Model for predicting game controls through EEG signals

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

L. Charan Sai Venkat Narayana	|	AP21110011621
S. Aditya Sesha Sai	|	AP21110011627
P. Mani Chandrika	|	AP21110011629

Course Name: Machine Learning Lab

Course Code: 336L

Under the Guidance of

Dr. Satya Pramod Jammy

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

May, 2024

Table of Contents

Abstract

This project is about training a machine learning model which can be used for the construction of a personalized game controller which has basic controls such as **up, down, left, right, and rest position** and can be controlled using brain signals, enabling a hands-free gaming experience. The dataset used to train this model will be one person's data from the BCI-2000 dataset which consists of the eeg signals. The basic idea is to collect the raw signals, preprocess the data and send it to the model which will predict the next move. This project employs various data exploratory methods which will be used to understand the data, then preprocess the data accordingly for better performance of the model.

Abbreviations

EDA	Exploratory Data Analysis
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
SVM	Support Vector Machine
RF	Random Forest
PCA	Principal Component Analysis
EEG	Electroencephalogram

List of Equations

Confusion Matrix:

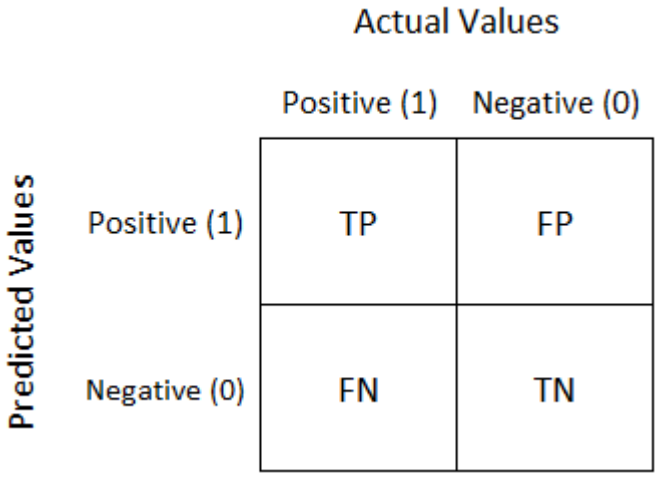


Figure 1: Confusion Matrix

Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$

Precision $= \frac{TP}{TP+FP}$

Recall $= \frac{TP}{TP+FN}$

F1 Score $= 2 \times \frac{Precision \times Recall}{Precision+Recall}$

1. Introduction

This study delves into unexplored area in the field of gaming innovation by attempting to build a predictive model that decodes game controls directly from EEG data. With an emphasis on using brainwave data from the BCI-2000 dataset, the primary aim is to develop a predictive model that provides personalized gaming interactions solely through cognitive input.

The creation of a reliable machine learning model that has been trained on EEG data is the central goal of this project. The model's goal is to use the inherent patterns in these signals to predict basic game controls like up, down, left, right, and rest position. This will enable neural instructions to revolutionize gaming interaction.

The methodology of this study revolves around the development of a robust predictive model trained on EEG data. The study prioritizes the interpretation of EEG signals by the machine learning models, aiming to showcase its efficacy in predicting game controls accurately and efficiently.

In addition to model development, this study places significant emphasis on data exploration and analysis. By thoroughly investigating the BCI-2000 dataset, the study seeks to uncover underlying patterns and trends within the EEG signals. Through this comprehensive approach, the study aims to gain deeper insights into the complexities of neural activity and refine the predictive capabilities of the model by elucidating the relationship between EEG patterns and game controls.

2.Dataset Description

Electroencephalography (EEG) is an electrophysiological monitoring method to record electrical activity of the brain. Brain consists of four lobes which is frontal, parietal, temporal and occipital lobe. Each lobe has their own respective function and it release different rhythmic wave when carry out different actions.

Dataset includes over 1500 one and two-minute EEG recordings, received from 109 volunteers. Subjects performed distinct motor/imagery duties at the same time as 64-channel EEG were recorded using the BCI2000 gadget. Every subject is made to perform 14 experimental runs which included two one-minute baseline runs with eyes open and eyes closed and three two-minute runs of each of the four following tasks:

A target or a signal appears on left side or right side of the screen and the subject opens and closes the corresponding fist till the target disappears. And in the next case the Subject imagines opening and closing the respective fist.

A target or a signal appears on top portion or on the bottom portion of the screen. The subject opens and closes both fists if it is on top and opens and closes both feet if it is on bottom. And in the next case the Subject imagines opening and closing fists or feet.

In precise, the experiments are: eyes open and eyes closed as base cases, open and close left or right fist as the first task, imagining it is task two, open and close both fists or both feet as task three, imagining it is task 4. These tasks are repeated further for testing the individual respectively. The records are supplied right here in EDF+ layout, containing 64 EEG indicators, each sampled at a hundred and sixty samples in keeping with 2d, and an annotation channel. Each annotation consists of one in

every of 3 codes (T0, T1, or T2). T0 , corresponding to rest, T1 corresponding to onset of movement (actual or imagined) of the left fist (in runs 3, four, 7, 8, eleven, and 12), Both fists (in runs five, 6, nine, 10, 13, and 14), T2 corresponding to onset of movement (real or imagined) of the right fist (in runs 3, 4, 7, eight, eleven, and 12), Both feet (in runs 5, 6, nine, 10, 13, and 14). "Within the BCI2000-layout versions of these documents, which can be available from the participants of this records set, those annotations are encoded as values of zero, 1, or 2 in the Target Code nation variable. The EEGs have been recorded from 64 electrodes as consistent with the global 10-10 system". The numbers beneath every electrode name indicate the order in which they appear within the information; be aware that alerts within the data are numbered from 0 to 63, even as the numbers within the discern range from 1 to 64.

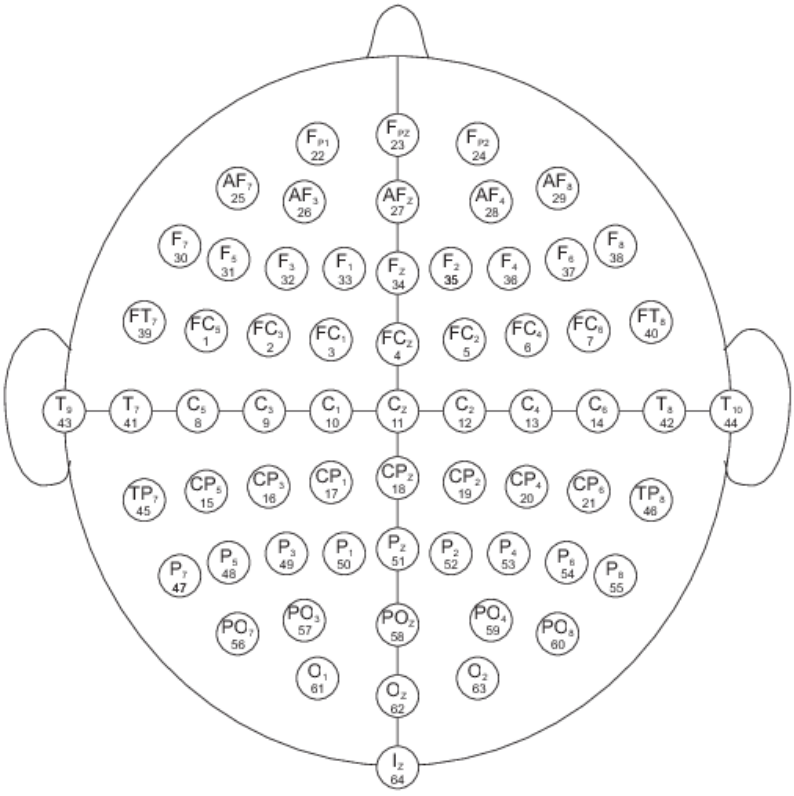


Figure 2: Illustration for EEG signals capture

3. Machine Learning Models

3.1 Support Vector Machine (SVM):

Although it is primarily known for its classification abilities, Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for regression and classification tasks. As seen in Fig. 5, SVM looks for the optimal hyperplane to maximize the margin between data points of different classes in order to improve generalization. Both linearly and non-linearly separable data can be handled by SVM and it transforms that data into higher-dimensional features spaces by using kernel functions. SVM is most impactful in high-dimensional spaces and is relatively less affected by overfitting, making it suitable for complex datasets. One drawback of SVM is its computational complexity, particularly with large datasets, because it necessitates solving a quadratic optimization problem. However, with the advent of efficient optimization techniques, SVM remains a popular choice in various applications like optical character recognition (OCR) systems for recognizing handwritten characters and digits, binary and multiclass classification problems, and so on.

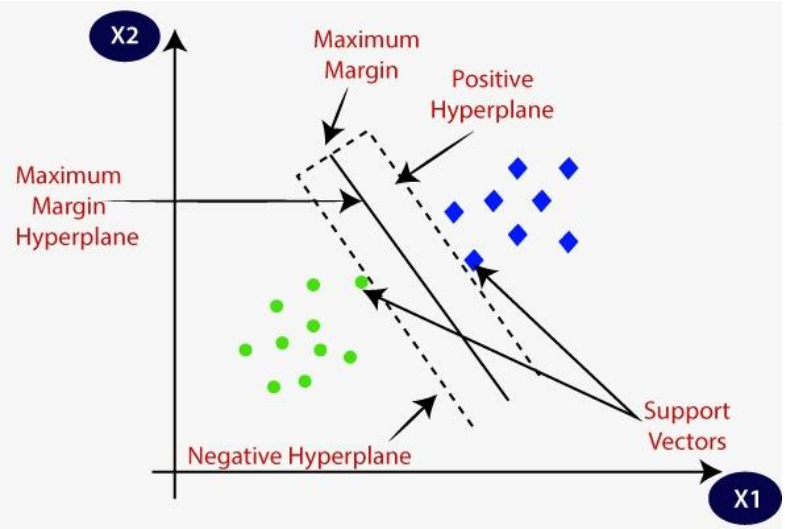


Figure 3: Illustration for SVM

3.2 Random Forest (RF):

Random Forest is a powerful ensemble learning method in machine learning that operates by constructing a multitude of decision trees during training and outputting the mode of the classes (in classification problems) or mean prediction (in regression problems) of the individual trees. Each decision tree in the forest is trained on a random subset of the training data and selects a random subset of features at each split point, hence the name "random forest."

By building multiple trees and aggregating their predictions, random forests mitigate overfitting and reduce variance, resulting in robust and accurate models. This approach enhances generalization performance compared to individual decision trees, making random forests widely used across various domains, from finance to healthcare and beyond. Additionally, the inherent randomness in the selection of data samples and features contributes to the model's resilience against noise and outliers, further bolstering its effectiveness in real-world scenarios.

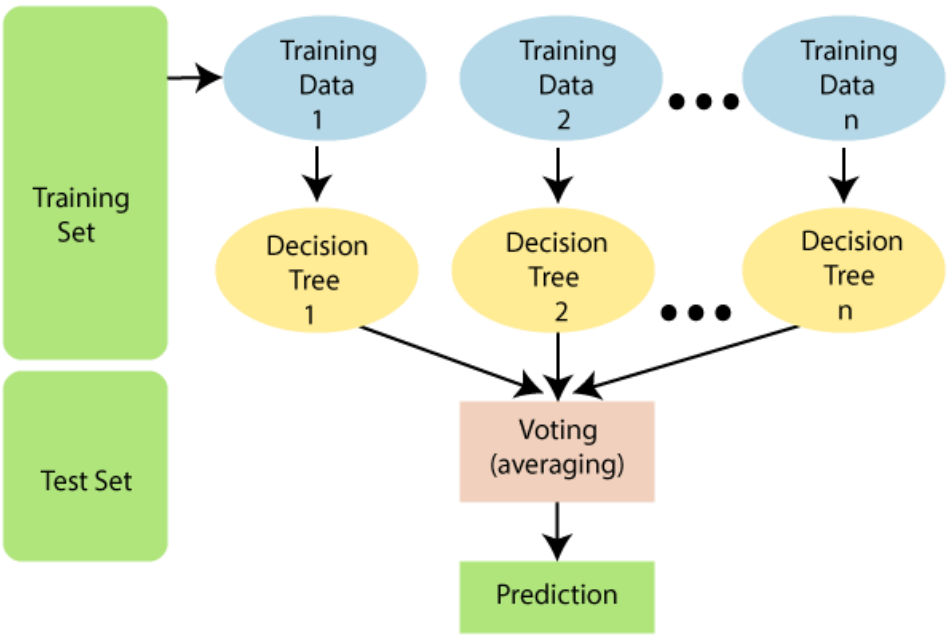


Figure 4: Illustration for Random Forest

3.3 Principal component Analysis (PCA):

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA is a type of dimensionality reduction technique, which is a feature extraction technique that aims to reduce the number of input features while retaining as much of the original information as possible.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality.

4. Methodology

Our project's methodology begins with "**read**" function in which we firstly read the raw EDF file and convert the data into a DataFrame, then we extracted events from the raw data, calculating durations for each action (up, down, left, right, rest). Based on the annotations for each action, we created a label vector. Then we appended the label vector as a column to the DataFrame and returned it, providing a structured dataset with annotations for further analysis.

Using the "read" function, we obtained separate DataFrames for real data and imaginary data, containing left, right, and rest actions, as well as for up, down, and rest actions. Following this, the respective DataFrames for real and imaginary data are merged separately to generate comprehensive datasets for each.

Next, we plotted the correlation matrices for real data and imaginary data and Feature Selection is conducted with a correlation threshold of 0.9 to remove highly correlated columns. Subsequently we normalized the data and performed PCA independently on both the real and imaginary datasets to reduce dimensionality while preserving most of the variance.

We trained SVM and RF classifiers on real and imaginary data across four stages: on Raw Data, Data after Feature Selection, Normalized Data and Data after PCA. The performance of each classifier is evaluated separately using standard evaluation metrics such as accuracy, precision, recall, and F1-score.

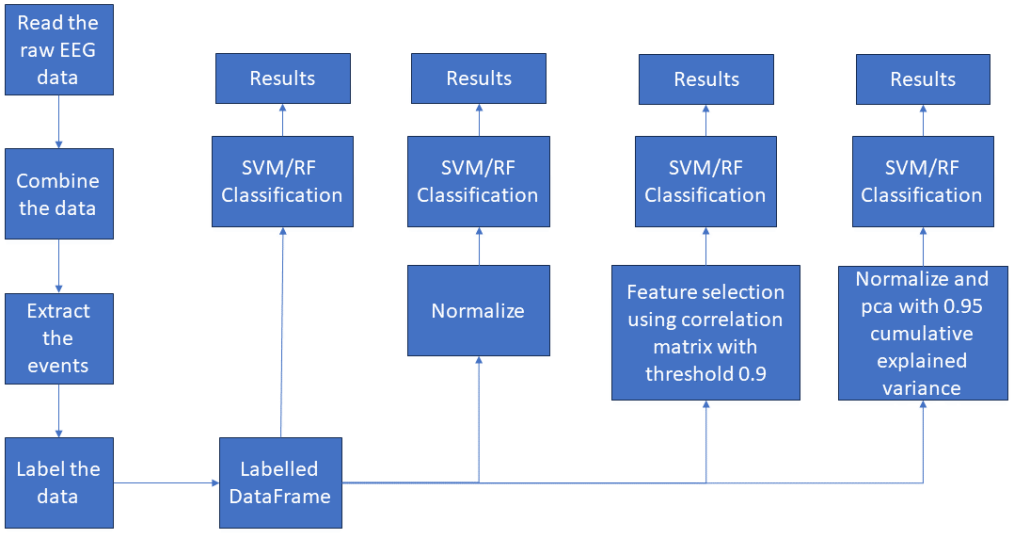
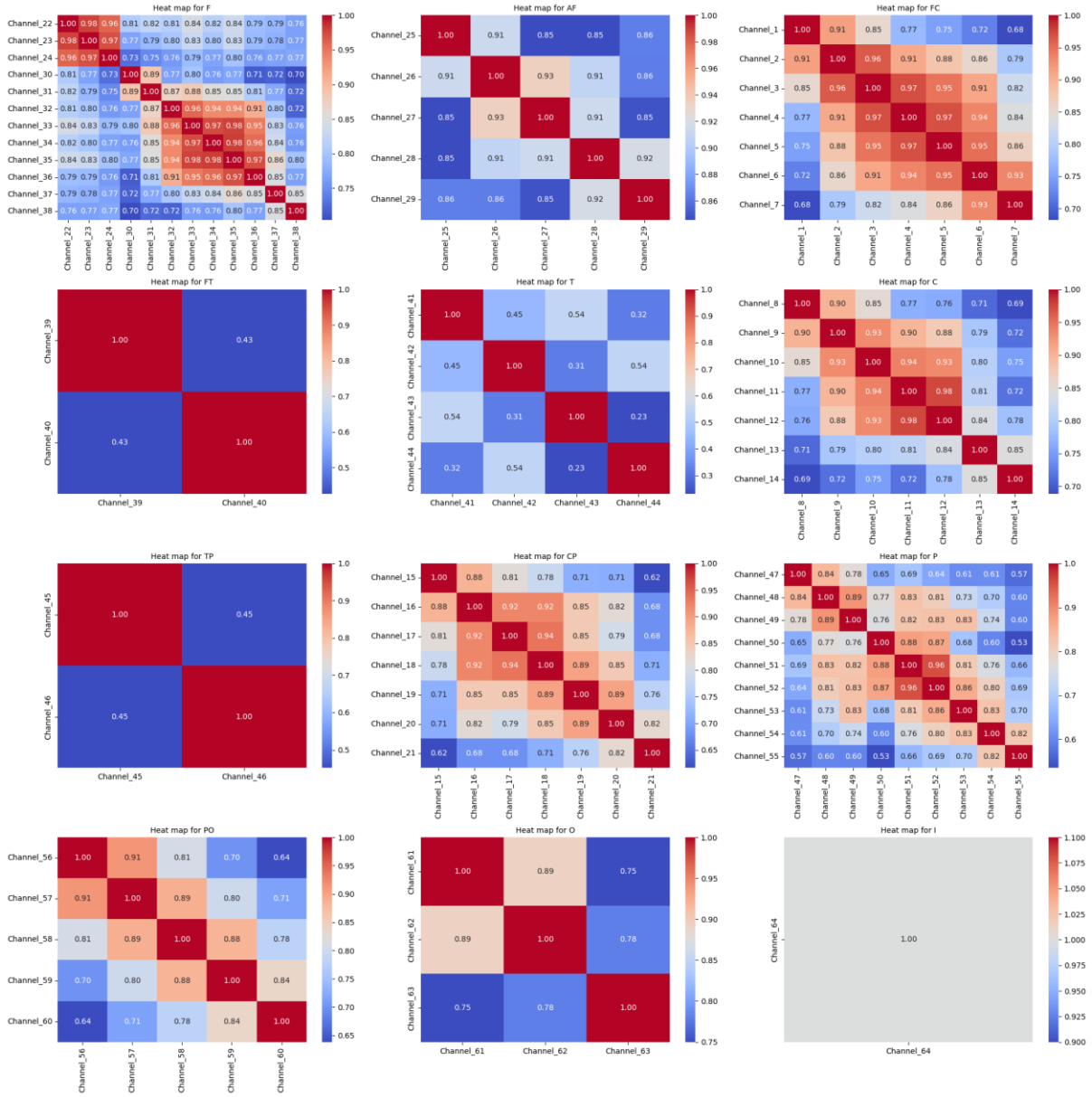


Figure 5: Methodology

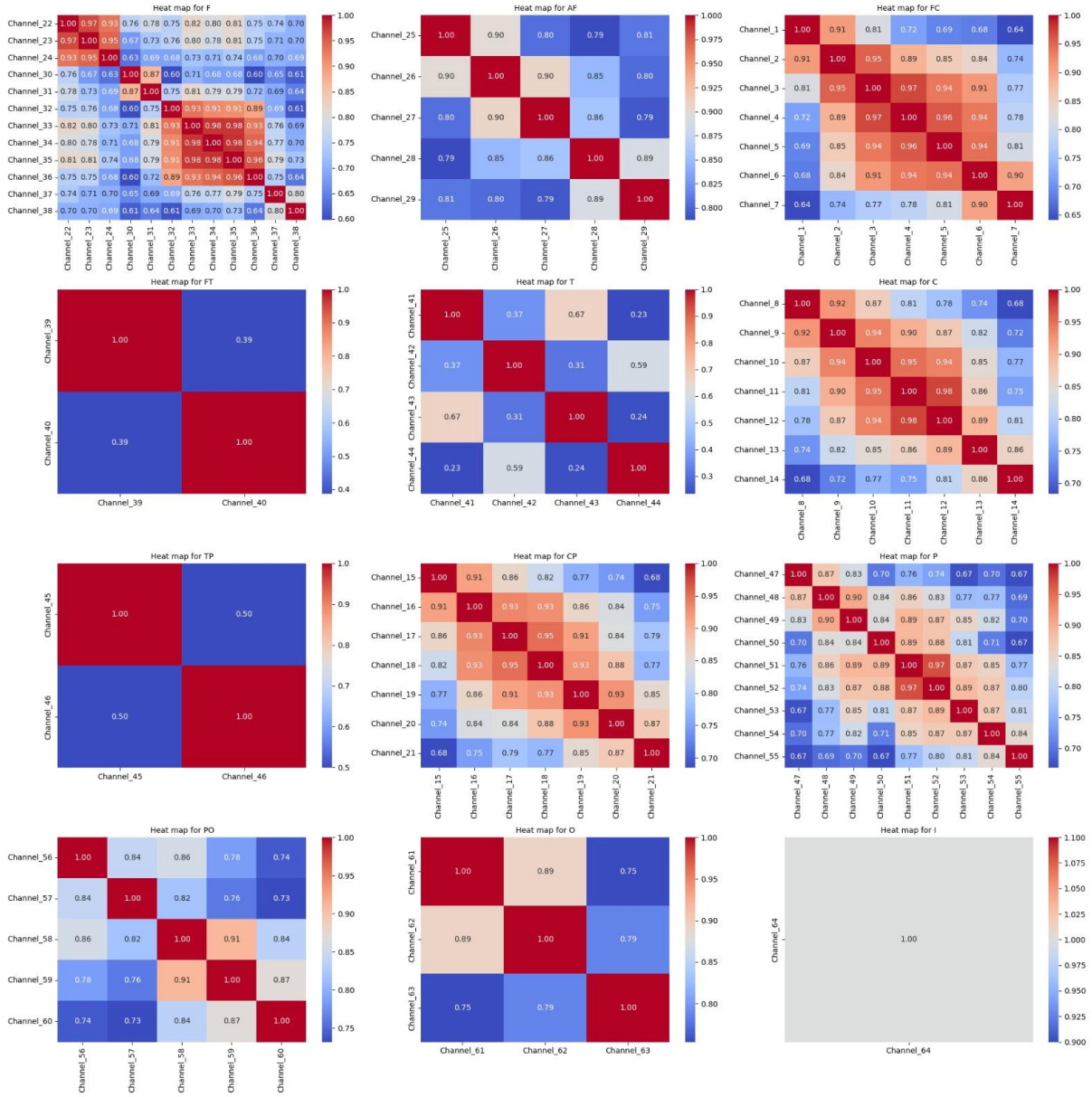
5. Data Analysis

The correlation matrix is plotted as a heatmap based on the different regions of the brain

For REAL DATA:



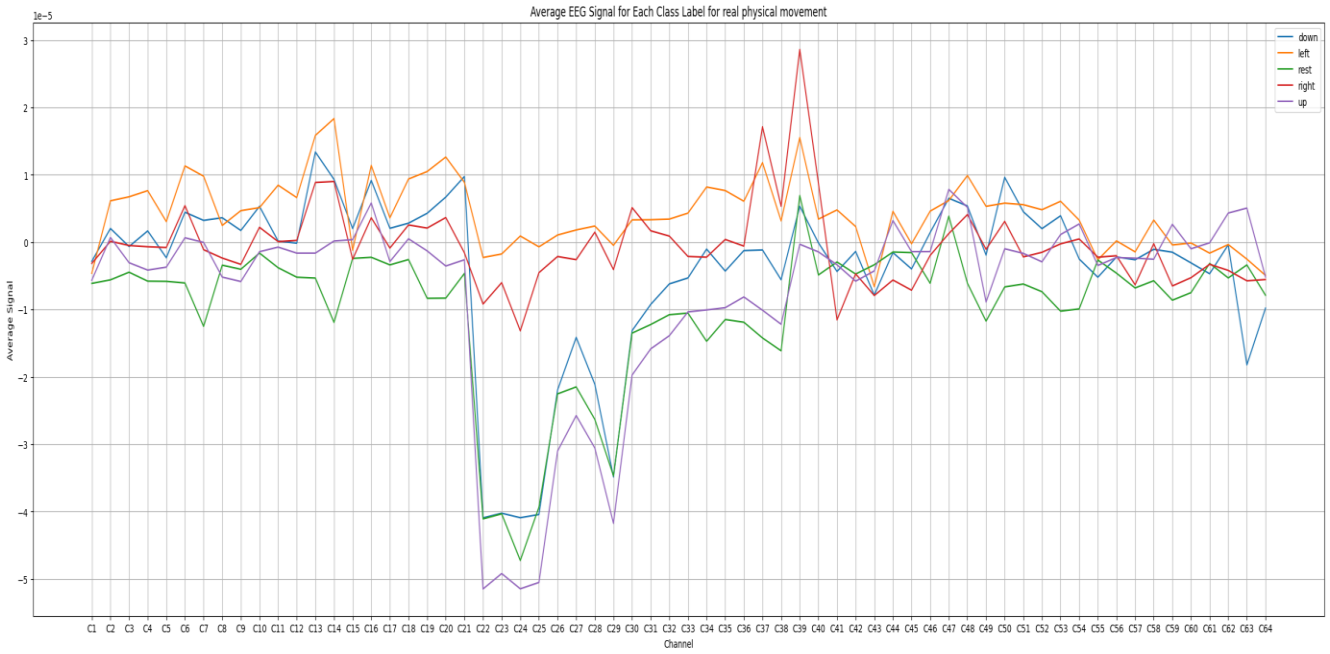
For IMAGINARY DATA:



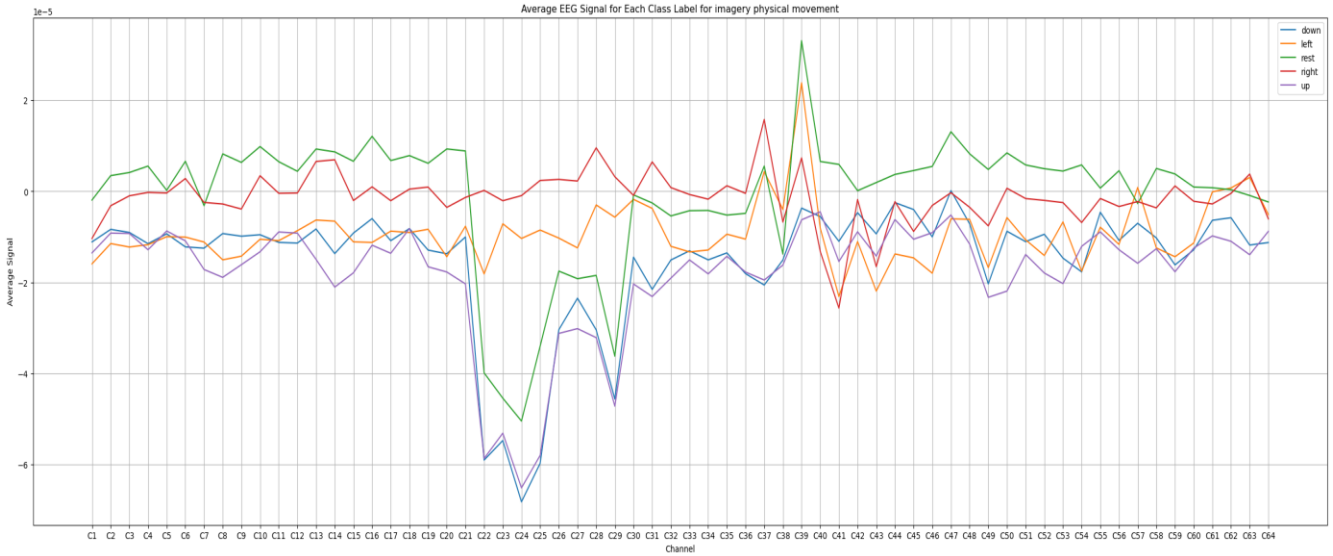
The correlation matrices' analysis indicates that there aren't any significant positive or negative correlations between electrodes located in the same brain region for both real and imaginary data.

Plotting average values of each channel and labels

Real tasks Avg value plot:

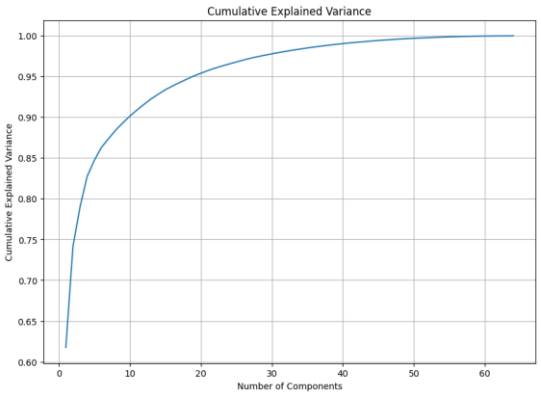


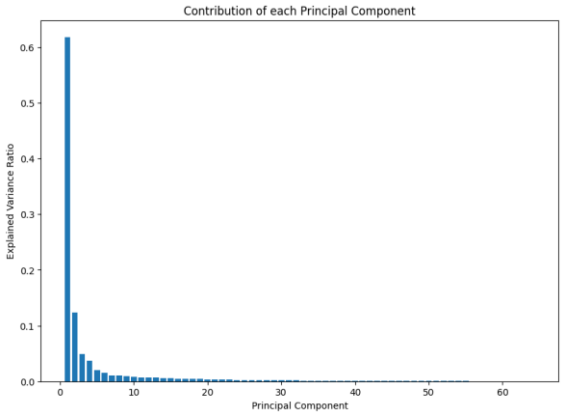
Imaginary tasks Avg value plot:



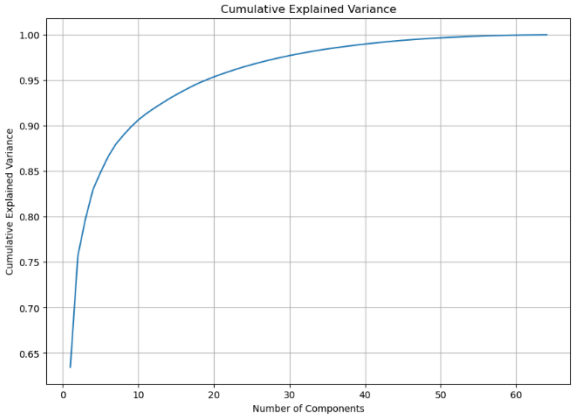
As shown above, the average values for up and down are comparable, whereas on the left and right have similar average values for their respective electrodes.

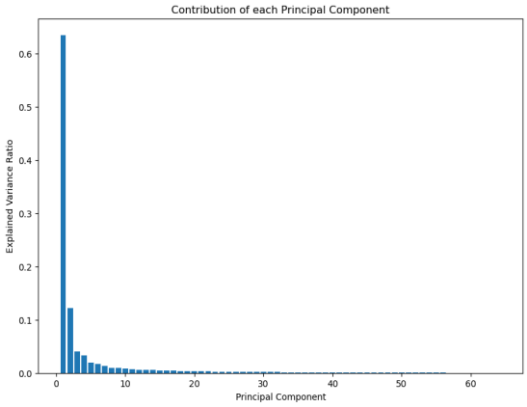
REAL DATA PCA:





IMAGINARY DATA PCA:

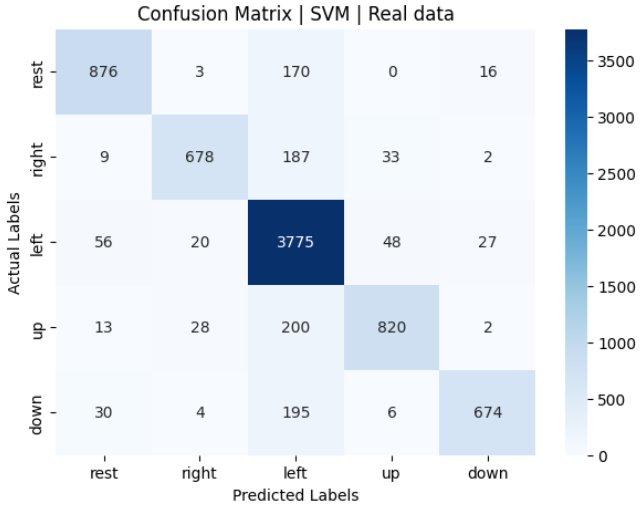




6. Results

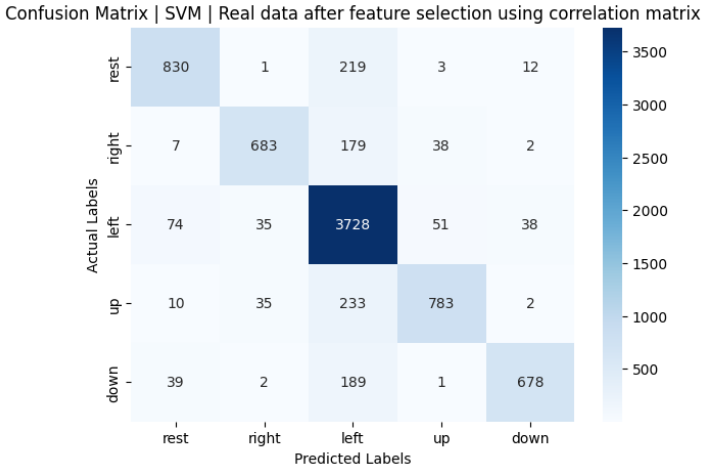
SVM CLASSIFIER-REAL DATA:

1. RAW DATA



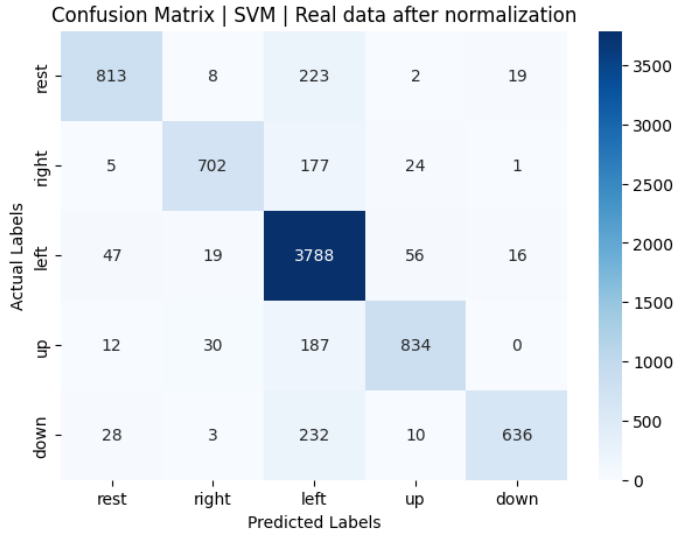
The SVM classifier trained on raw data achieved 87% accuracy with high precision for "up" (0.93) and recall for "rest" (0.96), but "left" and "right" classes showed lower recall.

2. DATA AFTER FEATURE SELECTION USING CORRELATION MATRIX OF THRESHOLD 0.9



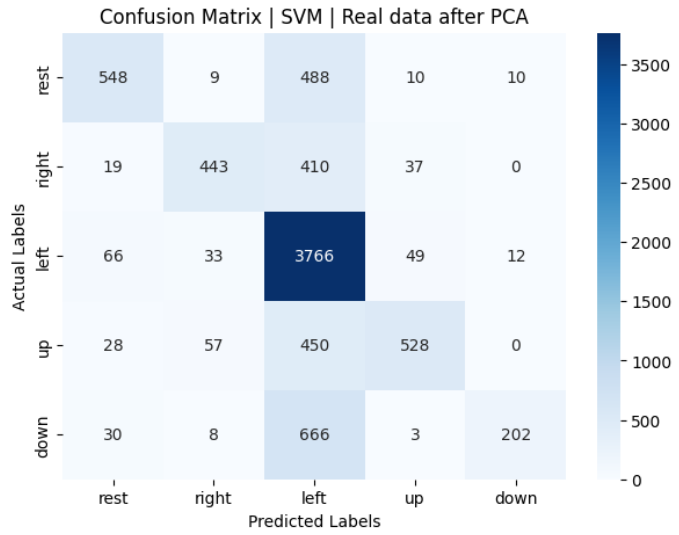
The SVM classifier on data after feature selection achieved a promising 85% accuracy with balanced F1 scores (0.82-0.83) for most classes. However, the "left" and "right" classes show lower recall.

3. AFTER NORMALIZING THE DATA



The SVM classifier on data after normalization achieved a strong 86% overall accuracy with balanced F1-scores (0.80-0.89) for most classes. However, the "up" class shows lower recall.

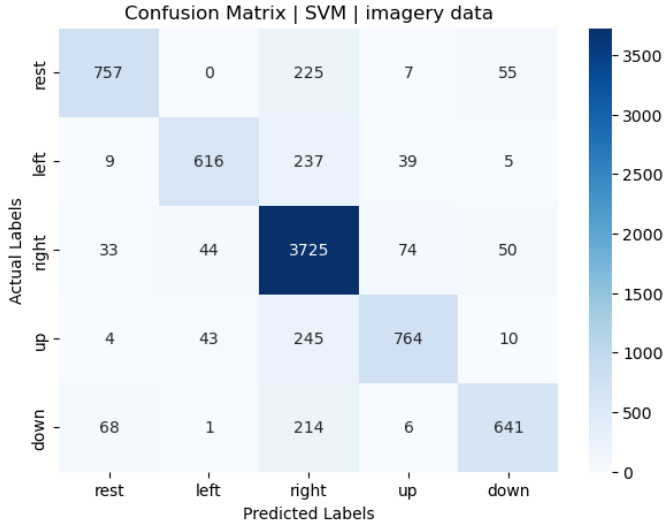
4. AFTER PCA



The SVM classifier after PCA on imaginary data achieved a moderate 70% overall accuracy. However, significant gaps exist between precision (0.79-0.90) and recall (0.22-0.96) for all classes, indicating challenges in identifying true positives.

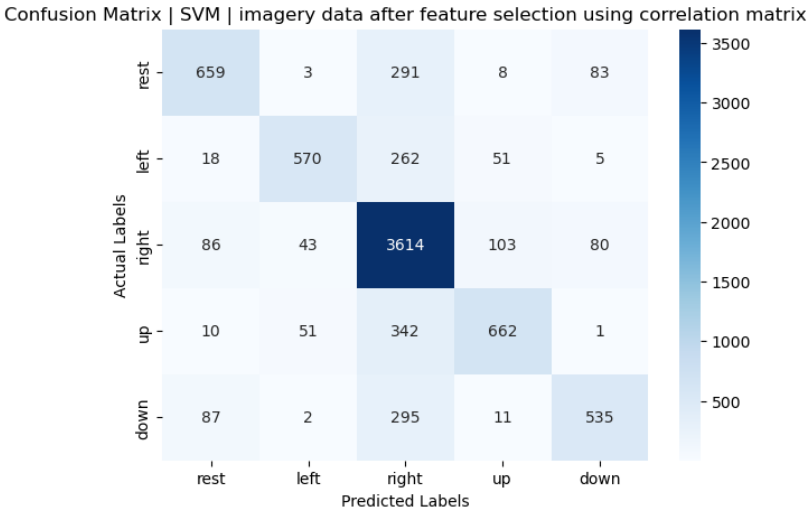
SVM CLASSIFER- IMAGINARY DATA:

1. RAW DATA



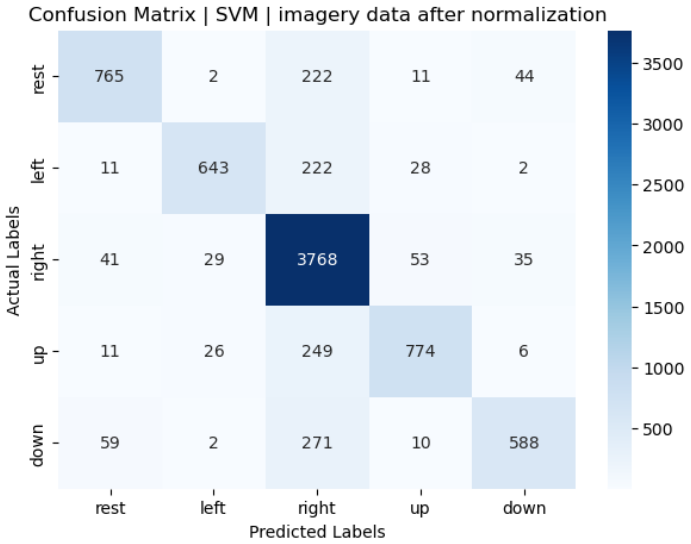
The SVM classifier on imagery raw data achieved a promising 83% overall accuracy with balanced F1-scores (0.76-0.87) for most classes.

2. DATA AFTER FEATURE SELECTION USING CORRELATION MATRIX PF THRESHOLD 0.9



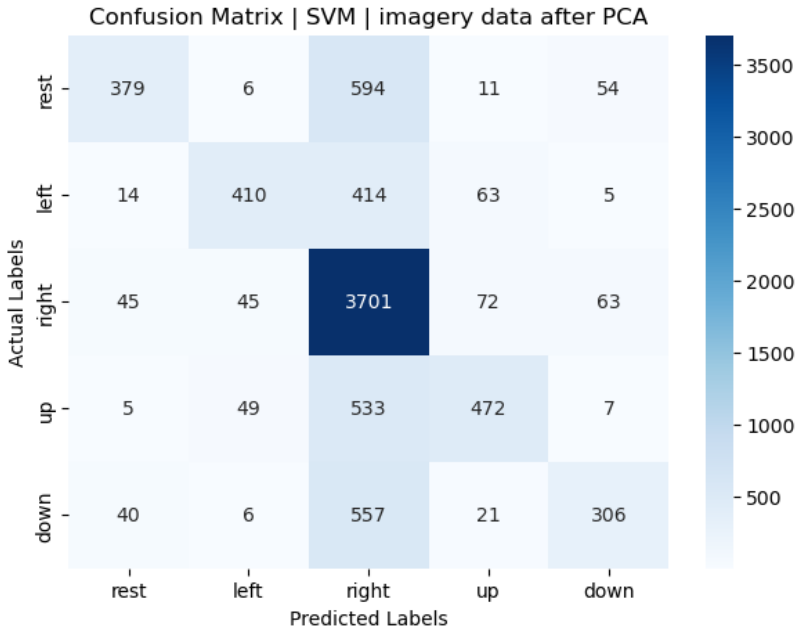
The SVM classifier on selected features achieved a moderate 77% overall accuracy. A significant disparity exists between precision (0.75-0.85) and recall (0.58-0.63) for all classes.

3. AFTER NORMALIZING THE DATA



The SVM classifier on normalized imagery data achieved a promising 83% overall accuracy with balanced F1-scores (0.73-0.80) for most classes. The confusion matrix (if available) could still reveal areas for improvement in recall, particularly for the "up" class (0.63).

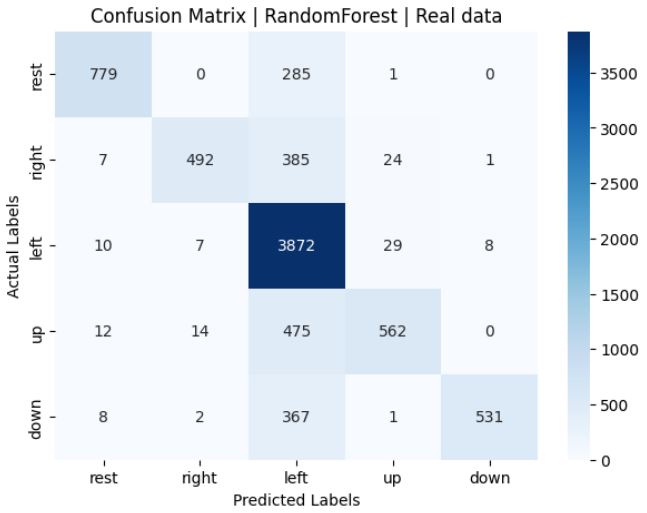
4. AFTER PCA



The SVM classifier on imagery PCA-transformed data achieved a moderate 67% overall accuracy. A significant gap exists between precision (0.70-0.79) and recall (0.33-0.45) for all classes, especially "up," "left," and "right."

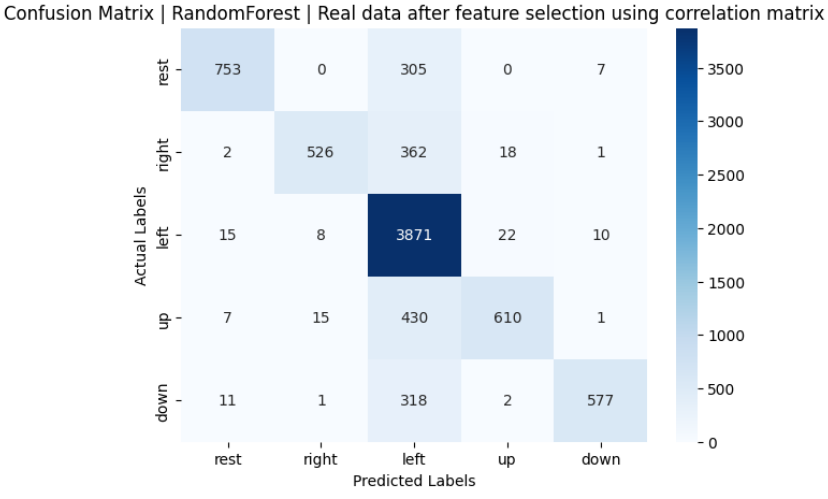
RANDOM FOREST- REAL DATA:

1. RAW DATA



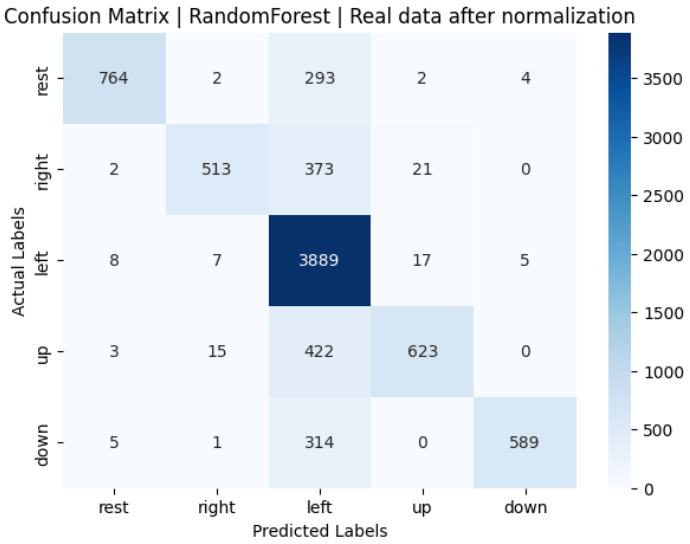
RF Classifier on raw data achieved a promising overall accuracy of 81%. While all classes exhibit high precision (0.94-0.98), a gap exists between precision and recall for "left", "right", and "up" classes (recall: 0.56-0.65). This difference indicates the model might be prioritizing precision over recall when classifying these specific classes in the raw data.

2. DATA AFTER FEATURE SELECTION USING CORRELATION MATRIX THRESHOLD 0.9



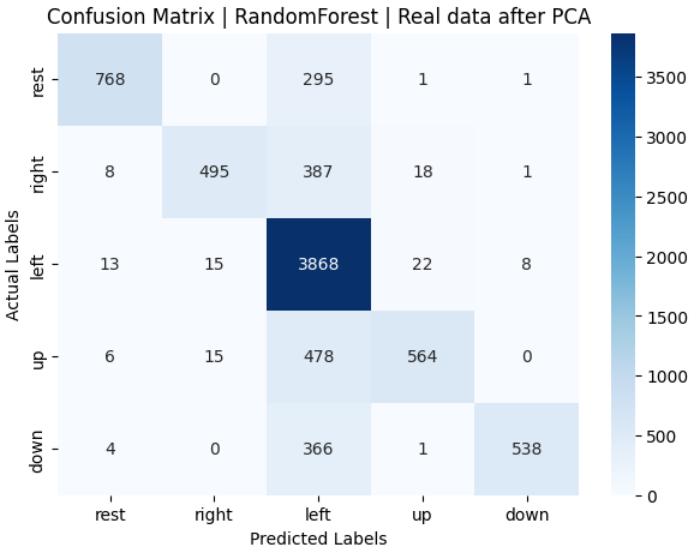
The Random Forest Classifier on data after feature selection achieved a promising 81% overall accuracy. All classes exhibit high precision (0.94-0.97). However, a gap exists between precision and recall for "left," "right," and "up" classes (recall: 0.57-0.63). Further analysis of these specific classes might be beneficial to understand this behavior.

3. AFTER NORMALIZING THE DATA



The Random Forest Classifier on normalized data achieved a strong 81% overall accuracy with high precision for all classes (0.94-0.97). However, a gap exists between precision and recall for "left," "right," and "up" classes (recall: 0.57-0.67).

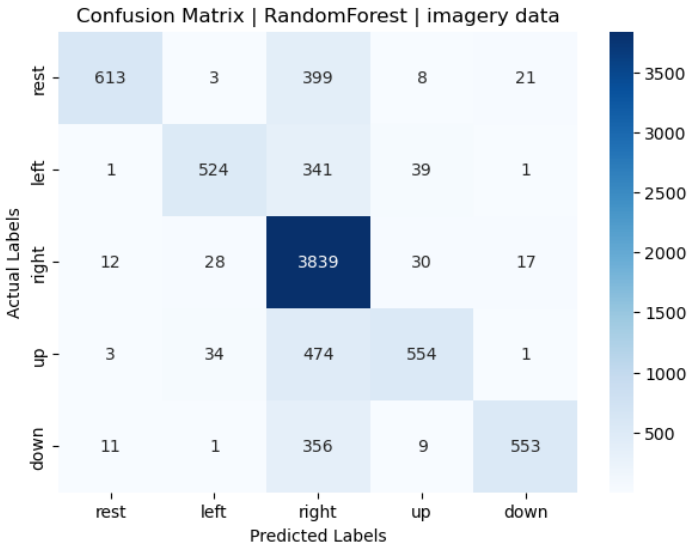
4. AFTER PCA



The Random Forest Classifier on PCA-transformed data achieved a 79% overall accuracy with high precision for most classes (0.91-0.98). However, a disparity exists between precision and recall, especially for "left," "right," and "up" classes (recall: 0.50-0.58).

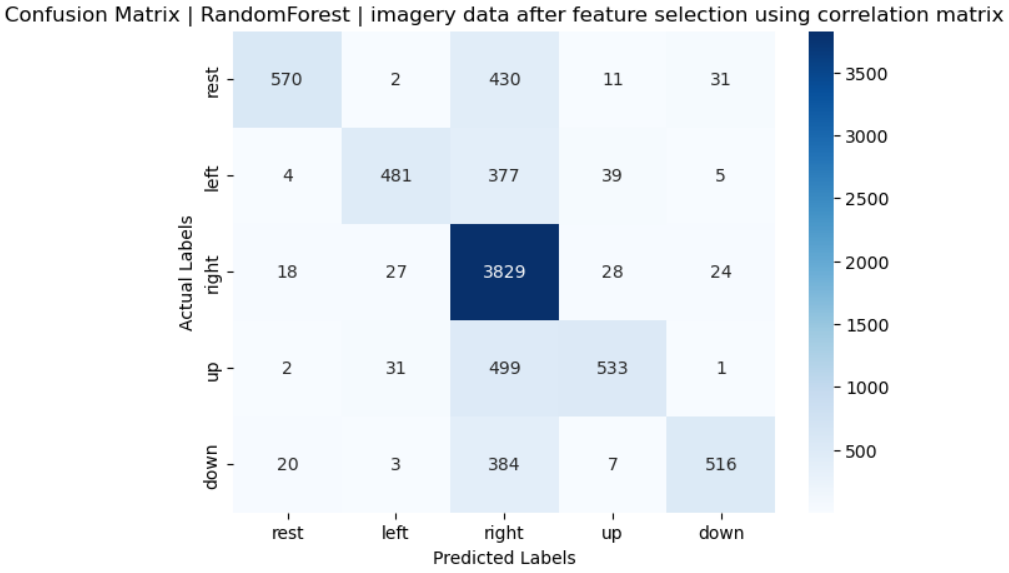
RANDOM FOREST- IMAGINARY DATA:

1. RAW DATA



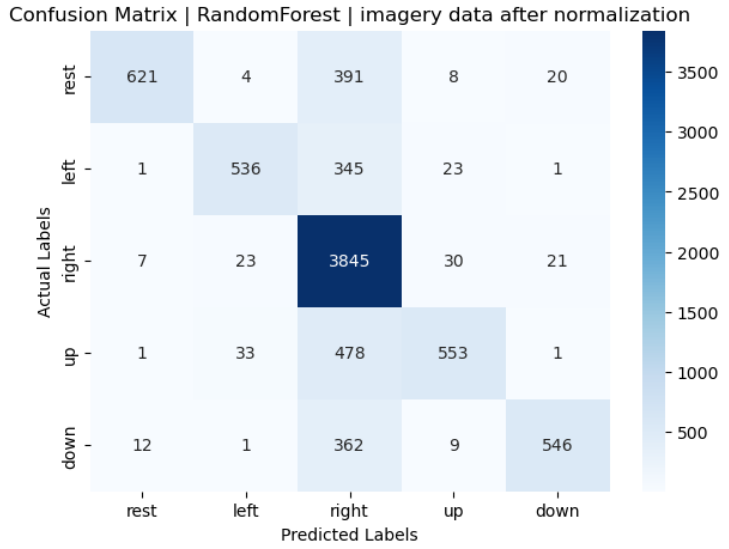
The Random Forest on raw data achieved a 77% overall accuracy with a disparity between precision (0.87-0.96) and recall (0.52-0.59) for all classes

2. DATA AFTER FEATURE SELECTION USING CORRELATION MATRIC OF THRESHOLD 0.9



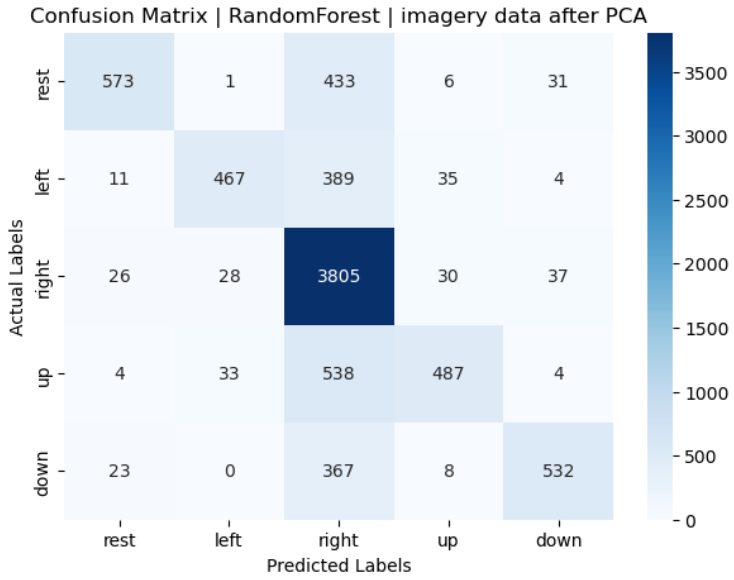
The Random Forest Classifier on data after feature selection achieved a moderate overall accuracy of 75%. While precision remains high for most classes (0.86-0.93), a gap exists between precision and recall (0.50-0.55) for "left," "right," and "up" classes.

3. AFTER NORMALIZING THE DATA



The Random Forest Classifier on normalized data achieved a moderate overall accuracy of 77%. While precision remains high for most classes (0.89-0.96), the confusion matrix indicates a gap between precision and recall for all classes, particularly "left," "right," and "up" (recall: 0.52-0.60).

4. AFTER PCA



The Random Forest Classifier on PCA-transformed data achieved a moderate overall accuracy of 75%. There's a disparity between precision (0.86-0.91) and recall (0.46-0.57) for all classes, particularly "left," "right," and "up" classes.

7.Conclusion

The project showed a promising result when using the EEG signals for predicting the control while the user performs certain tasks for each. The real task where the user performs the task actually has more impact in predicting the control the user wants to perform through his tasks than the set of imagining tasks where the user imagines the task. However, imagined movement data still yielded reasonable accuracy, particularly with SVM classifiers on raw data it also underscores the need for continued research and development efforts to realize the full potential of brain-computer interfaces.

we can unlock new possibilities for enhancing human-machine interaction and empowering individuals with diverse abilities to interact with technology seamlessly.

8. References

1. scikit-learn. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830
2. Goldberger, A., et al. "Physio Bank, Physio Toolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220." (2000).