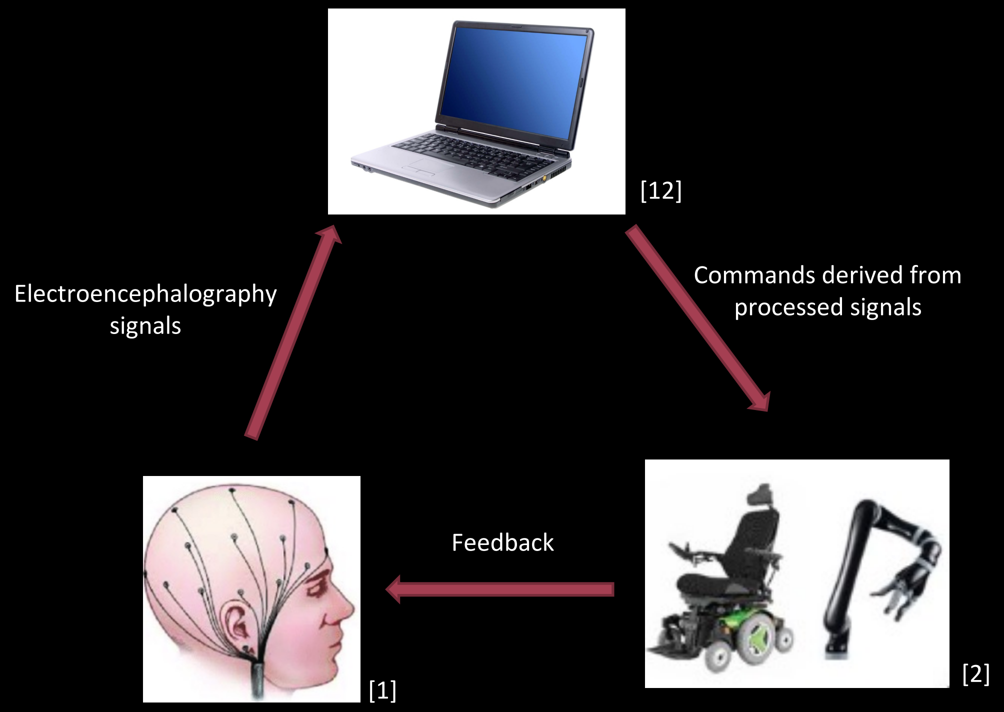
BRAIN-COMPUTER INTERFACE MOVEMENT DECODING

ADITYA BHAMIDIPATI

TABLE OF CONTENTS

INTRODUCTION

A brain-computer interface (BCI) is a system capable of extracting brain signals, interpreting them through signal processing, and creating corresponding digital commands that control output devices [1]. BCIs typically leverage electroencephalography (EEG) signals collected from brain activity through electrodes attached to an individual's scalp to create output commands enabling the individual to achieve a desired action via an assistive device such as a robotic prosthetic [3]. For individuals suffering from motor disabilities, BCIs are powerful tools that allow them to interact with the environment around them. Therefore, the intended purpose of this project is to develop a signal processing support vector machine (SVM) model that is capable of classifying EEG signals, captured through 204 individual electrodes, as either left or right movement. The SVM model is designed to be utilized within a larger BCI system to decode input EEG signals. In this project, the concepts of two-class linear SVM and two-level cross validation (CV) that performs hyperparameter optimization of the SVM model's regularization constant are leveraged to create a BCI signal classification algorithm. The algorithm constructs an SVM model hyperparameter tuned through the CV process to maximize the model's classification accuracy. In addition to developing an optimal SVM classifier for EEG signal classification, another goal of this project is to leverage the optimal SVM classifier to determine which of the 204 electrodes have the highest weightage when determining if an EEG signal refers to left or right motion. Two simulations where the BCI signal classification algorithm utilizes different datasets of EEG signals are performed to evaluate both channel weightage and SVM model performance.



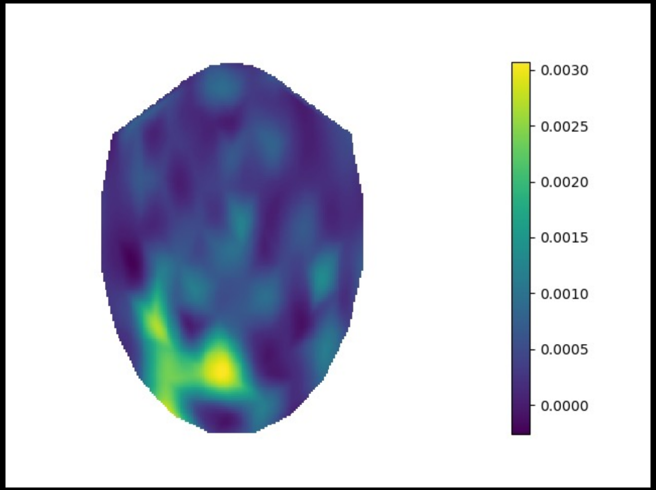
Brain-computer Interface System Architecture

INTRODUCTION

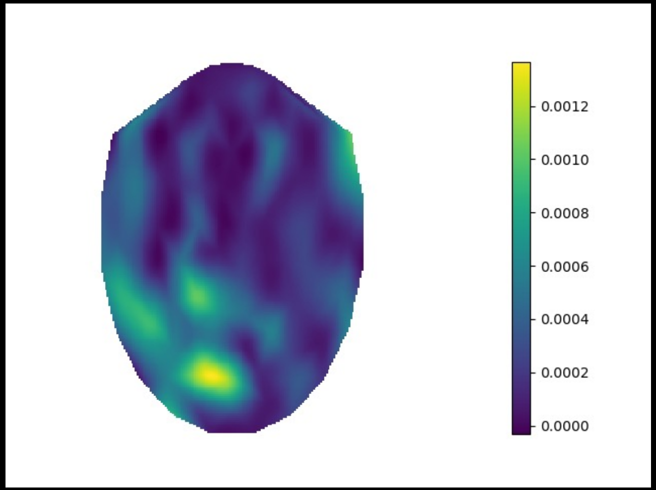
Key Findings:

1) Utilizing a smaller value for regularization hyperparameter of the SVM model leads to increased model accuracy.

2) Both the simulations illustrate that the dominant channels for left and right motion EEG signal classification are located in the bottom left-side of the human scalp



Simulation 1: Imaginary Data



Simulation 2: Overt Data

**Note: Project findings are further discussed in the conclusion

MATHEMATICAL FORMULATION

The BCI signal classification algorithm is formulated using two main mathematical frameworks/concepts:

1) Two-Class Linear Support Vector Machine| Pages 6-9
2) Two-Level Cross Validation| Page 10

After the linear SVM has been theorized using a convex optimization problem, the two-level cross validation process is utilized to find the SVM's optimal regularization parameter that leads to the highest probability of correct decision – accuracy.

The Python packages utilized to develop the algorithm are outlined on page 11.

TWO-CLASS LINEAR SVM

A two-class support vector machine (SVM) utilizing a linear kernel creates a decision statistic that is utilized to classifies new feature vectors as either positive or negative samples. More specifically, the classifier puts training feature vectors (data points) in a D-dimensional space where each feature in a feature vector is an axis of the space [4]. The classifier then creates a D-1 dimensional hyperplane – a linear decision boundary – that separates the positive and negative samples which is utilized in the decision rule to classify new, unseen feature vectors.

$$Decision\ Statistic: g(f) = v^T f + k \quad [4]$$

The calculated decision statistic which equates to the distance between a feature vector f and the hyperplane v is leveraged to classify the feature vector as either a positive or negative sample. To create an optimal hyperplane v and derive an optimal value for the offset k a convex optimization problem is formulated:

$$min_{v,k,\beta} \sum \beta_l + \alpha * v^T v$$

$$ST.\ s_i * (v^T f_l + k) \geq 1 - \beta_i$$

$$\beta_l \geq 0$$

$$(l = 1,2, \ldots, L) \quad [5]$$

s_i = {-1,1}
Depends on if the training feature vector f_l is a positive or negative sample

TWO-CLASS LINEAR SVM

This optimization problem seeks to maximize the distance between the hyperplane and feature vector closest to the hyperplane (margin) for linearly non-separable data and is solved utilizing Lagrangian multipliers [4]. Therefore, the convex optimization problem is converted to the following Lagrangian equation:

$$L(v,k,m,n) = \frac{1}{2}\|v\|^2 - \sum_{l=1}^{L} m_l \{s_l(v^T f_l + k) - 1 + \beta_l\} - \sum_{l=1}^{L} n_l \beta_l + \alpha \sum_{l=1}^{L} \beta_l \quad [5]$$

$\forall m_l \; and \; n_l$ (Lagrangian multipliers) $\geq$ 0; L = total number of training feature vectors

β_l is a slack variable introduced for misclassification.

$\alpha \sum_{l=1}^{L} \beta_l$ defines the regularization term. The regularization parameter α creates a trade-off between maximizing the margin and lowering misclassification. When α trends towards infinity, there is a high penalty for misclassification causing the the slack variable β_l to trend towards 0 resulting in a decision boundary that produces no classification errors (margin is small) [5]. The regularization parameter α is the hyperparameter optimized through the two-level cross-validation method.

TWO-CLASS LINEAR SVM

Utilizing the Lagrangian, values for each Lagrangian multiplier m_l are derived. Each m_l corresponds to a training feature vector (f_l) subject to the following conditions:

- If m_l = 0, f_l does not contribute to creating the hyperplane [5]
- If m_l> 0, f_l is a support vector [5]

After the support vector classifier has been trained, only the feature vectors classified as support vectors are necessary to create the decision boundary (v) and the offset (k) utilized by the model to make classifications [4]:

$v = \sum_{l=1}^{L} m_l s_l f_l$ where L equals the total number of support vectors

Therefore, the decision statistic utilized by the model equates to:

$$Decision\ Statistic: g(f) = v^T f + k = \sum_{l=1}^{L} m_l s_l f_l^T f + b \quad [4]$$

The decision statistic is leveraged by the model to determine which side of the decision boundary the new feature vector (f) exists enabling for positive or negative classification.

TWO-CLASS LINEAR SVM

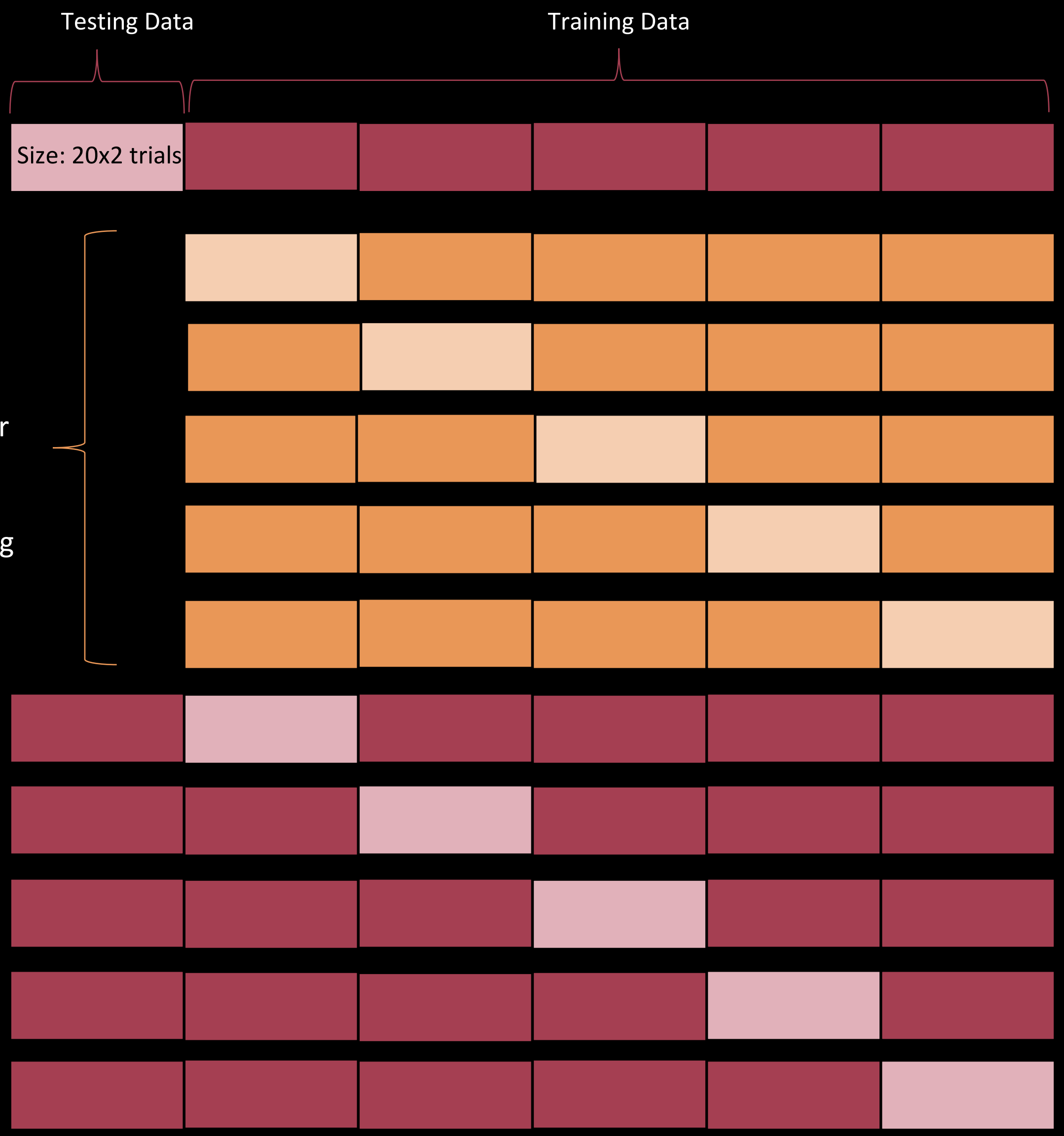
In the context of this project, the theorized two-class linear SVM model is meant to be leveraged within a BCI system to classify EEG signals as either referring to left (0) or right motion (1). The reason behind utilizing the SVM model is due to the high dimensionality of the EEG signal data. More specifically, the 204 electrodes are utilized to generate signal data producing 204 corresponding predictor variables that the model must consider when creating the output classification. Hence, each training feature vector contains a total of 204 features. However, as previously described, only training feature vectors classified as support vectors are considered by the SVM model when creating the decision boundary (hyperplane) utilized for classification allowing for increased computational efficiency when dealing with high dimensional data. Additionally, since SVM models utilize convex optimization when creating the optimal hyperplane that maximizes the margin, the solution yielded is the global maximum: the most ideal solution.

TWO-LEVEL CV

The goal of the two-level cross validation (CV) process is to combine both parameter optimization and model evaluation into a single method [6]. Therefore, the outer CV process (1st level) trains the SVM model via training data. The model then performs classifications on testing data after the inner CV process (2nd level) finds the optimal regularization parameter for the model.

The 1st level of CV determines the accuracy of the SVM model when trained utilizing the training data. The SVM model, after being hyperparameter tuned, creates classifications for the testing data which are evaluated to create an accuracy score. Since the CV process is performed iteratively, a total of six accuracy scores are produced.

The 2nd level of CV is utilized to find the ideal value for the regularization parameter α utilized by the SVM model after being training using the training data.



SOFTWARE PACKAGES LEVERAGED

1) Two-class Linear Support Vector Machine
   - Scikit-learn Library [8]: SVC method
   - Numpy Library [10]

2) Two-Level Cross Validation
   - Scikit-learn Library : StratifiedKFold and GridSearchCV methods
   - Scikit-learn Library: accuracy_score method
   - Numpy Library

EXPERIMENTAL

RESULTS

After the BCI measurement classification algorithm is developed in Python, to different signal data sets are utilized to evaluate and visualize algorithm performance:

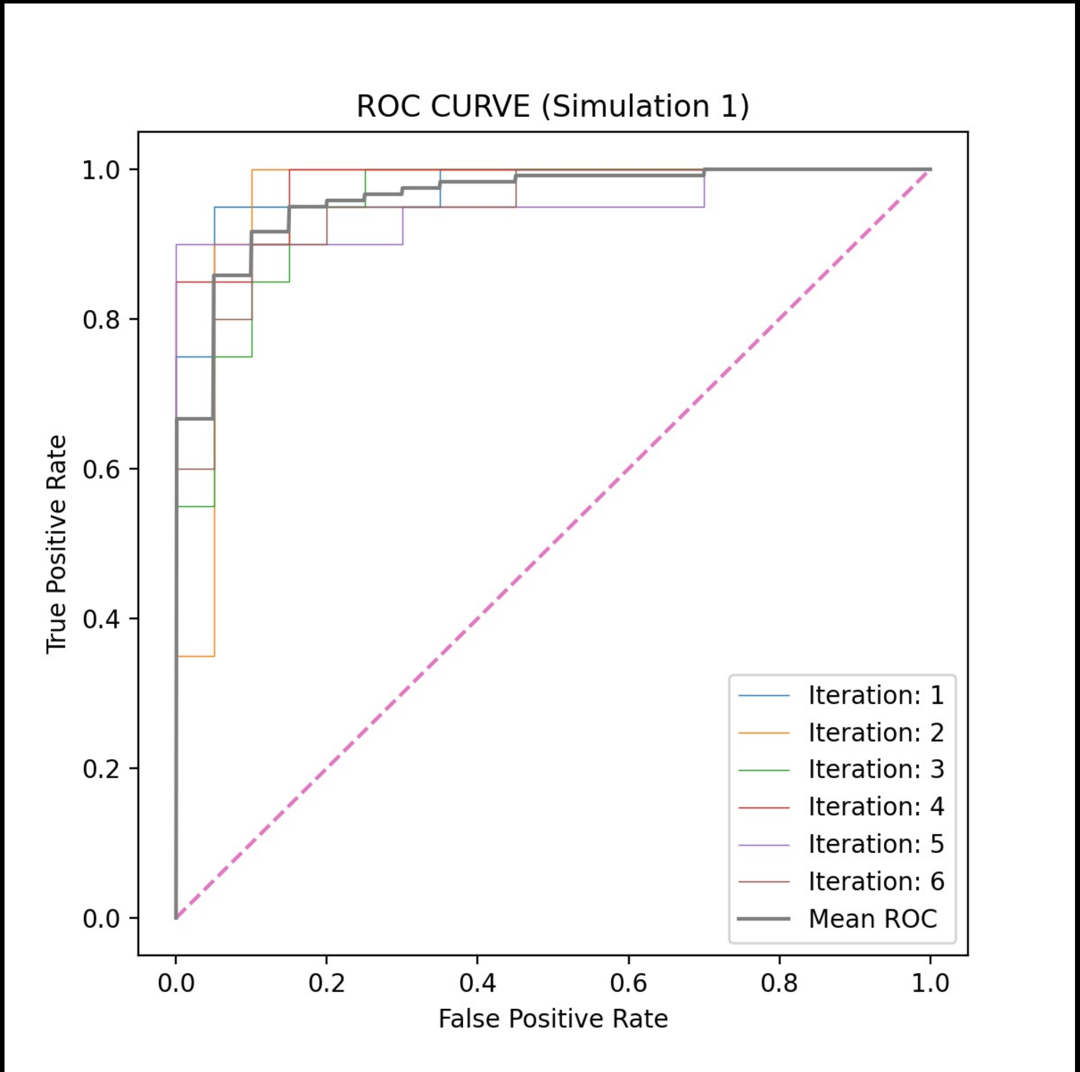
1) Imagined Movement| Pages 13-16
2) Overt Movement| Pages 17-20

The results produced by the two simulations are compared on pages 21 and 22.

The Python packages utilized to create the performance visualizations are outlined on page 23.

IMAGINED MOVEMENT

In this simulation, the BCI algorithm utilizes EEG signal data derived from imagined movement. Therefore, the CV process leveraged by the algorithm trains, hyperparameter tunes, and evaluates the SVM model iteratively six different times. The performance of SVM model after each CV iteration is visualized utilizing a receiver operating characteristic (ROC) curve which visualizes binary classification results utilizing two calculated parameters: true positive rate and false positive rate. True positive rate refers to the number of samples correctly classified by an SVM model while false positive rate refers to the number of samples incorrectly classified by the SVM model [7]. Additionally, a mean ROC curve is generated to generalize the cross-validated model performance [7]. It can be observed that the individual ROCs created after each iteration of the CV process are representative of the mean ROC with a certain level of deviation.



IMAGINED MOVEMENT

Values for the regularization parameter α examined by the 2^{nd} level of the CV process are equal to = [0.01, 1, 100, 10000].

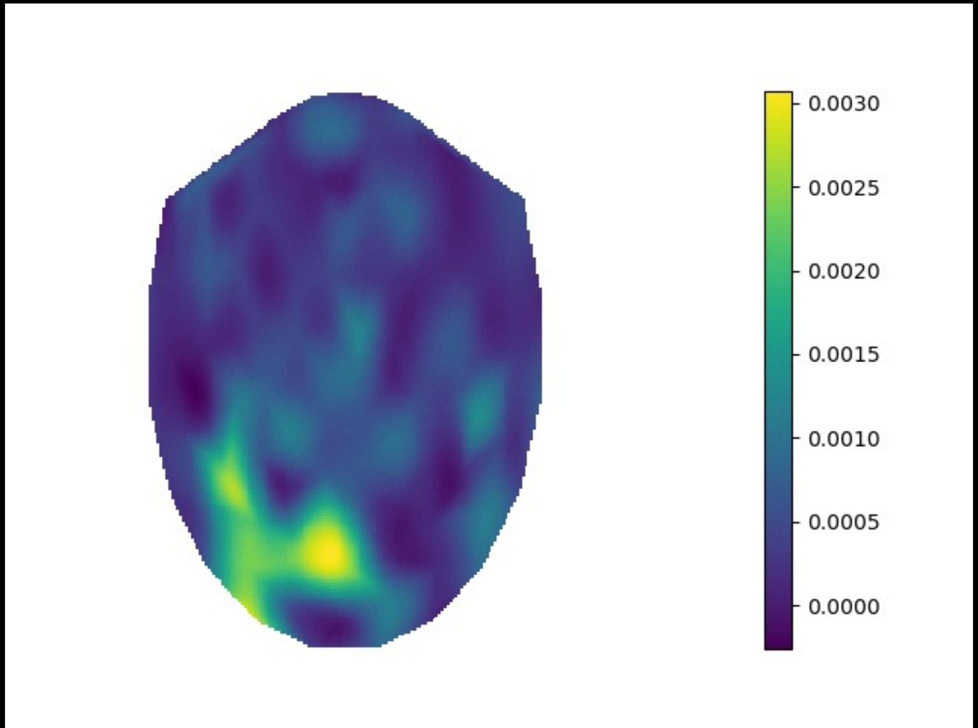
Accuracy evaluation and hyperparameter tuning results:

CV Iteration	Accuracy Score	α Value
1	0.925	0.01
2	0.925	0.01
3	0.875	0.01
4	0.9	0.01
5	0.95	0.01
6	0.875	0.01
-	Total Accuracy: 0.908	-

Similar to how the individual ROCs created during each CV iteration have a certain level of deviation they are still representative of the mean ROC, the individual accuracy scores produced by each CV iteration have slight variation but are still representative of the total accuracy.

IMAGINED MOVEMENT

The model weights of SVM model when trained, hyperparameter tuned, and evaluated by the first iteration of BCI algorithm's CV process are extracted to create a brain surface visualization. Since 204 electrodes (channels) are utilized to generate the EGG data, each feature vector has a total of 204 features (predictor variables). Similar to linear regression, the SVM model defines a linear hyperplane (v) that assigns weights to each predictor variable. In turn, these model weights are utilized as channel weights creating a spatial map visualizing the dominance of each electrode located on the scalp.

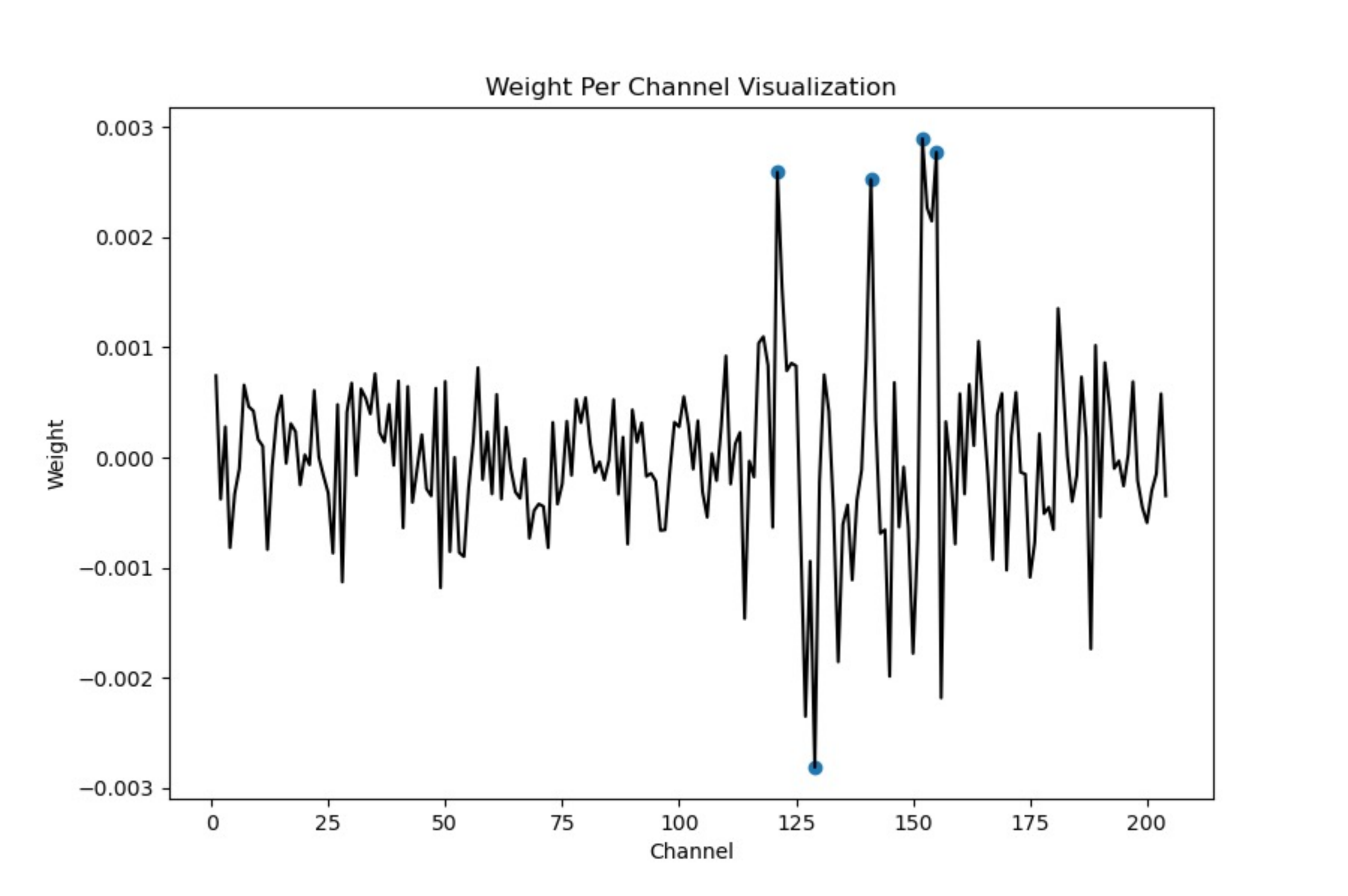


IMAGINED MOVEMENT

Five most dominant channels (electrodes)

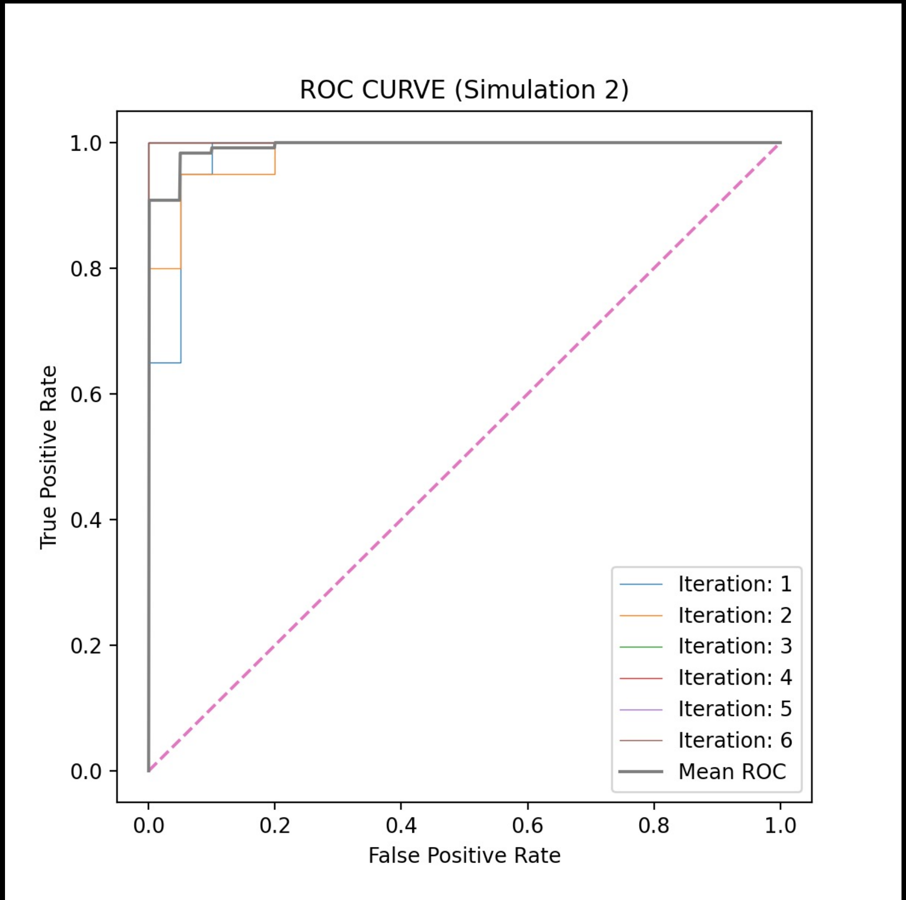
Dominant Channel	Weight
152	0.0028946
129	-0.00281046
155	0.00277214
121	0.00258997
141	0.00252323

Plot of all channel weights



OVERT MOVEMENT

In this simulation, the BCI algorithm utilizes EEG signal data derived from overt movement. Similar to the previous simulation, the CV process leveraged by the algorithm creates iteratively trains, hyperparameter tunes, and evaluates the SVM model six different times. The individual and combined model performance of these iterations are visualized utilizing ROC curves:



OVERT MOVEMENT

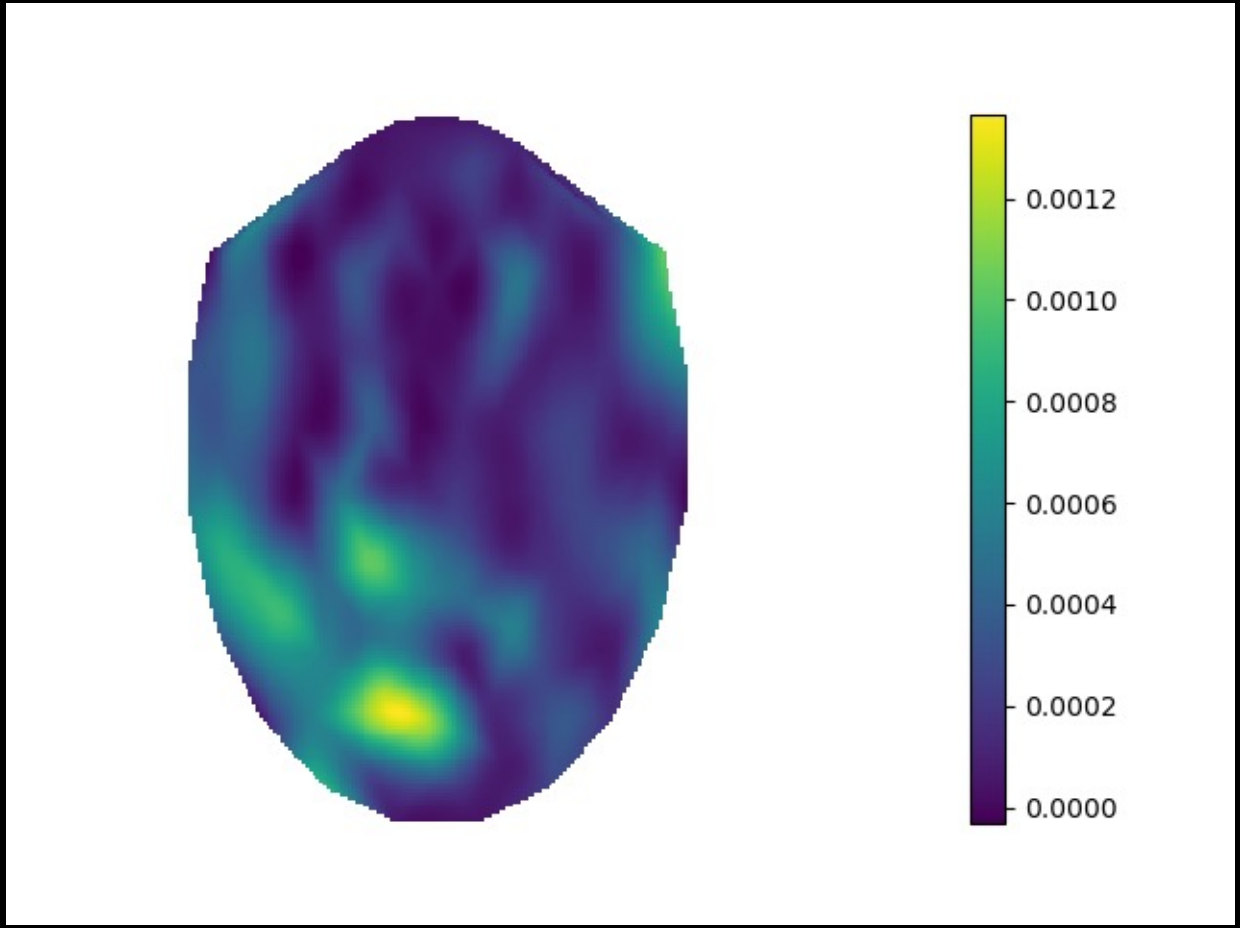
Values for the regularization parameter α examined by the 2nd level of the CV process are equal to = [0.01, 1, 100, 10000].

Accuracy evaluation and hyperparameter tuning results:

CV Iteration	Accuracy Score	α Value
1	0.95	0.01
2	0.925	0.01
3	0.975	0.01
4	0.975	0.01
5	1.0	0.01
6	0.925	0.01
-	Total Accuracy: 0.958	-

OVERT MOVEMENT

Spatial map visualizing the dominance of each electrode located on the scalp. Electrode weights are extracted from the SVM model when trained, hyperparameter tuned, and evaluated by the first iteration of BCI algorithm's CV process.

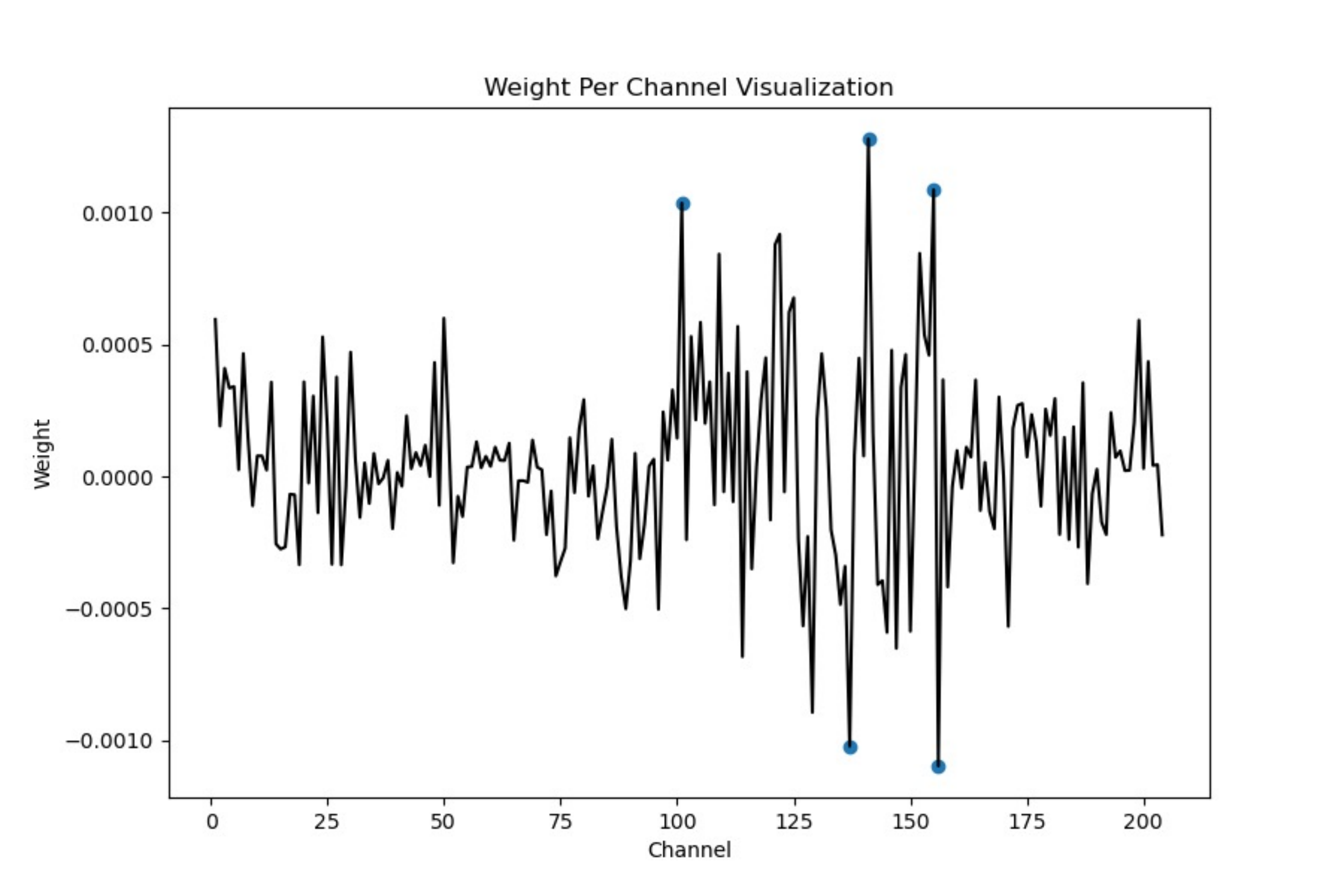


OVERT MOVEMENT

Five most dominant channels (electrodes)

Dominant Channel	Weight
141	0.00127829
156	-0.00109715
155	0.00108574
101	0.00103563
137	-0.00102183

Plot of all channel weights



IMAGINARY VS OVERT

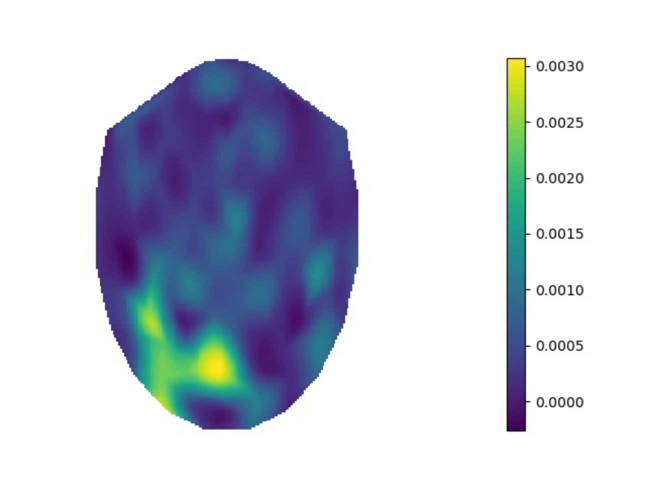
The value for optimal regularization parameter (α) produced by both simulations is equal to 0.01. 0.01 is the smallest value for the regularization parameter α examined by the 2nd level of the CV process. Therefore, the simulations will be performed again with a smaller value of 0.001 added within the 2nd level CV search hyperparameter search range. The results of this experiment will be discussed in detail in the conclusion section.

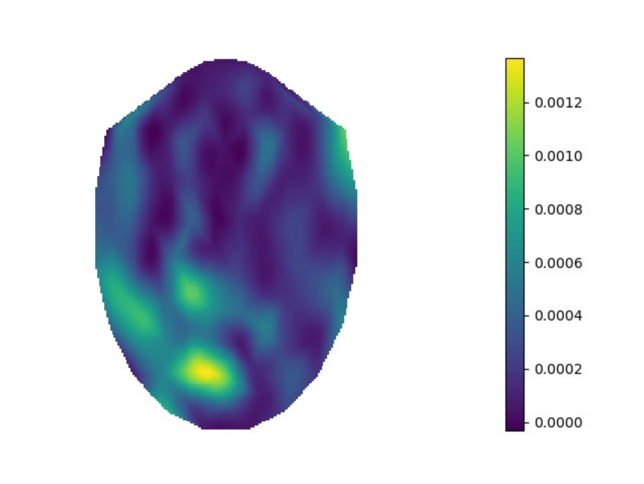
Imaginary Data Simulation Optimal α Value	Overt Data Simulation Optimal α Value
0.01	0.01

IMAGINARY VS OVERT

There is an overlap between the top five most dominant channels produced by each simulation: channels 141 and 155. Additionally, both simulations produce channels that are close in proximity and concentrated within a particular area on the human scalp: the bottom-left. This similarity indicates that this region of the brain is responsible for left and right motion.

Imaginary Data Simulation	**Overt Data Simulation**





Dominant Channels	**Dominant Channels**
152	141
129	156
155	155
121	101
141	137

SOFTWARE PACKAGES LEVERAGED

1) ROC Visualization
   - Scikit-learn Library [8] : roc_curve and auc methods
   - Matplotlib Library [9]
   - Numpy Library [10]

2) Brain Surface and Channel Weights visualizations
   - Matplotlib Library
   - Provided show_chanWeights.py script [11]

CONCLUSION

Comparing the experimental results of the imaginary data and overt data simulations, it can be observed that both simulations produce cross-validated accuracies above 90% (cross-validated accuracy is equal to the mean of accuracy scores produced by each SVM model in the CV process), however, the overt data simulation produces the higher accuracy score. This observation makes sense from a theoretical standpoint because the overt EEG signals utilized in the overt data simulation were generated from actual movement while the imaginary EEG signals utilized in the imaginary data simulation were generated from imagined movement. Therefore, the data utilized in the overt data simulation are stronger and lead to higher model performance in terms of accuracy. Additionally, the mean ROC curve produced by the overt data simulation has a larger area under it (AUC) indicating better model performance [7].

	Imaginary Data Simulation	Overt Data Simulation
cross-validated accuracy	0.908	0.958
AUC	0.961	0.993

CONCLUSION

A key insight uncovered by examining simulation results is the inverse relationship between the SVM model's regularization parameter α and SVM model performance/accuracy. Both the imaginary data and the overt data simulations yielded an optimal SVM regularization parameter α value of 0.01. Specifically, the SVM model utilized 0.01 each time after being hyperparameter tuned by the 2nd level of the CV (performed six different times) in both simulations. Since 0.01 was the smallest value examined by the 2nd level of the CV process, an additional value of 0.001 was added to the α search range to test if an even smaller hyperparameter would improve SVM performance in terms of accuracy. Correspondingly, the SVM model utilized 0.001 as the optimal regularization in both simulations which led to higher total accuracy. Hence, it can be concluded that as α trends to 0, the penalty for misclassification is reduced in turn maximizing the margin causing SVM performance to increase [5]. Accordingly, it can be concluded that larger values of α cause an SVM model to overfit creating a hyperplane that separates training data with very little misclassification (potentially no misclassification) but does not maximize the margin leading to inaccurate results when creating classifications on new testing data.

	Imaginary Data Simulation Total Accuracy	Overt Data Simulation Total Accuracy
$\alpha = 0.01$	0.908	0.958
$\alpha = .001$	0.912	0.971

CONCLUSION

Within both these simulations, utilizing a smaller SVM regularization parameter leads to more accurate model performance. This means that the EEG datasets utilized in both simulations are close to linear separability because even though a very small penalty is placed on misclassification, the SVM model's performance increases indicating there are not a large number of misclassifications. Hence when utilizing this algorithm to classify EEG datasets with more overlap, SVM model accuracy might decline. To address this shortcoming, it would be beneficial to implement a data exploration/processing precursor algorithm that reduces the chance of misclassification by performing actions such as removing extreme outliers. However, despite having a shortcoming, the BCI signal classification algorithm is capable of creating a hyperparameter tuned linear SVM model, leveraging non-linearly separable EEG data, which can classify EEG signals as referring to left or right movement with high accuracy.

REFERENCES

[1] Sadeghi, S. (2018, September 10). *Recent Advances in Hybrid Brain-Computer Interface Systems: A Technological and Quantitative Review*. Basic and Clinical Neuroscience. https://bcn.iums.ac.ir/browse.php?a_id=960&slc_lang=en&sid=1&ftxt=1&html=1.

[2] Brain-computer interface systems. (2020). Oregon Health & Science University. https://www.ohsu.edu/reknew/brain-computer-interface-systems

[3] P. Stegman, C. S. Crawford, M. Andujar, A. Nijholt and J. E. Gilbert, "Brain–Computer Interface Software: A Review and Discussion," in IEEE Transactions on Human-Machine Systems, vol. 50, no. 2, pp. 101-115, April 2020, doi: 10.1109/THMS.2020.2968411.

[4] Bishop Christopher M. 2006. Pattern Recognition and Machine Learning. Springer.

[5] H. Funaya and K. Ikeda, "A statistical analysis of soft-margin support vector machines for non-separable problems," The 2012 International Joint Conference on Neural Networks (IJCNN), 2012, pp. 1-7, doi: 10.1109/IJCNN.2012.6252443.

[6] M. J. Abdulaal, A. J. Casson and P. Gaydecki, "Performance of Nested vs. Non-Nested SVM Cross-Validation Methods in Visual BCI: Validation Study," 2018 26th European Signal Processing Conference (EUSIPCO), 2018, pp. 1680-1684, doi: 10.23919/EUSIPCO.2018.8553102.

REFERENCES

[7] E. Keedwell, "An analysis of the area under the ROC curve and its use as a metric for comparing clinical scorecards," 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2014, pp. 24-29, doi: 10.1109/BIBM.2014.6999263.

[8] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

[9] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," in Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, May-June 2007, doi: 10.1109/MCSE.2007.55.

[10] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

[11] Tantum, Stacy (2011) show_chanWeights.py [Source code]. https://sakai.duke.edu/access/content/group/80fd5f18-2013-403e-b9da-8aa5556047f9/MP2/show_chanWeights.py

[12] computer | History, Parts, Networking, Operating Systems, & Facts. (2022, April 7). Encyclopedia Britannica. https://www.britannica.com/technology/computer