BCI Robot Motor Imagery Rehabilitation

Emerald Zhang
emerald.zhang@utexas.edu

Julian Weaver
julian.weaver@utexas.edu

Nihita Sarma
nihitasarma@utexas.edu

I. Introduction

In this project, we aim to develop and train a sophisticated brain-computer interface (BCI) decoder capable of accurately distinguishing between Motor Imagery (MI) commands for 'Reach' and 'Rest'. Subjects were fitted to a Harmony Exoskeleton and presented with specific cues to imagine reaching out or to remain at rest. Over the course of several sessions, the subjects attempted to gain greater MI accuracy. To facilitate learning of this task, closed-loop feedback was achieved through upper-arm application of Functional Electrical Stimulation (FES), with FES providing proprioceptive feedback for 'Reaching'.

II. Relevant Work

The BCI-MI rehabilitation space has seen considerable development, especially work targeting the upper limbs. Contemporary research has a focus on binary task distinctions, such as contrasting active motor tasks and rest states within a specific muscle group or limb. This includes differentiation between grasp and rest in hand muscles and differentiation between extension and rest of an arm. [1] Analysis of complex movements and differentiability across vastly different body parts remains less explored. However, work has been done in the field of multi-class decoding strategies, and has been applied to differentiating hand gestures or different types of reaching motions. [1] [2]

A. Pre-processing

Pre-processing of EEG data is a critical component of Neural Engineering research, reducing noise and allowing for effective feature extraction. Using a band-pass filter on the EEG data to eliminate noise and irrelevant frequencies is a standard method among studies. Further data cleaning is often achieved with common Matlab toolboxes such as EEGLAB. [2] These toolboxes allow for processes such as artifact rejection. Additionally, studies have been conducted to identify frequency range and EEG channel combinations yielding the greatest discriminability for MI decoding. [4]

B. Feature Extraction

Feature extraction is a critical step to derive meaningful information from EEG data, assisting in decoding. Commonly extracted features include features derived from the temporal, spatial, and frequency domains, which have shown considerable utility in enhancing the accuracy of MI decoding. It has been shown that using features such as Mean Absolute Value (MAV), Root Mean Square (RMS), Waveform Length (WL), Autoregressive (AR), and Common Spatial Patterns (CSP) can allow for a successful decoding model. [8] Similar results have been reported by those using characteristics of Power Spectral Density (PSD) and spatial-temporal channel weighting. [5] [6]

C. Model Selection

Simple classifiers such as Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM) have historically performed well on binary classification tasks. [2]However, recent work has expanded into deep learning methods for classification. Convolutional Neural Networks (CNNs) have, for instance, been shown to surpass the performance of SVM models in distinguishing right and left hand movements across various contexts. [7]

III. Elaborated Hypothesis

We posit that subjects will successfully learn to control the upper limb harmony exoskeleton using MI to induce reaching movements. We expect this training to manifest as a statistically significant increase in performance between consecutive online sessions. We also hypothesize that after an inability to maintain MI commands, causing the exoskeleton to stop, an error-related potential (ErrP) will be induced due to the mismatch between expected and actual robot behavior. However, this secondary hypothesis was not adequately explored in our current setup, leaving room for further study.

IV. Methods

A. Collecting and Loading Data

The data for this project was acquired from three subjects, with each subject collecting data on two consecutive days. EEG (electroencephalogram) was recorded from 64 channels distributed over the scalp according to the 10/20 standard electrode positioning. EOG (electrooculogram) and gyroscopic data were also collected as well, with the EOG recording eye blinks and other artifacts. The first day of data collection consisted of a resting state calibration, an offline session, and an online session, while the second day of data collection consisted of a resting state calibration and an online session. These components were organized as follows:

- Resting State Calibration:
  - Baseline rest EEG was recorded from subjects during both eyes-open and eyes-closed rest.
- Offline Session:
  - Three runs of EEG recording were conducted, at 20 trials each.

  - In half of these trials, subjects performed MI of a reaching movement with their dominant hand, while in the other half subjects were at rest.
  - Subjects received "fake" FES feedback during MI, as there was no BCI decoder.
- Online Session:
  - Subjects participated in multiple (3 - 7) runs, at 20 trials each.
  - Half of these trials involved active BCI operation where subjects delivered MI commands to control an exoskeleton, while the other half involved rest.
  - Successful decoding of MI commands resulted in the robot arm moving towards the target. If the BCI decoder identified a rest state or failed to decode any class with enough confidence in 7 seconds, the robot remained stationary.
  - Subjects were required to sustain their MI or rest state for a predetermined duration (6 seconds). Failure to maintain led to the robot performing an opposite action, such as retracting during a reach MI or initiating movement during a rest state.

To work with the EEG data in Matlab, we first converted the given .gdf files to .mat files. We then subdivided each subject's data into online and offline sessions, attaching corresponding resting-state calibration data to the day 1 offline and day 2 online session structs.

For the offline session, we created separate fields for rest calibration and the offline runs. Using event codes, each run signal was split into trials, and labels were stored that indicated whether the trials involved MI or rest. The rest calibration was split into eyes-open and eyes-closed data. The online sessions were similarly structured to the offline sessions, but with only the second online session having a field for rest calibration. In addition to MI/rest trial labels, the online sessions included labels for tracking the success of the command's initial delivery, the success of sustaining the command, and the sample within the trial that corresponded to the beginning of sustainment (if applicable).

It is important to note the failures in data acquisition that occurred for Subject 2, for whom the FES feedback system failed. This may have had a significant impact on the ability for the subject to learn to control the upper limb exoskeleton.

B. Fisher Scores & Discriminability

Once we have our data loaded into structs, we begin preprocessing. We first calculate fisher scores for each EEG channel to measure the channel's discriminability between offline resting state calibration and the offline reach task.

To remove as much inter-trial variability as possible, data for all reach task trials within the three offline runs are averaged together. For each non-EOG channel, a Fisher score for the channel is calculated as follows:

$$\text{Fisher Score} = \frac{(\text{mean}_{reach} - \text{mean}_{rest})^2}{\text{var}_{reach} + \text{var}_{rest}}$$

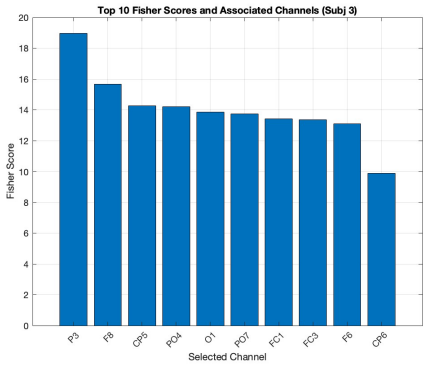

Fig. 1. The 10 channels with the highest fisher scores for the subject. The largest value was found from channel P3, on the parietal lobe.

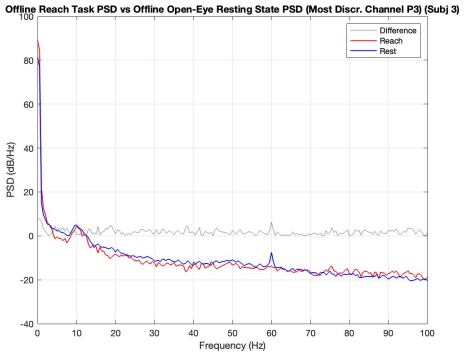

Fig. 2. A comparison between two PSDs- that of offline baseline open-eye rest, and that of an average of all offline reach trials. A plot showing the absolute value of the difference between reach and rest is also included. There is no easily identifiable band that shows good discriminability between reach and rest.

The 10 channels with the largest fisher scores are selected. This number of channels was arbitrarily chosen to significantly reduce the dimensionality of the data for classification.

Many channels chosen are physiologically relevant. [1] (P, CP, PO) channels are on the parietal lobe, roughly surrounding the sensorimotor cortex, which would significantly correlate to the MI activity measured.

C. PSD & Bandpass Filter

To reduce signal SNR, we decided to apply a bandpass filter to the EEG data. To find this band, we conducted a Power Spectral Density comparison between offline open-eye resting state calibration PSD and offline reach task PSD, focused on only the most discriminable channel calculated from fisher scores. In this comparison, the 'offline reach task' was an average of all reach task trials in the offline runs.

To account for the non-stationarity of EEG signals, PSD was calculated with window size 2 seconds and overlap 50%.

Given the lack of visibly discriminable regions in the PSD, as seen in the relatively flat difference plots, PSD did not give us much insight into good bandpasses for us to use. [2] As such, we underwent a systematic assessment of different

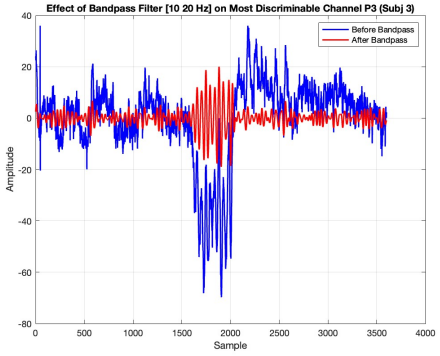

Fig. 3. The effects of the band-pass filter. The data shown comes from the first trial of the first run of the offline runs done by the subject. The channel chosen was the most discriminable channel as calculated from the fisher scores.

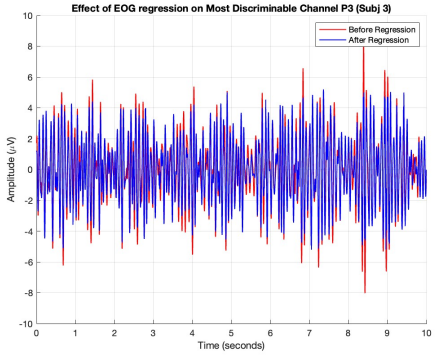

Fig. 4. The effects of regressing out EOG signals, from the offline baseline open-eye rest performed by the subject. The channel with the highest fisher score (P3) is isolated. Peaks that occur due to blinks and other motor movements are noticeably reduced.

bandpass parameters centered in the beta frequencies (8-30 Hz) in an attempt to find a bandpass that was most conducive to classification. We concluded that the band [10 Hz, 20 Hz] was most effective. [3]

This chosen band is physiologically relevant, as beta frequencies are associated with motor activity.

D. EOG Removal

While we were directed to avoid any unnecessary movement during data collection, including blinking, we were unable to achieve complete stillness. Because of this, artifacts may be present within the data. To remove these artifacts, we conducted a regression between the EOG channels (EOG, sens7, sens8, sens9) and every other channel, subtracting from each non-EOG channel the estimated artifacts based on the EOG channels. For a single channel, the regression was conducted as follows:

- First, conduct a polynomial (cube) regression between the EOG channel and the current channel.
- Estimate the artifact based on this regression model and subtract it from the channel data.
- For each remaining point outside of 2.5 sd of the mean, replace it with the average value of a window around the point.
- Repeat this process for each EOG channel.

This process was repeated for all EEG data.

As is seen in the figure , large artifacts in the data likely due to blinks are successfully removed. [4]

E. Rest Removal

Another factor of concern is the baseline difference in brain state between the two data collection days for each subject. To minimize this concern, another regression step is applied between each session's EEG data and its corresponding open-eye baseline resting state data, subtracting the estimated artifact channel by channel. For a single channel, this regression was conducted as follows:

- First, conduct a linear regression between the baseline open-eye rest data channel and the current channel, trimming the baseline data length to match the length of the current channel.
- Estimate the artifact based on this regression model and subtract it from the channel data.

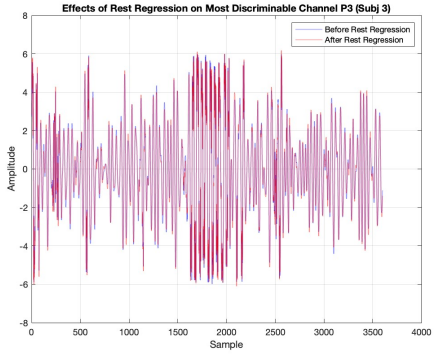

Fig. 5. The effects of regressing out baseline rest signals from the first trial of the first offline run done. The channel with the highest fisher score (P3) is isolated. There is relatively little change done by the regression, possibly due to the previous EOG regression.

This process was repeated for all EEG data.

As is evident from the plot, this rest regression step had a negligible effect on the data. This is likely due to the several prior filtering and artifact removal steps. [5]

F. Input to Feature Extraction

The input to feature extraction follows the same structure as described in data loading, but with filtered and artifact-adjusted data. Additionally, only the top 10 most discriminable channels remain. This is the final state of the data prior to feature extraction and classification.

G. Feature Implementation

In order to reduce the dimensionality, enhance patterns, improve interpretability, and facilitate classification, feature extraction was performed on this preprocessed data. For each of the top 10 most discriminable channels, signals were segmented using a windowing size of 256ms and an overlap of

0.5, chosen through a grid search with a window size range from 102.4ms to 1126.4ms and an overlap range from 0.25 to 0.75, with ranges chosen based on overall signal length. Then, for each of these windows 13 features were evaluated: Mean Absolute Value (MAV), Variance (VAR), Root Mean Square (RMS), Wave Length (WL), Autoregressive (AR), Zero Crossing (ZC), Slope Sign Change (SSC), Power Spectral Density (PSD), Higuchi Fractal Dimensionality (HFD), Symbolized Shannon Entropy (SSE), Global Dissimilarity (GD), Global Field Power (GFP), and Continuous Wavelet Transform (CWT). For each of these features, one value was calculated per datapoint per channel and all values were appended together, causing each feature to have 1,350 values on average.

When deciding which features to implement, it was imperative to choose multiple features from each of the domains, temporal, spatial, and frequency, in order to retain as much information as possible when going from the pre-processed data to features [3].

In the temporal domain, the features MAV, VAR, RMS, WL, ZC, and SSC were implemented. MAV was chosen as one of the temporal domain features because it provides a useful measure of the deviation of a signal from the mean, which could be useful in discriminating between reach and rest because reach may fluctuate more than rest. VAR was chosen because it also shows the spread and dispersion of data around the mean, making it useful for discriminating for the same reason that MAV is. RMS was chosen because it provides a measure of the overall signal amplitude over time, which could aid in discriminating between reach and rest because reach may have higher overall amplitudes due to more activity. WL was chosen because by calculating the length of the waveform at different segments, it provides a measure of signal complexity over time which could aid in discriminating between reach and rest as reach may be more complex due to it being an action versus rest which is in a meditative state. ZC was chosen because it provides a measure of the rate at which the signal changes its sign, which could aid in discrimination as reach may have more frequent changes in neural activity than the rest state. SSC was chosen because, similar to ZC, it measures the rate at which the signal's slope changes sign, also aiding in discrimination as reach may have more frequent neural activity changes than rest. These features were calculated as described in the temporal domain feature equations section of the supplementary materials (equations 1-6).

In the frequency domain, the features PSD, SSE, and CWT were implemented. PSD was chosen because it provides a measure of the power distribution of the signal across different frequencies, aiding in discriminability as different frequencies are more likely to be active in rest as compared to reach as those 2 states would evoke different brain wave types (most likely alpha and beta waves respectively). SSE was chosen because it measures the uncertainty and randomness in a signal's amplitude distribution, aiding in discriminability as the reach state may have more complexity in frequency and amplitude distribution than reach in the motor regions. CWT

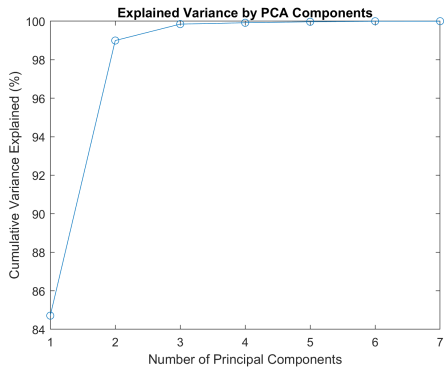

Fig. 6. The results of running principle component analysis on all 13 of our features. 2 of the 13 principal components are used to explain 95% of the variance observed in this feature matrix.

was chosen because it decomposes a signal into its wavelet components, which captures changes in frequency content and temporal dynamics, aiding in discrimination as reach will most likely have a different distribution or magnitude of wavelet coefficients than the rest state. These features were calculated as described in the frequency domain feature equations section of the supplementary materials (equations 7-9).

In the spatial domain, the features HFD, GD, and GFP were implemented. HFD was chosen because it estimates the fractal dimension of a signal, quantifying the spatial irregularity and complexity of the signal, aiding in discriminability by potentially detecting higher spatial complexity in reach compared to rest states. GD was chosen because it measures the overall differences between 2 signals, in our case between the same data point between channels, aiding in discriminability by potentially detecting signal differences between motor and non-motor channels when comparing reach versus rest states. GFP was chosen because it quantifies the overall magnitude of EEG activity over all 10 channels assessed at a given time point, showing global synchronization, potentially aiding in discriminability as the reach state may have more synchrony in the motor channels while the rest state may have more overall synchrony due to being a more meditative state and having overall lower activity. These features were calculated as described in the frequency domain feature equations section of the supplementary materials (equations 10-12).

H. Feature Selection

Once these features in all 3 domains were implemented, principal component analysis was performed to select the ones contributing the most variance to the overall feature set. This selection of a smaller group of features was done in order to reduce the dimensionality of our inputs into our classification model while still retaining as much feature information as possible.

As seen above in figure [6], PCA revealed that at a minimum 2 components were needed to explain over 95% of the feature matrix's variance. However, when PCA was run on all of the online and offline sessions, it was shown that up to

5 components were needed to explain the overall variance from the set of 13 features, leading us to conclude that we should use 5 features to keep the data necessary to explain overall variance for all subjects and sessions. However, the top features chosen by PCA were different across the online and offline sessions and between runs, causing a need to use the components that were the top features overall. To assess which features were the overall top, the top 5 features were aggregated for each session and run and the count of each feature was assessed to find the 5 features used as top features the most throughout the data. After running this analysis, it was found that the features that explained the most variability overall were VAR, WL, ZC, SSE, and HFD, containing at least 1 feature from each domain type.

I. Final Feature Matrix

After the features explaining most data variance were chosen through PCA, these features were combined into a feature matrix with columns representing each feature and rows representing each data point. These feature matrices were calculated for each signal and run and then were used as input for our machine learning model.

J. Model Selection

An important note should be made on the data fed to the classifier. Because the feature extraction pipeline and the classifiers require consistent dimensions, and each window may produce a different number of features depending on the result of the trial (timeout or sustain), some data segmentation and selection must be performed. We chose to make the assumption that the signals must be discriminable during the task period to even make a decision, so then we cut out the sustain to maintain a comparable window length across all trials. Although this did give us significant results from classification, we do acknowledge there is a loss of data from the sustain periods and from any time periods that are lost.

To select the best model, we decided to test an array of known classifiers, including some used in relevant literature. These were the linear classifier, the quadratic classifier, SVM - one vs one, and the Gaussian kernel. To accurately assess the performance of these classifiers, we first performed run-wise cross validation for each individual type of model. We chose to do this run-wise to preserve the independence of data in each run and to prevent data leakage between runs. Because of the non-linearity of biological signals, it's important to account for inter-trial differences. From these classifiers(one for each run), we collected data representing the average and maximum classification accuracy. We looked at these metrics in order to also get an idea of the spread of the data. Fortunately, the Gaussian kernel had both the highest mean and highest max accuracy. In the event that one classifier type had the highest mean but a larger standard deviation, further statistical analysis would have to be done on these metrics to choose a classifier. However, a potential drawback to this comparison is that the low number of runs may contribute to a difficult and uncomprehensive analysis. Once the type of classifier was selected, we trained it on the entire offline dataset to produce our final classifier to be used for testing and evidence accumulation. This ensures that we utilize the train/offline dataset to its maximum capacity.

TABLE I
THE MAXIMUM ACCURACY ACHIEVED AND AVERAGE ACCURACY ACROSS ALL RUNS POST RUN-WISE CROSS-VALIDATION.

	Model			
	Linear	Quadratic	SVM	Gaussian Kernel
Max	54.370	50.333	50.203	72.666
Mean	48.531	50.203	49.981	57.561

TABLE II
RUN-WISE ACCURACIES FOR EACH ONLINE SESSION, TESTED ON THE FINAL GAUSSIAN KERNEL MODEL TRAINED ON THE FULL OFFLINE(TRAIN) DATASET. ONLINE SESSION 1 IS MISSING RUNS 6 AND 7 BECAUSE THERE WERE ONLY 5 RECORDED RUNS ON DAY 1.

	Run					
	1	2	3	4	5	6
Session 1	0.5881	0.5200	0.6261	0.6481	-	-
Session 2	0.5411	0.4897	0.4910	0.4895	0.5627	0.5362

The resulting accuracies, as shown in the figure above, were quite consistent with the accuracies obtained from cross-validation, although none exceeded the maximum accuracy of 72.66%. However, this is expected, as we purposely separated offline and online data into train and test splits respectively due to both the nature of the experiment and the inherent differences in the signal sets.

Furthermore, we see in figures [7] and [8] that the results are heavily skewed towards the rest motion, as most of the rest motions are classified correctly, but the reach motions are also wrongly classified as rest. It is interesting that during the training and cross validation, results were still skewed, but far less biased towards rest than during testing.

V. RESULTS AND DISCUSSION

A. Evidence Accumulation Paradigm

The leaky integrator solution cited in a previous work was implemented for evidence accumulation. We chose smoothing parameter α of 0.7 by observation. Because α controls the degree of smoothing applied to the output probabilities and represents the weighting of the current observation relative to past observations, the choice of a larger α would smooth less and place more emphasis on the current observation. We noticed that our biased classifier pushed the classification probabilities toward extremes, and thus chose to rely more on the state of the most recent observation.

To produce a full-trial based classification result, threshold values for each task were chosen. A value of 0.75 was chosen for rest tasks, and 0.55 for reach tasks. To do this, we displayed the probability distributions of each class for all 20 trials from run 1, along with the expected class label. The thresholds were produced by visually taking a holistic overview of both the expected label, the trajectory of the distribution curves, and the thresholds used in our real-time recordings. We hoped to

TABLE III
THE NUMBER OF HITS(CORRECT TRIAL LABELS) USING OUR EVIDENCE ACCUMULATION INTERPRETATION. THE SECOND ROW BELOW EACH SESSION INDICATES THE TOTAL CALCULATED ACCURACY PER RUN. THE NUMBERS IN EACH SESSION CELL ARE IN THE FORM (REST HITS, REACH HITS)

	Run			
	1	2	3	4
Session 1	2, 4	5, 5	2, 6	6,5
total Accuracy	0.3	0.5	0.4	0.55
Session 2	3,5	6,6	5,9	4,3
total Accuracy	0.4	0.6	0.7	0.35

correct the bias towards the rest of our classifier by choosing a higher threshold for the resting state than the reaching state.

For each trial in each run, the probability distributions were compared at each time step. The trial result corresponded to the class that crossed threshold first. We chose to implement this style of paradigm rather than looking at the final aggregate timestep or an average due to the fact that the distribution values floated towards 0 for reach and 1 for rest. We do acknowledge that this may be an area of future improvement. As the table shows, our proposed paradigm does not result in a biased classification towards rest, as some runs show a greater classification accuracy of reach motions than rest. However, this does simultaneously increase the probability that a misclassification for rest may happen.

B. Paired t-tests

In order to probe our hypothesis, we wondered if there would be significant increase in accuracy between the two online sessions.

To assess this, a paired t-test was performed between the accuracies of the runs of each session. The two sets of data arise from the same subject and a paired t-test is designed to compare the means of the same group or item under two separate scenarios. For session 2, because there are more runs, we took the first 2 runs and the final 2 runs to perform a t-test at $\alpha = 0.1$ to see if there was a significant increase which may indicate that learning happened in between sessions.

We performed such a t-test twice, once on the trial-based accuracies computed from evidence accumulation from table III, and once for the accuracies computed from treating the online runs as testing data for the classifier. Both tests were statistically significant, with a $p-value$ of 0.0378 and 0.0605, respectively. This indicates that there is a significant difference in accuracy across the two online sessions.

However, a drawback here is that there exists an inconsistency between the final runs, for we chose the final runs with higher accuracies to perform our t-test on. This may prove to be faulty, and in the future, lowering the value of significance to 0.05 may provide more precise results and scientific value.

VI. CONCLUSION AND POTENTIAL FOR FUTURE WORK

Based on our analyses of our classification accuracies between sessions, we were able to show that subjects were able to successfully learn to control the upper limb harmony exoskeleton, therefore supporting our hypothesis. In order to further assess our hypothesis, future work could include training an RNN to see if classification accuracy improves and if we can observe learning with a better model as RNNs are better equipped to deal with time series data than the models we have used in this paper. Researchers could also test other deep learning models as well to see if they have a better accuracy due to their automated feature extraction through weightage based on what aids in classification accuracy. Along with this, in order to assess which signals are more associated with incorrect classification, ErrP analysis could be run. Lastly, our hypothesis could be extended to see if this learning can also be observed and sustained in more long term settings.

VII. STATEMENT OF WORK

- ***Julian* (33%):** Data Loading + Pre-processing Pipeline + Feature Extraction Grid Search
- ***Nihita* (33%):** Feature Implementation + Principal Component Analysis + Feature Extraction
- ***Emerald* (33%):** Classification Pipeline + Feature Extraction framework + Statistical Analysis/Evidence Accumulation

VIII. Supplementary Material

A. Temporal Domain Feature Equations

$$MAV = \frac{1}{N}\sum_{i=1}^{N}|x_i| \tag{1}$$

N = number samples in signal
x_i = signal value at sample i

$$VAR = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{2}$$

μ = mean of signal

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}x_i^2} \tag{3}$$

$$WL = \sum_{i=1}^{N-1}|x_{i+1} - x_i| \tag{4}$$

$$ZC = \frac{1}{N-1}\sum_{n=1}^{N-1}\begin{cases} 1 & \text{if } x[n]\cdot x[n-1] < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$sign(x)$ = sign of function

$$SSC = \sum_{i=2}^{N-1}\begin{cases} 1 & \text{if } (\text{sign}(x_i - x_{i-1}) \neq \text{sign}(x_{i-1} - x_{i-2})) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Frequency Domain Feature Equations

$$PSD(f) = |F(f)|^2 \tag{7}$$

$F(f)$ = signal Fourier transform at frequency f

$$SSE = -\sum_{i=1}^{N}P(x_i)\log_2(P(x_i)) \tag{8}$$

$P(x_i)$ = probability of occurrence of x_i in signal

$$CWT(a,b) = \int_{-\infty}^{\infty}x(t)\cdot\psi^*\left(\frac{t-b}{a}\right)dt \tag{9}$$

$x(t)$ = input signal at time$t$
ψ = analyzing wavelet function
$\psi^*(t)$ = complex conjugate of ψ
a = scale parameter/wavelet width
b = translation parameter/wavelet shift along time axis

Spatial Domain Feature Equations

$$HFD(k) = \frac{\log\left(\frac{N}{k}\right)}{\log(k)} \tag{10}$$

k = scale factor/number length k subseries used in calculation

$$GD(X,Y) = \frac{1}{N}\sum_{i=1}^{N}d(x_i, y_i) \tag{11}$$

x_i, y_i = data points at index i in signals x and y
$d(x_i, y_i)$ = distance measure between x_i and y_i

$$GFP(t) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}|x_i(t)|^2} \tag{12}$$

M = total channel number

Confusion Matrices for the classification of the runs of each online session on the final classifier

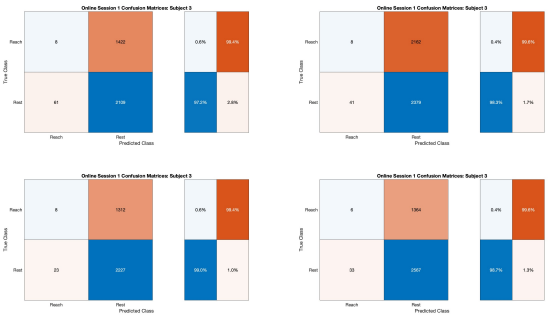

Fig. 7. The confusion matrices for each run of online session 1.

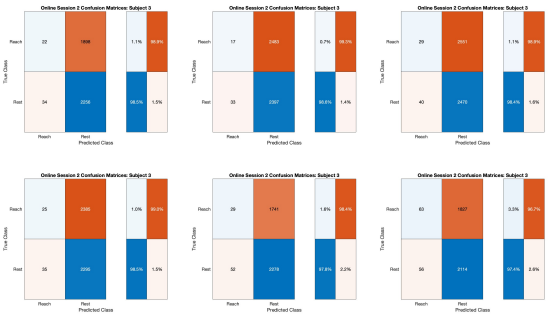

Fig. 8. The confusion matrices for each run of online session 2.

References

[1] Brunner, I., Lundquist, C.B., Pedersen, A.R. et al. Brain computer interface training with motor imagery and functional electrical stimulation for patients with severe upper limb paresis after stroke: a randomized controlled pilot trial. J NeuroEngineering Rehabil 21, 10 (2024).

[2] Shiman, Farid et al. "Classification of different reaching movements from the same limb using EEG." *Journal of neural engineering* vol. 14,4 (2017): 046018.

[3] Singh, A.K., and Krishnan, S.. "Trends in EEG signal feature extraction applications." Frontiers in Artificial Intelligence 5 (2023): 1072801.

[4] Saad M. Sarhan, Mohammed Z. Al-Faiz, Ayad M. Takhakh, A review on EMG/EEG based control scheme of upper limb rehabilitation robots for stroke patients, Heliyon, Volume 9, Issue 8, 2023, e18308, ISSN 2405-8440

[5] K. Mitra, F. S. Racz, S. Kumar, A. D. Deshpande and J. Del R. Millán, "Characterizing the Onset and Offset of Motor Imagery During Passive Arm Movements Induced by an Upper-Body Exoskeleton," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, MI, USA, 2023, pp. 3789-3794, doi: 10.1109/IROS55552.2023.10342492.

[6] Song, Y., Jia, X., Yang, L., and Xie, L., "Transformer-based Spatial-Temporal Feature Learning for EEG Decoding", ¡i¿arXiv e-prints¡/i¿, 2021. doi:10.48550/arXiv.2106.11170.

[7] Tang, Zhichuan et al. "An Upper-Limb Rehabilitation Exoskeleton System Controlled by MI Recognition Model With Deep Emphasized Informative Features in a VR Scene." *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society* vol. 31 (2023): 4390-4401. doi:10.1109/TNSRE.2023.3329059

[8] Al-Faiz, M., Al-hamadani, A., "Analysis and Implementation of Brain Waves Feature Extraction and Classification to Control Robotic Hand", *Iraqi Journal of ICT*, vol. 1, no. 3, pp. 31–41, Feb. 2019,https://doi.org/10.31987/ijict.1.3.35 doi: 10.31987/ijict.1.3.35.