Predicting Video Game Stimuli Using EEG Scans

Aris Chalini, Ronald Lee, Jesus Fernandez

Description

- We would like to see if it is possible to predict the category of video game a person is playing (boring, calm, horror, and funny) given a single EEG reading.
- What kind of prediction would that model make on the EEG scans of people who were presented with different audio/visual stimuli?

Problem Statement

As technology advances and brainwave monitoring devices become more affordable and convenient to use, brainwaves will become another important data source for various future applications. This research focuses on brainwave signals generated by human emotions.

Boring(LANV) Calm(LAPV) Horror(HANV) Funny(HAPV)

Objective

1. The main objective is to train a model to distinguish between four distinct emotional states stimulated by four corresponding video games
2. The secondary objective is to have the model effectively predict on new, unseen data from a different study

Evaluation Parameters

1. Our primary evaluation parameter will be accuracy (number of correct predictions divided by the total number of samples). We will consider a video-game-category identification accuracy of 72% to be a successful model.
2. Our secondary evaluation parameter will be loss.

Data Sources

GAMEEMO: Twenty-eight participants had their EEG readings taken (14 channels read from different regions of the brain) while playing a video game for 5 minutes that was either categorized as boring, calm, horror, or funny. This means that the data originates from 112 (28*4) unique scenarios. For each of those unique participant-genre combinations, there are 38253 EEG readings that were recorded over the course of those 5 minutes of gameplay. With a sampling rate of 128 Hz, this represents approximately 5 minutes of activity.

DREAMER: Twenty-three participants had their EEG readings taken (14 channels corresponding to the same regions recorded in the GAMEEMO dataset) while watching 18 different film clips that range from 1.08 to 6.55 minutes long..

Machine Learning Algorithms

Baseline - KNN

Multi-layer Perceptron Neural Network (MLPNN)

Random Forest

CNN

LSTM

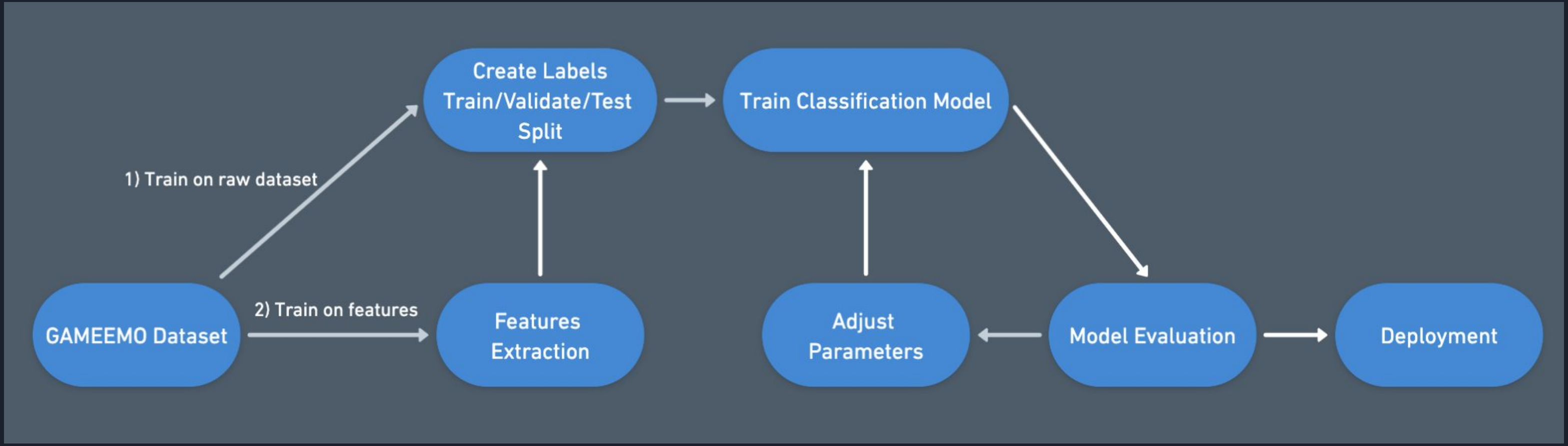
Combinations of CNN + LSTM + MLPNN

*Dimensionality reduction will be attempted where appropriate.

Approach / Methodology

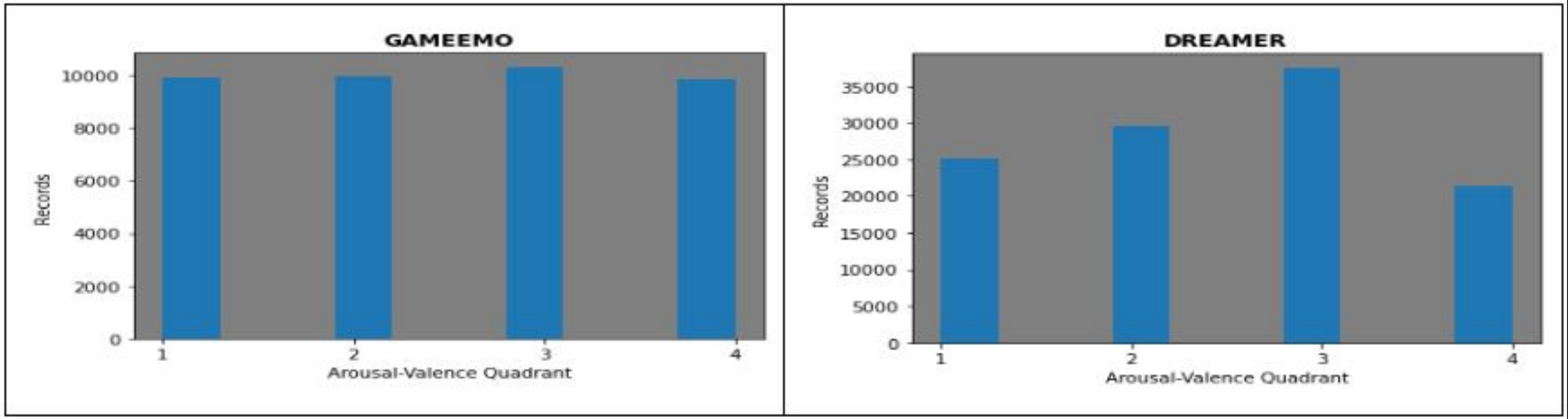
We will train and test our model on the GAMEEMO dataset. This will be done with a 60/20/20 train/validate/test split

We want to compare how our model labels different data with different stimuli

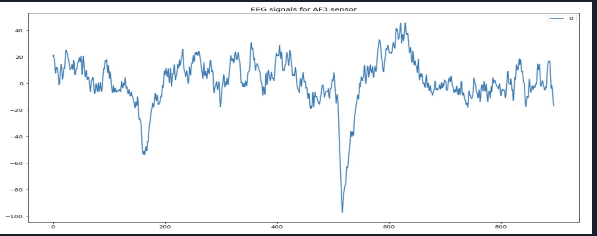


Labeling in the DREAMER dataset was mapped to match the labeling of the GAMEEMO dataset.

Fig. 1. Arousal-valence emotion model [43].



The Nature of Raw EEG Data



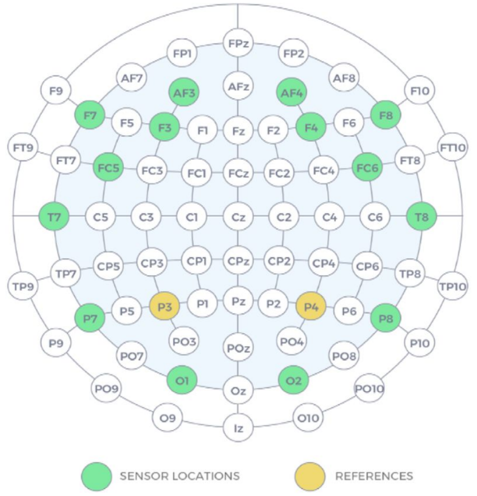


Fig. 2. Sensor and reference electrode locations [12].

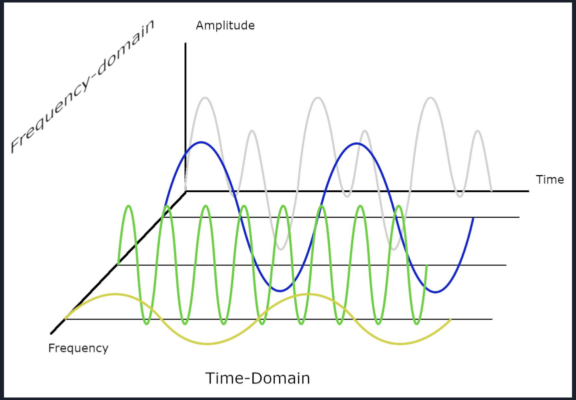
AF3	AF4	F3	F4	F7	F8	FC5	FC6	O1	O2	P7	P8	T7	T8
-4.1025	-9.7437	5.1279	0	0.5127	-6.667	-15.3848	-4.1025	14.8716	-0.51318	6.1538	0	2.0513	0.5127
4.1025	-24.1025	-9.8721	-15	-3.5898	-21.667	-30.2563	7.1797	-0.12842	4.1025	3.5898	7.1797	4.1025	6.667
-10.7693	-15.8977	2.5642	0	1.0256	-8.2053	-15.2563	-6.6667	14.8716	-3.0769	12.3074	-1.0256	7.6921	3.0769
4.2307	-14.6155	0.25659	-7.949	3.8459	-23.2053	-15.1282	0.76929	-0.12842	-0.76929	2.8206	4.3591	-0.25659	4.8718
1.0256	-23.5896	-4.1023	-22.949	1.5383	-35.3845	-13.3333	5.1282	-7.1794	-1.0256	7.1794	8.2053	3.5901	5.1282
-7.436	-8.5896	7.9487	-7.949	0.25635	-20.3845	1.6667	-7.436	7.8206	-12.0513	11.5381	-6.7947	3.333	-0.76953
7.564	-16.9231	-6.1538	-22.949	2.0515	-35.3845	-13.3333	7.564	-7.1794	2.9487	1.0256	8.2053	-1.0256	14.2305
-2.0513	-11.7949	4.1025	-8.7183	0	-28.2053	-8.2051	0	6.6665	1.5386	5.1279	5.1279	-1.5386	6.1538
2.0513	-4.1025	4.1025	6.2817	-2.0513	-17.9487	-0.5127	-3.5898	8.2051	-3.5898	2.0513	-4.1025	0.5127	5.1284
4.3591	-19.1025	-5.8972	-8.7183	4.8718	-32.9487	-12.5642	0.76929	-6.7949	-0.76929	2.8206	3.8464	4.3591	6.4104
-6.6665	-7.1797	9.1028	6.2817	18.4614	-23.5898	2.4358	-13.3335	8.2051	-15.7693	14.8716	-8.2051	4.6152	-4.6152
8.3335	-17.1792	-0.25635	-8.7183	16.6665	-38.5898	-9.4873	1.6665	-6.7949	-2.3076	0.25635	2.8208	1.7949	5.8975
2.3076	-21.2817	7.436	-23.7183	22.3076	-53.5898	5.3848	-6.9229	2.8208	-5.8975	0.25635	-4.3589	5.3848	-0.25635
8.718	-9.7439	12.3074	-8.7183	30.2561	-51.2822	17.9485	-8.718	1.5383	-10.7693	1.5383	-8.718	1.0256	-7.1794
14.6155	-21.282	3.8459	-20.2563	29.4871	-66.2822	2.9485	2.8206	-13.4617	-1.7947	-2.3079	1.282	-0.76929	0.76929
12.0513	-20.2563	5.3848	-15.6406	30.5132	-81.2822	4.3594	0.25635	-15.6406	-3.333	-0.25635	0.76953	0.25635	-0.25635
9.7437	-18.4614	7.6924	-9.7437	32.8203	-77.9487	8.7178	-2.0513	-10.7695	-4.6157	2.0513	0.5127	2.0513	-0.5127
7.1794	-16.9231	9.7434	-4.1028	34.8718	-72.8206	12.8206	-4.1028	-5.6414	-5.6414	4.6155	0.51294	3.5896	-0.51294
3.8459	-16.1541	11.0256	0.76929	36.1536	-68.4617	15.6409	-7.4358	-1.7952	-7.949	5.8972	-0.76929	3.8459	-1.7952
-0.76904	-17.1797	10.5127	3.3335	35.1284	-65.8975	17.1797	-12.0513	0.76904	-11.5386	5.8975	-3.3335	2.8203	-4.8716
-3.8462	-15.6411	12.564	8.4614	36.6665	-61.2822	21.2817	-14.6152	4.8716	-13.0771	7.436	-3.8462	3.8462	-5.3848
11.1538	-22.0513	5.1284	-6.5386	39.4873	-76.2822	11.2822	-2.0513	-10.1284	-3.5898	1.0259	0.5127	0.5127	-0.5127
6.9233	-30	5.8975	-21.5386	31.5386	-91.2822	14.6152	-7.9487	-14.1025	-2.8203	4.3589	-0.25635	1.7949	0.25635
1.5383	-15	12.3079	-6.5386	33.3333	-76.2822	28.718	-13.8459	0.89746	-10.7693	4.6155	-9.2307	-1.5383	-3.0769
14.1025	-30	-2.6921	-21.5386	28.9741	-91.2822	13.718	1.1541	-14.1025	3.333	-2.3076	5.7693	2.3076	4.8716
-0.89746	-27.4358	2.3079	-6.5386	29.4871	-76.2822	9.4871	-13.0769	0.89746	-2.8206	8.9744	0.76929	14.6155	-1.7947
9.231	-31.7947	-8.7178	-21.5386	31.2822	-91.2822	-1.5386	-4.1025	-8.2051	1.5386	6.1538	9.231	11.7949	4.1025
0.76929	-37.1794	-9.4871	-36.5386	33.0769	-103.8459	6.9231	-3.8459	-3.8459	-0.76929	7.949	8.9744	6.4104	1.282
1.5386	-22.1794	2.564	-21.5386	31.2822	-88.8459	17.436	-13.3335	6.1538	-9.7437	9.231	-1.5386	4.1025	-1.5386

Engineered Features and Raw Arrays

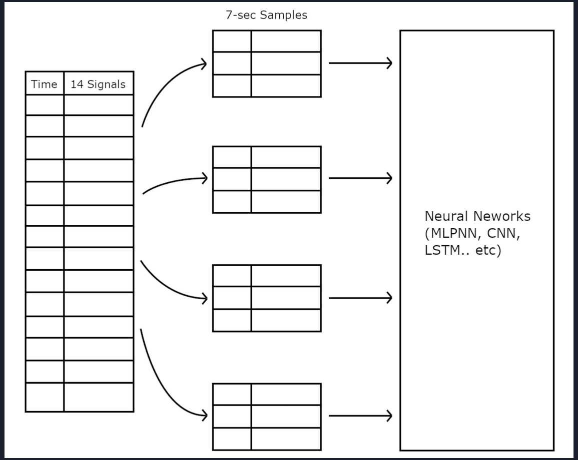
For each 1 second, the following features are computed:

- Min/max/mean/std
- Skewness and Kurtosis
- Covariance
- Eigenvalues
- Logarithmic covariance
- Fast Fourier Transform

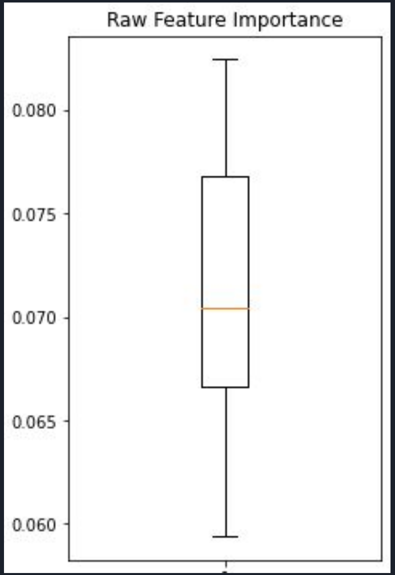
Domain Change



7-second Sampling

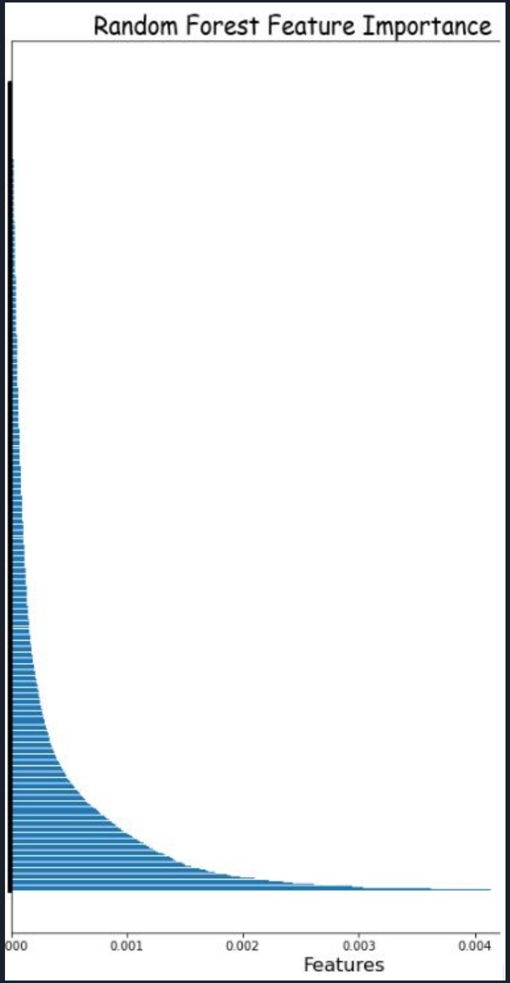


Feature Importances



Little variance in EEG sensor importance.

Most of the 3738 features contribute relatively little information to this classification task.



Models Trained on Raw EEG Data (7-sec)

Model Name	Accuracy	Loss	Training Time	Hyper params (not a complete list)
MLPNN	72.26%	1.065	36 sec	2 Dense layers (512,256 units), SGD optimizer, relu activation.
CNN	75.56%	7.54	45 sec	1 Conv layer (1024 filters), kernel 3, stride 1, Adam optimizer, relu activation.
CNN + MLPNN	75.98%	1.32	26 sec	4 Conv layers (100,100,100,100 filters), kernel 3, stride 1, 1 Dense layer (512 units), SGD optimizer, relu activation.
CNN + LSTM + MLPNN	76.62%	1.05	52 sec	3 Conv layers (100,100,100 filters), kernel 3, stride 1, 1 LSTM layer (256 units), 1 Dense layer (512 units), SGD optimizer, relu activation.
LSTM	70.03%	1.09	64 sec	1 LSTM layer (15 units), Adam optimizer
LSTM + MLPNN	75.56%	0.92	112 sec	1 LSTM layer (256 units), 1 Dense layers (512 units), SGD optimizer, relu activation.
LSTM + CNN + MLPNN	75.35%	1.26	110 sec	1 LSTM layer (256 units), 3 Conv layers (100, 100, 100 units), kernel 3, stride 1, 1 Dense layer (512 units), SGD optimizer, relu activation.

Models Trained with Feature Engineering

Model Name	Accuracy	Loss	Training Time	Hyper params (not a complete list)
KNN	72.00%		0.00755 sec	1 neighbor
MLPNN	92.65%	0.50	703 sec	2 Dense layers (512,64 units), Adam optimizer, relu activation.
MLPNN (PCA)	76.33%	1.33	83 sec	Same as MLPNN, PCA 95% of variance retained.
CNN	79.86%	0.87	179 sec	3 Conv layers (50,50,50 filters), kernel 3, stride 1, SGD optimizer
CNN (PCA)	67.96%	2.22	120 sec	Same as CNN, PCA 95% of variance retained.
CNN + MLPNN	88.82%	0.44	344 sec	3 Conv layers (100,100,100 filters), kernel 3, stride 1, 1 Dense layer (512 units), SGD optimizer, relu activation.
CNN + MLPNN (PCA)	80.51%	0.83	162 sec	Same as CNN+MLPNN, PCA 95% of variance retained.
CNN + LSTM + MLPNN	89.56%	0.41	1187 sec	3 Conv layers (100,100,100 filters), kernel 3, stride 1, 1 LSTM layer (256 units), 1 Dense layer (512 units), SGD optimizer, relu activation.
CNN+LSTM+MLPNN (PCA)	79.98%	0.85	399 sec	Same as CNN+LSTM+MLPN, PCA 95% of variance retained.
LSTM	78.23%	0.79	1539 sec	1 LSTM layer (15 units), Adam optimizer.
LSTM (PCA)	69.93%	1.27	556 sec	Same as LSTM, PCA 95% of variance retained.
LSTM + MLPNN	85.88%	0.80	1648 sec	1 LSTM layer (15 units), 3 Dense layers (100,100,100 units), Adam optimizer, relu activation.
LSTM+MLPNN (PCA)	83.96%	0.82	588 sec	Same as LSTM+MLPNN, PCA 95% of variance retained.
LSTM + CNN + MLPNN	89.63%	0.40	3351 sec	1 LSTM layer (256 units), 3 Conv layers (100, 100, 100 units), kernel 3, stride 1, 1 Dense layer (512 units), SGD optimizer, relu activation.
LSTM + CNN + MLPNN (PCA)	79.53%	0.87	1230 sec	Same as LSTM+CNN+MLPNN, PCA 95% of variance retained.
Random Forest	81.00%		15.9 sec	min_samples_leaf=20, n_estimators=1000, max_depth = 150, bootstrap=True, oob_score=True, n_jobs=-1, random_state=seed, max_features=0.2

Best Model - MLPNN - 92.65% Accuracy

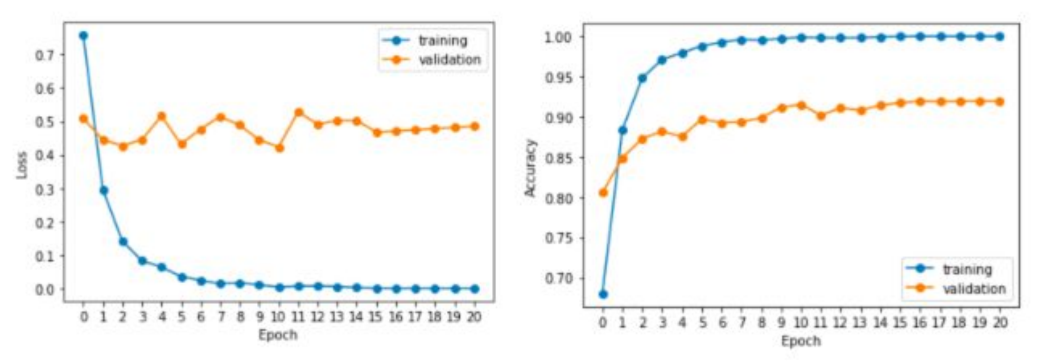
```
optimizer: Adam ; activation: relu
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 batch_normalization (BatchN  (None, 3738)             14952
 ormalization)

 fc_0 (Dense)                (None, 512)               1914368

 fc_1 (Dense)                (None, 64)                32832

 Activation (Dense)          (None, 4)                 260

=================================================================
Total params: 1,962,412
Trainable params: 1,954,936
Non-trainable params: 7,476
```



However, when this model is tested on DREAMER dataset, accuracy is only at around 30%...

Limitations of the Study

- Different Stimuli
- Activity level
- Limitations of a Four-Quadrant model
- Emotional State Labels
- Subject diversity

Future Work

- Collect more data that better represents the population

Thank you

References

MEEMO:
https://www.sciencedirect.com/science/article/abs/pii/S1746809420301075?via%3Dihub

DREAMER:
S. Katsigiannis, N. Ramzan, "DREAMER: A Database for Emotion Recognition Through EEG and ECG Signals from Wireless Low-cost Off-the-Shelf Devices," IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 1, pp. 98-107, Jan. 2018. doi: 10.1109/JBHI.2017.2688239

Logarithmic covariance: http://www.stat.rice.edu/~jrojo/4th-Lehmann/slides/Deng.pdf

Feature engineering code: https://github.com/jordan-bird/eeg-feature-generation