## Model Results

* Input: all features (all frequencies, power, rewards, channels, for the time 0-1000 ms) split by the 135 samples.
* Output: Difference in Reaction Time, Standard Deviation for Reward and No Reward, Probability for WSLS
* Losses are stored in box folder (see GRU\_loss\_scores\_stdrew.csv, etc.)
* Best results are low losses, which are the wsls models.
  + They are organized so that they contain the number of times they were fed through to train the machine and the loss was calculated using Mean Squared Error

## Issues

* All results ended up overfitting the data.
  + Ex. R2 score of training data for WSLS models would be .98 where the R2 score of validation data would result in -.38
  + Cut down on the epochs and even still saw bad splits of R2 scores
* When the features were cut down the overfitting stopped being so prevalent, but losses skyrocketed
* Not enough samples for deep learning as is with the 0-1000 ms time sequence

## Possible Changes

* Cut down time sequence
  + Currently 0-1000 ms is a sequence length of 251. This could be too much
    - Might be cut down via correlations data
* Using trial-by-trial data
  + Concern is biases in data could lead to problems of generalizing and not end up solving the overfitting problem
* Try shallow models
  + Currently the methods being tried are RNNs, all of which are data hungry
    - Might be able to change to something that may require less data (Support Vector Regression Machine, Ridge Regression, etc.)
    - This would also be more along the lines of XAI, although since it is a loosely defined term this isn’t a reason to leave RNNs
  + More complexity would generally require more data and since we are overfitting the baseline models building more complex models would be inadvisable
* Possibly might find a way to determine the important times
  + Would likely need to use the time steps as the output, but this needs more thought behind it
  + Could be very useful, but may become redundant if we get data per the correlation results for the time lags