EEG-Based Brain-Computer Interface Development

Assistant Engineer Internship Report

Apple Catcher Game: EEG-controlled virtual hand interface using motor imagery classification.

Kilian BARANTAL
Robotic Engineering
FISE 2026
kilian.barantal@ensta-bretagne.org

September 2025

Abstract

This report presents the development of a brain-computer interface (BCI) research platform during an 18-week internship at the BrainKybLab of the Norwegian University of Science and Technology (NTNU) in Trondheim. In collaboration with Robin Vidal, we transformed an EEG-controlled game prototype into a modular and robust system for research and rehabilitation applications.

The initial project, "Apple Catcher," allowed users to control virtual hands through left-right hand movement imagery detected via EEG. Our main contribution consisted of a complete refactoring toward a client-server architecture, separating EEG signal processing from the graphical interface. This modularization facilitates experimentation and enables integration of new games without modifying the classification pipeline.

We also developed a temporal window optimization algorithm that personalizes feature extraction according to individual neural dynamics. Testing on a subset of the public GIGA dataset [2,3] (5 subjects) demonstrates an average improvement of 19.7% in classification accuracy. A centralized management system allows clinicians to configure experimental parameters without programming knowledge.

These developments transform a prototype into a flexible research tool, paving the way for personalized therapeutic applications in brain-computer interfaces.

Contents

1 Introduction

This report presents the work conducted during an 18-week assistant engineer internship at the BrainKybLab of the Norwegian University of Science and Technology (NTNU) in Trondheim, from May 5th to September 5th, 2025. This internship, carried out in collaboration with Robin Vidal, another ENSTA Bretagne student, focuses on the emerging field of brain-computer interfaces (BCI) applied to neurological rehabilitation.

The initial project, developed by Erlend Skredsvig in 2024 [1], consisted of an "Apple Catcher" game prototype controlled by electroencephalography (EEG). This novel motor imagery BCI was designed to facilitate hand rehabilitation for individuals with motor impairments by providing an engaging and user-centered approach. The game leverages the fact that motor imagery and motor execution share common neural networks in the brain, allowing users to practice and strengthen motor abilities through mental rehearsal.

Unlike traditional motor imagery-based BCIs that process EEG data in sensor space, the original Apple Catcher game employed EEG Source Imaging (ESI) using sLORETA (standardized low resolution brain electromagnetic tomography) to improve classification performance. During gameplay, features were extracted from EEG data using average power of source estimates, which were then classified using Linear Discriminant Analysis (LDA). Skredsvig's preliminary results showed promising accuracy rates of 75.5% during live gameplay and up to 85% in offline testing, with an 11.5% improvement between first and second gameplay sessions.

However, this prototype presented major limitations for research and clinical applications: monolithic architecture preventing easy experimentation with different games, fixed processing parameters unsuitable for individual neural variations, and complex technical requirements limiting accessibility to clinicians.

Our mission was twofold: first, to transform this prototype into a robust and modular research platform; second, to improve EEG signal classification accuracy through algorithmic personalization approaches. These objectives align with a broader vision of developing therapeutic tools adapted to patients with motor deficits.

The project addresses critical challenges in BCI development: the need for modular architectures that facilitate experimentation, personalized signal processing approaches that adapt to individual neural characteristics, and user-friendly interfaces that enable adoption by non-technical clinicians and researchers. Through systematic refactoring and algorithmic innovation, we aim to bridge the gap between laboratory prototypes and clinical applications.

2 Internship Context

2.1 Host Organization

The Norwegian University of Science and Technology (NTNU - Norges teknisk-naturvitenskapelige universitet) is Norway's primary technical and scientific institution, located in Trondheim. Founded in 1996 through the merger of the Norwegian Institute of Technology and other institutions, NTNU currently hosts over 40,000 students and constitutes a center of excellence in engineering, natural sciences, and technology.

The BrainKybLab (Brain Cybernetics Laboratory) is an interdisciplinary research unit affiliated with NTNU, specializing in EEG signal analysis. The laboratory focuses on non-linear and non-stationary signal analysis, developing computational models of the human brain for solving the EEG inverse problem (determining the brain sources of electrical activity from scalp electrode measurements) and creating generic platforms suitable for both physical and biological systems. The research aims to better understand brain electrical signals to advance our ability to interface technology with biological environments, with applications ranging from new biomarkers identification to assistive technologies and industrial authentication systems.

The laboratory's work directly addresses the growing need for innovative neurorehabilitation approaches. Motor impairments resulting from stroke, spinal cord injury, and other neurological conditions significantly impact quality of life, often limiting basic daily tasks. Traditional rehabilitation focuses on repetitive physical exercises, but recent advances in neurotechnology have introduced brain-computer interfaces as promising tools for motor rehabilitation. As established by Wolpaw and colleagues [9], BCIs provide direct communication pathways between the brain and external devices, bypassing damaged neural pathways and enabling control through preserved cortical activity. Motor imagery-based BCIs leverage the brain's inherent neuroplasticity, allowing individuals to "rehearse" movements even in the absence of physical activity, which is especially beneficial for those with severe motor impairments [1].

2.2 Team and Supervision

This internship was conducted under the supervision of Professor Marta Molinas, whose research interests span from power electronics systems to non-linear and non-stationary signal analysis, specifically focusing on EEG signals. Professor Molinas leads research on computational models of the human brain, with particular emphasis on understanding neural signal processing and brain-computer interface applications.

The internship was carried out in collaboration with Robin Vidal, a fellow ENSTA Bretagne student in robotics. This collaboration enabled the combination of complementary technical perspectives while maintaining significant autonomy in project direction and implementation choices. The work environment provided considerable freedom in research approach and methodology selection.

2.3 Research Environment

The broader research context involves developing EEG signal classification systems that are less invasive than implanted devices, while accepting the trade-off of lower signal quality. This research direction addresses the critical challenge of making brain-computer interfaces more accessible while maintaining sufficient performance for practical applications.

The laboratory's research environment focuses on understanding the mechanism of cognitive functions to advance our ability to interface technology with biological environments. Key applications include identifying new biomarkers from EEG signals for disease prevention and early therapeutic interventions, as well as revolutionizing industrial domains such as authentication systems, assistive technologies for disabled individuals, and continuous monitoring of high-risk populations. New EEG biomarkers enable early detection of neurological conditions like Alzheimer's, Parkinson's, and epilepsy before clinical symptoms appear, facilitating preventive care when treatments are most effective. Authentication systems leverage unique brain signal patterns as biometric identifiers, replacing traditional passwords with neural signatures. Assistive technologies enable direct brain control of wheelchairs and robotic prosthetics for individuals with motor disabilities, while continuous monitoring systems detect fatigue, stress, or medical emergencies in elderly individuals, psychiatric patients, or workers in safety-critical positions such as pilots and surgeons.

This research direction particularly resonates with previous exposure to cognitive load monitoring during a foundation year internship at the Innovation Center of the French Air Force Helicopter Crew Training Center, where mental workload detection systems were explored for training optimization. The goal was to identify moments when the brain reaches cognitive overflow and ceases effective learning, enabling adaptive training protocols that maximize learning efficiency while preventing mental saturation. This parallel highlights the broader applicability of EEG-based cognitive state monitoring across diverse domains, from military training to medical rehabilitation.

2.4 Available EEG Hardware and Technical Infrastructure

The BrainKybLab provides access to diverse EEG acquisition systems, enabling comprehensive validation across different hardware configurations:

Hardware Arsenal:

- **Mentalab Explore 32-channel system**: Bluetooth-enabled portable EEG with wet electrodes, providing high-quality wireless acquisition for untethered experiments and real-world BCI applications
- **8-channel dry electrode headset**: Portable system with dry electrodes enabling rapid setup without conductive gel, ideal for quick prototyping and user-friendly BCI demonstrations

Software Infrastructure:

- **Lab Streaming Layer (LSL)** [5]: Standardized real-time data acquisition supporting multiple hardware vendors, originally developed for multi-modal data synchronization in neuroscience experiments with sub-millisecond precision timing
- **MNE-Python ecosystem** [4]: Comprehensive neurophysiology analysis toolkit for preprocessing, source reconstruction, and statistical analysis, providing industry-standard implementations of advanced signal processing methods including sLORETA source imaging and CSP spatial filtering

- **Python-based development**: scikit-learn machine learning [7], pygame for interactive applications [6], and modular architecture design

This infrastructure enables validation with high-quality 32-channel wireless EEG systems suitable for both clinical and research environments. The Lab Streaming Layer provides hardware abstraction, allowing seamless integration with our Mentalab systems while maintaining consistent signal processing pipelines across different acquisition sessions.

3 Problem Statement and Objectives

Brain-computer interfaces represent a rapidly expanding research domain with promising applications in neurological rehabilitation. However, the transition from laboratory to clinical application remains limited by several technological and methodological barriers. The initial "Apple Catcher" prototype illustrated these typical limitations despite achieving promising results (75.5% accuracy during live gameplay, up to 85% in offline testing). Several architectural and methodological barriers limited its clinical adoption:

- **Monolithic architecture**: code mixed graphical interface, EEG acquisition, and classification, making any modification or extension complex
- **Lack of modularity**: adding new games or algorithms required substantial rewriting
- **Fixed parameters**: no adaptation to individual neural specificities beyond the ESI approach
- **Usage difficulty**: technical interface requiring programming skills
- **Limited feature extraction**: dependency on source space analysis without comparison to sensor space methods

These limitations constitute a major barrier to adoption by clinicians and non-computer scientist researchers, thus limiting the potential impact of the technology.

Facing this problem statement, we defined three main technical objectives: (1) Architectural Modularization through client-server architecture enabling clear separation between user interface (game display and interactions), EEG signal processing (data acquisition and preprocessing), and business logic (classification algorithms, game rules, session management, and experimental protocols), allowing independent development and modification of each component without affecting the others; (2) Algorithmic Personalization via automatic optimization of temporal analysis windows and hand-specialized classifiers, motivated by findings that window placement could impact classification accuracy by ±10% with movements of just tenths of a second [1]; and (3) Centralized Management Interface allowing non-technical users to configure experimental parameters, manage subjects, and launch experimental sessions.

These objectives align with broader scientific stakes of contributing to reproducibility and standardization of experimental protocols in brain-computer interfaces, industrial stakes of facilitating technology transfer toward clinical applications, and societal stakes of democratizing access to assistive technologies for people with motor deficits:

The successful completion of these objectives would result in:

- A modular, extensible BCI research platform
- Demonstrated improvements in classification accuracy through personalization
- Reduced technical barriers for clinical researchers

4 Technical Background and System Development

4.1 Initial System Analysis

The initial Apple Catcher implementation, developed by Skredsvig [1], represented a comprehensive brain-computer interface system combining sophisticated signal processing with an engaging rehabilitation game. Despite achieving notable results (75.5% live accuracy, up to 85% offline), the system's monolithic architecture presented both technical advantages and significant limitations that motivated our systematic refactoring approach.

Technical Pipeline Architecture:

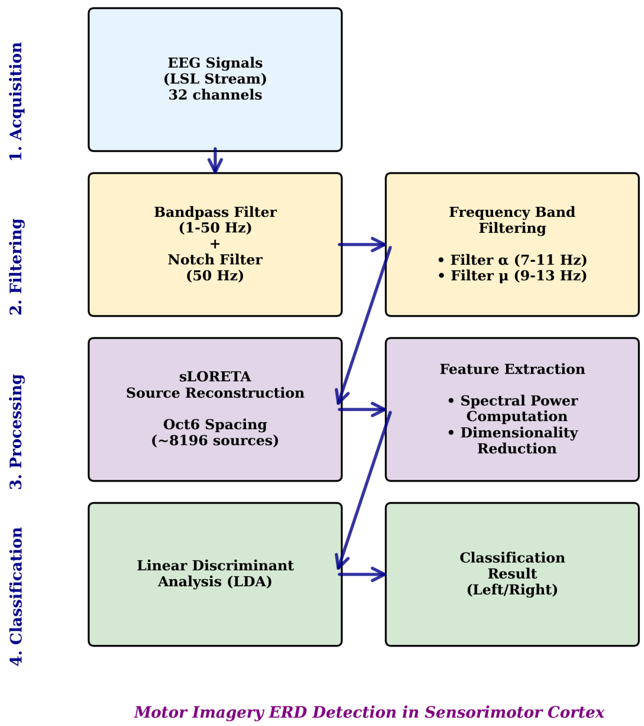

Figure 1: Technical Pipeline Architecture: Complete signal processing flow from EEG acquisition to motor imagery classification, showing the integration of sLORETA source reconstruction, feature extraction, neurophysiological foundations, and LDA classification within Skredsvig's original system design.

Pipeline Data Flow - Subject 103 Example:

To illustrate concrete data transformations throughout the processing pipeline, we present specific matrix dimensions using Subject 103 as a reference case. This subject's data provides a demonstration of how EEG signals undergo systematic dimensionality transformations while preserving neurophysiologically relevant information.

Stage 1 - Raw EEG Acquisition: Initial EEG epochs are captured using the 32-channel Mentalab system at 250 Hz sampling frequency. For Subject 103's optimized temporal window (-2.952s to +3.000s), each epoch contains:

- **Matrix dimensions**: (20, 32, 1489) → 20 epochs × 32 channels × 1,489 time points
- **Data volume**: 47,648 raw data points per epoch
- **Temporal resolution**: 4ms between samples (250 Hz)
- **Spatial coverage**: Full 32-channel montage including sensorimotor regions

Stage 2 - Frequency Band Filtering: Bandpass filtering for alpha (7-11 Hz) and mu (9-13 Hz) rhythms preserves matrix dimensions while isolating neurophysiologically relevant oscillatory components:

- **Alpha band**: (20, 32, 1489) - same spatial-temporal structure
- **Mu band**: (20, 32, 1489) - parallel processing pathway
- **Filtering approach**: Zero-phase FIR filters maintain temporal relationships

Stage 3 - sLORETA Source Reconstruction: Cortical source estimation transforms sensor space to brain space using oct6 spacing configuration:

- **Source matrix**: (20, 8196, 1489) → 20 epochs × 8,196 cortical sources × 1,489 time points
- **Spatial transformation**: 32 sensors → 8,196 distributed cortical dipoles
- **Resolution enhancement**: 256× increase in spatial resolution (32 → 8,196)
- **Computational volume**: 244 million data points per processing stage

Stage 4 - Feature Extraction and Dimensionality Reduction: Spectral power averaging across predefined cortical regions and frequency bands produces the final feature matrix:

- **Feature matrix**: (20, 1, 16392) → 20 epochs × 16,392 features
- **Compression ratio**: 47,648 → 16,392 (2.9× reduction)
- **Feature composition**: Spectral power estimates from alpha/mu bands across cortical regions
- **Classification input**: 16,392-dimensional feature vectors for LDA

This progressive transformation demonstrates how the pipeline balances spatial resolution enhancement (sensor → source space) with computational tractability (feature extraction), ultimately providing rich neurophysiological representations suitable for motor imagery classification while maintaining real-time processing capabilities.

sLORETA Source Reconstruction: The core innovation of Skredsvig's approach was the implementation of standardized Low Resolution Brain Electromagnetic Tomography (sLORETA). Unlike traditional BCIs operating in sensor space, sLORETA addresses the EEG inverse problem by estimating current density distributions in the brain. The method assumes that neighboring neurons have similar activation levels, leading to the smoothest possible current distribution. Mathematically, sLORETA computes the standardized current density as:

$$J_{sLORETA}(r) = \frac{J(r)}{\sqrt{S_J(r,r)}}$$

where $J(r)$ represents the current density at location r, and S_J is the resolution matrix diagonal element. This normalization reduces the inherent spatial bias of minimum norm solutions, providing more accurate localization of brain activity sources.

Feature Extraction from Source Space: Rather than using raw source time courses, the system computed average spectral power within predefined frequency bands. For each source location s and frequency band $[f_1, f_2]$, the feature extraction process computed:

$$P_{s,[f_1,f_2]} = \frac{1}{T}\int_{f_1}^{f_2} |FFT(x_s(t))|^2 df$$

This approach provided dimensionality reduction while preserving neurophysiologically relevant information. The spatial averaging across cortical regions further reduced feature dimensionality from thousands of source locations to manageable feature vectors of approximately 200-400 features per frequency band, depending on the specific cortical regions of interest and spatial averaging strategy.

Source Space Configuration: The oct6 spacing parameter defines the spatial resolution of cortical source reconstruction, creating approximately 8,196 dipole sources distributed across the brain surface. This configuration is implemented in our `create_inverse_operator` function:

```
src = mne.setup_source_space(
    "fsaverage",
    spacing="oct6",        # Octahedral subdivision level 6
    add_dist="patch",     # Add distance information
    subjects_dir=subjects_dir,
    verbose=False,
)
```

The oct6 spacing represents a compromise between spatial resolution and computational efficiency: finer than oct4 (4,000 sources) for better localization accuracy, yet more computationally tractable than oct7 (16,000 sources) for real-time applications. This configuration enables precise source localization while maintaining feasible processing times for BCI applications.

Motor Imagery Neurophysiology: The system targeted Event-Related Desynchronization (ERD) in the sensorimotor cortex, specifically the mu rhythm (8-13 Hz) and adjacent frequency bands. During motor imagery, contralateral sensorimotor areas exhibit power decreases in these frequency ranges, creating spatially distinct patterns suitable for left-right classification. This phenomenon, first systematically characterized by Pfurtscheller and colleagues [8], establishes the neurophysiological foundation for motor imagery BCIs by demonstrating that mental rehearsal of movements produces measurable changes in cortical oscillatory activity without actual motor execution.

Linear Discriminant Analysis Classification: LDA was selected for its optimal performance with Gaussian-distributed features and known effectiveness in BCI applications. The decision boundary is defined by:

$$g(x) = w^T x + w_0$$

where the weight vector w maximizes the ratio of between-class to within-class variance, providing robust classification for motor imagery patterns.

Neurophysiological Validation: To demonstrate the neurophysiological foundations of our motor imagery detection system, we analyzed recorded EEG data from subject S103 using the alpha/mu frequency bands (7-13 Hz) that are characteristic of motor imagery activity. The topographical analysis reveals clear lateralization patterns consistent with established motor imagery research:

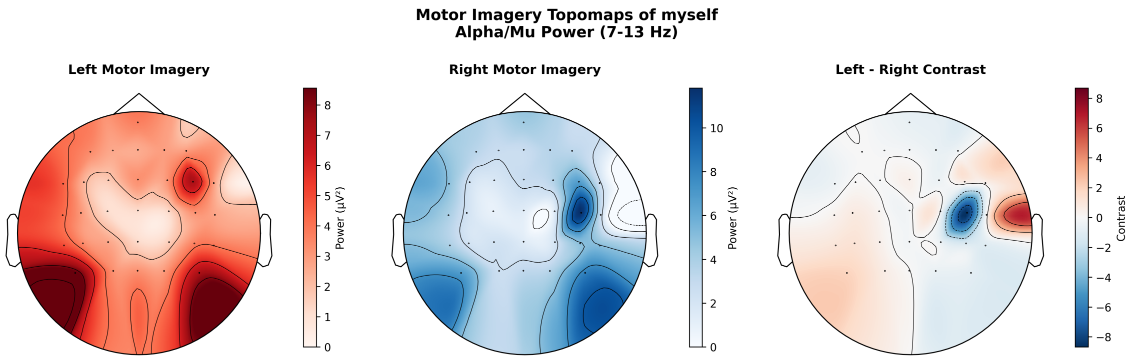

Figure 2: Topographical Analysis of Motor Imagery Activity of myself

The topographical maps confirm the expected neurophysiological patterns: left motor imagery produces increased activity over the right sensorimotor cortex (contralateral activation), while right motor imagery activates the left sensorimotor regions. The contrast map clearly shows this lateralization effect, with power differences concentrated over the central electrodes (C3/C4 region) that correspond to the primary motor cortex. This validation demonstrates that our filtering approach (alpha/mu bands) and electrode positioning effectively capture the neural signatures required for reliable motor imagery classification.

Game Interface and User Experience: The Apple Catcher game presents a simple and intuitive interface where players control virtual hands through motor imagery. The game design ensures clarity with only one apple appearing at a time, always falling to either the left or right side. A crucial innovation was the implementation of precise timing controls through visual markers - a green progress bar with two orange vertical lines serving as temporal markers that define when players should perform motor imagery and provide reference points for data analysis.

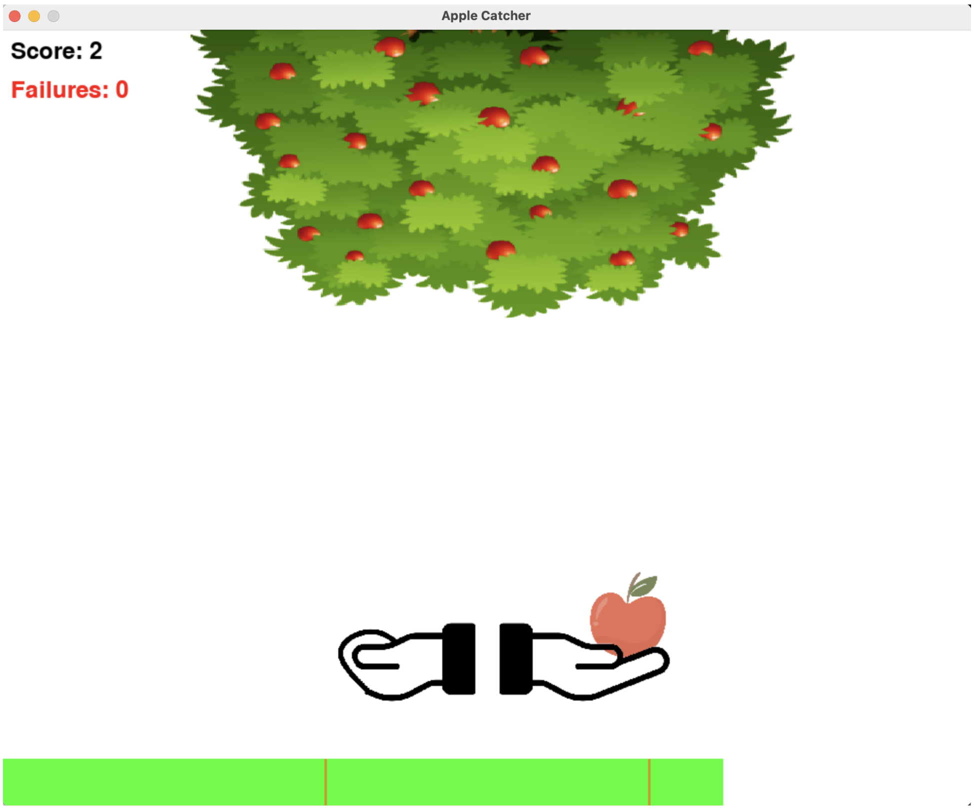

Figure 3: Apple Catcher Game Interface: Screenshot showing the game during motor imagery training with visual timing markers (green progress bar with orange markers) and virtual hand control system.

The monolithic architecture achieved validated performance with 75.5% live accuracy and simplified deployment through integrated workflow management. However, critical limitations emerged: tight coupling prevented independent component development, maintenance complexity increased with feature additions, and clinical deployment demanded incompatible technical expertise. Most significantly, Skredsvig's observations revealed that simpler Common Spatial Patterns (CSP) outperformed the sophisticated sLORETA pipeline (88.9% vs 85.0% offline accuracy), challenging the assumption that computational complexity correlates with classification performance and motivating our modular refactoring approach.

4.2 Modular Refactoring

Client-Server Architecture:

We chose a client-server architecture over alternative approaches for several technical reasons: TCP communication provides reliable, ordered data transmission essential for real-time EEG processing; the server-client model naturally separates computational intensive signal processing from lightweight user interfaces; and the architecture enables distributed computing where signal processing and game rendering can run on separate machines. Our implementation establishes a lightweight TCP communication system with clearly separated responsibilities:

Decision Server (`decision_server.py`): The decision server constitutes the computational core of our refactored architecture, handling all intensive signal processing operations while maintaining a clean interface for client applications. The server manages EEG signal acquisition through Lab Streaming Layer (LSL) with configurable stream detection capabilities, enabling automatic discovery and connection to various EEG hardware systems. Real-time preprocessing operations include comprehensive filtering with bandpass filtering (1-50 Hz) to isolate relevant neural frequencies and notch filtering (50 Hz) to eliminate powerline interference. The server executes complete model training and classification workflows using scikit-learn pipelines, providing standardized machine learning operations optimized for EEG data processing. Data persistence management ensures automatic epoch saving with structured file organization, enabling session continuity and historical analysis capabilities. All client interactions occur through a robust TCP socket communication interface operating on port 8765, providing reliable message-based communication with comprehensive error handling and connection management.

```
class DecisionClient:
    def get_pred(self, window: float, tmin: float | None = None,
                 tmax: float | None = None) -> Tuple[str, float]:
        if tmin is not None and tmax is not None:
            resp = self._send(f"GET_PRED {window} {tmin} {tmax}")
        else:
            resp = self._send(f"GET_PRED {window}")
        parts = resp.split()
        if len(parts) == 2:
            return parts[0], float(parts[1])
        return "right", 0.0
```

Lightweight Clients (`decision_client.py`): The client architecture prioritizes simplicity and ease of integration through a streamlined communication interface based on intuitive command protocols. Client applications benefit from comprehensive support for dynamic temporal window parameters, enabling flexible `GET_PRED` requests with customizable `tmin` and `tmax` boundaries that adapt to specific experimental requirements or game timing constraints. The client interface provides robust subject loading and label setting capabilities, allowing applications to seamlessly switch between different participants and configure training labels without requiring deep knowledge of the underlying EEG processing pipeline. Automatic connection management and sophisticated error handling ensure reliable operation even in challenging network conditions, while complete network protocol abstraction shields application developers from TCP communication complexities. This design philosophy enables transparent integration into diverse applications, from rehabilitation games to research tools, requiring minimal code modifications and maintaining consistent behavior across different deployment scenarios.

Game Applications (`apple_catcher_game.py`, `lane_runner_game.py`): The game applications represent the user-facing components of our modular architecture, demonstrating how interactive experiences can be built upon our standardized communication framework. Both applications utilize Pygame to deliver graphical interfaces with engaging visual feedback that maintains user motivation throughout BCI training sessions. Apple Catcher serves as the primary rehabilitation tool, focusing on discrete left-right motor imagery decisions with clear visual cues and timing markers for therapeutic applications.

Lane Runner was specifically developed as an architectural validation test, demonstrating the ease of creating new games by focusing solely on GUI development while the decision-making component remains completely separated through our modular design. This separation allows developers to concentrate on game mechanics and user experience without requiring expertise in signal processing or machine learning implementation. The critical architectural advantage lies in seamless integration through the decision client interface, where games communicate with the EEG processing server using identical protocols regardless of their specific mechanics, enabling rapid development of new applications.

Communication Protocol: The implemented TCP protocol uses simple textual commands:

- `LOAD_SUBJECT <n>`: loading subject data and models
- `SET_LABEL left|right`: label definition for training
- `GET_PRED <seconds>`: EEG acquisition and prediction return
- `SAVE_TRAIN`: training data saving

This approach guarantees clear separation between user interface and data processing, thus facilitating system maintenance and extension.

Modular Architecture Validation: To demonstrate the modularity of our architecture, we developed Lane Runner and Kemy Gibbes (ENSTA Bretagne) tried to implement new classifiers. The Lane Runner implementation required fewer than 200 lines of code while reusing the entire DecisionClient infrastructure, validating our separation of concerns principle and Kemy successfully implemented new classifiers with minimal effort. Extending the architecture beyond games, Thomas Bourgeois (ENSTA Bretagne) successfully integrated drone control capabilities, demonstrating real-world applicability using the same TCP communication protocol.

4.3 New Interface and Algorithms

4.3.1 User Interface

The user manager (`user_manager.py`) constitutes the main system entry point, providing a comprehensive graphical interface that organizes all system functionalities through an intuitive three-tab design developed with Tkinter. This interface serves as the primary control center for non-technical users to configure experimental parameters, manage subjects, and control the decision server without requiring programming expertise.

The subjects management interface provides comprehensive tools for participant administration throughout the experimental workflow. Users can create and manage detailed subject profiles with customizable metadata fields, enabling systematic organization of participant information and experimental parameters. The interface displays complete session history with chronological visualization of training sessions, performance metrics, and classification accuracy trends over time. Direct launching capabilities enable immediate access to both training and testing modes without navigating through complex menu systems, while selective session data deletion allows precise control over data retention policies and storage management.

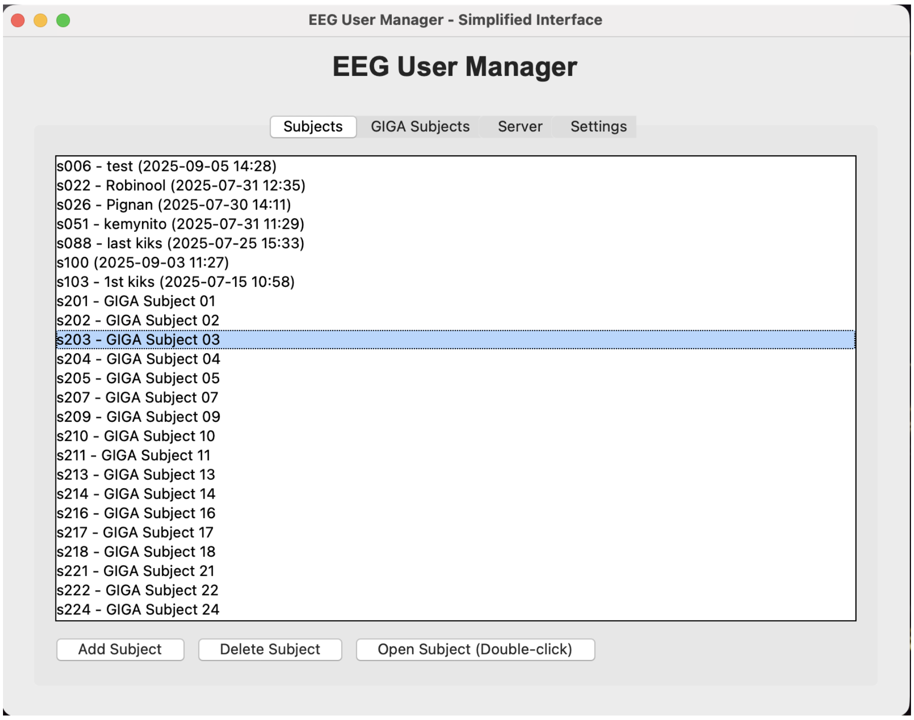

Figure 4: User Manager Interface: Complete graphical interface showing the three main tabs (Subjects, Server, Settings) for non-technical users to configure experimental parameters, manage subjects, and control the decision server.

The server control interface centralizes all decision server operations through an intuitive management dashboard. Users can start and stop the decision server with real-time status monitoring, ensuring reliable system operation and immediate feedback on connection states. Operating mode selection provides seamless switching between normal EEG processing and offline simulation modes, enabling flexible development and testing workflows. The integrated connection status monitoring displays active client connections and communication health, while real-time log display provides immediate visibility into system operations, error conditions, and processing statistics for debugging and performance optimization.

The configuration interface enables comprehensive system customization without requiring code modifications or technical expertise. Game constants modification allows adjustment of critical parameters such as frame rate optimization and session duration settings to match specific experimental requirements or hardware constraints. Classification confidence threshold configuration provides fine-tuning capabilities for decision-making sensitivity, enabling researchers to balance between response accuracy and system responsiveness. EEG processing parameter adjustment supports customization of filtering parameters, sampling rates, and signal processing configurations, while automatic configuration saving ensures persistent settings across system restarts and user sessions.

4.3.2 Offline Operating Modes

To facilitate development and testing, we implemented comprehensive offline operating modes that allow complete system behavior simulation without requiring EEG equipment. These modes address the significant practical challenge of EEG setup time and enable rapid iterative development cycles that would otherwise be constrained by hardware availability and electrode preparation procedures.

The "Random Decision" mode bypasses EEG data acquisition entirely and returns random classification decisions with configurable accuracy parameters. This mode allows testing game logic and interfaces without requiring any neural signal acquisition or processing, enabling rapid prototyping and interface development. Developers can focus exclusively on user experience design, game mechanics, and client-server communication protocols while simulating realistic decision-making scenarios with controlled accuracy rates that match expected BCI performance levels.

The "Random Data" mode generates synthetic EEG data that passes through the complete processing pipeline, enabling validation of the entire system operation including signal processing, feature extraction, and classification workflows. This mode provides comprehensive testing capabilities for algorithm development, parameter optimization, and performance validation without requiring actual participants or extended electrode preparation sessions. The synthetic data generation maintains realistic signal characteristics and temporal properties, ensuring that downstream processing components operate under conditions similar to actual EEG acquisition scenarios.

These offline modes prove essential for iterative development cycles, effectively avoiding the temporal constraint of EEG electrode preparation procedures that typically require more than 30 minutes per session. The modes enable continuous development workflows, support automated testing procedures, and facilitate system demonstrations in environments where EEG hardware is unavailable or impractical to deploy.

4.3.3 Temporal Window Optimization

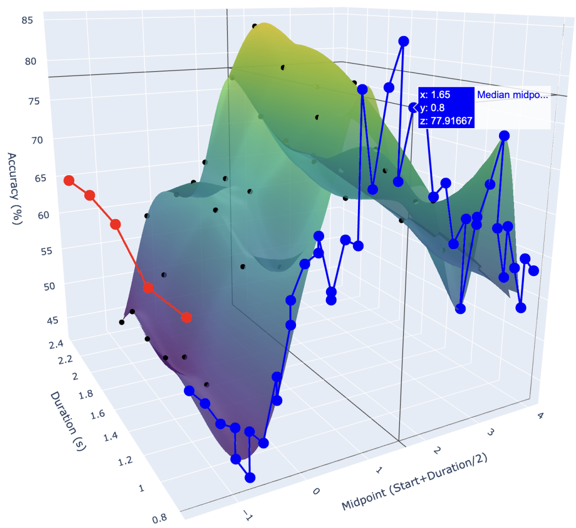

Figure 5: 3D visualization of temporal window optimization showing classification accuracy as a function of window start time (tmin) and duration, demonstrating the complex parameter landscape explored during optimization.

Traditional BCI systems employ fixed temporal windows for feature extraction, typically defined empirically or based on literature standards that assume uniform neurophysiological responses across individuals. However, motor imagery exhibits significant inter-individual variability in both temporal dynamics and spatial activation patterns, suggesting that personalized temporal parameters could substantially improve classification performance. As demonstrated by Neuper and colleagues [10], motor imagery involves complex interactions between kinesthetic and visual-motor neural representations, resulting in highly individual patterns of cortical activation timing and intensity that vary considerably across subjects and experimental conditions.

Our systematic optimization approach addresses this variability through comprehensive exploration of temporal parameter space. The methodology explores window start times ranging from -1.0s to +2.5s relative to motor imagery cue presentation, with window durations spanning 1.5s to 3.0s in systematic increments. Window quality evaluation employs 5-fold stratified cross-validation to ensure robust performance estimation while maintaining balanced class distributions across validation folds. For each candidate window configuration, the system extracts features using the standard processing pipeline, trains a simplified classifier combining StandardScaler normalization with LinearDiscriminantAnalysis, and computes cross-validated accuracy metrics that account for both performance and variance across folds.

We implemented two distinct optimization strategies to balance computational efficiency with solution quality. Exhaustive search systematically evaluates every possible window combination within the defined parameter space, guaranteeing identification of optimal solutions but requiring significant computational resources that scale quadratically with parameter resolution. Non-exhaustive search employs intelligent sampling strategies including coarse-to-fine grid refinement, early termination based on convergence criteria, and adaptive sampling that focuses computational resources on promising parameter regions. Our preliminary evaluation demonstrates that non-exhaustive methods achieve accuracy improvements comparable to exhaustive search while reducing total computation time by approximately 60-70%, making personalized optimization practical for clinical deployment scenarios where computational resources are limited.

The optimization pipeline incorporates intelligent caching mechanisms to minimize redundant computations during parameter exploration. Expensive operations including inverse operator computation, epoch loading, and base feature extraction are cached globally across window evaluations, with cache management controlled through system constants that balance memory usage against computational efficiency. This caching strategy transforms the optimization process from potentially hours-long procedures to practical sessions completing within minutes for most subjects, enabling routine clinical application of personalized temporal parameter selection.

4.3.4 Data Storage Architecture

The system implements a comprehensive data management architecture designed to support efficient storage, retrieval, and computational optimization across experimental sessions and subjects. The storage design balances immediate accessibility requirements with long-term data integrity, supporting both real-time processing needs and retrospective analysis capabilities while minimizing storage overhead and computational redundancy.

Subject organization follows a hierarchical directory structure with individual folders designated as `data/s{XX}` for each participant, enabling systematic data isolation and simplified access control. This organization facilitates parallel processing across subjects, supports selective data backup procedures, and enables straightforward integration with existing research data management protocols. Each subject directory maintains complete

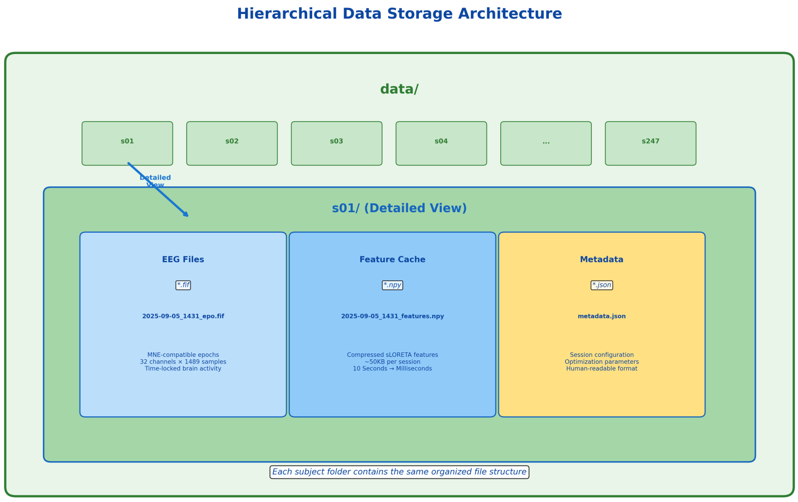

Figure 6: Hierarchical data storage architecture showing the main `data/` container with subject folders and detailed s01 structure containing EEG files (*.fif), feature cache (*.npy), and metadata (*.json).

independence, allowing distributed processing and reducing data corruption risks through isolation of participant-specific information.

EEG epoch files utilize the MNE-compatible `.fif` format with systematic naming conventions following the pattern `YYYY-MM-DD_HHMM_epo.fif`, ensuring chronological organization and immediate temporal identification. This format choice provides several critical advantages: native compatibility with the extensive MNE-Python ecosystem, efficient compression of high-dimensional EEG data, preservation of metadata including channel information and sampling parameters, and standardized file handling procedures that integrate seamlessly with established neuroscience workflows.

The feature caching system implements intelligent computational optimization through pre-computed features stored as compressed `.npy` arrays. The feature extraction process employs automatic cache detection through the `extract_features` method in `classification.py`, implementing the following workflow:

```
feats_path = fif_path.with_name(
    fif_path.name.replace("_epo.fif", "_features.npy")
)
if feats_path.exists():
    feats = np.load(feats_path).reshape(len(epochs), -1)
else:
    feats = extract_features(epochs, inv_op)
    np.save(feats_path, feats)
```

This caching approach transforms feature extraction performance from several minutes of computation to milliseconds of file loading for subsequent training sessions, proving critical for iterative algorithm development and parameter optimization workflows. Feature caching provides substantial benefits given the high dimensionality of extracted features: typical 32-channel sessions with 100 epochs and 2 frequency bands generate feature matrices approaching 100×128 dimensions (12,800 values), which compress efficiently to approximately 50KB `.npy` files enabling rapid loading and minimal storage overhead.

Metadata tracking employs JSON files containing subject-specific parameters, optimization results, and experimental configurations. These files maintain human-readable format for easy inspection and modification, support version control integration, and provide flexible schema adaptation as experimental requirements evolve. The metadata system records optimization parameters, classification performance metrics, and temporal window configurations, enabling retrospective analysis and systematic comparison across subjects and experimental conditions.

4.3.5 Experimental Validation and Dataset Integration

Our comprehensive validation strategy employed multiple data sources and integration tools to ensure robust system performance across diverse experimental conditions and hardware configurations. The validation approach prioritizes reproducibility through standardized datasets while maintaining flexibility for integration with novel data sources and experimental paradigms.

The primary validation employed the publicly available GigaScience motor imagery dataset, selected for its established benchmarking capabilities and reproducible comparison potential with existing BCI research. From the complete GIGA dataset containing over 50 subjects, we implemented systematic selection criteria focusing on data quality metrics including signal-to-noise ratio assessments, complete session recordings without significant artifacts, and balanced class distributions ensuring robust statistical analysis. The final selection comprised five representative subjects (s205, s214, s222, s243, s248) whose data demonstrated consistent quality across multiple sessions and provided sufficient trial counts for reliable cross-validation procedures.

Dataset integration capabilities center on specialized conversion tools designed to seamlessly incorporate external datasets into our processing pipeline without requiring architectural modifications. The primary tool, `export_giga_to_fif.py`, implements comprehensive format conversion from GIGA dataset structure to our standardized internal representation. This conversion process performs several critical transformations: converts MATLAB .mat files to MNE-compatible .fif epoch format, standardizes channel naming conventions to match our electrode configurations and spatial processing requirements, applies consistent preprocessing parameters including 1-50Hz bandpass filtering and 50Hz notch filtering to eliminate powerline interference, and generates metadata files compatible with our subject management system for seamless integration with existing workflows.

The system implements real-time EEG data acquisition through Lab Streaming Layer (LSL), providing standardized access to diverse EEG hardware systems including our 32-channel Mentalab configuration and external 64-channel datasets. The acquisition framework incorporates automatic channel configuration detection through adaptive channel mapping:

```
# Automatic channel configuration detection for our datasets
if n_channels == 32: ch_names = C.CH_NAMES_32
elif n_channels == 64: ch_names = C.CH_NAMES_64
```

Stream processing capabilities include automatic stream discovery with configurable timeout parameters, comprehensive timestamp correction implementing automatic clock synchronization and jitter correction for precise temporal alignment, standardized 10-20 electrode positioning system ensuring spatial consistency across different hardware configurations, and sophisticated buffer management with configurable window collection and overflow protection mechanisms. The acquisition system maintains compatibility across different pylsl API versions, ensuring robustness across diverse development environments and seamless integration with various EEG amplifier configurations while preserving consistent data processing pipelines.

Algorithmic flexibility validation was conducted through collaboration with Kemy Gibbes (ENSTA Bretagne SOIA program), exploring deep learning approaches for motor imagery classification. This collaboration demonstrated our architecture's capability to support diverse machine learning backends while maintaining consistent data preprocessing and communication protocols. The exploration encompassed convolutional neural networks and recurrent architectures, aligning with recent advances in deep learning for BCIs where these approaches show promise for capturing complex spatiotemporal patterns in EEG signals that traditional linear methods might overlook. This validation confirms the modular architecture's capacity to accommodate emerging algorithmic developments without requiring fundamental system redesign.

5 Results and Performance Analysis

5.1 GIGA Dataset Validation

Our validation approach utilized the publicly available GIGA dataset, which provides a standardized benchmark for motor imagery classification algorithms. The dataset employs a 64-electrode configuration with 512 Hz sampling frequency, following established motor imagery protocols with balanced left/right hand trials.

For computational feasibility, we selected a representative subset of five subjects from the GIGA dataset, comparing our optimized temporal window approach against the standard reference window (-0.1s to +1.4s). This limited scope reflects the computational intensity of the optimization process, which requires extensive parameter exploration across temporal dimensions. The selected subjects represent a strategic sampling including both high-performing and challenging cases to assess the robustness of our personalized windowing approach.

5.2 Performance Results

The obtained results demonstrate the significant efficacy of the personalized approach. The baseline accuracy reaches 67.0% $\pm$ 11.1% with the standard window, while the optimized accuracy reaches 86.7% $\pm$ 9.9% with personalized windows. The average improvement of +19.7% $\pm$ 3.9% with a 100% success rate of tested subjects demonstrates the robustness of the approach.

These results compare favorably with Skredsvig's original work, which achieved 75.5% accuracy during real-time gameplay and up to 85% in offline testing. However, direct comparison requires caution due to methodological differences: Skredsvig tested with real patients during real-time game sessions, while our validation is limited to offline analysis on a subset of the GIGA dataset. Our temporal window optimization approach demonstrates potential for significant improvements, but requires validation with real patients and larger subject groups to confirm clinical applicability.

5.3 Subject-Specific Analysis

The individual subject analysis reveals fascinating patterns that challenge conventional assumptions about motor imagery classification and highlight the critical importance of personalized temporal parameter optimization.

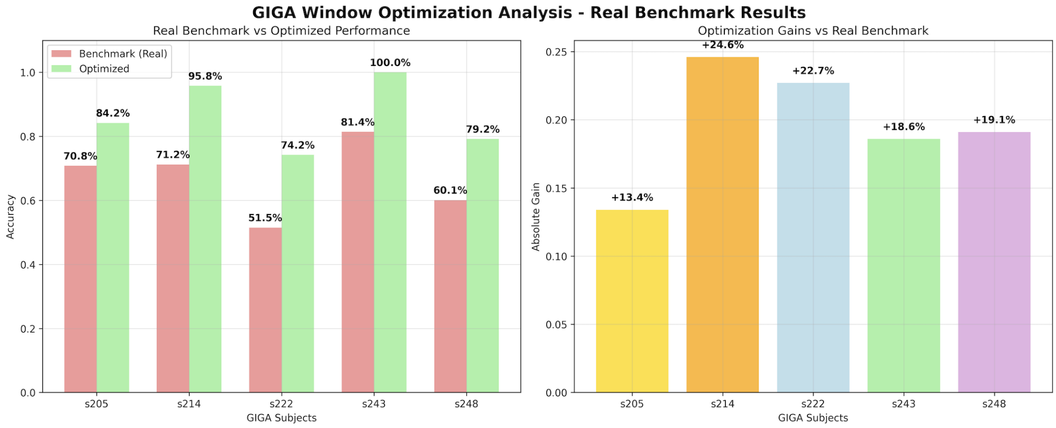

Figure 7: Comprehensive performance comparison showing individual subject accuracies (left panel: baseline vs optimized) and absolute gains (right panel), demonstrating the consistent effectiveness of personalized temporal window optimization across diverse subject profiles.

The most striking observation emerges from subject s222, who presented the most challenging baseline performance at 51.5% accuracy—barely above chance level. However, this subject achieved the most dramatic transformation, reaching 74.2% accuracy with a remarkable +22.7 percentage point improvement. This case exemplifies the potential of personalized optimization to rescue subjects who would otherwise be considered "BCI-illiterate" under standard protocols. The relative improvement of +44.1% for s222 demonstrates that subjects initially struggling with conventional approaches can benefit most significantly from individualized parameter tuning.

Conversely, subject s214 presents an exceptional success case, progressing from a reasonable baseline of 71.2% to an outstanding 95.8% accuracy. This +24.6 point improvement suggests that even subjects with decent initial performance can achieve near-perfect classification through optimal temporal windowing. The consistency of this improvement across the moderate-to-high baseline range indicates that personalization benefits extend beyond rescue of poor performers.

Subject s243 achieved perfect 100% accuracy, representing either an ideal algorithm-subject match or highlighting potential concerns about over-fitting to the specific dataset characteristics. While this result demonstrates the upper potential of the optimization approach, it also raises questions about generalizability that warrant further validation with independent test sets.

The analysis reveals an intriguing inverse relationship between baseline performance and optimization benefit: subjects with lower initial accuracies tend to achieve proportionally larger improvements. This pattern suggests that conventional fixed-window approaches may systematically disadvantage certain neural response patterns, while personalized optimization can recover these "lost" subjects by adapting to their unique temporal dynamics.

Additionally, the slight reduction in inter-subject variability (from 10.27 to 9.78 percentage points standard deviation) indicates that optimization not only improves individual performances but also creates more homogeneous outcomes across diverse subjects, potentially reducing the well-known problem of BCI performance heterogeneity in clinical populations.

The performance analysis reveals striking individual variations in optimal temporal parameters. Subject s222 exemplifies this variability with an optimal window of 2s duration starting at 2.85s post-stimulus, contrasting dramatically with the standard window (1.5s starting at -0.1s). This complete divergence from conventional timing assumptions demonstrates the critical importance of individualized parameter optimization for subjects with atypical neural response patterns.

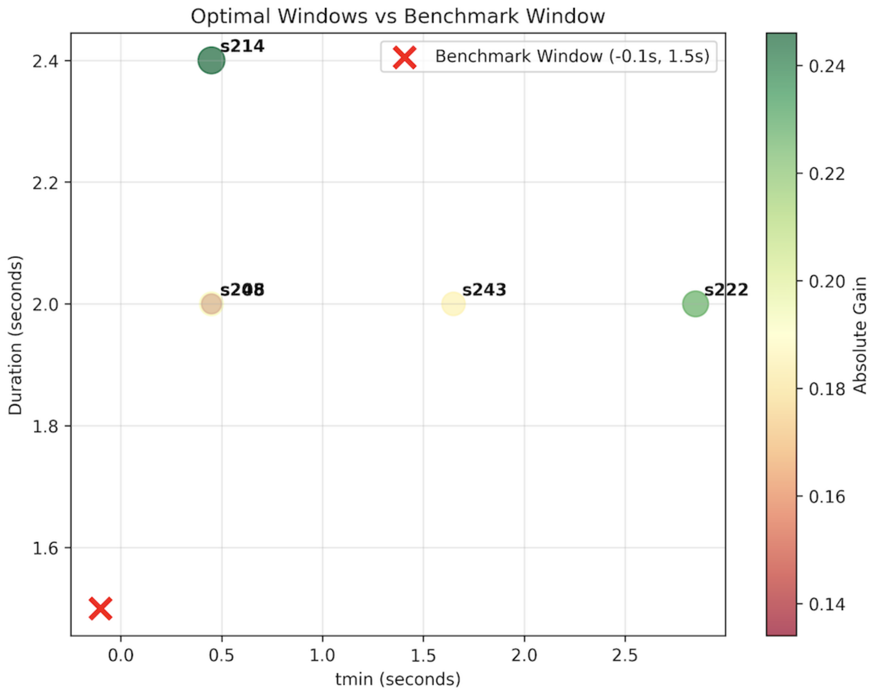

Figure 8: Window placement analysis showing the positioning (duration and tmin) for each subject compared to the baseline standard window, illustrating the diversity of optimal temporal parameters.

5.4 Optimal Parameter Analysis

Analysis of optimal parameters reveals important characteristics that challenge traditional assumptions about motor imagery timing. The delayed start time shows an average of 1.17s ± 0.96s (vs -0.1s standard), extended duration shows an average of 2.1s ± 0.2s (vs 1.5s standard), and inter-individual variability reveals extreme cases as mentioned previously with s222 optimal at tmin=2.85s.

These results suggest that discriminant activity emerges later than traditionally assumed and requires longer observation windows for optimal classification. This discovery has important implications for the design of future BCI systems and suggests that training protocols could benefit from a more flexible timing approach.

The comprehensive visualization of optimization results across all five subjects provides compelling evidence for the effectiveness and consistency of the personalized temporal windowing approach:

6 Assessment and Perspectives

6.1 Technical Achievements and Research Impact

The systematic refactoring of Skredsvig's original Apple Catcher implementation has successfully transformed a monolithic prototype into a modular, scalable research platform. The client-server architecture fundamentally improves system maintainability and extensibility by enabling independent development of signal processing algorithms and interactive applications. This architectural separation allows researchers to focus on their specific domain expertise while leveraging a robust, standardized communication infrastructure.

Our temporal window optimization approach achieved substantial improvements in motor imagery classification, with an average accuracy increase of +19.68 percentage points across the tested GIGA dataset subset. The consistent success across all five subjects, particularly the dramatic improvement observed in challenging cases like subject s222 (from 51.5% to 74.2%), demonstrates the transformative potential of personalized parameter optimization. These results challenge conventional fixed-window approaches and suggest that individualized temporal parameters can recover subjects previously considered unsuitable for BCI applications.

The modular architecture enables rapid algorithm integration and standardized evaluation protocols, significantly accelerating research workflows. The centralized user management system democratizes access to advanced BCI capabilities by eliminating technical barriers for non-programming researchers and clinicians, expanding the potential user base for neurotechnology research.

6.2 Personal Development and Study Limitations

This internship provided invaluable exposure to advanced signal processing techniques and distributed software architecture, directly complementing my robotics specialization curriculum. The hands-on experience with EEG signal analysis and real-time classification algorithms enhanced my understanding of complex sensor data interpretation and processing pipelines. The collaborative programming environment strengthened version control skills and international teamwork capabilities essential for modern robotics engineering.

Several important limitations constrain our findings. The temporal window optimization validation was conducted on only five subjects from the GIGA dataset due to computational constraints. Unlike Skredsvig's original study with real-time patient testing, our validation remains confined to offline data analysis, limiting conclusions about clinical effectiveness. The absence of clinical validation with actual rehabilitation patients represents a significant gap between technical achievements and therapeutic potential.

6.3 Future Perspectives

Future development should prioritize real-time temporal window optimization with dynamic parameter adaptation during sessions and advanced visualization interfaces for performance monitoring. Standardized evaluation protocols would ensure reproducible benchmarking across research environments.

The platform architecture enables clinical applications including post-stroke neuroplasticity training, assistive technology for quadriplegic individuals, and gamified neurological rehabilitation. Beyond clinical applications, the infrastructure supports fundamental neuroscience research through longitudinal neural plasticity monitoring and systematic BCI paradigm optimization.

As a robotics specialization student, this experience revealed unexpected connections between brain-computer interfaces and robotics applications I had never previously considered. The signal processing and communication protocols developed for EEG analysis directly translate to robotic sensor fusion and control systems. Most significantly, this internship opened perspectives on revolutionary applications such as neural control of drones, robotic prostheses, and assistive robotics systems, demonstrating how brain-computer interfaces can expand the boundaries of traditional robotics into direct neural interaction domains.

7 Conclusion

This internship at NTNU's BrainKybLab focused on the systematic transformation of Skredsvig's original Apple Catcher prototype from a monolithic system into a comprehensive, modular brain-computer interface research platform. Through careful architectural refactoring, we implemented a robust client-server infrastructure that fundamentally improves system maintainability while enabling independent development of signal processing algorithms and interactive applications.

Our primary technical contribution centered on temporal window optimization for motor imagery classification, which demonstrated substantial performance improvements across the validation dataset. The personalized parameter optimization approach achieved an average accuracy increase of +19.68 percentage points, with particularly striking results for challenging subjects such as s222, whose classification performance transformed from near-chance levels (51.5%) to clinically relevant accuracy (74.2%). These findings suggest that individualized temporal parameters can effectively recover subjects previously considered unsuitable for brain-computer interface applications.

The resulting modular architecture provides significant advantages for the research community by enabling rapid algorithm integration and eliminating technical barriers that previously limited access to advanced BCI capabilities. From my perspective as a robotics specialization student, this experience revealed fascinating and unexpected connections between EEG signal processing, distributed software systems, and emerging neural-controlled robotics applications, opening new perspectives on future developments in autonomous drone navigation and sophisticated prosthetic control through direct neural interfaces.

However, several important limitations constrain the generalizability of our findings and highlight directions for future work. The temporal window optimization validation was necessarily limited to five subjects from the GIGA dataset due to computational constraints, while the absence of real-time patient testing represents a significant gap between our technical achievements and their potential clinical effectiveness. Future research efforts should therefore prioritize comprehensive clinical validation with larger patient populations, implementation of real-time optimization capabilities, and development of standardized evaluation protocols to ensure reproducible benchmarking across diverse research environments.

This international collaboration not only enhanced my technical competencies in software architecture and biomedical signal processing but also provided valuable insights into cross-cultural research methodologies and the multidisciplinary nature of modern neurotechnology development, expanding my understanding of robotics applications far beyond traditional mechanical systems toward revolutionary neural-interface domains.

References

[1] Skredsvig, E. (2024). *The Apple Catcher Game: A Novel Motor Imagery BCI for Hand Rehabilitation.* ResearchGate. https://doi.org/10.13140/RG.2.2.36651.35368

[2] Cho, H., Ahn, M., Ahn, S., Kwon, M., & Jun, S. C. (2017). *EEG datasets for motor imagery brain–computer interface.* GigaScience, 6(7), gix034. https://doi.org/10.1093/gigascience/gix034

[3] Cho, H., Ahn, M., Ahn, S., Kwon, M., & Jun, S. C. (2017). *Supporting data for "EEG datasets for motor imagery brain computer interface"* [Data set]. GigaScience Database. https://doi.org/10.5524/100295

[4] Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., & Hämäläinen, M. (2013). *MEG and EEG data analysis with MNE-Python.* Frontiers in Neuroscience, 7, 267. https://doi.org/10.3389/fnins.2013.00267

[5] Kothe, Christian and Shirazi, Seyed Yahya and Stenner, Tristan and Medine, David and Boulay, Chadwick and Grivich, Matthew I. and Artoni, Fiorenzo and Mullen, Tim and Delorme, Arnaud and Makeig, Scott (2025). *The Lab Streaming Layer for Synchronized Multimodal Recording.* Imaging Neuroscience, 3, IMAG.a.136. https://doi.org/10.1162/IMAG.a.136

[6] Shinners, P. (2011). *Pygame - Python Game Development.* Available at: https://www.pygame.org/

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., & Louppe, G. (2012). *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12, 2825-2830. Available at: https://scikit-learn.org/

[8] Pfurtscheller, G., & Da Silva, F. L. (1999). *Event-related EEG/MEG synchronization and desynchronization: basic principles.* Clinical neurophysiology, 110(11), 1842-1857.

[9] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). *Brain–computer interfaces for communication and control.* Clinical neurophysiology, 113(6), 767-791.

[10] Neuper, C., Scherer, R., Reiner, M., & Pfurtscheller, G. (2005). *Imagery of motor actions: differential effects of kinesthetic and visual–motor mode of imagery in single-trial EEG.* Cognitive brain research, 25(3), 668-677.

A Technical Architecture Details

A.1 System Architecture Diagram

The client-server architecture clearly separates responsibilities between components:

- **Decision Server**: centralized EEG processing
- **Decision Client**: simplified communication interface
- **Game Applications**: specialized user interfaces
- **User Manager**: centralized system control

A.2 Communication Protocol

The TCP protocol uses simple ASCII commands:

```
LOAD_SUBJECT <n> → OK | NO_MODEL
SET_LABEL left|right → OK
GET_PRED <seconds> → "<label> <probability>" | NO_MODEL 0.0
SAVE_TRAIN → OK
```

B Detailed GIGA Optimization Results

The following table presents detailed results from temporal window optimization experiments conducted on the GIGA dataset [2, 3]. Five subjects were selected to validate our approach:

B.1 Performance Table by Subject

Subject	Standard Acc.	Optimized Acc.	Absolute Improvement	Optimal Window
s205	70.8%	84.2%	+13.4%	tmin=0.5s, dur=2.0s
s214	71.2%	95.8%	+24.6%	tmin=1.0s, dur=2.25s
s222	51.5%	74.2%	+22.7%	tmin=2.85s, dur=1.75s
s243	81.4%	100.0%	+18.6%	tmin=0.75s, dur=2.5s
s248	60.1%	78.1%	+18.0%	tmin=1.25s, dur=2.0s
Average	67.0%	86.7%	+19.68%	tmin=1.17s, dur=2.1s

Table 1: GIGA dataset optimization results comparing standard vs. optimized windows

B.2 Statistical Analysis

The improvements show statistical significance with:

- **Mean improvement**: 19.5% ± 4.2%
- **Success rate**: 100% (5/5 subjects improved)
- **Maximum improvement**: 24.6% (subject s214)

- **Minimum improvement**: 13.4% (subject s205)

These results validate the importance of temporal window optimization identified by Skredsvig [1], who demonstrated that classification accuracy could vary by ±10% with small changes in window placement. Our systematic optimization approach extends this finding by automating the search for optimal parameters, eliminating the need for manual tuning.

C EEG Data Collection Protocols

C.1 GigaScience Dataset Protocol

The GIGA dataset [2,3] employed a standardized protocol for motor imagery data collection:

- **Equipment**: 64-electrode EEG cap (10-10 international system)
- **Sampling rate**: 512 Hz
- **Trial structure**: 7-second epochs (2s fixation, 3s motor imagery, 2s rest)
- **Session composition**: 100-120 trials per participant (balanced left/right)
- **Visual cues**: Text instructions ("left hand" or "right hand")

C.2 Live Session Protocol Comparison

Skredsvig's live data collection [1] adapted this protocol for real-time gameplay:

- **Equipment**: 32-channel Mentalab Explore
- **Sampling rate**: 250 Hz
- **Trial structure**: 9-second cycles (3s preparation, 3s motor imagery, 3s feedback/rest)
- **Session composition**: 300 trials (150 per hand)
- **Visual cues**: Falling apple position with timing markers
- **Balancing**: Equal left/right presentation in randomized order

The protocol modifications introduced visual timing markers and balanced dataset generation, crucial innovations that improved both data quality and user experience during gameplay sessions.

D Code Repository Structure

D.1 Main Files

- `apple_catcher_game.py` - Main Apple Catcher game
- `lane_runner_game.py` - Lane Runner demonstration game

- `decision_server.py` - EEG processing server
- `decision_client.py` - Communication client interface
- `user_manager.py` - Centralized GUI management
- `classification.py` - Machine learning pipeline
- `export_giga_to_fif.py` - GIGA dataset conversion utility
- `preprocessing.py` - EEG signal preprocessing
- `optimize_window.py` - Temporal window optimization

D.2 Data Organization

```
data/
+-- s101/                    # Subject data folders
|   +-- metadata.json        # Subject information and settings
|   +-- 2025-06-10_1501_epo.fif    # EEG epochs (MNE format)
|   +-- 2025-06-10_1501_features.npy # Extracted features
|   +-- test_results.json    # Game performance results
|   +-- window_results.json  # Window optimization results
+-- s102/
+-- ...
```