Weekly Work Report 11/1/2024

**Response to Last Week’s Feedback:**

* Implement **other models** from the paper as originally applied, specifically maintaining the **window size, using temporal signals** instead of summaries, and utilizing **the same frequency bands** for each channel.
* Use **summary** **features** to train a **fully connected neural network**.

**This Week:**

* **The other 3 models and FCNN are used for classification.**
  + **Classification of Impasse vs. Rest States.**
  + **Classification of Aha! vs. Rest States.**
  + **Classification of Impasse vs. Rest States.**

**Their** **Task:** classify EEG signals into different states **(Rest state or Task State)**

**Their Input:**

**A screenshot of a computer

Description automatically generated** **A screenshot of a computer

Description automatically generated**

Data processing: (72, x, 21, 500) ; where x is inconsistent

* 72: samples/subjects \* 2 states
* x: inconsistent
* 12: channels
* 500: time points / epoch (window size = 1s) / sampling frequency

Their input shape: (x, 21, 17)

* Shape of X\_train: (6940, 21, 17)
* Shape of X\_test: (1736, 21, 17)
* Shape of y\_train: (6940, 2)
* Shape of y\_test: (1736, 2)

This shape corresponds to (samples/subjects, segments/trials, channels/electrodes, time points/samples per segment)

My input Shape

A screenshot of a computer code

Description automatically generated

**Features:**

Feature extraction:

from scipy.signal import welch

**def extract\_frequency\_features(data, sfreq):**

**n\_samples, n\_channels, n\_times = data.shape**

**freq\_bands = {**

**'delta': (0.5, 4),**

**'theta': (4, 8),**

**'alpha/mu': (8, 12),**

**'low\_alpha': (8, 10),**

**'high\_alpha': (10, 12),**

**'beta': (12, 30),**

**'low\_beta': (12, 15),**

**'mid\_beta': (15, 20),**

**'high\_beta': (20, 30),**

**'gamma': (30, 100),**

**'low\_gamma': (30, 50),**

**'high\_gamma': (50, 100),**

**'epsilon': (0.1, 0.5),**

**'sigma': (12, 16),**

**'high\_frequency\_oscillations': (100, 500),**

**'ripples': (80, 200),**

**'fast\_ripples': (200, 500)**

**}**

freq\_features = np.zeros((n\_samples, n\_channels, len(freq\_bands)))

for i in range(n\_samples):

for j in range(n\_channels):

*#* ***Compute power spectral density (PSD) using Welch's method***

f, psd = welch(data[i, j, :], sfreq, nperseg=min(256, n\_times))

*# Extract power in specified frequency bands*

for band, (f\_low, f\_high) in freq\_bands.items():

idx\_band = np.where((f >= f\_low) & (f < f\_high))[0]

if len(idx\_band) > 0:

power\_in\_band = np.mean(psd[idx\_band])

else:

power\_in\_band = 0.0 *# If no frequencies in band, set power to 0*

freq\_features[i, j, list(freq\_bands.keys()).index(band)] = power\_in\_band

return freq\_features

X\_freq\_features = extract\_frequency\_features(X, sfreq=500)

print('Shape of X\_freq\_features:', X\_freq\_features.shape)

A screenshot of a computer program

Description automatically generated

**Modeling and Classification**

* **EEGNet:** A compact convolutional neural network tailored for EEG signal classification.
* **Tsception:** A temporal convolutional neural network designed for time-series data.
* **~~ATCNet:~~** ~~Attention-based Temporal Convolutional Network focusing on important time-series features.~~
* **LSTM RNN:** A Long Short-Term Memory Recurrent Neural Network to capture temporal dependencies in the EEG signals

**Impasse vs Rest**

* **EEGNet**

**A screenshot of a computer

Description automatically generated**

A graph with a red line and a blue line

Description automatically generated

* **TSCeption**

**A screenshot of a computer screen

Description automatically generated**

A graph of a curve

Description automatically generated

* **LSTM RNN**

**A screenshot of a computer

Description automatically generated**

A graph of a curve

Description automatically generated with medium confidence

* **Result Table**

**A black text on a white background

Description automatically generated**

**Aha vs Rest**

* **EEGNet**

**A screenshot of a computer

Description automatically generated**

A graph with a red line and a blue line

Description automatically generated

* **TSCeption**

**A screenshot of a computer

Description automatically generated**

A graph with red line and white text

Description automatically generated

* **LSTM RNN**

**A screenshot of a computer

Description automatically generated**

A graph with red line and white text

Description automatically generated

* **Result Table**

**A black numbers and a white background

Description automatically generated**

**Impasse (0) vs Aha (1)**

* **EEGNet**

**A screenshot of a computer

Description automatically generated**

A graph with a red line and a blue line

Description automatically generated

* **TSCeption**

**A screenshot of a graph

Description automatically generated**

A graph of a positive rate

Description automatically generated with medium confidence

* **LSTM RNN**

**A screenshot of a report

Description automatically generated**

* **Result Table**

A close-up of numbers

Description automatically generated

**Best Result:**

**A black numbers and a white background

Description automatically generated**

A screenshot of a graph

Description automatically generated

**FCNN: Summary features only**

**Impasse vs Rest**

**A white sheet with text and numbers

Description automatically generated with medium confidence**

A graph of a curve

Description automatically generated

**Aha vs Rest**

A table with text and numbers

Description automatically generated

A graph of a curve

Description automatically generated

**Aha vs Impasse**

A screenshot of a computer

Description automatically generated

A graph of a curve

Description automatically generated

**Overall Result**

**A close-up of numbers

Description automatically generated**

**Next Week:**

* Finish processing additional signals, including EDA and pupil data.
* **Result Comparison: with-in different time\_window**

**This Month:**

* Compare Aha!/Impasse classification using
  + physiology signals
  + EEG
  + physiology signals + EEG

**Final Goal:**

* Explore the dynamics of **Attention**, **Impasse**, and the **"Aha!"** moment.