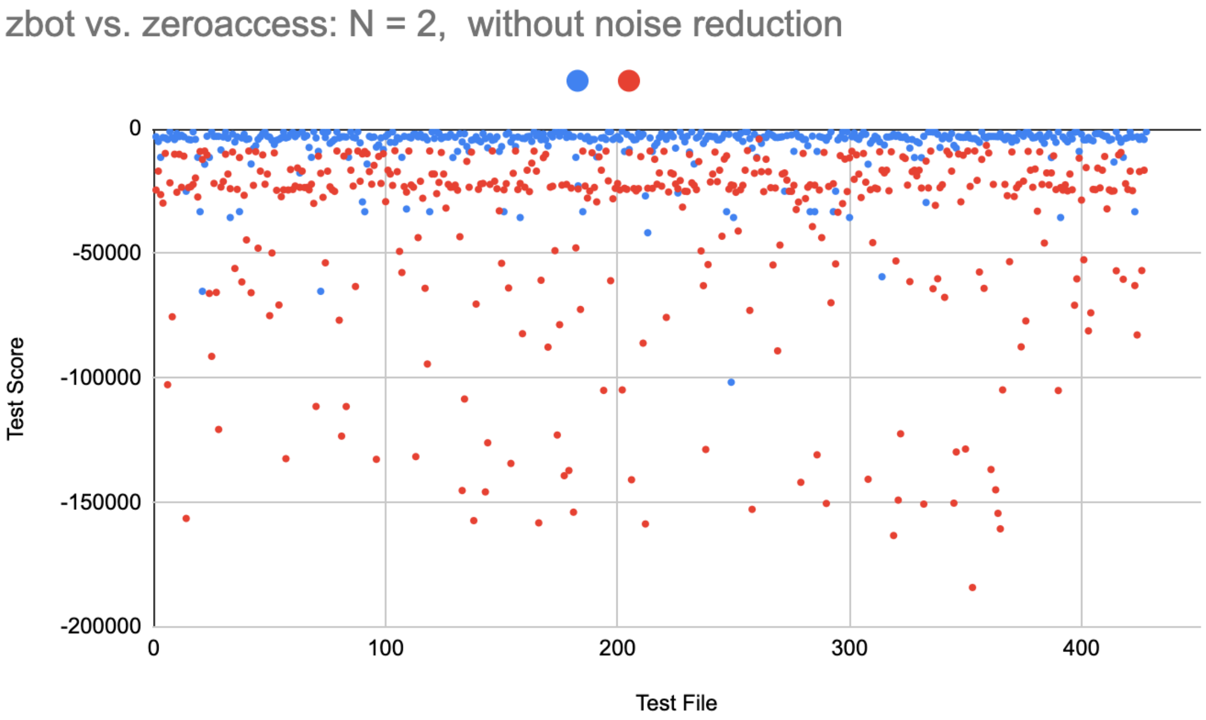
Then in order to reduce some disperse scores those value are very different(smaller in this case) from the class average score, I applied noise reduction by setting M = 30. Result is shown as below graph:



From the graph, it still shows the HMM model classifies the data from two malware families successfully. Also, comparing to the result of the last experiment without appling noise reduction, if I decresed the M value, the test scores range is getting narrow down. Most of the scattered points will gather toward their families/ get closer to the average.

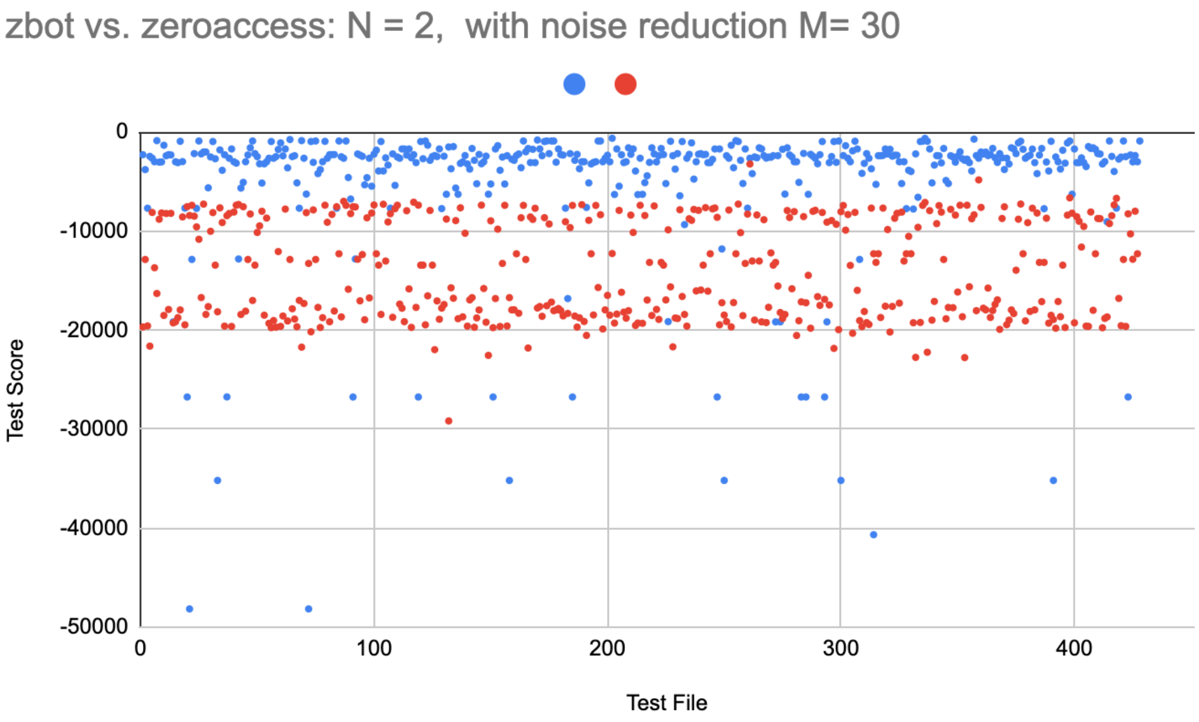
For two hidden state N = 2:

Score result which I applied M without noise reduction is shown as below graph:



Comparing with the result at N = 1(no noise reduction), there is a slight difference: the scores of each family are slightly closer togather at N = 2 than at N = 1.

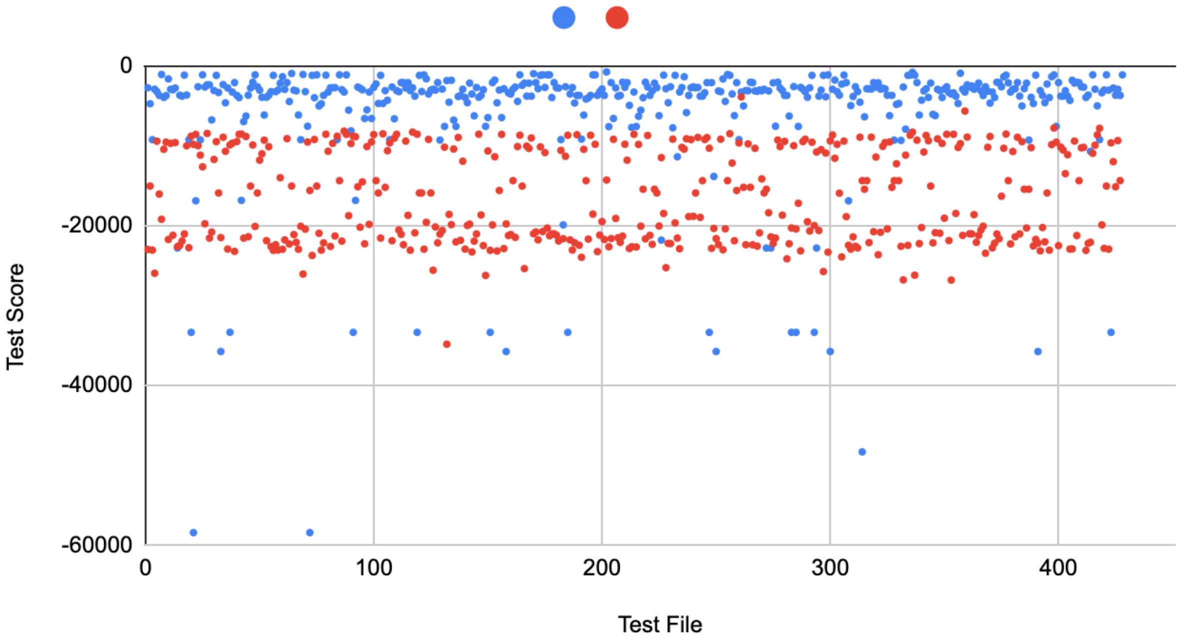
Applied noise reduction by setting M = 30. Result is shown as below graph:



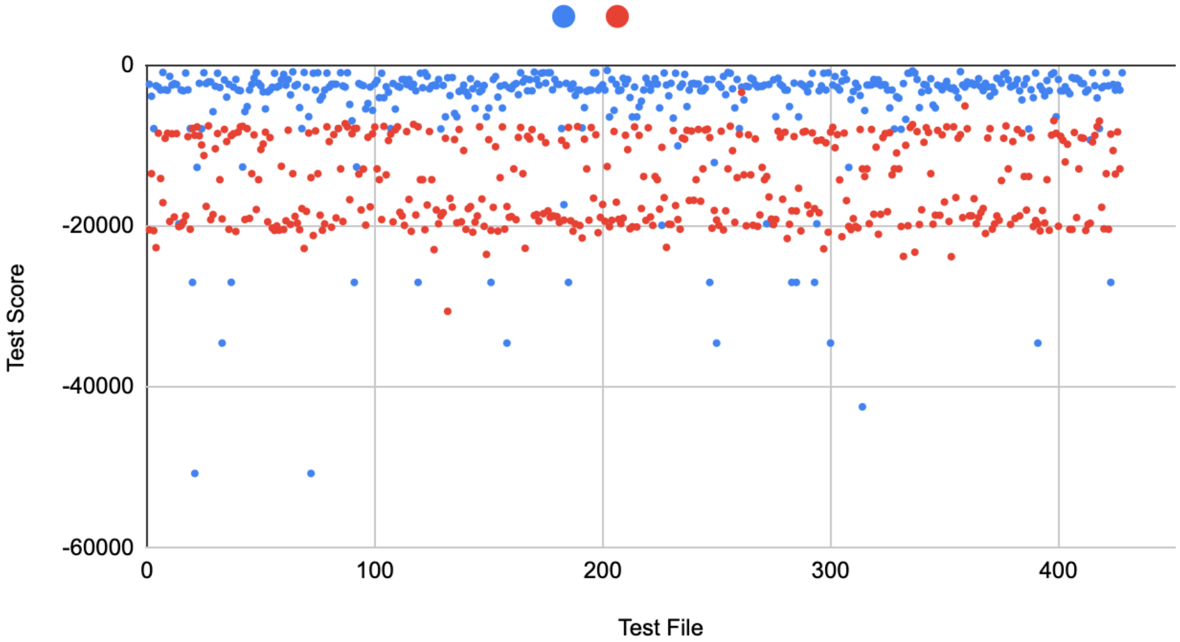
Comparing with the result at N = 1 with noise reduction M = 30, the test scores range get narrow and the score(points) from two families are getting closer at N = 2 than at N = 1.

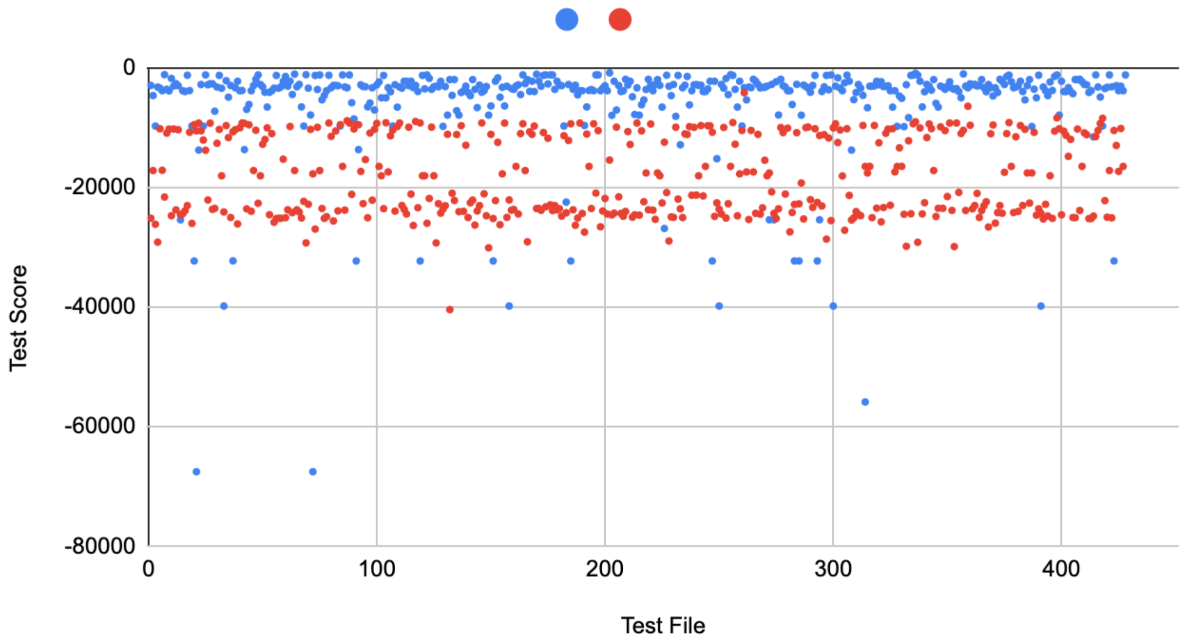
The following graphs show the results of applying different M values for noise reduction:

zbot vs. zeroaccess: N = 1,  with noise reduction M= 31

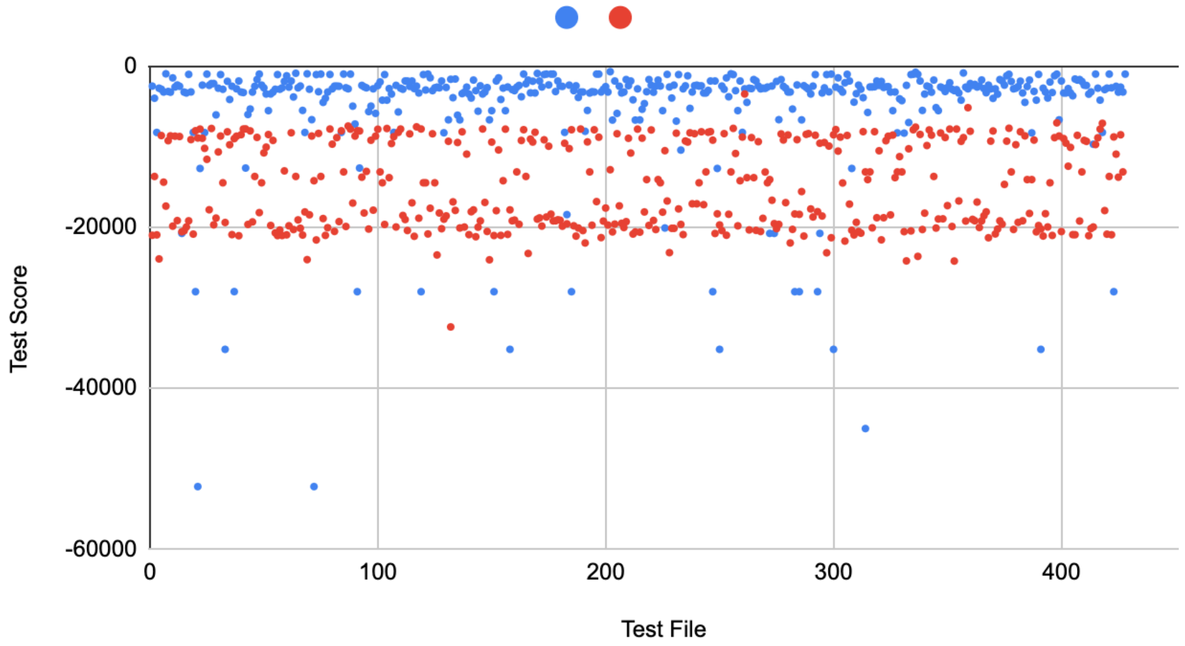


zbot vs. zeroaccess: N = 2,  with noise reduction M= 31

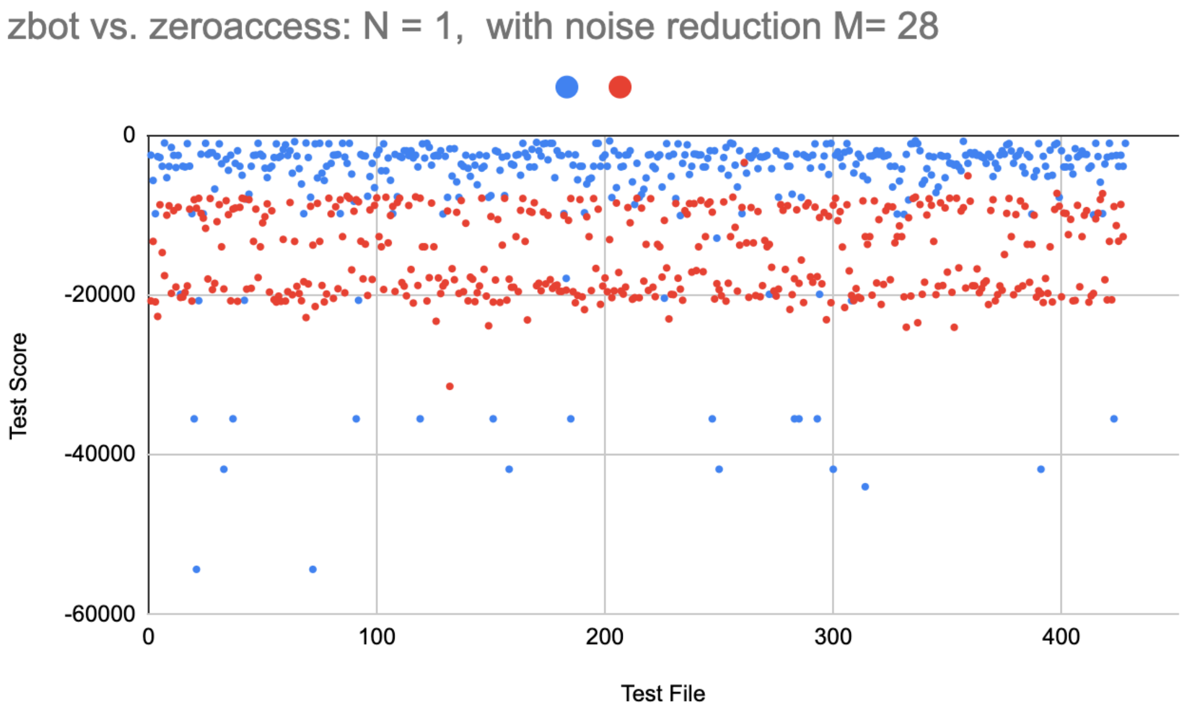


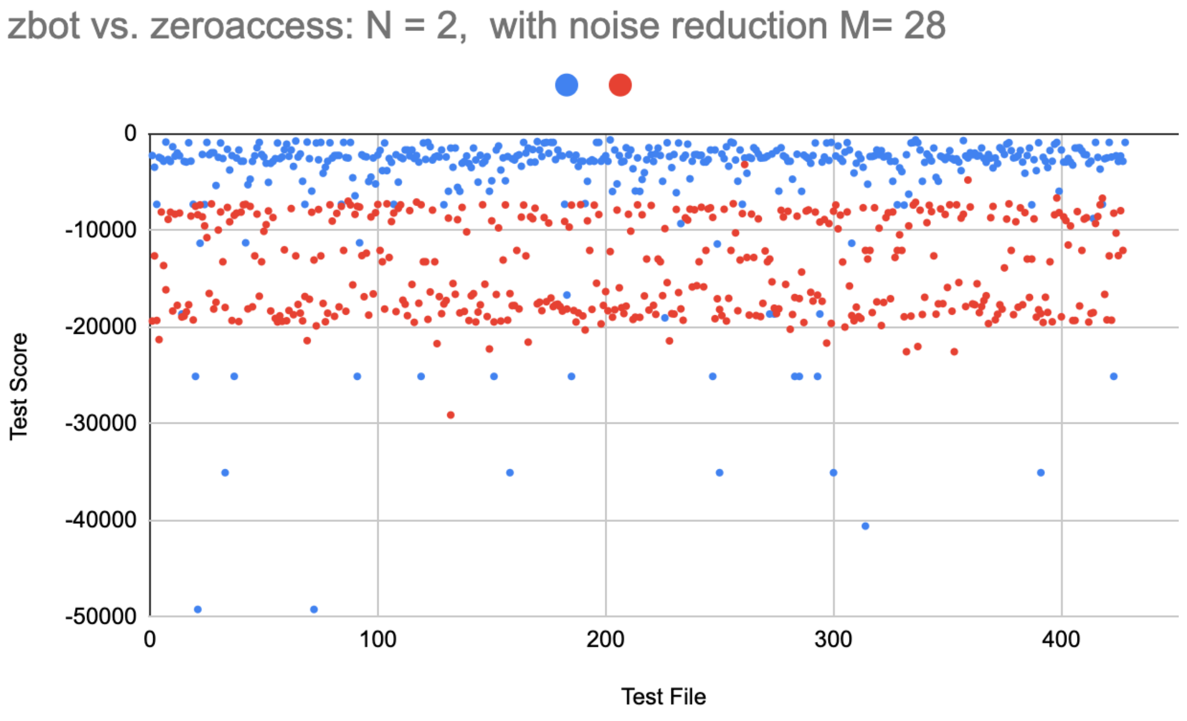


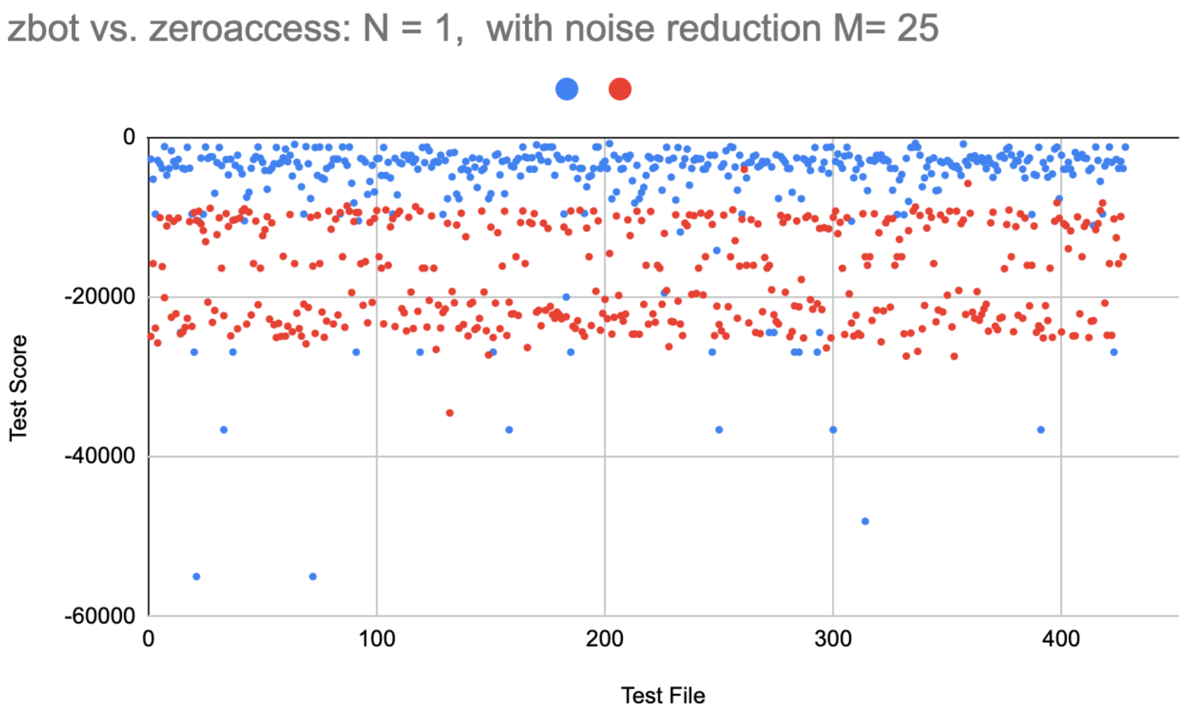
zbot vs. zeroaccess: N = 1,  with noise reduction M= 33

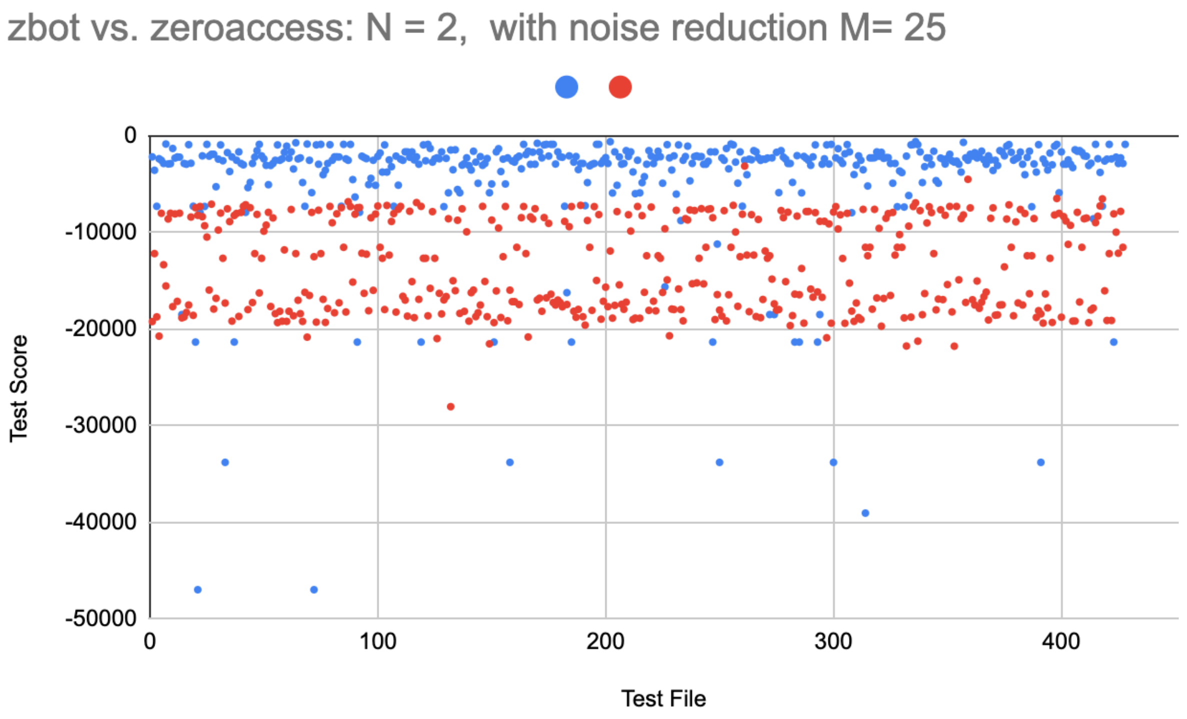


zbot vs. zeroaccess: N = 2,  with noise reduction M= 33


zbot vs. zeroaccess: N = 1,  with noise reduction M= 28
● ●
Test Score
0
-20000
-40000
-60000
0
100
200
300
400
Test File


zbot vs. zeroaccess: N = 2,  with noise reduction M= 28
● ●
Test Score
0
-10000
-20000
-30000
-40000
-50000
0
100
200
300
400
Test File





Through the observation, the model can help narrowing down the range of the test score which focus on classifying the data from two distinct families.

In conclusion, My HMM model can separate the data from different groups after the training process. Condensing the observation category by applying noise reduction method, we can optimaize the result of classifying the distinct classes, especially emphasize the class by concentrate the relevent data obviously.