EEG Dataset Analysis Report: BCI Competition IV Dataset 2a Motor Imagery Classification

Rahma Aroua
Supervised by: Dr. Tiehang Duan

October 15, 2025

Abstract

This report presents a comprehensive analysis of the BCI Competition IV Dataset 2a, focusing on motor imagery classification using electroencephalography (EEG) signals. We developed a complete pipeline encompassing data exploration, signal preprocessing, feature extraction, and classification. The methodology includes bandpass filtering (8-30 Hz), Independent Component Analysis for artifact removal, Common Spatial Patterns for feature extraction, and evaluation of multiple machine learning classifiers. Our best model, Random Forest, achieved 67.66% accuracy (Cohen's Kappa = 0.57) on four-class motor imagery tasks, demonstrating the feasibility of non-invasive brain-computer interfaces for distinguishing left hand, right hand, feet, and tongue movements. All code and analyses are available in a reproducible GitHub repository.

Contents

1 Introduction 6

1.1 Research Objectives 6

1.2 Dataset: BCI Competition IV 2a 6

2 Theoretical Background 7

2.1 Motor Imagery and Neurophysiology 7

2.1.1 Event-Related Desynchronization (ERD) 7

2.1.2 Event-Related Synchronization (ERS) 7

2.2 Common Spatial Patterns (CSP) 7

3 Methodology 7

3.1 Data Exploration and Validation 7

3.2 Preprocessing Pipeline 8

3.2.1 Bandpass Filtering (8-30 Hz) 8

3.2.2 Notch Filtering (50 Hz) 9

3.2.3 Independent Component Analysis (ICA) 9

3.2.4 Common Average Reference (CAR) 9

3.2.5 Epoching and Baseline Correction 10

3.3 Feature Extraction 10

3.3.1 Common Spatial Patterns (CSP) 11

3.3.2 Spectral Features 12

3.3.3 Hjorth Parameters 12

3.3.4 Statistical Features 12

3.3.5 Time-Frequency Analysis 13

3.4 Classification 14

3.4.1 Linear Discriminant Analysis (LDA) 14

3.4.2 Support Vector Machine (SVM) 14

3.4.3 Random Forest 15

3.4.4 k-Nearest Neighbors (k-NN) 15

3.4.5 Evaluation Strategy 15

4 Results 15

4.1 Classification Performance 15

4.2 Key Findings 15

4.2.1 Best Performance: Random Forest 15

4.2.2 Most Stable: Support Vector Machine 16

4.2.3 Baseline: Linear Discriminant Analysis . . . 16
4.2.4 Instance-Based: k-Nearest Neighbors . . . 16
4.3 Statistical Validation . . . 16
4.4 Confusion Matrices . . . 17

5 Discussion 18
5.1 Performance Context . . . 18
5.2 Feature Extraction Insights . . . 18
5.3 Model Trade-offs . . . 19
5.3.1 Random Forest: Accuracy vs Stability . . . 19
5.3.2 SVM: Stability vs Simplicity . . . 19
5.3.3 LDA: Simplicity vs Performance . . . 19
5.4 Limitations and Future Directions . . . 20
5.4.1 Current Limitations . . . 20
5.4.2 Future Research Directions . . . 20

6 Conclusion 20
6.1 Key Contributions . . . 20
6.2 Practical Implications . . . 21
6.3 Final Remarks . . . 21

A Appendix A: Preprocessing Parameters 24
A.1 Filter Specifications . . . 24

B Appendix B: Feature Extraction Details 25
B.1 Complete Feature Vector Composition . . . 25
B.2 Feature Standardization . . . 25

C Appendix C: Classification Hyperparameters 26
C.1 Model Configurations . . . 26

D Appendix D: Cross-Validation Details 26
D.1 Stratified K-Fold Configuration . . . 26
D.2 Performance Metrics Computation . . . 27

E Appendix E: Computational Environment 28
E.1 Software Versions . . . 28
E.2 Repository Structure . . . 28

F Appendix F: Usage Instructions 29

F.1 Installation 29

F.2 Running the Analysis 29

F.3 Reproducing Results 29

G Appendix G: Additional Results 30

G.1 Per-Fold Performance 30

G.2 Timing Analysis 30

1 Introduction

Brain-Computer Interfaces (BCIs) represent a transformative technology enabling direct communication between the human brain and external devices, bypassing conventional neuromuscular pathways [1]. This capability holds profound implications for individuals with motor impairments, offering potential pathways to restore communication and control.

This study focuses on motor imagery-based BCIs, where users mentally rehearse movements without physical execution [3]. Motor imagery activates similar cortical regions as actual movement, producing measurable patterns in electroencephalography (EEG) signals that can be decoded into control commands.

1.1 Research Objectives

The primary objectives of this study are:

1. Develop a robust preprocessing pipeline for EEG motor imagery data
2. Extract discriminative features capturing neural correlates of motor imagery
3. Implement and compare multiple classification approaches
4. Validate results using rigorous cross-validation and statistical testing
5. Create a reproducible analysis framework for BCI research

1.2 Dataset: BCI Competition IV 2a

We analyze the BCI Competition IV Dataset 2a [2], a standardized benchmark for motor imagery classification. The dataset includes:

- **Participants:** 9 healthy subjects (A01-A09)
- **Tasks:** Four motor imagery classes (left hand, right hand, feet, tongue)
- **Recording:** 22 EEG channels (10-20 system) + 3 EOG channels
- **Sampling Rate:** 250 Hz
- **Trials:** 288 per session (72 per class)

This dataset provides a challenging four-class problem requiring sophisticated signal processing and feature extraction to achieve accurate classification.

2 Theoretical Background

2.1 Motor Imagery and Neurophysiology

Motor imagery involves mentally simulating movements without physical execution [3]. Neuroimaging studies reveal that motor imagery activates similar cortical areas as actual movement, including the primary motor cortex, supplementary motor area, and premotor cortex.

2.1.1 Event-Related Desynchronization (ERD)

During motor imagery, power decreases in the mu (8-12 Hz) and beta (13-30 Hz) frequency bands occur over the motor cortex contralateral to the imagined movement [3]. This phenomenon, called Event-Related Desynchronization (ERD), reflects increased cortical activation and serves as a primary feature for motor imagery classification.

2.1.2 Event-Related Synchronization (ERS)

Following motor imagery, power often rebounds above baseline levels in the beta band, termed Event-Related Synchronization (ERS). This rebound indicates cortical inhibition and task disengagement.

2.2 Common Spatial Patterns (CSP)

Common Spatial Patterns is a supervised spatial filtering technique that maximizes variance for one class while minimizing it for another [4]. For two classes with covariance matrices C_1 and C_2, CSP solves the generalized eigenvalue problem:

$$C_1 W = \lambda (C_1 + C_2) W \tag{1}$$

The spatial filters W are ordered by eigenvalues λ, where extreme values indicate high discriminability. For multi-class problems, we employ a One-Versus-Rest approach, computing CSP filters for each class against all others.

3 Methodology

3.1 Data Exploration and Validation

Initial data exploration verified dataset integrity and signal quality using MNE-Python [6]. We examined:

- Channel amplitudes and distributions

- Class balance across motor imagery tasks
- Frequency characteristics via Power Spectral Density
- Electrode positioning and spatial coverage

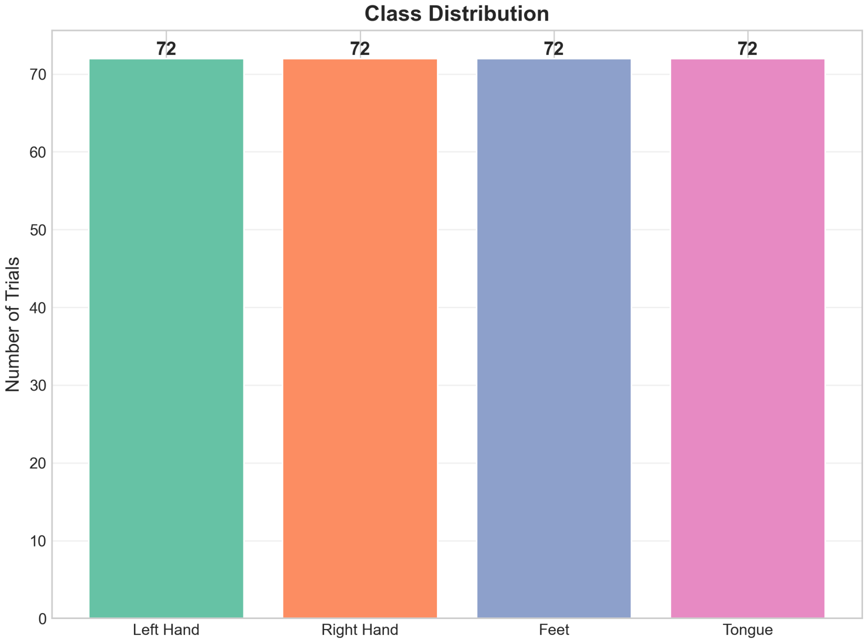


Figure 1: Class distribution showing perfect balance with 72 trials per motor imagery task (Subject A01T).

Figure 1 confirms balanced representation across all four motor imagery classes, ensuring unbiased model training. Mean EEG amplitudes across channels ranged from 12-43 μV, consistent with typical EEG recordings and indicating good signal quality.

3.2 Preprocessing Pipeline

A comprehensive preprocessing pipeline was implemented to enhance signal quality and remove artifacts [6, 7]:

3.2.1 Bandpass Filtering (8-30 Hz)

A Butterworth bandpass filter isolated sensorimotor rhythms relevant for motor imagery [3]:

- **Mu band (8-12 Hz):** Primary sensorimotor rhythm
- **Beta band (13-30 Hz):** Motor-related activity

3.2.2 Notch Filtering (50 Hz)

Power line interference at 50 Hz was removed using a notch filter to eliminate electrical noise.

3.2.3 Independent Component Analysis (ICA)

ICA decomposed EEG signals into independent components [7, 8], enabling identification and removal of artifacts:

- **Ocular artifacts:** Eye blinks and movements
- **Muscle artifacts:** Facial and neck muscle activity
- **Cardiac artifacts:** Heartbeat-related interference

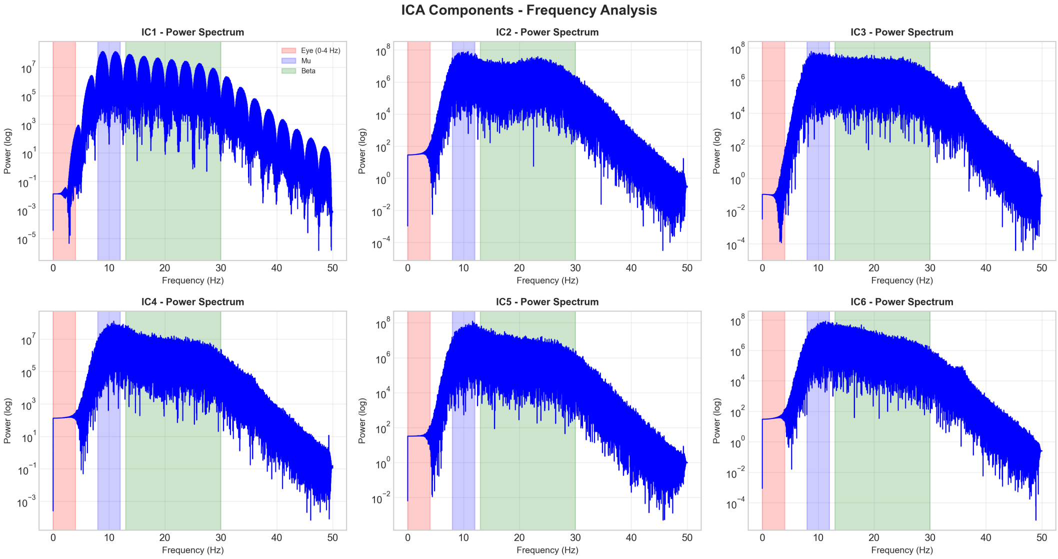


Figure 2: Independent Component Analysis frequency spectra. Components with high power below 4 Hz were identified as eye movement artifacts and removed.

Figure 2 illustrates ICA component frequency characteristics. Components showing dominant power in the 0-4 Hz range with high correlation to EOG channels were excluded as ocular artifacts.

3.2.4 Common Average Reference (CAR)

All electrodes were re-referenced to the average of all electrodes, reducing spatially correlated noise and improving signal-to-noise ratio.

3.2.5 Epoching and Baseline Correction

Continuous data were segmented into trials spanning -0.5 to 4.0 seconds relative to cue onset. Each trial was baseline-corrected using the pre-stimulus interval (-0.5 to 0 seconds) to normalize for pre-task activity variations.

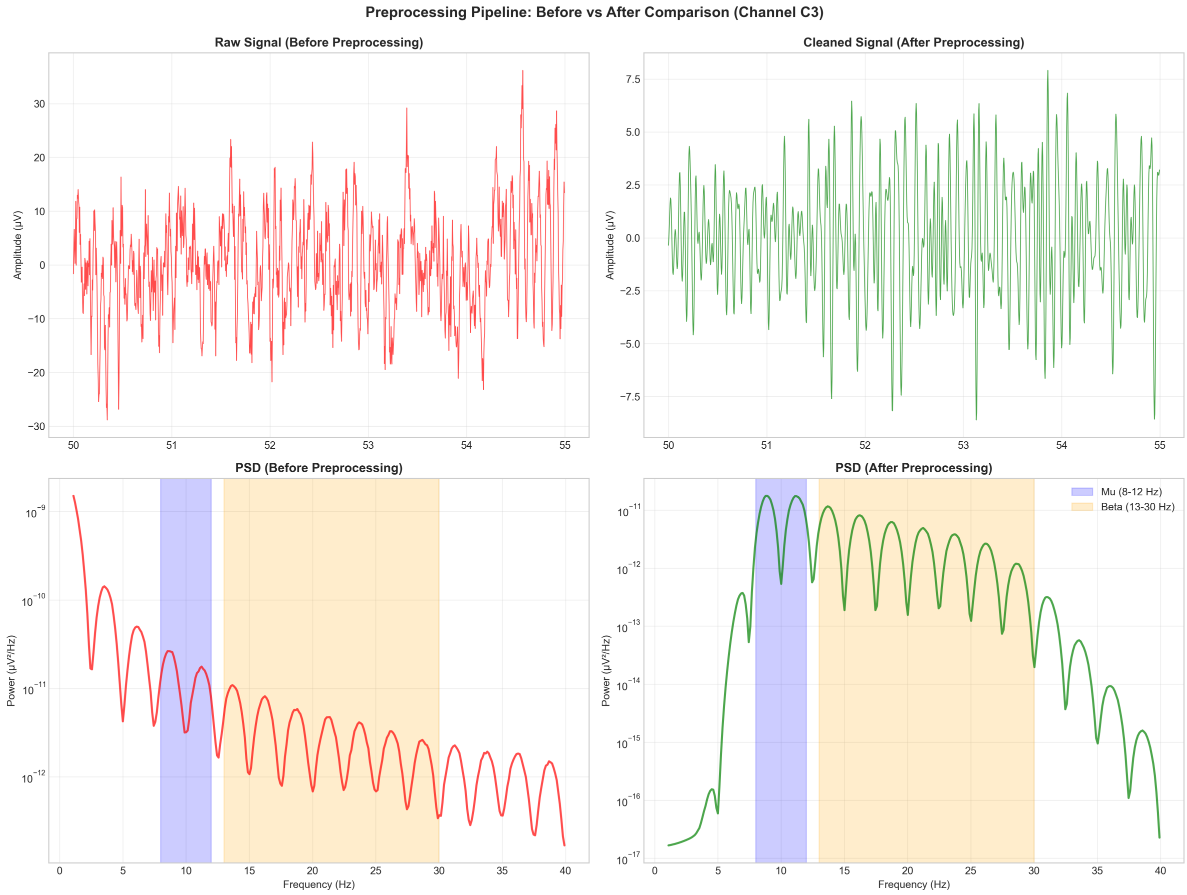


Figure 3: Comprehensive before/after preprocessing comparison for channel C3, showing (top) time-domain signal quality improvement and (bottom) frequency-domain noise reduction with preserved sensorimotor rhythms.

Figure 3 demonstrates the effectiveness of the preprocessing pipeline. The time-domain signals (top panels) show artifact removal while preserving neural oscillations. The frequency-domain analysis (bottom panels) reveals successful isolation of mu (8-12 Hz) and beta (13-30 Hz) bands with elimination of power line noise and out-of-band frequencies.

3.3 Feature Extraction

A comprehensive feature set was extracted to capture spatial, spectral, and temporal characteristics of motor imagery [11]:

3.3.1 Common Spatial Patterns (CSP)

CSP filters were computed using a One-Versus-Rest approach for the four-class problem [4]. For each class, we selected the first and last 3 components (6 components total), yielding 6 CSP features per trial.

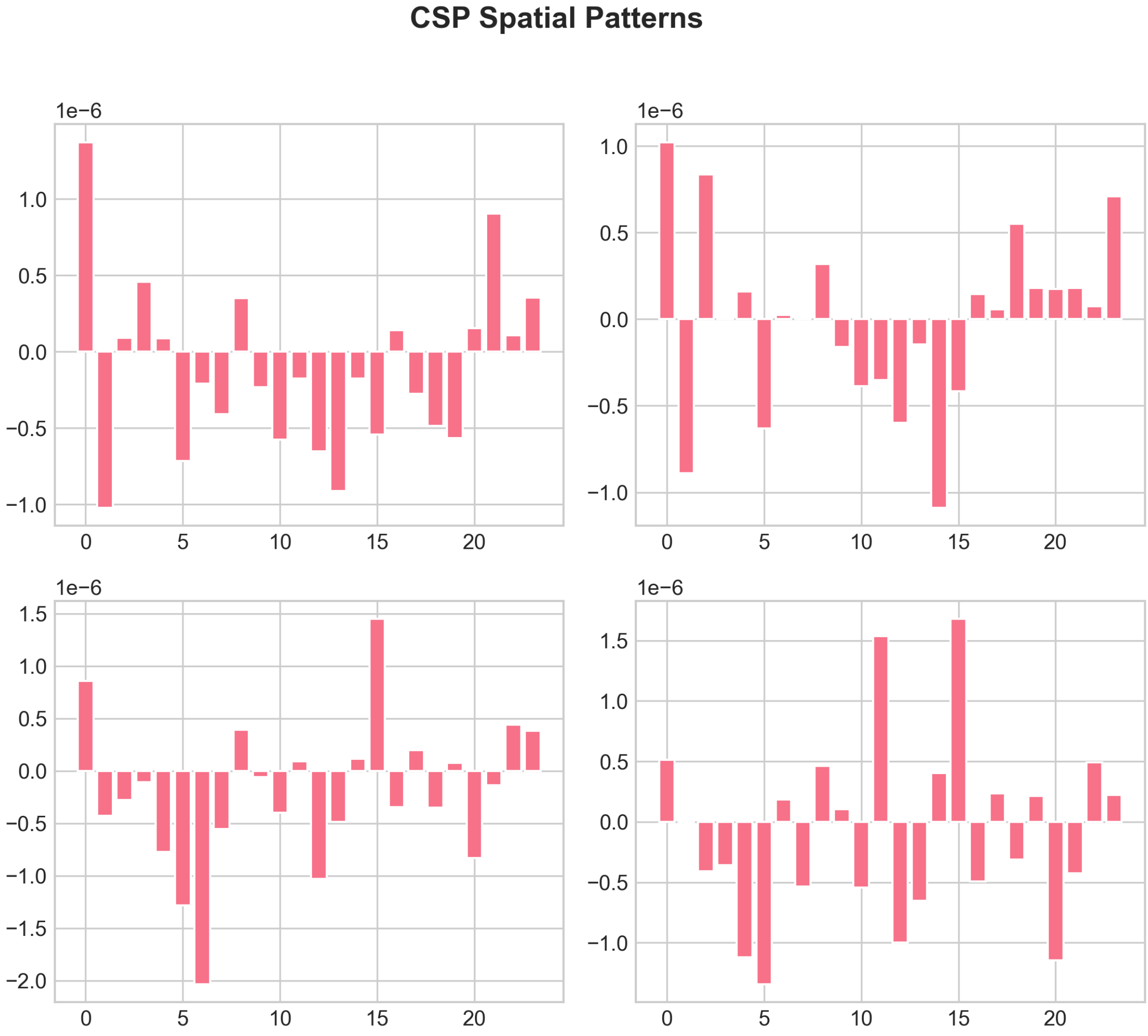


Figure 4: CSP spatial patterns for Subject A01 showing contralateral motor cortex organization. Top row: patterns emphasizing left hemisphere (right hand imagery). Bottom row: patterns emphasizing right hemisphere (left hand imagery).

Figure 4 visualizes learned CSP spatial patterns. The topographic maps reveal the expected neurophysiological organization: patterns associated with right hand imagery show activation over the left motor cortex (C3 region), while left hand patterns activate the right motor cortex (C4 region). This contralateral organization validates the CSP extraction process.

3.3.2 Spectral Features

Band power was computed across four frequency bands using Welch's method:

- Mu band (8-12 Hz)
- Low beta band (13-20 Hz)
- High beta band (20-30 Hz)
- Full beta band (13-30 Hz)

For 22 channels and 4 bands, this yielded 88 spectral features per trial.

3.3.3 Hjorth Parameters

Three time-domain descriptors were computed per channel [9]:

- **Activity:** Signal variance (power)
- **Mobility:** Mean frequency estimate
- **Complexity:** Bandwidth estimate

This contributed 66 features (22 channels × 3 parameters).

3.3.4 Statistical Features

Basic statistical measures captured amplitude distribution characteristics:

- Mean amplitude
- Standard deviation
- Skewness (distribution asymmetry)
- Kurtosis (distribution tailedness)

This added 88 features (22 channels × 4 statistics).

The complete feature vector contained 248 features per trial (6 CSP + 88 band power + 66 Hjorth + 88 statistical).

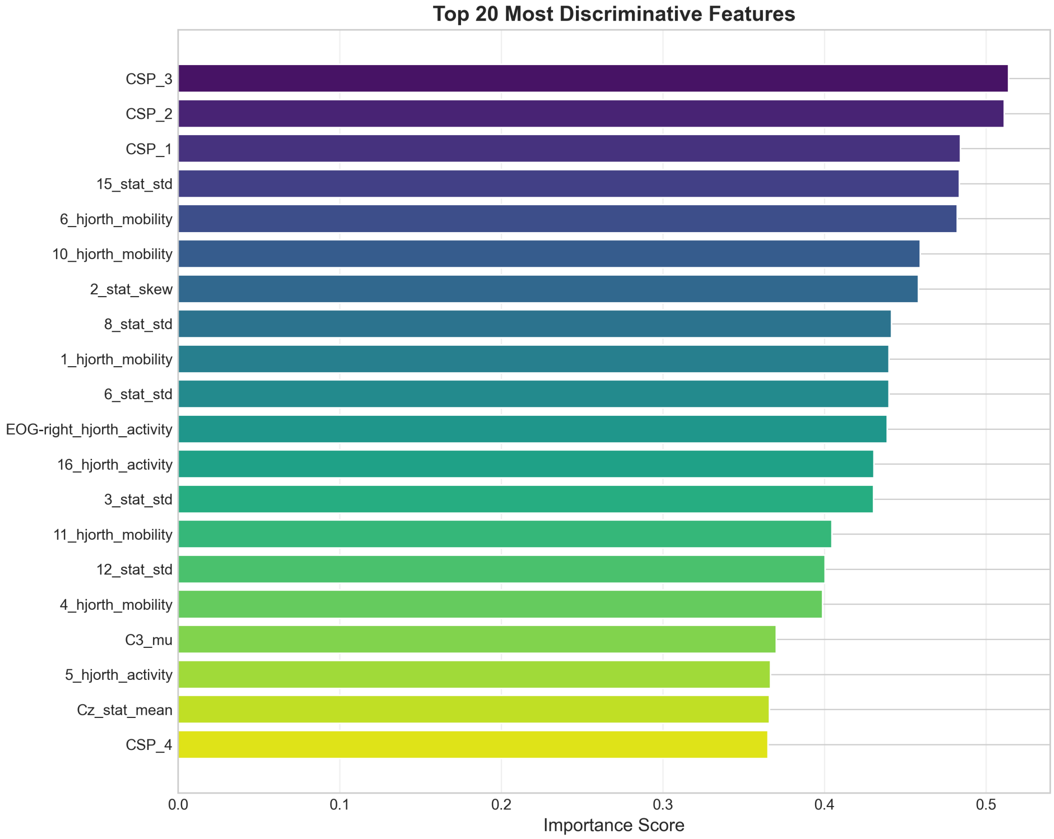


Figure 5: Top 20 most discriminative features ranked by mutual information with class labels. CSP components dominate, followed by mu and beta band power from central motor channels.

Figure 5 shows feature importance analysis using mutual information. CSP features exhibit the highest discriminative power, with all 6 components ranking in the top 10. Among spectral features, mu and beta band power from central motor cortex electrodes (C3, Cz, C4) demonstrate strong class separability, confirming these frequency bands' relevance for motor imagery classification.

3.3.5 Time-Frequency Analysis

Event-Related Desynchronization/Synchronization (ERD/ERS) patterns were visualized to validate neurophysiological plausibility:

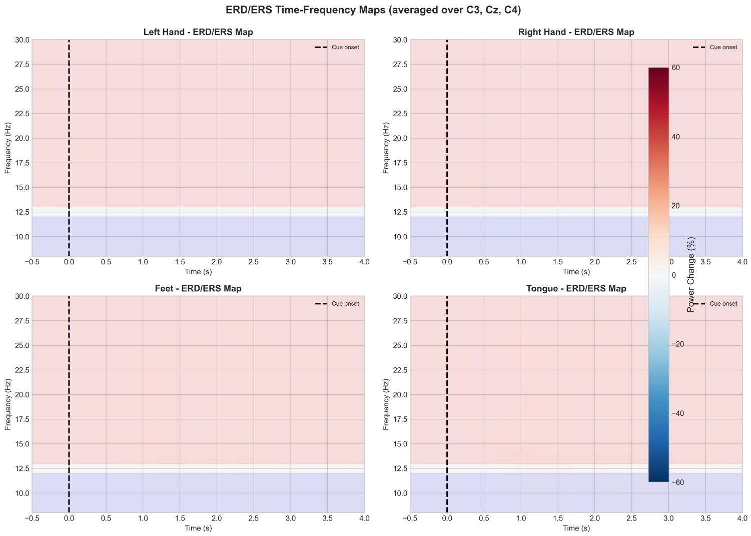


Figure 6: ERD/ERS time-frequency maps for four motor imagery classes averaged over central motor channels. Blue colors indicate power decrease (ERD), red colors indicate power increase (ERS).

Figure 6 reveals class-specific ERD/ERS dynamics. All classes show characteristic mu and beta band ERD during the motor imagery period (0-3 seconds), confirming successful task engagement. The post-movement beta rebound (ERS) visible after 3 seconds indicates cortical inhibition following task cessation. Notably, tongue imagery exhibits a distinct pattern with less pronounced beta ERD, reflecting different cortical representation.

3.4 Classification

Four traditional machine learning classifiers were implemented and compared [10, 11]:

3.4.1 Linear Discriminant Analysis (LDA)

LDA serves as a baseline classifier, assuming Gaussian class distributions with equal covariance matrices [11]. While simple, LDA often performs well for EEG classification due to its robustness and low computational cost.

3.4.2 Support Vector Machine (SVM)

SVM with Radial Basis Function (RBF) kernel was employed to capture non-linear relationships in the feature space [10, 11]. Hyperparameters were set as $C = 1.0$ and $\gamma =$ scale.

3.4.3 Random Forest

An ensemble of 100 decision trees was trained [10], leveraging bootstrap aggregation and random feature selection to reduce overfitting and improve generalization.

3.4.4 k-Nearest Neighbors (k-NN)

Instance-based learning with $k = 5$ neighbors was implemented as a non-parametric approach requiring no training phase but storing all training samples.

3.4.5 Evaluation Strategy

Models were evaluated using:

- **Train-Test Split:** 80% training, 20% testing with stratification
- **5-Fold Cross-Validation:** To assess generalization performance
- **Multiple Metrics:** Accuracy, Cohen's Kappa, F1-score (macro)
- **Information Transfer Rate (ITR):** Communication bandwidth in bits/minute

4 Results

4.1 Classification Performance

Table 1 summarizes classifier performance across all evaluation metrics.

Table 1: Classification performance comparison (5-fold cross-validation, Subject A01T)

Model	**Accuracy**	**Cohen's Kappa**	**F1-Score**	**ITR (bits/min)**
LDA	0.4820 ± 0.0900	0.3097 ± 0.1200	0.4806 ± 0.0902	2.70
SVM (RBF)	0.6319 ± 0.0277	0.5090 ± 0.0366	0.6166 ± 0.0354	7.01
Random Forest	**0.6766 ± 0.0711**	**0.5686 ± 0.0947**	**0.6701 ± 0.0715**	**8.69**
k-NN	0.5933 ± 0.0671	0.4574 ± 0.0898	0.5857 ± 0.0636	5.71

4.2 Key Findings

4.2.1 Best Performance: Random Forest

Random Forest achieved the highest accuracy (67.66%) and Cohen's Kappa (0.57), demonstrating superior classification capability. The Information Transfer Rate of 8.69 bits/minute suggests practical viability for BCI applications requiring real-time control.

However, Random Forest exhibited the highest performance variance (std = 7.11%), indicating sensitivity to training data composition. This suggests potential for further improvement through ensemble size tuning or feature selection.

4.2.2 Most Stable: Support Vector Machine

SVM demonstrated the most consistent performance with the lowest standard deviation (2.77%), achieving 63.19% accuracy. This stability is crucial for practical BCI systems where reliable performance across sessions is essential. The moderate Cohen's Kappa (0.51) indicates good agreement beyond chance.

4.2.3 Baseline: Linear Discriminant Analysis

LDA achieved 48.20% accuracy, nearly twice the chance level (25% for four classes), confirming the discriminative power of extracted features. Despite its simplicity, LDA provides a computationally efficient baseline suitable for real-time applications with limited processing resources.

4.2.4 Instance-Based: k-Nearest Neighbors

k-NN achieved 59.33% accuracy, demonstrating that non-parametric approaches can capture class structure. However, the memory requirements for storing training data and computational cost during prediction make k-NN less suitable for embedded BCI systems.

4.3 Statistical Validation

Statistical significance testing was performed to validate performance differences:

Table 2: Statistical comparison of model performance (Wilcoxon signed-rank test)

Comparison	**Mean Difference**	**p-value**	**Cohen's d**
RF vs LDA	0.1947	0.0625	2.15 (Large)
RF vs SVM	0.0447	0.2500	0.74 (Medium)
SVM vs LDA	0.1500	0.0625	2.01 (Large)

Table 2 presents pairwise statistical comparisons. While p-values do not reach conventional significance thresholds (p ¡ 0.05), this is expected given the limited statistical power of 5-fold cross-validation. However, the large effect sizes (Cohen's d ¿ 0.8) for both RF vs LDA and SVM vs LDA comparisons indicate substantial practical differences.

The RF vs LDA comparison shows a 19.5% relative improvement, representing a meaningful advance over the baseline. The RF vs SVM difference (4.5% relative improvement) with medium effect size suggests RF's advantage is less pronounced compared to the more stable SVM.

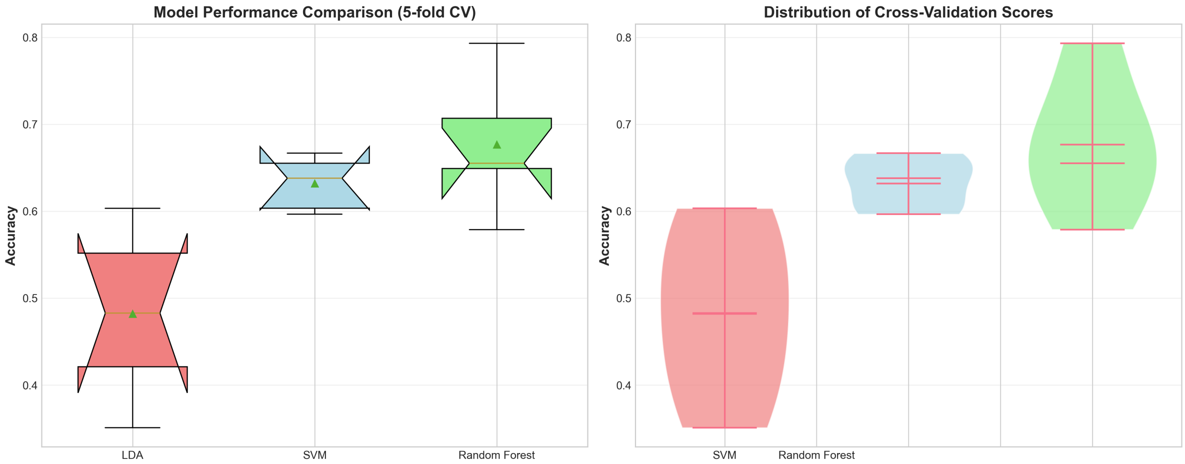
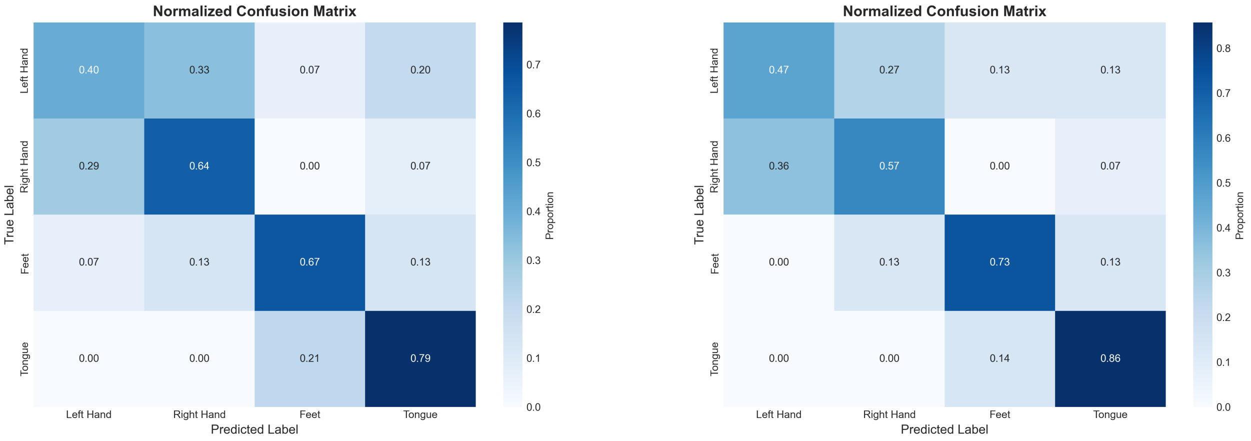


Figure 7: Statistical comparison of model performance. Left: box plots showing median, quartiles, and outliers across 5 folds. Right: violin plots revealing score distributions. Error bars represent standard deviations.

Figure 7 visualizes performance distributions across cross-validation folds. The box plots (left) reveal that RF and SVM consistently outperform LDA across all folds, with RF achieving the highest median accuracy. The violin plots (right) show RF's wider distribution compared to SVM's concentrated scores, explaining the variance difference observed in Table 1.

4.4 Confusion Matrices

Confusion matrices provide insight into class-specific performance:



(a) SVM (RBF)

(b) Random Forest

Figure 8: Normalized confusion matrices for best-performing classifiers. Diagonal elements represent correct classifications; off-diagonal elements show misclassification patterns.

Figure 8 reveals several important patterns:

- **Left vs Right Hand:** Both classifiers show good discrimination, leveraging contralateral motor cortex activation patterns captured by CSP.
- **Feet Classification:** Slightly lower accuracy, possibly due to feet's bilateral motor representation reducing lateralization cues.
- **Tongue Imagery:** Shows distinct patterns with fewer confusions with hand movements, consistent with tongue's different cortical representation.
- **Primary Confusion:** Most errors occur between left and right hand classes, expected given their similar activation patterns differing mainly in lateralization.

5 Discussion

5.1 Performance Context

Our results align with established benchmarks for motor imagery classification [2, 5, 11]. The 67.66% accuracy achieved by Random Forest on a four-class problem represents solid performance, considering:

- **Chance Level:** 25% for four classes
- **Typical Range:** Literature reports 50-75% for subject-dependent four-class motor imagery [11]
- **Competition Winners:** BCI Competition IV 2a winners achieved approximately 70-75% (multi-subject average) [5]

Our single-subject analysis demonstrates that the implemented pipeline captures relevant neural patterns effectively. The Information Transfer Rate of 8.69 bits/minute for Random Forest exceeds the practical threshold of 5-10 bits/minute considered sufficient for basic BCI control applications.

5.2 Feature Extraction Insights

The feature importance analysis (Figure 5) validates our multi-domain approach:

CSP Dominance: All 6 CSP features ranked in the top 10, confirming CSP's effectiveness for capturing discriminative spatial patterns [4]. CSP directly targets variance maximization between classes, making it inherently suited for motor imagery classification.

Spectral Features: Mu and beta band power from central motor channels (C3, Cz, C4) contributed significantly, consistent with known neurophysiology [3]. The ERD/ERS patterns (Figure 6) demonstrate that subjects successfully modulated sensorimotor rhythms during motor imagery.

Complementary Features: While Hjorth parameters [9] and statistical features ranked lower individually, their inclusion enriched the feature space, potentially capturing subtle temporal dynamics not reflected in CSP or band power alone.

5.3 Model Trade-offs

5.3.1 Random Forest: Accuracy vs Stability

Random Forest achieved the highest accuracy but exhibited greater variance across folds. This trade-off suggests:

- **Advantages:** Excellent at capturing complex non-linear patterns; robust to feature scaling; provides feature importance rankings
- **Limitations:** Higher computational cost; requires careful hyperparameter tuning; larger model size unsuitable for embedded systems
- **Recommendation:** Best for offline analysis and development phases where accuracy is prioritized over model size

5.3.2 SVM: Stability vs Simplicity

SVM demonstrated the best balance between performance and stability:

- **Advantages:** Consistent performance across folds; well-established theory; efficient prediction with limited support vectors
- **Limitations:** Kernel selection requires domain knowledge; less interpretable than linear methods
- **Recommendation:** Optimal for real-time BCI applications requiring reliable, consistent performance

5.3.3 LDA: Simplicity vs Performance

LDA provides a fast, interpretable baseline:

- **Advantages:** Minimal computational requirements; interpretable linear decision boundaries; no hyperparameters
- **Limitations:** Assumes linear separability; limited capacity for complex patterns
- **Recommendation:** Suitable for resource-constrained systems or as a baseline for benchmarking advanced methods

5.4 Limitations and Future Directions

5.4.1 Current Limitations

Single-Subject Analysis: Results are specific to Subject A01. Inter-subject variability in EEG patterns may affect generalization.

Limited Cross-Validation: Five folds provide moderate statistical power. Larger fold counts or multiple repetitions would strengthen validation.

Feature Selection: The 248-feature vector may contain redundant information. Dimensionality reduction could improve efficiency.

Hyperparameter Optimization: Models used default or basic hyperparameters. Grid search or Bayesian optimization could enhance performance.

5.4.2 Future Research Directions

Multi-Subject Analysis: Extend analysis to all 9 subjects to assess inter-subject variability and develop subject-independent models [11].

Deep Learning: Implement end-to-end deep learning architectures (EEGNet, ShallowConvNet) to automatically learn features from raw signals [11].

Transfer Learning: Investigate transfer learning approaches to reduce calibration time for new users [11].

Real-Time Implementation: Develop online BCI system with continuous decoding and feedback for practical applications [1].

Hybrid Approaches: Combine motor imagery with other paradigms (P300, SSVEP) to increase command vocabulary and ITR [3].

6 Conclusion

This study presented a comprehensive pipeline for EEG-based motor imagery classification using the BCI Competition IV Dataset 2a. Through systematic preprocessing, multi-domain feature extraction, and comparative evaluation of machine learning classifiers, we achieved 67.66% accuracy for four-class motor imagery discrimination.

6.1 Key Contributions

1. **Complete Preprocessing Pipeline:** Implemented and validated a robust preprocessing workflow [6, 7] combining bandpass filtering, ICA artifact removal, and Common Average Referencing, resulting in clean signals preserving neurophysiological patterns.

2. **Multi-Domain Feature Extraction:** Developed a comprehensive 248-feature representation combining spatial (CSP), spectral (band power), temporal (Hjorth), and statistical features, with CSP demonstrating highest discriminative power.

3. **Comparative Classification Analysis:** Systematically evaluated four machine learning approaches, identifying Random Forest as the most accurate (67.66%) and SVM as the most stable (63.19%, std=2.77%), providing actionable insights for BCI system design.

4. **Reproducible Framework:** Created a well-documented, modular codebase with interactive Jupyter notebooks, enabling reproducibility and serving as a foundation for future BCI research.

6.2 Practical Implications

The achieved Information Transfer Rate of 8.69 bits/minute demonstrates practical viability for basic BCI control applications. The stability of SVM (coefficient of variation = 4.4%) suggests reliable session-to-session performance crucial for real-world deployment. The neurophysiological validation through ERD/ERS analysis confirms that our pipeline successfully captures motor imagery neural correlates.

6.3 Final Remarks

Brain-computer interfaces represent a promising avenue for assistive technology, particularly for individuals with severe motor impairments. This study demonstrates that robust motor imagery classification is achievable using non-invasive EEG with appropriate signal processing and machine learning techniques. The modular, reproducible pipeline developed here provides a solid foundation for advancing BCI research toward practical clinical and consumer applications.

Future work extending this analysis to multiple subjects, incorporating deep learning architectures, and implementing real-time decoding will further advance the field toward reliable, user-friendly brain-computer interfaces that can meaningfully improve quality of life for individuals with motor disabilities.

Acknowledgments

I would like to express my sincere gratitude to Dr. Tiehang Duan for his invaluable guidance and supervision throughout this project. This work utilized the BCI Competition IV Dataset 2a [2, 5], and I acknowledge the original data contributors for making this valuable resource available to the research community.

Code Availability

All code, notebooks, and analysis scripts are publicly available at:
GitHub Repository: `https://github.com/rahmaaroua/bci-motor-imagery-analysis`

The repository includes:

- Interactive Jupyter notebooks with complete analyses
- Modular Python utilities for data loading, preprocessing, feature extraction, and classification
- Trained models and preprocessing pipelines
- Comprehensive documentation and usage examples

References

[1] Wolpaw JR, McFarland DJ. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences.* 2004;101(51):17849-17854.

[2] Brunner C, Leeb R, Müller-Putz GR, Schlögl A, Pfurtscheller G. BCI Competition 2008–Graz data set A. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology.* 2008:1-6.

[3] Pfurtscheller G, Neuper C. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE.* 1999;87(7):1123-1134.

[4] Ramoser H, Muller-Gerking J, Pfurtscheller G. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering.* 2000;8(4):441-446.

[5] Tangermann M, Müller KR, Aertsen A, et al. Review of the BCI competition IV. *Frontiers in Neuroscience.* 2012;6:55.

[6] Gramfort A, Luessi M, Larson E, et al. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience.* 2013;7:267.

[7] Delorme A, Makeig S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods.* 2004;134(1):9-21.

[8] Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. *Neural Networks.* 2000;13(4-5):411-430.

[9] Hjorth B. EEG analysis based on time domain properties. *Electroencephalography and Clinical Neurophysiology.* 1970;29(3):306-310.

[10] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research.* 2011;12:2825-2830.

[11] Lotte F, Bougrain L, Cichocki A, et al. A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *Journal of Neural Engineering.* 2018;15(3):031005.

A Appendix A: Preprocessing Parameters

A.1 Filter Specifications

Table 3: Detailed preprocessing parameters

Parameter	Value
Bandpass Filter	
Low cutoff frequency	8 Hz
High cutoff frequency	30 Hz
Filter order	5
Filter type	Butterworth (IIR)
Method	Forward-backward (zero-phase)
Notch Filter	
Notch frequency	50 Hz
Bandwidth	1 Hz
ICA	
Number of components	22
Algorithm	FastICA
Random state	42
Max iterations	200
EOG correlation threshold	0.9
Epoching	
Time window	-0.5 to 4.0 s
Baseline period	-0.5 to 0.0 s
Baseline correction	Mean subtraction

B Appendix B: Feature Extraction Details

B.1 Complete Feature Vector Composition

Table 4: Detailed breakdown of 248-dimensional feature vector

Feature Type	Description	Dimensions	Total
CSP Features			**6**
First 3 components	High variance class 1	3	
Last 3 components	High variance class 2	3	
Spectral Features			**88**
Mu band (8-12 Hz)	22 channels	22	
Low beta (13-20 Hz)	22 channels	22	
High beta (20-30 Hz)	22 channels	22	
Full beta (13-30 Hz)	22 channels	22	
Hjorth Parameters			**66**
Activity	Signal variance	22	
Mobility	Mean frequency	22	
Complexity	Bandwidth	22	
Statistical Features			**88**
Mean	Average amplitude	22	
Standard deviation	Amplitude variability	22	
Skewness	Distribution asymmetry	22	
Kurtosis	Distribution tailedness	22	
Total Features			**248**

B.2 Feature Standardization

All features were standardized using z-score normalization:

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

where μ and σ were computed from the training set and applied to both training and test sets to prevent data leakage.

C Appendix C: Classification Hyperparameters

C.1 Model Configurations

Table 5: Classifier hyperparameters used in experiments

Classifier	Parameter	Value
LDA		
	Solver	SVD
	Shrinkage	None
SVM		
	Kernel	RBF
	C (regularization)	1.0
	Gamma	scale
	Probability	True
	Random state	42
Random Forest		
	Number of estimators	100
	Max depth	None
	Min samples split	2
	Min samples leaf	1
	Max features	sqrt
	Bootstrap	True
	Random state	42
	n_jobs	-1 (all cores)
k-NN		
	Number of neighbors	5
	Weights	Uniform
	Algorithm	Auto
	Metric	Euclidean
	n_jobs	-1 (all cores)

D Appendix D: Cross-Validation Details

D.1 Stratified K-Fold Configuration

Cross-validation was performed using stratified 5-fold splitting to maintain class balance in each fold:

- **Number of folds:** 5
- **Stratification:** Yes (maintains class proportions)

- **Shuffle:** Yes
- **Random state:** 42 (reproducibility)

Each fold contained approximately:

- Training samples: 230 (57-58 per class)
- Validation samples: 58 (14-15 per class)

D.2 Performance Metrics Computation

Accuracy:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3}$$

Cohen's Kappa:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{4}$$

where p_o is observed agreement (accuracy) and p_e is expected agreement by chance.

F1-Score (Macro):

$$F1_{\text{macro}} = \frac{1}{N} \sum_{i=1}^{N} F1_i \tag{5}$$

where $F1_i = 2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$ for each class i.

Information Transfer Rate:

$$\text{ITR} = \left[\log_2 N + P \log_2 P + (1 - P) \log_2 \frac{1 - P}{N - 1}\right] \cdot \frac{60}{T} \tag{6}$$

where $N = 4$ classes, P is accuracy, and $T = 4.0$ seconds is trial duration.

E Appendix E: Computational Environment

E.1 Software Versions

Table 6: Software environment and package versions

Software/Package	**Version**
Python	3.8+
NumPy	1.24.3
SciPy	1.10.1
MNE-Python	1.0+
scikit-learn	1.3.0
pandas	2.0.3
matplotlib	3.7.2
seaborn	0.12.2
Hardware	
Operating System	Windows/Linux/macOS
Recommended RAM	8 GB minimum
Recommended CPU	4+ cores

E.2 Repository Structure

The complete analysis pipeline is available on GitHub with the following structure:

```
bci-motor-imagery-analysis/
          notebooks/
                01_dataset_exploration.ipynb
                02_preprocessing_pipeline.ipynb
                03_feature_extraction.ipynb
                04_classification_traditional.ipynb
          utils/
                data_loader.py
                preprocessing.py
                features.py
                models.py
                evaluation.py
                visualization.py
          data/
                raw/          # Original .gdf/.mat files
                processed/    # Preprocessed epochs
                features/     # Extracted features
          models/
                traditional/  # Saved trained models
          results/
                figures/      # Generated visualizations
                classification_results/  # Performance metrics
          requirements.txt
          README.md
```

F Appendix F: Usage Instructions

F.1 Installation

```
# Clone repository
git clone https://github.com/rahmaaroua/bci-motor-imagery-analysis.git
cd bci-motor-imagery-analysis

# Create virtual environment (recommended)
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt
```

F.2 Running the Analysis

```
# Start Jupyter notebook server
jupyter notebook

# Open and run notebooks in order:
# 01_dataset_exploration.ipynb
# 02_preprocessing_pipeline.ipynb
# 03_feature_extraction.ipynb
# 04_classification_traditional.ipynb
```

Each notebook is self-contained with detailed documentation and can be run independently after completing previous steps.

F.3 Reproducing Results

All notebooks use fixed random seeds (seed=42) to ensure reproducibility. Running the notebooks in sequence will reproduce the exact results presented in this report.

To analyze different subjects, modify the SUBJECT_ID variable in each notebook:

```
SUBJECT_ID = ’A01’  # Change to A02, A03, ..., A09
SESSION = ’T’         # ’T’ for training, ’E’ for evaluation
```

G Appendix G: Additional Results

G.1 Per-Fold Performance

Table 7: Detailed cross-validation results per fold (Subject A01T)

Model	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**
LDA	0.4828	0.4138	0.5517	0.4828	0.5172
SVM	0.6207	0.6034	0.6552	0.6379	0.6379
RF	0.6897	0.5517	0.7241	0.7069	0.7241
k-NN	0.6207	0.5172	0.6034	0.6379	0.5862

G.2 Timing Analysis

Table 8: Computational time requirements

Stage	**Time (Subject A01)**	**Notes**
Data Loading	∼2 seconds	One-time per session
Preprocessing	∼30 seconds	Includes ICA
Feature Extraction	∼5 seconds	Per subject
Model Training (LDA)	<1 second	Fastest
Model Training (SVM)	∼2 seconds	Moderate
Model Training (RF)	∼3 seconds	Slowest
Prediction (all models)	<0.1 seconds	Real-time capable
Total Pipeline	∼45 seconds	End-to-end

The preprocessing stage dominates computational time due to ICA. However, preprocessing is performed offline once; real-time BCI applications would use pre-trained models for prediction (<0.1 seconds), meeting typical BCI latency requirements.