Decoding Brain Dynamics: Brain Activity Patterns Predict Nature of the Stimulus

Capstone Project 2: Final Report

Chirag B. Limbachia

December 9, 2020

Contents

Problem Statement

Different stimuli evoke different brain response, which enables an organism to respond to a particular stimulus in a particular way. Interesting question about the brain-stimuli relationship is: can the nature of the stimulus be predicted based on the evoked brain activity?

In this project, dynamic brain activity patterns of human subjects captured by fMRI in response to approaching and retreating threats is used to predict the stimulus class (**approaching** or **retreating** threat).

Data

Data Collection

Functional magnetic resonance imaging (fMRI) data was collected from 61 human subjects. Typical fMRI data collection setup is shown in Figure 1A. The subject lies in the fMRI scanner with a head coil around their head. The subject views a visual stimulus on a projection screen via a mirror mounted to the head coil (Figure 1B and C).

fMRI is a time-series data. Every frame of the fMRI data is a 3-dimensional (volumetric) screenshot of the brain representing a single time-point. Every volume of the fMRI data is made up of several thousand voxels, and each voxel has an associated time-series (Figure 2). The fMRI data is commonly collected at a temporal resolution between 500-2500ms. Present fMRI data was collected at a temporal resolution of 1250ms. Forty-five minutes of fMRI was collected on every subject.

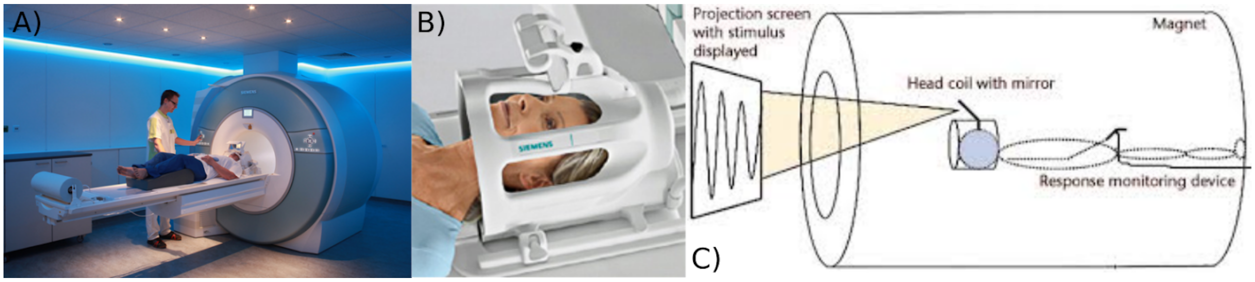
A)
B)
SIEMENS
Projection screen with stimulus displayed
Magnet
Head coil with mirror
Response monitoring device
C)

Figure 1. Typical fMRI setup. A) MRI scanner. B) Head coil with a mounted mirror. C) Schematic of an fMRI experiment setup. Subject is presented with a visual stimulus on a projection screen placed outside the scanner above the subject's head. The subject is able to see the visual stimulus via a tilted mirror mounted on the head coil.

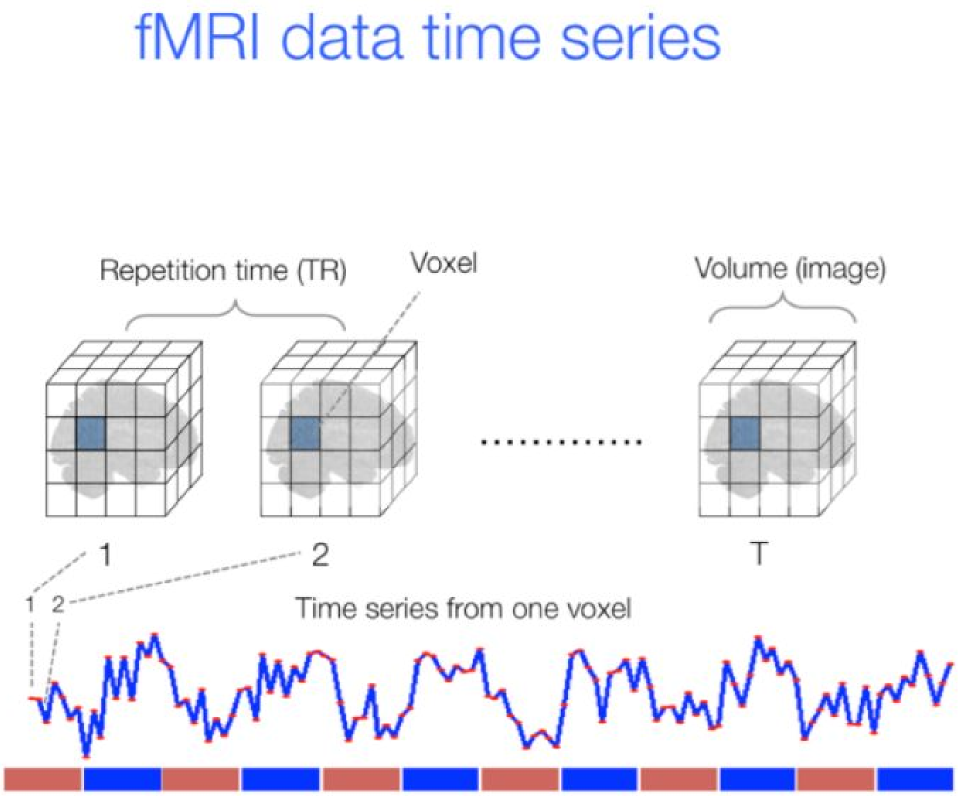

Figure 2. fMRI time-series data.

Visual Stimulus

During the fMRI experiment, participants were presented with a visual stimulus on the projection screen. Two circles moved around randomly on the screen. When the circles collided, a mild but unpleasant electric shock was delivered to the index and middle fingers of the subject's left hand. The screen turned white and a red dial appeared around the circles when the circles touched to indicate shock delivery. To check out a short clip of the visual stimulus, click here. Shock delivery was meant to induce fear of circle collision. Several "near-miss" events occurred at random times during the experiment. Near-miss events are defined as those when the circles approach each other at least for 8 seconds, come very close (i.e., distance less than 1.5 times the circle diameter), but miss and start to retreat at least for 8 seconds. Near-miss events were included to investigate the brain's dynamic response during approaching and retreating threats.

Data Preprocessing

fMRI data suffers from a lot of unwanted signals (noise) such as, physiological signals associated with respiratory and cardiac cycles, head motion, and scanner drifts. Such noise was filtered out from the data using ICA implemented in fMRI data analysis software, FSL (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/). ICA decomposes the data into independent spatio-temporal components. Based on the spatial and the associated temporal pattern, components are classified into noise or signal. Components identified as noise are regressed out. Figure 3 shows some examples of most commonly encountered noise.

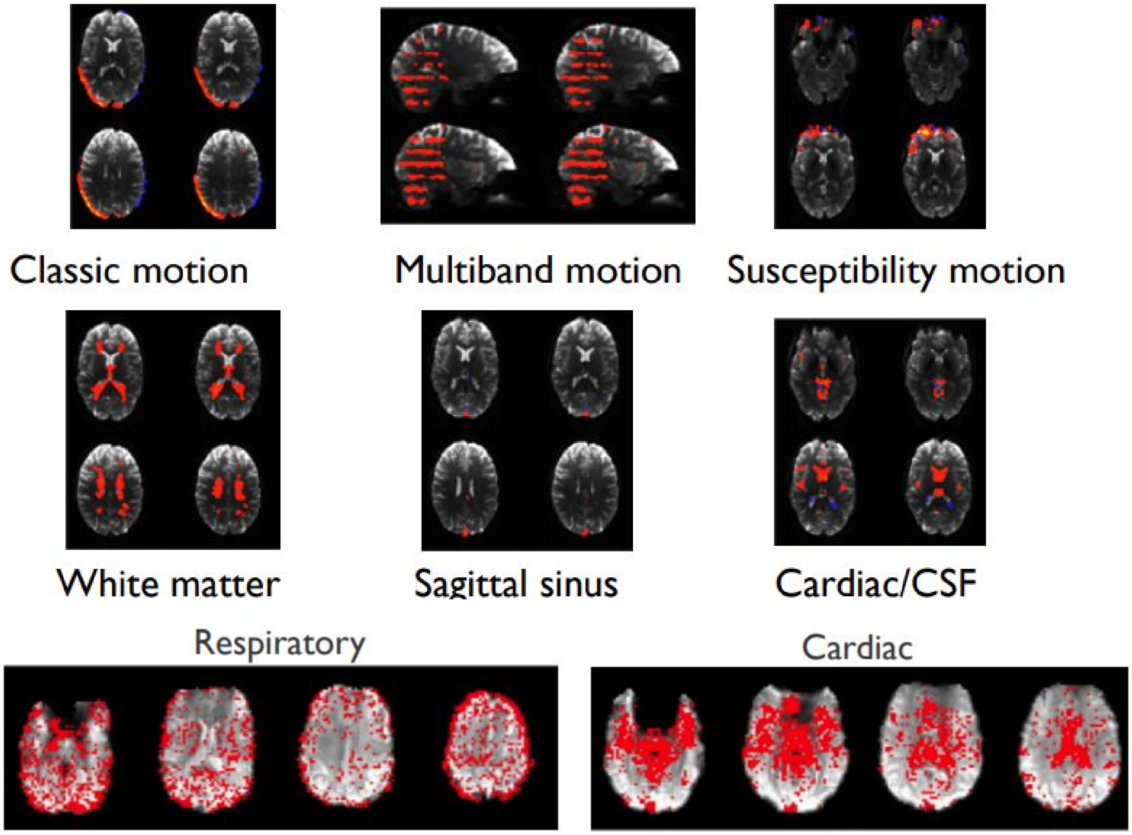

Figure 3. Noise commonly found in fMRI data.

Feature Selection

Time-series sequences corresponding to near-miss events were used in this project. Each near-miss sequence was 14 time-points long (14 x 1.25 = 17.5 seconds): first 7 time-points corresponded to the early period when the circles approached; and next 7 time-points corresponded to the later period when the circles reatreted. Every subject had 46 near-miss sequences. The near-miss sequences were extracted from 316 brain regions of interest (ROI) (Figure 4).

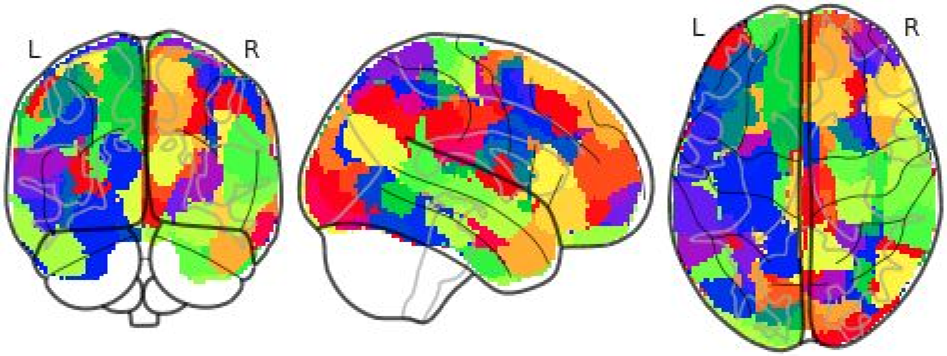

Figure 4. 316 brian regions of interest (ROI) displayed on a glass brain model. Each ROI is shown with a different color.

Figure 5 shows an example of a near-miss sequence extracted from the parietal cortex of one subject. Time-points 0-6 correspond to the period when the circles approach, and time-points 7-13 correspond to the period when the circles retreat. Every near-miss sequence was divided into two training examples: first 7 time-points of the sequence labeled as “approach” and last 7 time-point labeled as “retreat” (Table 1). Out of 61 subjects, only 42 subjects’ data was used to train and validate the predictive model. Remaining 19 subjects data was used to test the model performance.

Table 1. Shows two training examples created from a near-miss sequence.

#	Training Sequence							Label
1	TP00	TP01	TP02	TP03	TP04	TP05	TP06	“approach”
2	TP07	TP08	TP09	TP10	TP11	TP12	TP13	“retreat

TP = Time-point

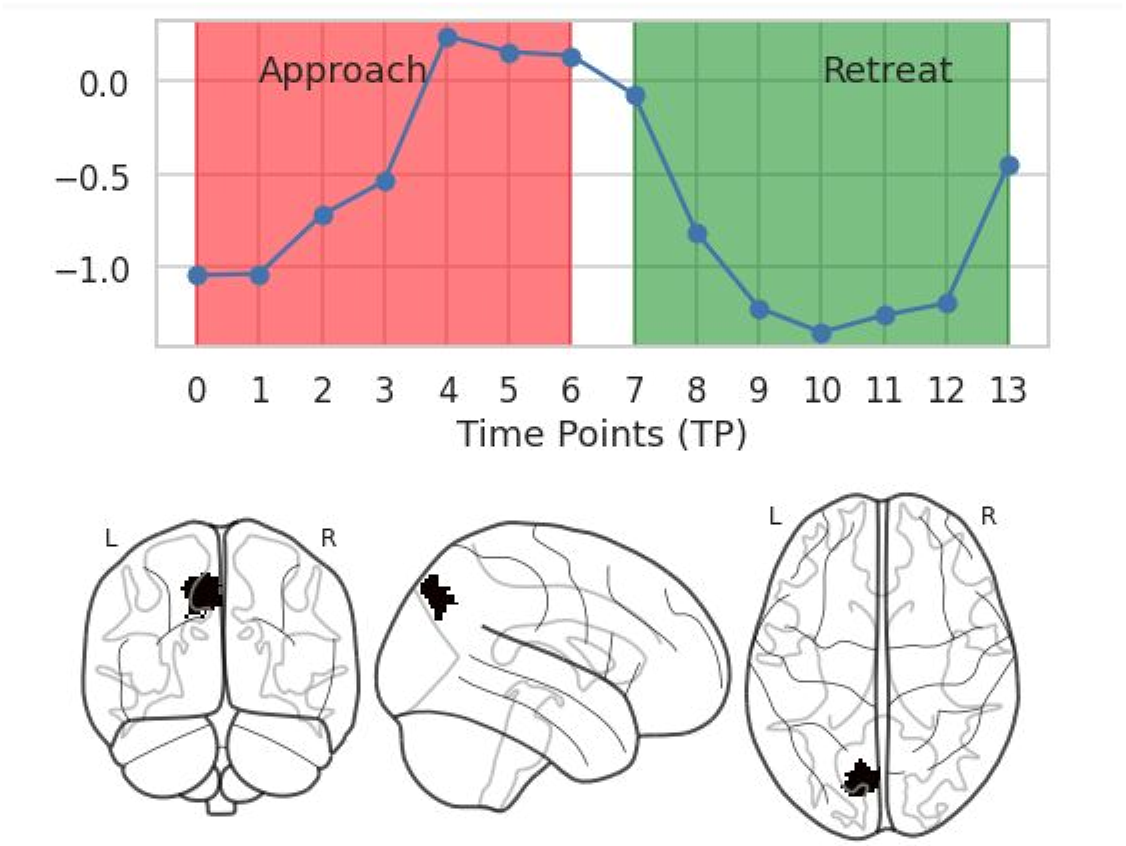

Figure 5. (Top) Near-miss sequence from the (bottom) parietal cortex of the brain. The two periods, approach and retreat, of the near-miss are indicated in red and green respectively.

Modeling

Model Architecture

A recurrent neural network architecture with gated recurrent units (GRU) was employed to build and train the predictive model. The architecture had three hidden layers, each with 16 GRU units, and an output time-distributed dense layer (DL) with single sigmoid activation unit. The time-distributed DL returns prediction for every time-point. Figure 6 shows the model schematic. The model input (x_t) is a batch of approach/retreat sequences. The dimension of x_t is (batch_size x time-points x features), where batch_size = 32, time-points = 7 time-points, and features = 316 ROIs. The hidden state outputs (h_t) from the GRU architecture are passed through the DL layer to get class predictions (y_t).

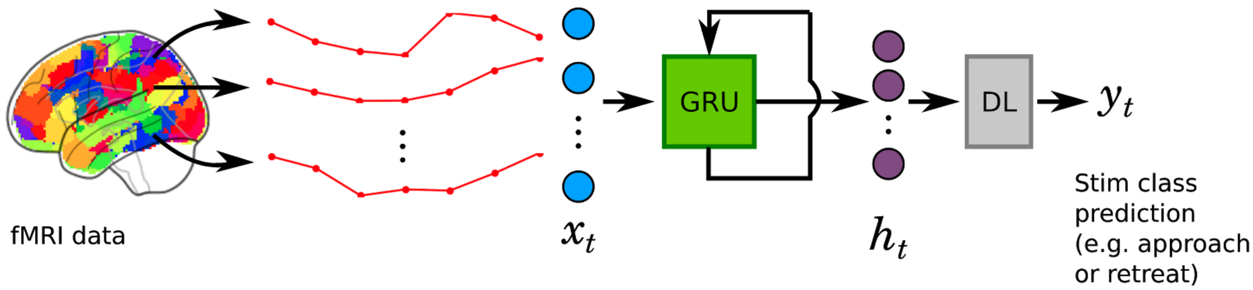

Figure 6. Model Schematic: x_t is the input of dimension (batch_size x time-points x features), which goes into the GRU architecture; h_t are hidden state outputs of the GRU architecture that are passed through DL to get class predictions (y_t).

Model Fine Tuning

In order to make the model generalizable and avoid overfitting, hyperparameters like: 1) L2 regularization, 2) dropout, and 3) learning rate were optimized using a k-fold nested cross-validation method. Seventy-five different combinations of the aforementioned hyperparameters were used to train and validate the model. Figure 7 shows the training and validation set performance of the 75 models. The models are arranged in the descending order of their mean validation accuracy. The error bars indicate standard deviation across folds. As the number of classes in the dataset were balanced, accuray was an appropriate metric to evaluate the model performance.

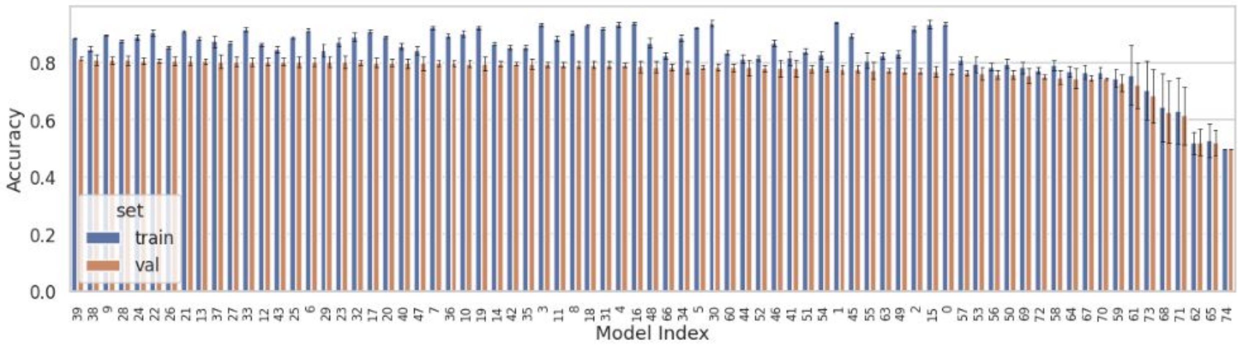

Figure 7. Training and validation accuracies of 75 different combinations of the following hyperparameters: 1) L2 = [0, 0.001, 0.003, 0.01, 0.03], 2) dropout = [0, 0.1, 0.2, 0.3, 0.4], and 3) learning rate = [0.001, 0.003, 0.006].

Model Performance

Accuracy

Best performing model yielded mean training and validation accuracies of 0.89 and 0.82, respectively. Best hyperparameters were: L2 = 0.003, dropout = 0.3, and learning rate = 0.001. Trained model was tested on near-miss sequences of the 19 held-out subjects. Figure 8 shows temporal and overall accuracies achieved on the held-out subjects' data. The model performed reasonably well from the 1st time-point itself, with a mean accuracy of 0.79. The accuracy gradually increased to 0.89 by the last (7th) time-point. The "overall" accuracy is the average accuracy across time-points, which is 0.83.

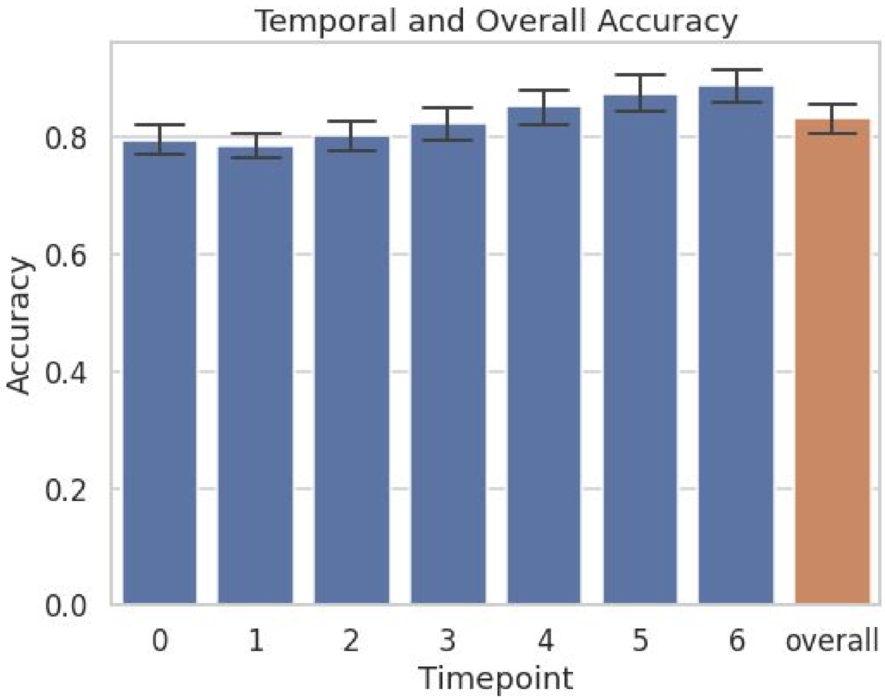

Figure 8. Performance of the trained model on near-miss sequence data of the held-out subjects. The barplot shows accuracy at every time-point, and overall accuracy across time-points.

Probability of Predicting the True Class

To better appreciate how the model's ability to separate the two classes improves with time, see Figure 9 that shows the probability of predicting the true class. The probability in favor of the true class increases with every time-point.

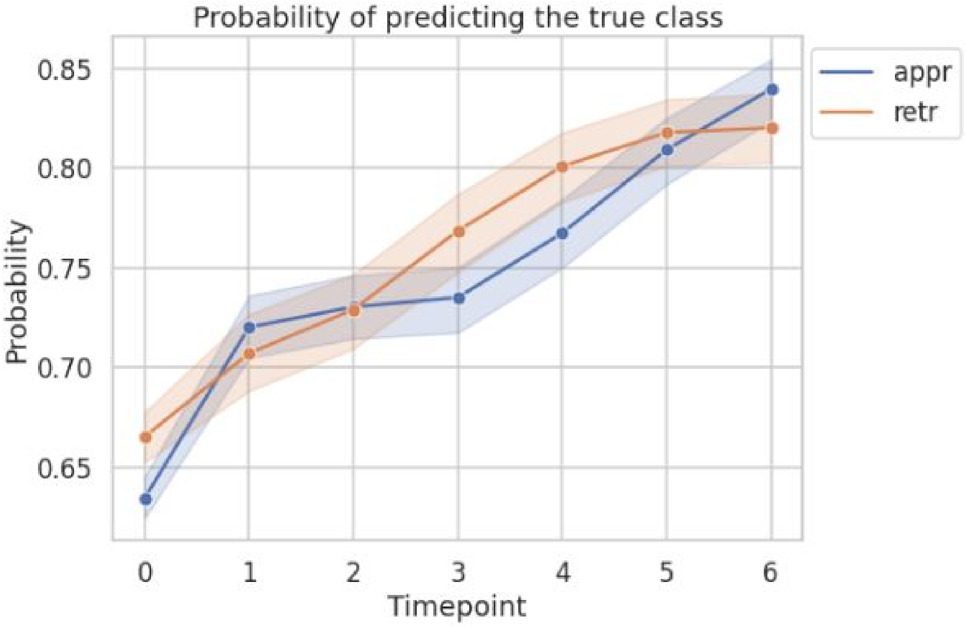

Figure 9. Probability of belonging to the true class: approach (appr) and retreat (retr).

Model Evaluation

Observed vs. Chance Accuracy

To assess significance of the observed test accuracy of 0.83, the observed test accuracy was compared against chance accuracy. Chance accuracy is obtained when the model predicts one of the two classes at random. To simulate a chance accuray distribution, the model with best hyperparameter settings was trained on the training data a hundred times, each time with shuffled labels. At every iteration, the model was tested on the test data with non-shuffled (i.e., true) labels, and the accuracy was recorded. From the chance accuracy distribution, it was found that the chance of achieving an accuracy of at least 0.83 was less than 0.009 (see Figure 10).

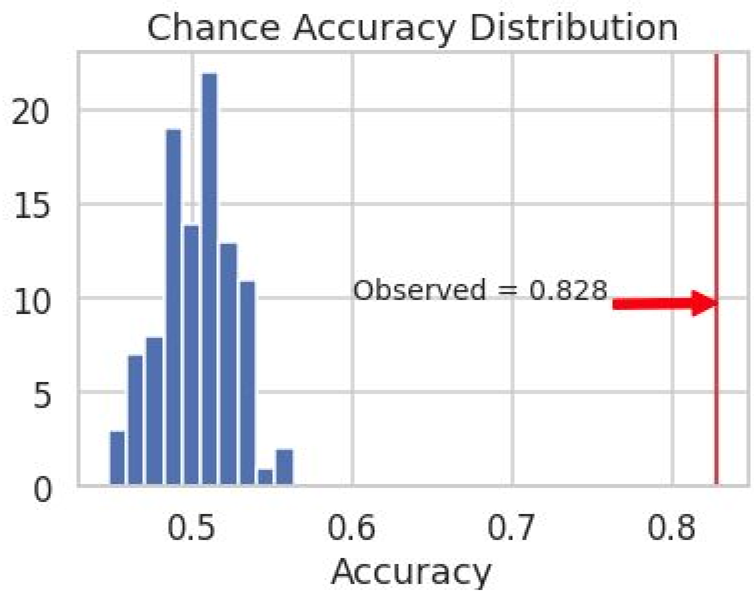

Figure 10. Chance vs. observed accuracy.

Comparison with Random Forest

As GRU belongs to the family of recurrent neural networks, it learns class separability from the sequential aspect of the data. Since the data used in this project is fMRI time-series data, GRU was a reasonable model choice. An interesting question that can be asked is, how well would a model that does not take into account the sequential aspect of the data perform? Would it perform as well as the GRU model? Or would it perform poorly? To make this comparison, a Random Forest classifier was trained on the current data. The Random Forest classifier was also fine tuned using the nested cross validation method, and best hyperparameters (n_estimators = 1500, max_feature = ‘sqrt’) were obtained. The test accuracy of Random Forest classifier was only 0.58 which is significantly low compared to that of the GRU model.

Conclusion

The GRU model was successful in characterizing the spatio-temporal patterns of the brain during approaching and retreating threats. Check out the project’s github and interactive jupyter-book.

Demonstration

The video in this link demonstrates model performance on one of the held out participant’s fMRI data. The video shows the visual paradigm presented to the participant during his/her fMRI scan, along with model predictions at the top-right of the screen. Prediction is either “Approach” or “Retreat”. If the prediction is correct, the color of the text remains green; and if it is incorrect, it turns red. Note that when the circles touch, the screen turns white and a red wheel appears around the circles to indicate delivery of the physical shock. Also note that the speed of the video has been increased by 4x for quick demonstration purposes.