SOFTWARE ENGINEERING PROJECT

Self-Supervised Learning for EEG-Based Motor Imagery Classification

BY

Nantawat Suksirisunt
Naytitorn Chaovirachot

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING
KASETSART UNIVERSITY

Academic Year 2565

Self-Supervised Learning for EEG-Based Motor Imagery Classification

BY

Nantawat Suksirisunt
Naytitorn Chaovirachot

**This Project Submitted in Partial Fulfillment of the Requirement for Bachelor Degree of Engineering (Software Engineering)
Department of Computer Engineering, Faculty of Engineering KASERTSART UNIVERSITY
Academic Year 2565**

Approved By:

Advisor .... **Date** ....
(Asst. Prof.Dr. Thanawin Rakthanmanon)

Co-Advisor .... **Date** ....
(Assoc. Prof. Dr. Theerawit Wilaiprasitporn)

Head of Department .... **Date** ....
(Assoc. Prof.Dr. Punpiti Piamsa-Nga)

Abstract

Put your abstract paragraph here.

Acknowledgement

Put your acknowledgement paragraph here.

Nantawat Suksirisunt
Naytitorn Chaovirachot

Table of Contents

Content **Page**

List of Tables

List of Figures

Page

Chapter 1

Introduction

1.1 Background

Electroencephalography (EEG) has become a vital tool in neuroscience, clinical diagnostics, and brain-computer interface (BCI) development. EEG measures the electrical activity of the brain through electrodes placed on the scalp, capturing neural oscillations and providing insights into cognitive functions and neural disorders. Recent advancements have driven significant research into EEG-based motor imagery (MI), where individuals imagine movements without physical execution, creating distinctive EEG patterns that can be leveraged for control and communication purposes.

1.2 Problem Statement

Despite extensive research, EEG-based MI classification faces critical challenges. Traditional supervised learning methods typically yield limited accuracy (around 50%) due to high signal variability, noise, and individual differences among subjects. Additionally, the scarcity and high cost of obtaining labeled EEG data hinder the development of robust, generalizable classification models. Consequently, there is a clear need for innovative methodologies capable of enhancing accuracy and reliability while reducing reliance on labeled data.

1.3 Solution Overview

To address these challenges, this project proposes the implementation of Self-Supervised Learning (SSL) techniques tailored specifically for EEG MI classification. SSL uses unlabeled data to learn meaningful representations through internally generated tasks (pretext tasks), significantly improving classification performance and robustness without extensive labeled datasets.

1.3.1 Features

1. Unified Preprocessing Pipeline: Standardizes and integrates multiple diverse EEG datasets, enhancing sample size and data consistency.
2. Advanced SSL Framework: Employs MixNet architecture with tailored pretext tasks designed explicitly for EEG data.
3. User-Friendly Interface: Interactive HTML/Tailwind-based analytics platform for intuitive EEG data exploration and visualization.

1.4 Target User

The primary users of this software include biomedical and neuroscience researchers, clinical practitioners, brain-computer interface developers, and the machine learning community.

- **Demographics:** Researchers and clinicians with backgrounds in neuroscience, biomedical engineering, and related fields.
- **Skill Level:** Varied technical proficiency, ranging from novice clinicians to expert researchers and software developers.
- **Industry or Domain:** Primarily neuroscience, clinical diagnostics, neurorehabilitation, and brain-computer interface development.

1.5 Benefit

The proposed SSL-based EEG MI classification system offers several key benefits:

- Enhanced accuracy (targeting above 80%) compared to traditional supervised methods.
- Reduced dependence on costly, labeled EEG datasets.
- Improved model robustness and generalizability across diverse subjects and experimental conditions.
- Intuitive interface for easy data analysis and visualization, accessible to a broad range of users.

1.6 Terminology

- **Electroencephalography (EEG):** A method for recording electrical activity of the brain using scalp electrodes.
- **Motor Imagery (MI):** The mental simulation of physical movements without actual muscle activity.
- **Brain-Computer Interface (BCI):** A direct communication pathway between the brain and an external device.
- **Self-Supervised Learning (SSL):** Machine learning methodology where models learn from unlabeled data through internally generated tasks.
- **MixNet:** A neural network architecture combining classical and modern deep-learning methods, specifically optimized for EEG classification.

Chapter 2

Literature Review and Related Work

This chapter presents related research efforts and existing solutions addressing the classification of EEG-based Motor Imagery (MI) signals. The review includes a technical comparison of frameworks using supervised and self-supervised learning (SSL), followed by a literature analysis of state-of-the-art systems. These insights help position our work within the current research landscape and identify key gaps we aim to address.

2.1 Competitor Analysis

To evaluate current EEG MI classification frameworks, we analyzed their model architecture, feature extraction approaches, performance, learning strategies, and support for multi-dataset generalization. The results are summarized in Table 2.1.

Observations:

- **TRIPNet** delivers the highest reported accuracy and excels in subject-independent tasks through tailored pretext tasks.
- **MixNet** balances classical signal processing with deep learning but is limited by manual preprocessing across datasets.
- **EEG-SSL** offers excellent scalability and dataset handling via BIDS format, with slightly lower classification accuracy.
- **EEGNet**, while lightweight and widely used as a baseline, lacks generalization and modern learning strategies.

Our proposed framework aims to combine the strengths of these systems—accurate feature extraction, dataset scalability, and robustness—into a self-supervised model that is also user-accessible through a visual interface.

Table 2.1: Technical Comparison of EEG MI Classification Frameworks

Framework	**Model Architecture**	**Feature Extraction**	**Accuracy**	**Learning Type**	**Multi-Dataset Support**
MixNet (2024)	Multi-task Autoencoder with FBCSP and Triplet Loss	Filter Bank CSP, latent space embeddings	85.4% (BCIC2a)	Supervised + Metric Learning	Partial (manual alignment)
TRIPNet (2024)	Triple-path CNN (spectral, spatial, temporal) + Statistician Module	Pretext-task-driven SSL (band prediction, spatial noise, temporal trend)	88.6% (OpenBMI)	Self-Supervised	Full (4 paradigms)
EEG-SSL (2024)	Modular BIDS-compatible SSL framework	Raw EEG + transformations (crop, mask, time-shift)	80.1% (Resting-state)	Self-Supervised	Full (BIDS format)
EEGNet (2018)	Depthwise Separable ConvNet (compact CNN)	Raw time-domain signals	69.5% (BCIC2a)	Supervised	No (Single-dataset)

2.2 Literature Review

MixNet (Autthasan et al., 2024): MixNet (Autthasan et al., 2024) is a novel framework that integrates Filter Bank Common Spatial Patterns (FBCSP) with a deep metric learning structure, combining autoencoders and triplet loss to optimize EEG-based Motor Imagery (MI) classification. This approach is designed to perform well in both subject-dependent and subject-independent scenarios, addressing challenges related to subject-specific learning and multi-task learning. By utilizing spectral-spatial signal integration, adaptive gradient blending, and multi-task autoencoders, MixNet demonstrates improved classification performance compared to previous models such as MIN2Net, EEGNet, and DeepConvNet.

However, a limitation of MixNet is its reliance on manual dataset alignment, which hinders its scalability and practical use in real-world applications. The lack of automated preprocessing tools further limits its usability, as it requires careful manual intervention for dataset preparation. Despite these challenges, MixNet stands out for its efficiency, accuracy, and potential for deployment in low-density EEG systems, making it suitable for wearable devices and mobile healthcare applications.

While MixNet outperforms state-of-the-art methods in terms of F1-score and computational efficiency, especially in the low-density EEG dataset BCIC IV2b, the need for manual preprocessing underscores the need for further development in automated tools to streamline its application in practical scenarios.

TRIPNet (Ko et al., 2024): TRIPNet (Ko et al., 2024) introduces a deep learning framework designed specifically for EEG signals, utilizing a three-pathway architecture that captures spectral, spatial, and temporal characteristics of the data. The model leverages custom self-supervised tasks for pretraining, enabling it to learn meaningful EEG representations without the need for costly human-centered annotations. These self-supervision strategies are carefully crafted to be neurophysiologically plausible, ensuring that the network can learn EEG features effectively while remaining consistent with neurobiological principles.

A key innovation in TRIPNet is the integration of the statistician module, which adaptively normalizes the input features based on their statistics. This module plays a crucial role in controlling the variability of the EEG signals, enhancing the stability and performance of the model across different conditions. The inclusion of this module also ensures that the framework is well-suited to generalize across various EEG paradigms, making it applicable to a wide range of EEG-based applications.

Through extensive empirical experiments, the authors validate the framework's effectiveness, highlighting the individual contributions of each component, such as the self-supervision methods and the statistician module. These results demonstrate that TRIPNet excels in representing EEG signals by efficiently managing the signal variability and capturing neurophysiologically meaningful patterns, especially through its stopped band prediction pretext task. This task involves band-stop filtering of predefined EEG frequency ranges ($\delta, \theta, \alpha, \beta, \gamma$ bands) to represent neural oscillations. However, it should be noted that the consistent use of predefined frequency ranges may complicate spectral representation learning due to subject-specific variations and environmental influences.

Looking ahead, the authors suggest that future research could explore hyperspectral synthesis and augmentation of EEG samples in a more rigorous image-processing manner, which could lead to novel pretext tasks for self-supervised learning. This would potentially eliminate the reliance on prior knowledge of frequency ranges and further enhance the flexibility of the model in diverse environments.

In conclusion, TRIPNet offers a significant step forward in EEG representation, demonstrating its ability to provide reliable decision-making with zero-calibration data and a user-friendly system. Despite its promising results, there is still room for improvement, particularly in exploring new pretext tasks for self-supervision.

EEG-SSL (Truong et al., 2024): The proposed Self-Supervised Learning (SSL) framework effectively addresses the challenges of large-scale EEG analysis by leveraging BIDS-formatted datasets. This framework facilitates efficient data preprocessing, segmentation, and encoding, overcoming the variability of channel configurations across different datasets. By incorporating self-supervised transformations, such as masking and time-shifting, the framework enables robust representation learning without requiring labeled data. This approach demonstrated its potential through experiments like the Relative Positioning (RP) task, where the model successfully learned meaningful representations, with training loss curves validating its learning capability.

A key feature of the framework is its modularity and flexibility, allowing researchers to customize experiments based on their needs. It accepts various BIDS-formatted EEG datasets, enabling it to process datasets of different sizes, modalities, and configurations. The SSL tasks, such as temporal contrastive learning or masked predictive coding, and model architectures (e.g., CNNs, Transformers) are fully customizable to suit specific research objectives. Despite this flexibility, the framework maintains fixed components, such as the hierarchical parsing of BIDS datasets, standardized preprocessing pipelines, and a core training loop. This balance ensures reproducibility and scalability for large-scale experiments, supporting rapid experimentation and big data workflows.

While the SSL framework is suitable for a variety of EEG paradigms, it is better optimized for tasks like resting-state EEG rather than task-specific classification (e.g., Motor Imagery (MI)). The framework's design also addresses the challenge of channel harmonization, where inter-channel relationships are learned directly from raw data at the model level, providing flexibility for different EEG channel configurations.

In terms of future directions, the framework shows promise for downstream tasks, such as cognitive state classification, anomaly detection, and clinical diagnosis. However, further research is needed to compare its performance against supervised models trained on labeled data. The key advantage of SSL lies in its ability to leverage vast amounts of unlabeled EEG data, which is typically unavailable for labeled data tasks. This could lead to better generalization, especially in scenarios with limited labeled data or high inter-subject variability. Future studies should benchmark this approach to quantify the impact of pretraining with SSL on downstream task performance.

Overall, this framework's ability to process large-scale EEG data with minimal human annotation, while providing a flexible and customizable architecture, positions it as a strong foundation for future neuroinformatics research.

Research Gap: While previous works such as MixNet, TRIPNet, and the Self-Supervised Learning (SSL) framework contribute strong models and scalable frameworks for EEG analysis, no existing approach fully integrates multi-dataset SSL training, efficient Motor Imagery (MI) task handling, and user accessibility within a single system. MixNet excels in subject-independent MI classification but is limited by manual dataset alignment and preprocessing, which hinders scalability and usability in real-world scenarios. Similarly, TRIPNet utilizes a robust three-pathway architecture for spectral, spatial, and temporal features but is more suited for general EEG applications rather than task-specific classifications like MI. The SSL framework, while highly flexible and modular, is optimized for resting-state EEG tasks and lacks direct support for task-specific classifications like MI, making it less suitable for time-sensitive applications like motor imagery.

This research gap forms the foundation of our proposed work, which aims to develop a unified system that supports multi-dataset SSL training, efficient MI task handling, and user accessibility. Our system will integrate the strengths of these previous works while addressing their limitations, offering a more comprehensive and adaptable solution for EEG-based applications.

Chapter 3

Requirement Analysis

This chapter outlines the analytical framework used to determine the needs, roles, and interactions involved in the research project. It defines the stakeholders, scenarios of system usage, and initial user interface concepts for interacting with the EEG-based Motor Imagery (MI) classification platform.

3.1 Stakeholder Analysis

Stakeholders in this research include individuals and groups who interact with, contribute to, or benefit from the system developed for EEG-based MI classification using self-supervised learning.

- **Research Students:** Graduate and undergraduate students conducting EEG or BCI-related research who require tools for preprocessing, modeling, and visualizing EEG signals.

- **Academic Advisors and Mentors:** Supervisors who guide the project, evaluate methodologies, and assess results.

- **BCI Researchers:** Specialists in Brain-Computer Interface systems who are interested in integrating the framework into larger research workflows.

- **Collaborative Institutions:** External collaborators or co-authors who may contribute datasets, methods, or validation tools.

3.2 User Stories

User stories are derived from research scenarios rather than business objectives. These help define how each type of stakeholder would interact with the system.

- **As a research student**, I want to upload raw EEG datasets from various MI tasks so that I can run preprocessing and train SSL models.
- **As an academic advisor**, I want to review model performance summaries and confusion matrices so that I can evaluate the effectiveness of the SSL training.
- **As a BCI researcher**, I want to compare the accuracy of traditional supervised models with the SSL framework so that I can determine its viability for cross-subject generalization.
- **As a collaborator**, I want to integrate new datasets into the framework pipeline so that I can validate the model across different data sources.

3.3 Use Case Diagram

The system supports various interactions between researchers and the components of the EEG SSL framework. Below is a use case diagram representing primary research activities.

3.4 Use Case Model

This section outlines key use cases using a brief or casual structure appropriate for a research prototype.

Use Case: Upload EEG Dataset

Type: Brief **Actor:** Researcher **Goal:** To upload a raw EEG dataset for preprocessing and model training.

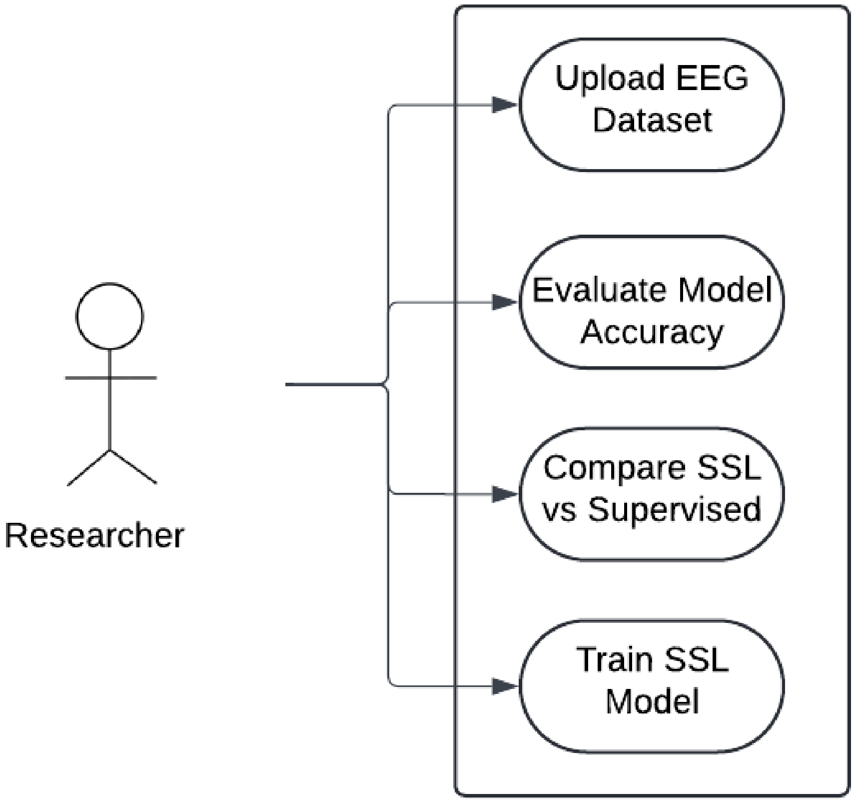

Figure 3.1: Use Case Diagram for EEG SSL Framework

Use Case: Train SSL Model

Type: Casual **Primary Actor:** Researcher **Preconditions:** Dataset has been preprocessed. **Main Flow:**

1. User selects the preprocessed dataset.
2. User configures SSL model parameters.
3. System begins training and logs progress.
4. Upon completion, results are visualized and saved.

Use Case: Evaluate Model Accuracy

Type: Casual **Actor:** Researcher **Goal:** To view performance metrics of the trained model.

Use Case: Compare SSL vs Supervised

Type: Brief **Actor:** Researcher **Goal:** Compare results of SSL model with a baseline supervised model.

3.5 User Interface Design

The user interface is designed to support interaction with the model training pipeline, prediction, evaluation, and model comparison. It aims to simplify research workflows such as dataset selection, preprocessing, training configuration, and performance analysis.

The UI is divided into multiple functional views, as shown below:

The UI is implemented using HTML and Tailwind CSS for a clean, responsive layout. Each screen is tailored to specific research tasks, ensuring usability for academic researchers with limited software engineering experience.

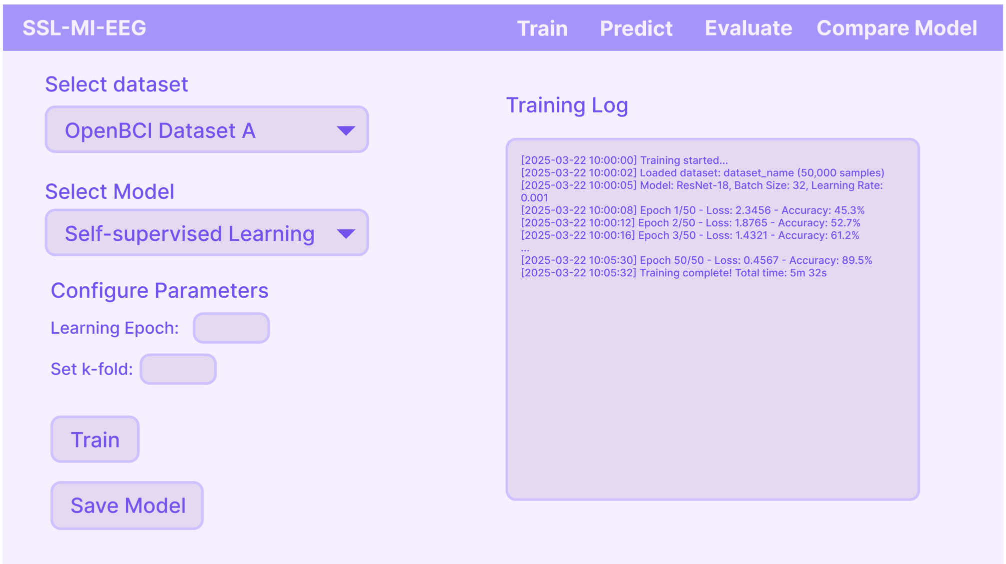

Figure 3.2: Training Interface – Dataset selection, SSL configuration, and training log

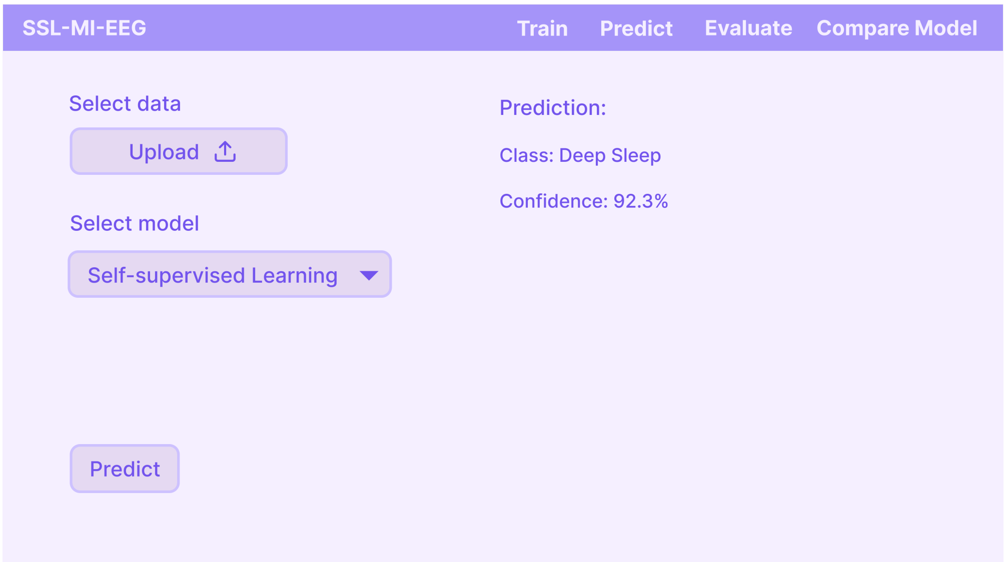

Figure 3.3: Prediction Interface – Uploading input data and viewing output class with confidence

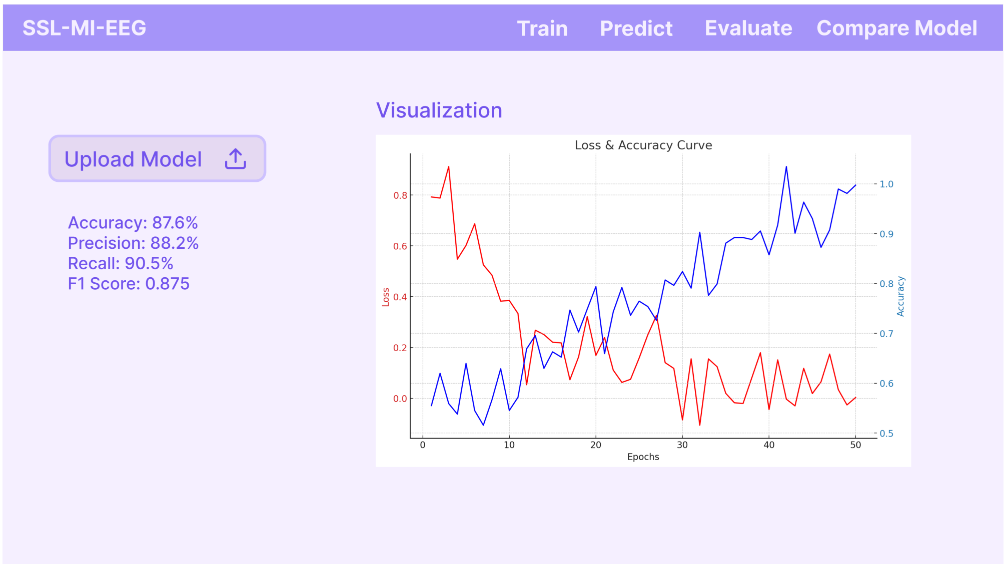

Figure 3.4: Evaluation Interface – Upload trained model and view performance metrics and curve

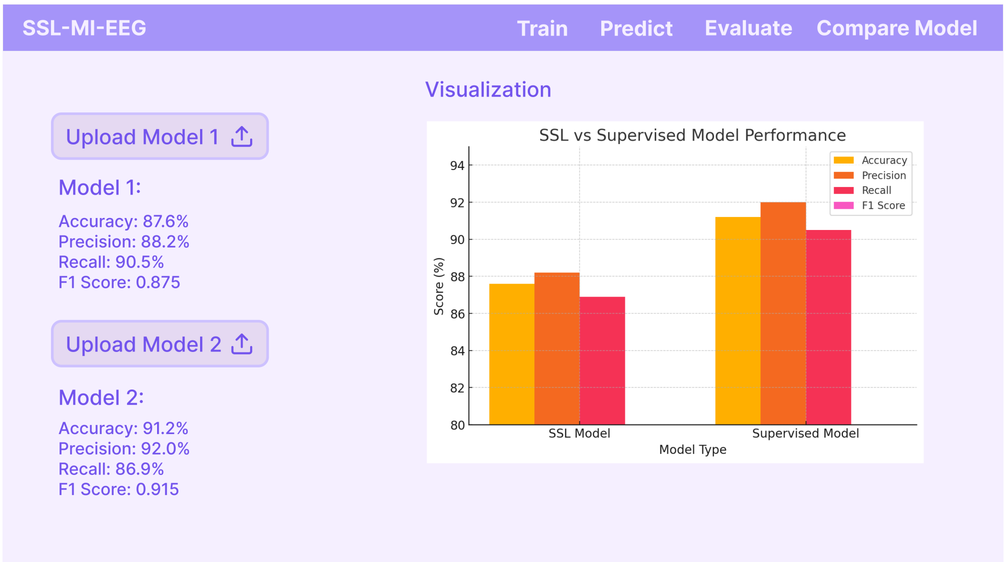

Figure 3.5: Comparison Interface – Visual comparison between SSL and Supervised models

Chapter 4

Software Architecture Design

This chapter describes the software architecture design of the EEG-based Motor Imagery (MI) classification system using Self-Supervised Learning (SSL). The system is designed to support research workflows, including dataset integration, preprocessing, SSL model training, evaluation, and visualization. Unified Modeling Language (UML) diagrams are used to represent structural and behavioral aspects of the system architecture.

4.1 Domain Model

The domain model captures core research entities and their relationships in the context of EEG analysis and SSL-based classification. These entities are grounded in scientific workflows rather than business operations.

- **EEGDataset:** Represents structured EEG data collected from experiments, including metadata and raw signals.
- **Preprocessor:** Responsible for applying signal filtering, artifact removal, and standardization.
- **SSLModel:** Represents the core self-supervised learning module, including its architecture and training process.
- **EvaluationReport:** Contains metrics (accuracy, F1-score, ROC-AUC) derived from model evaluation.
- **Researcher:** A user interacting with the system to upload data, configure models, and interpret results.

domain-model-eeg.png

Figure 4.1: Domain Model for EEG SSL Research Framework

4.2 Design Class Diagram

The class diagram provides a technical breakdown of system components. Classes are grouped by functionality into modules such as Data Handling, Model Training, and Evaluation.

- **DataLoader:** Loads EEG datasets and parses metadata.
- **PreprocessingPipeline:** Applies band-pass filters, ICA, normalization, and data segmentation.
- **SSLTrainer:** Implements self-supervised learning algorithms and manages training sessions.
- **ModelEvaluator:** Validates model outputs, computes evaluation metrics, and stores performance data.
- **Visualizer:** Displays results through plots and summaries for analysis.
- **UserInterface:** Connects researchers with backend components for configuring and visualizing workflows.

4.3 Sequence Diagram

The following sequence diagram illustrates the typical interaction for training a new SSL model on a selected EEG dataset:

Scenario: SSL Model Training Workflow

1. Researcher selects and uploads EEG data.
2. The system applies preprocessing to the dataset.
3. SSL model is initialized with pretext task configurations.
4. Training begins, and progress is logged.
5. Evaluation metrics are computed and visualized.

design-class-diagram-eeg.png

Figure 4.2: Design Class Diagram for EEG SSL System

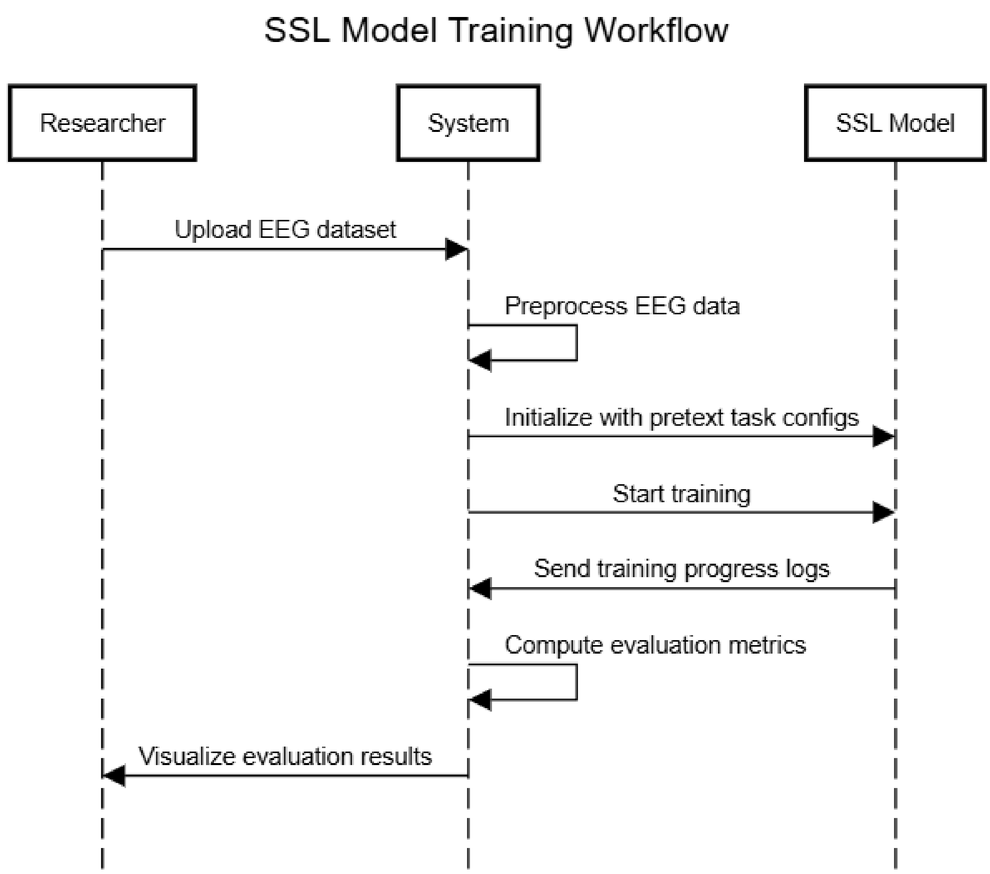

Figure 4.3: Sequence Diagram – SSL Model Training Flow

4.4 Algorithm

The following pseudocode outlines the simplified self-supervised learning training loop used in the project. It includes feature transformation, encoding, and contrastive loss calculation.

```
Algorithm 1: Self-Supervised EEG Feature Learning
Input: Unlabeled EEG samples X
Output: Learned representations Z

for epoch in range(num_epochs):
    for batch in X:
        x1, x2 = Augment(batch)
        z1 = Encoder(x1)
        z2 = Encoder(x2)
        loss = ContrastiveLoss(z1, z2)
        UpdateWeights(loss)
```

4.5 AI Component

The AI module in this system is built around a Self-Supervised Learning framework based on a dual-view contrastive encoder architecture. It supports flexible pretext tasks and multiple EEG datasets.

- **Encoder Network:** A convolutional network tailored to EEG signal structures (e.g., EEGNet or custom CNNs).
- **Pretext Tasks:**
  - Spectral Band Masking
  - Temporal Shuffling
  - Channel Dropout
- **Loss Function:** InfoNCE or triplet contrastive loss adapted to time-series EEG embeddings.
- **Fine-tuning Strategy:** Optionally transfers learned representations to downstream MI classification using supervised labels.

Chapter 5

Software Development

5.1 Software Development Methodology

<TIP: Describe your software development methodology in this section. />

5.2 Technology Stack

<TIP: Describe your technology stack here. See the following example from ThaiProgrammer.org />

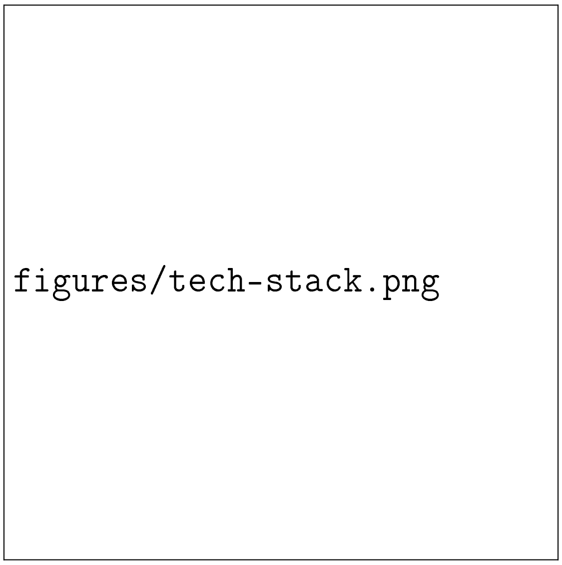

Figure 5.1: Example technology stack

5.3 Coding Standards

<TIP: Describe your coding standard for this project here. />

5.4 Progress Tracking Report

<TIP: Show that you have been working on this project overtime. It can be in the form of a burndown chart or a contribution graph from GitHub./>

Chapter 6

Deliverable

6.1 Software Solution

<TIP: Share a link to your Github repository. Showcase screenshots of the application and briefly describe each page here. />

6.2 Test Report

<TIP: Describe how you test your project. Place a test report here. If you use continuousintegration and deployment (CI/CD) tools, describe your CI/CD method here. />

Chapter 7

Conclusion and Discussion

<TIP: Discuss your work here. For example, you can discuss software patterns that you use in this project, software libraries, difficulties encountered during development, or any other topic. />

Reference

Bibliography

[1] Overleaf, "Learn latex in 30 minutes," https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes.

Appendix A

Appendix A: Example

<TIP: Put additional or supplementary information/data/figures in appendices. />

Appendix B

Appendix B: About LaTeX

LaTeX (stylized as LaTeX) is a software system for typesetting documents. LaTeX markup describes the content and layout of the document, as opposed to the formatted text found in WYSIWYG word processors like Google Docs, LibreOffice Writer, and Microsoft Word. The writer uses markup tagging conventions to define the general structure of a document, to stylize text throughout a document (such as bold and italics), and to add citations and cross-references.

LaTeX is widely used in academia for the communication and publication of scientific documents and technical note-taking in many fields, owing partially to its support for complex mathematical notation. It also has a prominent role in the preparation and publication of books and articles that contain complex multilingual materials, such as Arabic and Greek.

Overleaf has also provided a 30-minute guide on how you can get started on using LaTeX. [**?**]