**Modeling Rabin Fingerprinting using Machine Learning**

**Training Data Creation**

I attempted to create a model that could predict where breakpoints of Rabin fingerprinting would occur in a stream of random data. I first started by creating a data stream that had a length of 10,008 integers. I made my sliding window a length of 8 and made it so that there would be a break on average every 32 integers. A binary label was assigned 1 if a breakpoint occurred and 0 if there was not a breakpoint. This created 10,000 training examples where the input data was the 8 integers, and the output was a 0 or a 1.

**Model Design**

I started with an 80/20 split of training vs testing data. I decided to use a neural network with a structure of the input layer being 1 neuron, the hidden layers having 32 and 16 neurons, and the output layer of 1. All layers had the RelU activation function. The reason the last node was not a sigmoid activation function was because since there was a ratio of 31:1, the model would skew to predict 0 every time. This achieved a high level of accuracy but never predicted when a breakpoint would occur. I had to add a weight that changes the decision boundary based on the ratio of 0’s to 1’s and then using the sigmoid function on the result of the last node, causing more balanced predictions.

**Model Performance**

The total accuracy ends with around 45.65%. This is not an accurate representation of what we are trying to do. If we take the accuracy of true positives (predicted breakpoints) against all breakpoints predicted, the accuracy is 3.5%. This shows that the model does not have any significance as randomly guessing would result in an accuracy of 3.1%. This shows that trying to predict Rabin fingerprinting using an ML approach with *only randomly generated* data is not a feasible method.