

Ruchira Praveen Chamal Dedigamage

**A SPIKING NEURAL NETWORK BASED
PIPELINE FOR MOTOR IMAGERY EEG
CLASSIFICATION**

Faculty of Medicine and Health Technology
Master's Thesis
November 2024

ABSTRACT

Author : A Spiking Neural Network Based Pipeline for Motor Imagery EEG Classification
Master's Thesis
Tampere University
Biotechnology and Biomedical Engineering
November 2024

Motor imagery based electroencephalogram signals have been used for developing brain-computer interfaces because of their ease of execution and low-risk profile. These systems have achieved reliable performances with the emergence of machine learning. However, these machine-learning approaches are not focused on optimizing energy efficiency. Yet, energy efficiency is crucial for implementing portable devices since the continuous operational time depends on it.

The spiking neural network based classifiers are suitable and promising for achieving energy efficiency compared to the conventional machine learning algorithms while maintaining the same level of accuracy. Unlike traditional systems, where all computing units operate at each clock cycle, spike based systems process information asynchronously only when there are sufficient inputs to activate neurons; thus, these algorithms can be utilized on specialized hardware chips which can process spikes for achieving higher energy efficiency.

The primary objective of this thesis was to develop a spiking neural network based pipeline that incorporates a spatial filtering method called the common spatial pattern. Since motor imagery tasks are highly correlated with unique spatial areas, the common spatial pattern method can improve the separability between the motor imagery classes by enhancing the spatial features of the electroencephalogram channels. With a higher separability, input signals can be effectively encoded into spikes without losing the critical information for classification tasks.

To achieve this objective, three types of pipelines (rate coding, latency coding and delta coding) were implemented to classify the labelled ("left hand" and "right hand") motor imagery signals in a widely used open electroencephalogram dataset. In addition, to evaluate the contribution of the common spatial pattern method comparatively, another set of pipelines was implemented with the same steps, excluding the use of the common spatial pattern method. By using each pipeline, the classifier models were trained on different datasets (five individual and one multi-subject dataset) using a surrogate gradient decent based back propagation algorithm. 5-fold cross-validation was used to validate the performance of each pipeline model on each dataset.

Each trained model has demonstrated acceptable classification accuracy with the common spatial pattern method compared to conventional methods. Among them, the rate coding pipeline models recorded the highest mean accuracies on training and testing datasets, while latency coding pipeline models reflected the best energy efficiency. Results from the pipelines which do not utilize the common spatial pattern showed a drastic decrease in performance, highlighting the effectiveness of spatially filtered signals for the overall classification process. These findings direct future research to explore spike-friendly spatial filtering techniques and novel spiking neural network architectures to further enhance classification accuracy and energy efficiency in motor imagery electroencephalogram classification.

Keywords: Motor imagery (MI), electroencephalogram (EEG), spiking neural networks (SNN), Common Spatial Patterns (CSP), Spike encoding

The originality of this thesis has been checked using the Turnitin Originality Check service.

USE OF AI IN THESIS

I have utilised AI tools in my thesis:

- No
- Yes

The AI tools utilised in my thesis and their purposes are described below:

Names and versions of AI tools: ChatGPT 4o, ChatGPT 4o1-preview, Grammarly

Purpose of using AI tools: I have used the above-mentioned AI tools to find relevant resources and refine my texts to ensure correct grammar and better clarity.

Sections where AI tools were used:

1. INTRODUCTION
2. BACKGROUND
3. LITREATURE REVIEW
4. MATERIALS AND METHODS
5. RESULTS
6. DISCUSSION
7. CONCLUSION

I acknowledge that I am fully responsible for the entire content of my thesis, including the parts generated by AI, and accept accountability for any violations of ethical standards in publications.

PREFACE

This thesis marks the conclusion of my master's degree in biomedical sciences and engineering. Choosing this topic was a significant decision, one that required taking a considerable risk due to its innovative approach and uncertainty. However, I was determined to take on this challenge, and completing this work has been both challenging and fulfilling for me. This is a result of dedication, guidance, and support. So, I owe gratitude to many individuals who were with me throughout this journey.

First and foremost, I am deeply grateful to my supervisors, Dr. Tuomo Mäki-Marttunen and Docent, Marja-Leena Linne, for their invaluable guidance and encouragement throughout this journey. I also extend my gratitude to all the other members of the Computational Neuroscience (CNS) group as well. The collaborative environment of the group greatly enriched my research perspective on computational neuroscience. Additionally, I thank Tampere University for its support and resources for having the best student experience in my life.

I also want to express my heartfelt thanks to my master's level classmates and my beloved family, who have been my source of strength and support during this journey. Their encouragement has been influential in helping me accomplish both my thesis and my degree.

Finally, I want to express my profound gratitude to the free education systems in Finland and Sri Lanka and to the taxpayers committed to maintaining them. Their dedication to providing equal rights to education for everyone has had a significant impact on everyone's academic journey and has made the world a better place.

Tampere, 29 November 2024

Dedigamage Ruchira Praveen Chamal

CONTENTS

1. INTRODUCTION	1
1.1 Overview of Thesis Topic.....	1
1.2 Objectives.....	3
1.3 Thesis Structure	3
2. BACKGROUND	4
2.1 Principles of EEG Signals	4
2.2 Motor Imagery (MI) Tasks.....	6
2.3 EEG Signal Analysis with Neural Networks	6
2.4 Spike Driven Neural Processing	8
3. LITERATURE REVIEW.....	10
3.1 Performance of Second Generation ANNs as MI-EEG Classifiers	10
3.2 SNNs in MI-EEG Classification.....	12
3.3 Spike Encoding, Learning Rules and types of SNN architecture	13
4. MATERIALS AND METHODS	15
4.1 Dataset.....	15
4.2 Used Software Libraries	16
4.3 Implementation of Pipelines.....	17
4.3.1 Preprocessing Methods	17
4.3.2 Spike Encoding Schemes	21
4.3.3 SNN based Classifiers	25
4.4 Pipelines without the CSP algorithm	28
5. RESULTS	29
5.1 Preprocessing of Raw MI-EEG Signals.....	29
5.2 Effectiveness of Spike Encoding Schemes.....	32
5.3 Learning Curves	34
5.4 Activations of SNN Layers in Testing Phase.....	35
5.5 Performance Measures	38
6. DISCUSSION.....	40
6.1 The Potential of CSP for SNN based MI-EEG classifiers	40
6.2 Evaluation of each Pipeline.....	41
6.3 Impact of the CSP method.....	43
6.4 Limitations	44
6.5 Future Directions	45
7. CONCLUSIONS.....	46
8. REFERENCES	48
APPENDIX A : Determining the time of the next spike in LIF neurons	53
APPENDIX B : Hyperparameters.....	55

LIST OF SYMBOLS AND ABBREVIATIONS

1d-AX	One dimension-aggregate approximation
ANN	Artificial Neural Network
BCE	Binary Cross Entropy
BCI	Brain Computer Interface
BCIC	Brain Computer Interface Competition
BPF	Band-Pass Filter
CE	Cross Entropy
CNN	Convolutional Neural Network
CSP	Common Spatial Patterns
CWT	Continuous Wavelet Transform
DBN	Deep belief network
DWT	Discrete Wavelet Transform
DC	Direct Current
EEG	Electroencephalogram
EEMD	Ensemble Empirical Mode Decomposition
EM	Electromagnetic Waves
ERD	Event-Related Desynchronization
ERS	Event-Related Synchronization
FT	Fourier Transform
fMRI	functional Magnetic Resonance Imaging
fNIRS	functional Near Infrared Spectroscopy
LDA	Linear Discriminant Analysis
LIF	Leaky Integrate and Fire
LSTM	Long Short-term Memory
MI	Motor Imagery
ML	Machine Learning
PET	Positron Emission Tomography
PSD	Power Spectral Density
RNN	Recurrent Neural Network
SAE	Stacked Auto Encoder
SNN	Spiking Neural Networks
SNR	Signal-to-Noise Ratio
SMA	Supplementary Motor Areas
STFT	Short Time Fourier Transform
SVM	Support Vector Machine

1. INTRODUCTION

This thesis mainly suggests an innovative approach to classify motor imagery (MI) electroencephalogram (EEG) signals with spiking neural networks (SNNs) by incorporating common spatial patterns (CSP). The outcome inspires future work to explore spike-friendly spatial filters for MI-EEG signal classification tasks, providing a solid approach.

Firstly, this chapter introduces MI-EEG signals based brain computer interfaces (BCI). After that, it explains how the SNNs can improve the energy efficiency of BCIs and the impact of building energy efficient BCIs. At the end of the chapter, the objective of this work and the thesis structure are included to provide a concise outline.

1.1 Overview of Thesis Topic

EEG is a non-invasive neuroimaging method used to explore the brain's electrical activities. It is commonly used in clinical diagnostics and neuroscientific research for obtaining the effects of brain functions underlying human cognition and behaviors [1], [2]. Not only because of the non-invasiveness, EEG has been widely used for studying neural processes due to its high temporal resolution and comparatively low cost (both hardware production and operating cost) [3].

Because of these advantages, EEG has been widely utilized in specific studies such as MI studies. MI studies are a specific domain of brain sciences which focuses on the brain functions related to the imagination of movement without physical execution. Different MI actions correlated with different EEG activity patterns which are generated in certain brain areas [4]. Hence, MI-EEG signals can be used to translate MI actions to command signals. The ease of rendering MI activities makes MI-EEG a viable candidate for BCI applications (e.g., Motor rehabilitation therapies [5], assistive device controlling [6], gaming [7]) to produce command signals. Figure 1 illustrates the common steps of MI-EEG based BCI systems which produce commands for the applications.

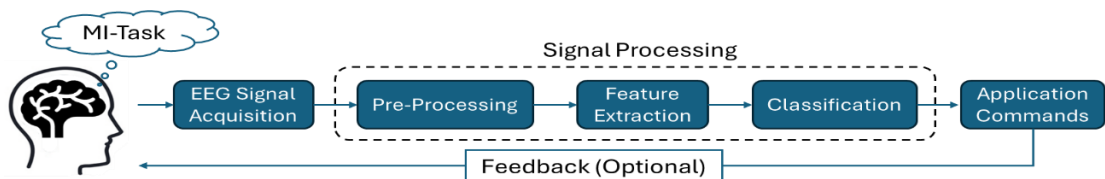


Figure 1. Common pipeline of MI-EEG based BCI systems.

Despite the potential of MI-EEG signals for BCIs, the processes of classification pose significant challenges due to the intrinsic properties of the EEG. The primary challenges arise from high noise susceptibility, high signal pattern variability (both within subjects and across subjects), and the complex temporal dynamics of EEG signals due to diverse brain activities. To overcome these challenges and to implement reliable and accurate methods for MI-EEG signal classification, specialized methods are essential [8].

Conventional EEG signal analysis methods employ time, frequency, and spatial domain features to implement MI-EEG classification models. Wavelet transformations and time frequency spectrograms are examples of using time and frequency domain features, and CSP is an example of using spatial domain features [9]-[11].

Since EEG signal patterns related to MI tasks are generated from spatially different brain regions, utilizing spatial features is more applicable to classify MI-EEG signals. Hence, the CSP method is one of the most suitable candidates for feature extraction. In brief, the CSP method enhances the EEG signals' spatial features by maximizing the signals' variance corresponding to the target classes while minimizing the variance for other classes [12].

In previous studies, the extracted MI-EEG features have been classified by using widely employed machine learning (ML) algorithms. Among these, Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA) have been widely adopted for their effectiveness in handling the complexities and non-linearities within EEG data [13],[14]. However, the high variability and the high non-stationary nature of EEG signals make it difficult to adapt and achieve reliable accuracy performance within a subject and across the subject. This necessitates alternative methods with high adaptability and generalization capability [15]. Neural networks are ideal candidates for satisfying those requirements. However, conventional neural networks are often not optimal in terms of energy efficiency.

As a neural network variant, SNNs are also capable of achieving both adaptability and generalization with improved energy efficiency. Like conventional neural networks, SNNs achieve the handling of the variability and the non-stationarity of EEG signals with (synaptic) weights dependent neural network architectures [16], [17]. However, unlike other conventional neural network types, SNNs mimic how biological neurons send information through discrete spikes. This bio-inspired, spiking approach allows SNNs to process temporal patterns more energy efficiently than conventional neural networks that work with continuous values. However, the true energy efficiency of SNNs can only be achieved when they are used spike based hardware processing units, also known as

neuromorphic hardware systems. These hardware systems are designed as networks of neurons and propagate information from inputs to outputs as electric spikes. While conventional systems consume energy in every clock cycle, neuromorphic systems consume energy only when there is enough potential to produce spikes [18].

Thus, it is clear that using SNNs as the classification algorithms for MI-EEG based BCI systems makes it possible to achieve improved energy efficient models while maintaining reliability. Previous SNN based MI-EEG studies have achieved accuracy levels similar to or higher than those of conventional MI-EEG classification methods, proving the reliability of SNN models for classifying MI-EEG signals [19]-[24]. However, SNNs require effectively encoded spike trains to represent MI-EEG signals to reach that level of accuracy. This thesis proposes an approach incorporating the CSP method for effective spike encoding, aiming to achieve reliable MI-EEG classification accuracy with SNN models.

1.2 Objectives

The primary objectives of this thesis are as follows:

Objective 1 : To design and implement a classification pipeline for motor imagery EEG signals using SNNs that integrate CSP spatial filtering techniques.

Objective 2 : To evaluate the effectiveness of the developed pipeline by analyzing its performance with respect to a motor imagery EEG dataset.

In this thesis, the goal of using SNN is to improve the energy efficiency of the MI-EEG classification process, and the goal of using the CSP method is to boost classification accuracy. Evaluating the effectiveness of the pipeline ensures the reliability of the approach and inspires future work to utilize spatial filtering for MI-EEG applications.

1.3 Thesis Structure

This thesis is organized into seven chapters. Following this introduction (Chapter 1), Chapter 2 provides background on EEG signals, MI tasks and the use of neural networks focusing on SNNs. Chapter 3 provides a comprehensive literature review of conventional and SNN based studies on classifying MI-EEG signals. The methodology adopted for developing the SNN based pipelines is detailed in Chapter 4, including dataset, preprocessing, and model training. Chapter 5 reports the transformations of each stage of pipelines and the overall results. Comprehensive discussions of each pipeline and the techniques employed have been included in Chapter 6 to interpret the obtained results and outline potential future work. The thesis concludes with Chapter 7, summarizing the key implications and contribution of the study.

2. BACKGROUND

This chapter provides an in depth overview of EEG signals, MI tasks, and SNNs. By reviewing relevant literature and recent advancements, this chapter delivers a basis for understanding the potential of SNNs in classifying MI-EEG signals.

2.1 Principles of EEG Signals

As mentioned in Section 1.1, EEG is a non-invasive neuroimaging technique that records electrical changes on the scalp resulting from cellular level activities in the brain. The communication activities which happen within the neuronal cells move charges from one place to another depending on the synaptic currents [25]. This creates electrical dipoles (see Figure 2) between neurons' bodies and apical dendrites. When a large population of neurons' synaptic communications occur simultaneously, the summed-up dipole effects can induce electromagnetic (EM) waves that can penetrate the surface of the scalp. As an effect, the potential of the scalp changes accordingly. The EEG electrodes record these potential changes as time varying signals [26], [27].

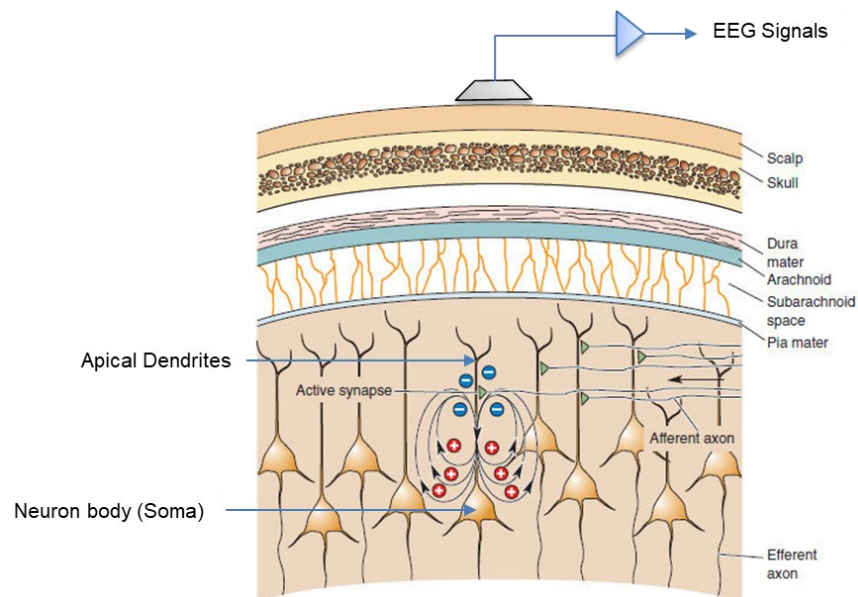


Figure 2. Generation of EEG signals. Image is adapted from [19].

EEG signals have a higher temporal resolution compared to the other neuroimaging techniques. This is mainly because of the short reaction time of the EEG systems and

the travelling speed of the induced EM waves (EEG signals) [28]. Current EEG acquisition systems can measure and record potential changes on the scalp on a millisecond or a microsecond scale [29].

However, EEG signals' spatial resolution is inherently low compared to other neuroimaging techniques like functional magnetic resonance imaging (fMRI) [30] and functional near-infrared spectroscopy (fNIRS) [31]. The major cause of this limitation is the diffuse nature of electrical fields as they travel through the insulation barriers such as brain tissues, cerebrospinal fluid, skull, and scalp (see Figure 2) before reaching the electrodes. Hence, the signal detected at the scalp represents the activity of thousands or millions of neurons over a relatively large area. This makes it difficult to find the exact source of brain activity. With multiple electrodes, EEG systems can overcome this limitation up to some extent and localize the region of the sources by using spatial filters [32]. Figure 3 shows the performance of EEG compared to other neuroimaging techniques in terms of temporal and spatial resolutions.

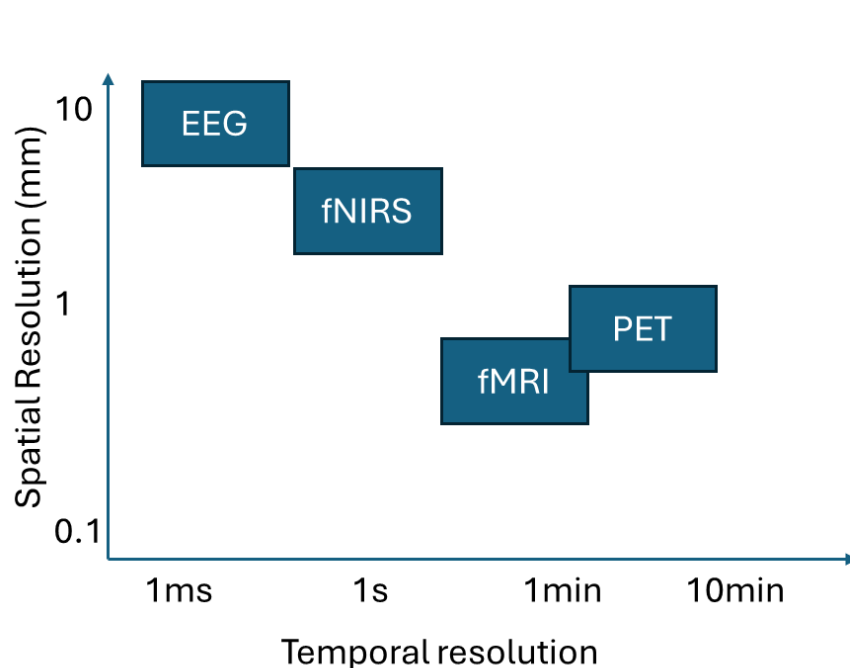


Figure 3. Illustration of temporal and spatial resolutions of different neuroimaging techniques, adapted from [33]. EEG – Electroencephalogram, fNIRS – Functional Near Infrared Spectroscopy, fMRI – Functional Magnetic Resonance Imaging, PET – Positron Emission Tomography.

Apart from the lower spatial resolution, maintaining a good signal-to-noise ratio (SNR) in EEG signals is also challenging, mainly because of their low amplitude range. Hence, EEG signals are susceptible to various types of noise [34] and artefacts, such as muscle movements, eye blinks, and external electromagnetic interference [35]. This susceptibil-

ity demands preprocessing steps to increase the SNR, otherwise enhancing the significant information. Common preprocessing steps include frequency domain filtering to remove high frequency noise components and direct current (DC) voltage offset. In addition, various types of filtering techniques, such as adaptive filtering techniques, optimal filtering techniques, and wavelet based filtering techniques (continuous wavelet transformation [CWT], discrete wavelet transformation [DWT]), are employed to reject artefacts and exclude signal components from unrelated sources depending on the application [36].

2.2 Motor Imagery (MI) Tasks

MI tasks are a type of cognitive task which refers to the mental simulation of movement without physical execution. This process activates neural pathways in the brain similar to the actual performance during the movement, excluding the pathways responsible for motor execution [37]. Studies have shown that this mental simulation can produce distinguishable patterns in EEG signals corresponding to different voluntarily imagined movements [38], [39]. Hence, MI tasks are perfect candidates for the development of BCI technologies intended to produce control commands.

The neural executions of both MI tasks and actual motor tasks happen within the motor cortex. Hence, the motor cortex is a key region in the brain when analyzing the MI-EEG. [40]. The motor cortex is located in the frontal lobe and is divided into three areas: 1.) The primary motor cortex, 2.) the premotor cortex, and 3.) supplementary motor areas (SMA) [41]. While the primary motor cortex is directly involved in the execution of movement, the premotor cortex and SMA are involved in the planning and preparation of movements. During motor imagery, these areas show activation patterns that closely match the patterns observed during the actual motor movements [42], [43]. These unique electrical activity patterns in the cortex generate EEG signals on the scalp, which can be detected as event-related desynchronization (ERD) and event-related synchronization (ERS) patterns, which mainly lie within the mu ($\sim 8 - \sim 13$ Hz) and beta ($\sim 14 - \sim 30$ Hz) frequency bands. These signals exhibit transient oscillations, highlighting the dynamic nature of the brain's activity during motor imagery [44]. This synchrony allows us to pinpoint the timing of these activities, which is highly beneficial for implementing reliable BCI systems.

2.3 EEG Signal Analysis with Neural Networks

As mentioned in Section 1.1, the complex and dynamic nature of brain activities and the way EEG captures them necessitates the use of complex non-linear algorithms, such as

neural networks, to analyze and categorize MI EEG signals in the context of BCIs [40], [41], [42]. Traditional linear models often fall short of capturing the intricate patterns and non-linear relationships inherent in EEG data [43]. However, Neural networks are well-suited for decoding MI-EEG signals because of their ability to learn high-dimensional relationships and flexibility for non-linear mappings. This capability is beneficial for BCIs, where the goal is to translate these signals into precise, reliable commands for external devices.

In classifying MI-EEG signals, feature extraction is also a crucial step because of the stochastic nature of the EEG signals. Feature extractions involve transforming raw EEG data into a set of representative features that capture the essential characteristics of brain activity associated with motor imagery. Effective feature extraction methods support the neural networks in distinguishing different classes with less effort. Power spectral densities (PSDs) of specific frequency bands (mu and beta bands) [39], time-domain statistical measures (mode, mean, variance) [44], connectivity measures between electrodes [45] and CSP components [45] are examples for commonly used feature extraction methods in MI-EEG classification tasks. These features have been selected for tasks by considering their relevance to the underlying neural processes and their discriminability in classifying MI tasks.

Moreover, advanced neural network architectures, such as convolutional neural networks (CNNs) [46] and recurrent neural networks (RNNs) like long short-term memory (LSTM) [47] networks, have the capability to perform automatic feature extraction by learning. These advanced architectures identify the most informative features directly from the data during the training process.

As in traditional methods, SNN based classification processes need extracted features from the preprocessed signals for achieving better accuracy performance. However, these extracted features must be converted into spikes before they feed to the SNN architecture. Hence, when selecting the features, it is essential to ensure that those feature values can be converted into spikes without losing significant information. If the generated spike trains are separable between classes, the spike driven neural processing mechanism in SNNs can accurately predict the classes.

2.4 Spike Driven Neural Processing

SNNs were introduced as the third generation of neural network models [16]. Figure 4 shows an abstract comparison of the first, second and third generations of Artificial Neural Networks (ANNs).

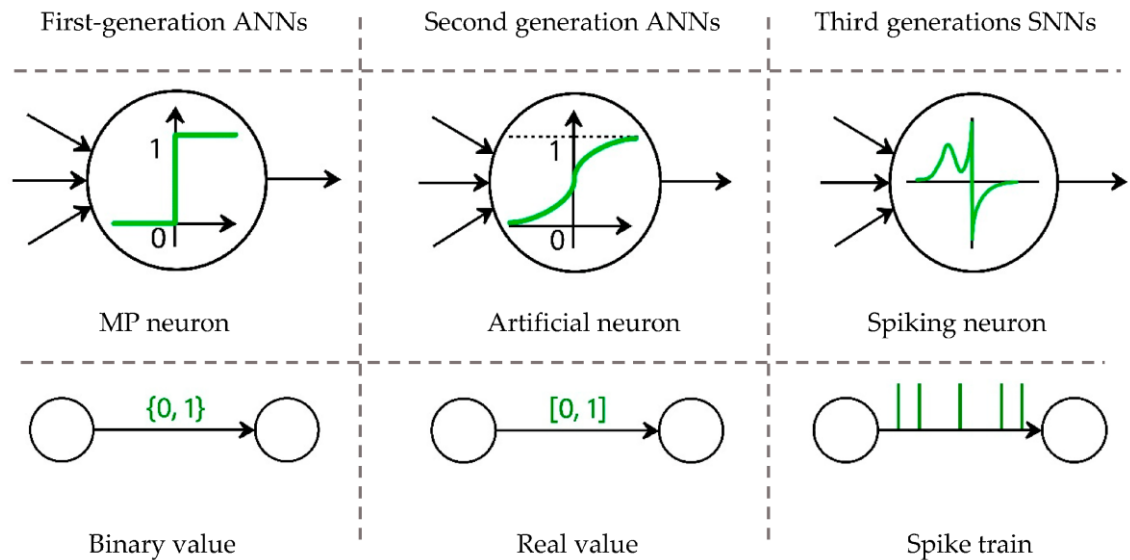


Figure 4. Illustrative comparison of neurons in first, second and third generations [46].

As mentioned in Section 1.1, the uniqueness of SNNs is that they try to mimic neural communication biological systems more closely than conventional neural network architectures [16]. SNNs employ spikes (discrete events in time which mimic the effects of action potentials) to communicate between neurons. This spike based communication is the major differences between SNNs compared to conventional neural networks [17]. As a result, neurons in SNNs do not transmit information at every cycle but rather send output spikes only when a specific membrane potential threshold is exceeded. This approach to information processing makes SNNs more biologically plausible and potentially more energy efficient, especially in tasks involving temporal data processing, like EEG signal analysis [18].

Like conventional neural networks, SNNs also require training to adjust their synaptic weights and perform specific tasks such as classification, prediction, or pattern recognition. The inherent operational dynamics of SNNs need suitable learning methodologies compatible with spike based communication mechanisms. Among such methodologies, Spike-Time Dependent Plasticity (STDP), Backpropagation using spikes (with surrogate gradients) and spiking time, and ANN-to-SNN conversion (often referred to as Shadow Training) stand out as prominent strategies with unique advantages (and also challenges) for different kinds of applications [17], [47].

Despite the theoretical advancements in SNNs' low power performance, applying them in real world encounter specific challenges due to the hardware which they needed. Firstly, their unique computation model, which includes time in processing and relies on the timing or frequency of events, demands novel architectural designs for implementing them on a large scale [48]. Secondly, the requirement for event based data (primarily available from specialized neuromorphic sensors like neuromorphic retinas) restricts the types of data SNNs can process and producing event based data is also challenging as these type of sensors are not widely available [49]. Lastly, while SNNs have the potential for high energy efficiency in neuromorphic hardware, the pace of developing suitable training algorithms lags behind the advancements in hardware [50], [51]. These issues underscore the need for continued research to harness the full potential of SNNs.

3. LITERATURE REVIEW

This chapter reviews previous works which have already been done on MI-EEG classification and the use of SNN in classification tasks and neuromorphic computing platforms. Additionally, the emerging neuromorphic infrastructure highlights have been mentioned here to prove the value of introducing spiking-based pipelines.

3.1 Performance of Second Generation ANNs as MI-EEG Classifiers

As mentioned in the previous chapters, machine learning methods have mainly been used to classify MI-EEG signals into separate classes. With deep learning architectures like CNNs, accurate and robust models have been introduced for classifying MI-EEG signals [52], regardless of their complexity and energy consumption. The table below presents a few works that have used second-generation ANNs for MI-EEG classifications.

Table 1. Prior works that use second generation ANNs for classifying MI-EEG signals.

Study	Architecture Type	Optimizer *	Dataset	Input Features	Mean Accuracy
[53]	SVM	-	BCIC IV 1 BCIC III 4a BCIC III 3a	Regularized CSP	81.6% 87.4% 91.9%
[54]	CNN	Gradient decent	BCIC IV 2a	CWT [†] images	83.2 %
[55]	CNN	Adam	BCIC IV 2a HGD	Time series	75.7% 95.4%
[56]	DBN	Adadelta	BCIC IV 2b	Time series	83.55%
[57]	LSTM	Adam	BCIC IV 2a	1d-AX [#]	71%
[58]	CNN+SVM	-	BCIC III 4a	DWT [†] , FT ⁺ , EEMD [□] , CSP	96.34%
[59]	CNN+SAE	Gradient decent	BCIC II 3 BCIC IV 2b	STFT [†] images	77.6% 90%
[60]	CNN+LSTM	Adam	PhysioNet	Time series	98.3%

* Gradient decent, Adam and Adadelta optimizers are algorithms which are used to update the weights in neural networks [61].

† CWT, DWT and short time Fourier transform (STFT) are signal processing techniques that provide time-frequency representations of signals.

One dimension-aggregate approximation (1d-AX) is a feature extraction technique for time-series data that normalizes and approximates each channel.

+ Fourier Transform (FT) which is a technique that decomposes a time domain signal into its frequency components.

▫ Ensemble Empirical Mode Decomposition is a signal processing technique that breaks non-stationary signals into simpler oscillatory components called intrinsic mode functions.

All the datasets included in Table 1 are publicly available for download. These datasets have often been used for MI-EEG studies [52],[62]. This consistent usage across various studies ensures the reliability of these datasets for training and testing MI-EEG classification models. The following table includes the key information of these datasets related to MI-EEG studies.

Table 2. Publicly available MI-EEG datasets.

Dataset	Class labels	No of EEG Channels	No of Subjects	No. of Trials
BCI Competition II 3 [63]	<ul style="list-style-type: none"> • Left Hand • Right Hand 	3	1	280
BCI Competition III 3a [64]	<ul style="list-style-type: none"> • Left Hand • Right Hand • Foot • Tongue 	60	3	240
BCI Competition III 4a [65]	<ul style="list-style-type: none"> • Right Hand • Foot 	118	5	1400
BCI Competition IV 1 [66]	<ul style="list-style-type: none"> • Left Hand • Right Hand / Foot 	59	7	1400*
BCI Competition IV 2a [67]	<ul style="list-style-type: none"> • Left Hand • Right Hand • Both Feet • Tongue 	22	9	576
BCI Competition IV 2b [68]	<ul style="list-style-type: none"> • Left Hand • Right Hand 	3	9	240
HGD [11]	<ul style="list-style-type: none"> • Left Hand • Right Hand • Both Feet • Rest 	128	14	14000
EEGMMID [69] (PhysioNet)	<ul style="list-style-type: none"> • Open a Fist • Close a Fist • Open both fists or feet • Close both fists or feet 	64	109	1500+
OpenBMI [70]	<ul style="list-style-type: none"> • Left Hand • Right Hand 	20	54	~21600

* 1400 is the number of trials which are included in the selected subset of the complete dataset.

For this thesis work subset of “BCI Competition IV 1” dataset has been used (see Section 4.1).

3.2 SNNs in MI-EEG Classification

Compared to second-generation ANNs and other machine learning methods, fewer studies have employed SNNs for classifying MI-EEG signals. However, despite a smaller research community, robust SNN based models have been developed that achieve sufficient classification accuracy in MI-EEG signal classification. Table 3 provides examples that highlight the performances in MI-EEG task classification achieved by SNNs.

Table 3. Examples of works related to MI-EEG classification tasks that use SNNs

Study	Dataset	Input Features	Learning Rule	Mean Accuracy
[19]	EEGMMID	2D matrices spatially mapped channels	Spatio-temporal back propagation	79.95%
[20]	BCIC IV 2b	Band Power Values	A Threshold Based Spike Rate Potentiation	75%
[21]	BCIC IV 2b	Spike representation of time-frequency	Mapping a trained CNN to a SNN (Shadow training)	75.63%
[22]	BCIC IV 2a	PSD, DWT	Particle Swarm Optimization	81.17%
[23]	BCIC II 3	Time-frequency PSD distributions	Gradient based optimization with backpropagation	93.9%
[24]	OpenBMI	Time domain signal windows	A surrogate gradient based backpropagation method	72.83%

Even though there are several works that use SNNs to classify MI-EEG signals, we could not find any that use the CSP method to condition the signals. The CSP method effectively transforms temporal MI-EEG signals into a space (by emphasizing the differences in the variances of their amplitudes) where they are more easily separable into classes [12]. This quality allows the signals to be encoded into spikes without losing critical information. This is the major reason that this thesis focuses on CSP when building the classification pipeline.

3.3 Spike Encoding, Learning Rules and types of SNN architecture

From the software perspective, researchers in the neuromorphic domain mainly examine encoding methods, learning rules and architecture for neural networks [17]. A good encoding method can select only the critical information from input signals while minimizing spike counts, which are proportional to energy consumption at the system level. Good learning rules and architecture act energy efficiently in the train phase as well as inferring phase by finding the best (or better) solution for the classification task (For example, some learning algorithms promote sparsity or efficient activation patterns, which reduce the number of computations required during both training and inference).

There are three commonly used efficient spike encoding methods [17] :

- Rate Coding – Information is encoded with the firing rate over a time interval.
- Latency Coding – Information is encoded in the timing of the first spike
- Delta Coding – Information is encoded according to the change compared to the previous sample (mostly used to encode time series data)

In Section 4.3.2, these methods are comprehensively explained. It is worth noting that novel encoding methods are mostly built upon these methods.

There are studies which have focused on inventing and modifying learning rules for training spike based neuromorphic architectures. Compared to the conventional learning rules, these rules focus on working with spikes to optimize the performance of SNNs while preserving energy efficiency. Table 4 presents a few learning rules introduced for training spike intelligence systems.

Table 4. Learning rules for training SNNs and their biological plausibility

Study	Learning Rule	Remarks	Biological Plausibility
[71]	Surrogate gradient based backpropagation through time	Use approximation to handle non-differentiability of spikes	Low
[72]	SpikeProp	Use temporal difference errors for adjusting weights	Low
[73]	Evolutionary algorithms	Inspired by natural selection, where potential solutions evolve generation by generation	Low
[74]	BCM Rule	Adjusts synaptic strength depending on the postsynaptic neuron's firing rate	High
[75]	STDP	Adjusts synaptic strength depending on the precise timing between spikes	High

In this work, surrogate gradient based backpropagation through time was employed to optimize the SNN architectures by considering its effectiveness on updating weights over other methods in general computational architectures.

Like the conventional neural networks' architectures, SNNs are also emerging with various types of architectures such as feedforward SNNs, recurrent SNNs, convolutional SNNs, liquid state machines and spiking DBNs by providing versatile options for addressing challenges [76]. These different architectures provide a range of flexible and powerful approaches to addressing various challenges. Consequently, many fields (such as pattern recognition, sensory processing and decision making) can harness the energy efficiency advantage of neuromorphic systems.

The optimal energy efficiency of SNNs cannot be accessible with conventional computing architectures (such as Von Neumann architectures), which are designed to work with synchronous digital signals. A major mismatch is that conventional architecture operates with clock signals, consuming power regardless of whether important information is being processed. In contrast, neuromorphic systems activate asynchronously and consume power when there is enough potential to produce spikes. Hence, implementation of SNNs in hardware requires new architectures. To address this need, several neuromorphic hardware chips have been developed specifically to process spikes and update weights. Table 5 includes some of these chips that are available in the market or available to use for research groups.

Table 5. *Neuromorphic hardware chips*

Ref.	Name of the Chip	Designer/Manufacturer
[77]	Loihi	Intel Labs
[78]	TrueNorth	IBM
[79]	SpiNNaker2	Technische Universität Dresden
[80]	Xylo	SynSense

This section and the entire chapter highlight the significant emergence of the neuromorphic field in both software and hardware. By mimicking biological neurons, this field is set to drive the development of energy efficient intelligent systems. Such advancements promise to make reliable, energy efficient wearable devices (like MI-EEG based wearable BCI systems) more accessible and affordable in the near future.

4. MATERIALS AND METHODS

This chapter presents the materials (dataset and software libraries) and methods (experimental procedures) utilized and implemented in this study. The purpose of this chapter is to provide a comprehensive understanding of the approach that was used to achieve the objectives of this thesis.

4.1 Dataset

The calibration dataset from the “BCI Competition IV—Dataset 1” was selected for this study to train and test the SNN based classifiers [66]. This sub-dataset consists of labeled EEG signal trials corresponding to two classes (right and left hand MI), recorded from five healthy subjects, with 100 trials available for each class per subject. Each recorded trial has a duration of 8 seconds. Figure 5 illustrates the paradigm which has been used to record the signals with visual cues (symbols in the black boxes).

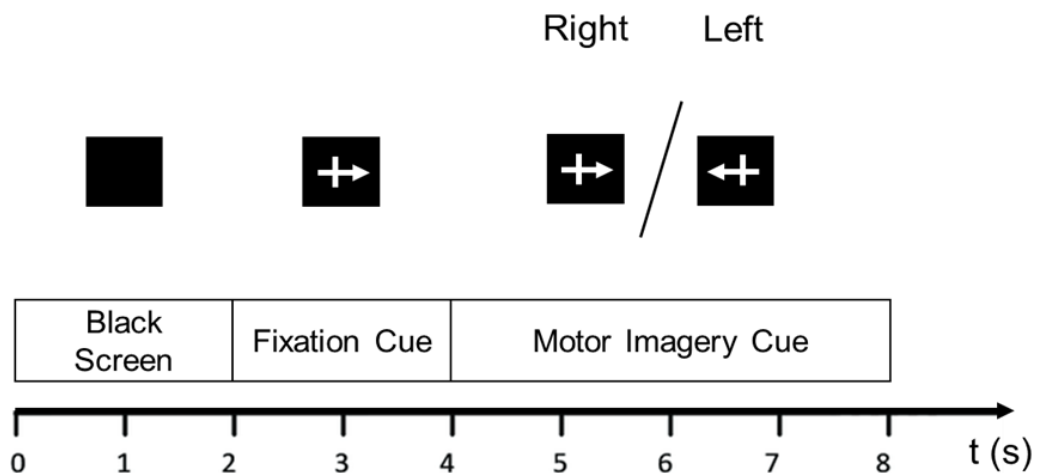


Figure 5. An illustration of the paradigm of recorded calibration trials. Black boxes show the symbols that appeared on the monitor as a visual cue for the subject. This figure has drawn according to the description provided by [66].

In general the time period between 4.5 s to 6.5 s shows dominant neural activities related to MI tasks [44]. Hence, for the development of classifiers in this thesis work, only the trimmed time windows from 4.5 s to 6.5 s were considered.

In this dataset, the EEG signals were acquired using 59 Ag/AgCl electrodes densely distributed over the sensorimotor areas to capture the electrical activities of the brain related to the motor imagery tasks. A “BrainAmp MR plus” amplifier was used to record the signals coming from the electrodes. The amplifier originally sampled the data at 1,000 Hz with 16-bit quantization, providing a 0.1 μ V accuracy. For post analysis the data were

down-sampled to 100 Hz. Additionally, they used a low-pass filter with a cutoff frequency at 49 Hz to eliminate powerline noise as well as other high-frequency noises. This dataset is publicly available for download at <https://www.bbc.de/competition/iv/download/index.html>.

4.2 Used Software Libraries

This thesis work is conducted in a software environment based on the Python (version 3.8.18) programming language. The following Python libraries were used :

- NumPy (version 1.24.3)

For numerical computing tasks, NumPy is a fundamental library in Python. It is primarily used for manipulating data structures such as arrays and matrices. NumPy provides a wide range of operations, including sorting and selecting data, reshaping arrays and matrices, and performing mathematical operations. In this work, NumPy was mainly used for preprocessing steps (especially for selecting the signal windows and constructing the CSP filter for spatial filtering).

- SciPy (version 1.10.1)

SciPy is a package built on NumPy. It provides high-level syntaxes for using the fundamental algorithms for scientific computing. In this work, SciPy was used to load EEG data from the data files and build the IIR filters to condition the EEG signals.

- SnnTorch (version 0.6.4)

This Python library is designed to implement SNNs and train them with gradient based learning methods. SnnTorch is built on PyTorch which is an open-source deep learning framework renowned for its intuitive Python interface and the capability to use GPU-accelerated tensor computation. Since SnnTorch inherits the functionalities from PyTorch, it is efficient for implementing SNN based pipelines and testing their performances.

4.3 Implementation of Pipelines

The first objective of this thesis is to implement a feasible SNN based pipeline for MI-EEG classification. As with conventional methods, the implementation of pipelines starts with the preprocessing steps, namely, applying BPF and CSP methods to the data. In contrast to conventional methods, preprocessed signals must be converted into spikes before feeding the SNN based classifiers. There are three major spike encoding methods (rate encoding, latency encoding and delta encoding) that convert signals into spikes. Each method has its own benefits and drawbacks. Hence, to find out the effect of each encoding method for the SNN based MI-EEG classifiers and to select the most suitable one for future works, three types of pipelines have been implemented based on each spike encoding method, as illustrated in Figure 6. This section provides detailed descriptions of each step in these pipelines to describe their implementation and discuss their suitability.

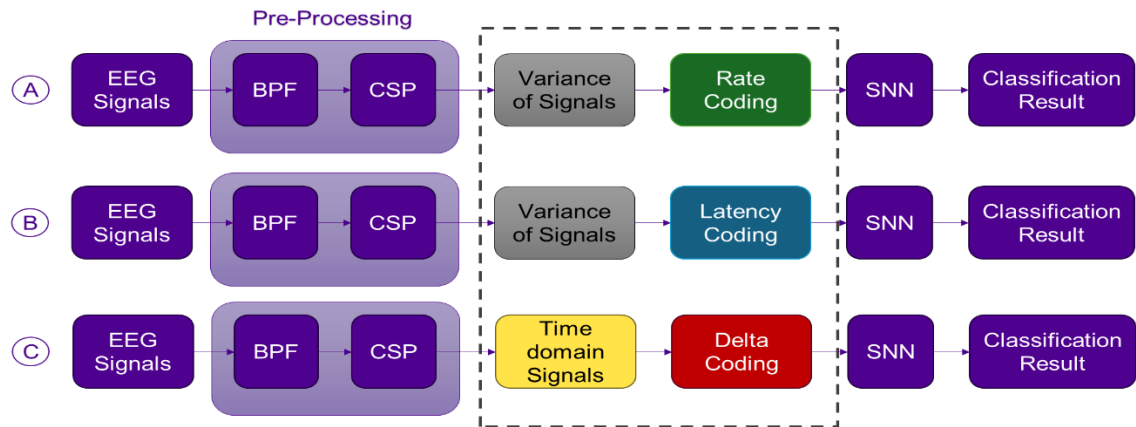


Figure 6. Block diagrams of the three implemented pipelines. Pipelines A, B, and C correspond to the rate coding, latency coding, and delta coding approaches, respectively.

4.3.1 Preprocessing Methods

Due to the non-stationary behavior of EEG signals, preprocessing techniques are essential for extracting relevant features from MI-EEG signals to make them separable into classes [30]. Preprocessing methods are intended to mitigate unwanted noises and enhance the feature selectivity in the signals, boosting the overall performance of systems. The types of preprocessing steps that are employed to condition the signals depend on the application and the type of noise affecting the signals. In the context of MI-EEG signals, frequency domain filters are essential for removing unwanted noise to improve the SNR value of the signals, while spatial filters are needed to enhance spatial features for the identification of the dominantly active brain regions [30].

- **Band-Pass Filter (BPF)**

The frequency bands of interest are roughly 8 Hz to 13 Hz and 14 Hz to 30 Hz for the MI-EEG signals (see Subsection 2.2). However, when analyzing the MI-EEG signals in the dataset, we observed that some of the low frequency components (lower than 8 Hz) correlate with MI activities. Thus, the 2 Hz – 30 Hz frequency band was selected as our frequency region of interest instead of 8 Hz – 30 Hz. To filter data in this selected band region from the other data, we integrated 6th order IIR Butterworth filters with a passband of 2 Hz to 30 Hz to the pipelines. The order and type of the filters were selected considering the widths of pass and stop bands and the previous experiments [81], [82]. These BPFs cut the DC offsets and slow wave artifacts from the lower end of the spectrum and reduce the effects of the noise components beyond 30 Hz.

- **Common Spatial Pattern (CSP)**

Since the MI-EEG signals from different classes are generated in different brain areas, spatial filtering can potentially increase the separability of the data between classes. The CSP algorithm was used in the implemented pipelines as another preprocessing step after the BPFs. The main expectation was that the CSP algorithm would enhance the spatial features of MI-EEG signals associated with each class [45]. Along with Figure 7, the numerical implementation of the CSP algorithm is described below.

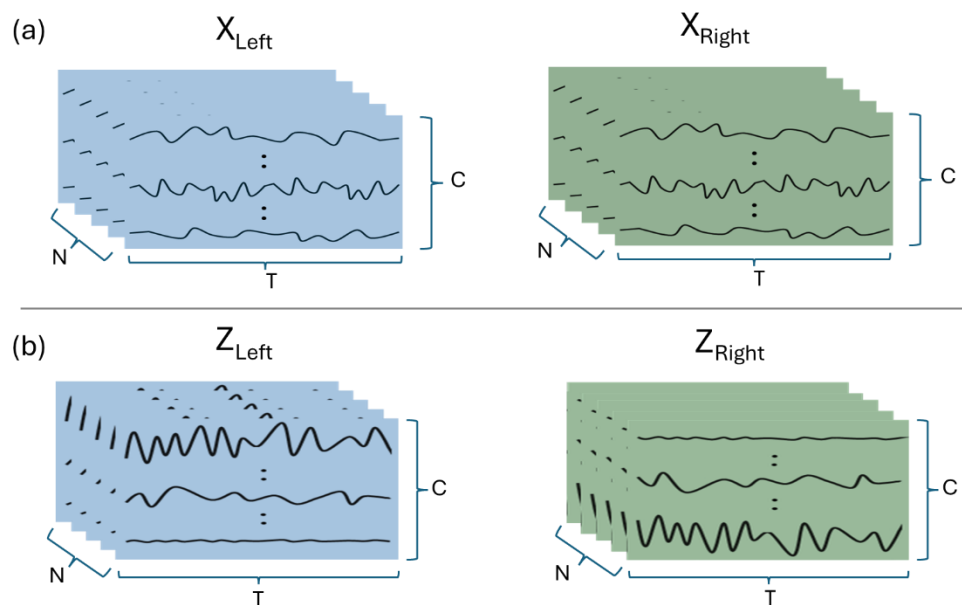


Figure 7. Demonstration of the input and output signals of the CSP algorithm. (a) X_{Left} and X_{Right} matrices represent the prepared matrices that include each class's trial data. (b) Z_{Left} and Z_{Right} matrices represent the output signals of the CSP algorithm. N , T , and C denote the number of trials, sample points per trial, and EEG channels, respectively.

Figure 7(a) shows the matrices that should be prepared to execute the CSP algorithm. The number of trials per class (\mathbf{N} , which is 100 in this work) proportionally affects the effectiveness of the CSP spatial filters and should be nearly equal for each class. Otherwise, spatial filters will be biased to the class with the larger value. The number of channels (\mathbf{C} , which is 59 in this work) also has a positive effect on the filters' output since the high number of channels provides more information about the signal sources. The number of sample points (\mathbf{T} , which is 200 in this work) should be selected carefully, considering the period in which activation of relevant sources (relevant areas in the motor cortex) is dominant.

Firstly, the demeaned signals in trimmed 2-second time windows by the BPFs were included in the two 3-dimensional matrices as illustrated in Figure 7(a) considering their labelled classes (Left and Right). After that, the following operations were performed on them sequentially, as explained in [12].

Operation 1 - Computing the average covariance matrices:

- Compute the covariance matrix for each trial

$$R_i = \frac{X_i X_i^T}{\text{trace}(X_i X_i^T)} \quad (1),$$

- Compute the average covariance matrix for each class

$$R_{Left} = \left(\frac{1}{N}\right) \times \sum_{i=0}^{N-1} R_i \quad (2)$$

$$R_{Right} = \left(\frac{1}{N}\right) \times \sum_{i=0}^{N-1} R_i \quad (3).$$

In Equations 1, 2 and 3, i denotes the trial number. R_i represents the covariance matrix of the i^{th} trial and X_i represents the input data of the i^{th} trial (T number of time samples from C number of EEG channels as illustrated in Figure 7). N is the number of trials in a class. R_{Right} and R_{Left} are the average covariance matrices of each class which summarize each class's overall spatial variance patterns.

Operation 2 - Forming composite covariance matrix (R_C):

$$R_c = R_{Right} + R_{Left} \quad (4).$$

The composite covariance matrix provides a unified reference for whitening the data, ensuring that both classes are normalized in the same spatial domain.

Equation 1 makes R_i matrices symmetric (which have same number of columns and rows). Hence, the matrices R_{Right} , R_{Left} and R_c inherit this property according to Equation 2-4, and become symmetric matrices too.

Operation 3 – Since R_c is a symmetric matrix, it can be decomposed by using the “eigenvalue decomposition”:

$$R_c = U \Lambda U^T \quad (5),$$

where U is an orthogonal ($UU^T = I$) matrix of eigenvectors and Λ is the diagonal matrix which has corresponding eigenvalues [83],.

Operation 4 - Compute the whitening transformation matrix P :

$$P = \Lambda^{-0.5} U^T \quad (6).$$

Calculation of $\Lambda^{-0.5}$ can be performed using the piece-wise square-root operation on Λ , since it is a diagonal matrix and all the elements in it are non-zero positive values. Whitening transformation simplifies the elements in the R_c by normalizing them and eliminating their statistical dependencies (decorrelation) [84].

Operation 5 - Whiten one of the average Covariance Matrices using P :

$$S_{Left} = P R_{Left} P^T \text{ or } S_{Right} = P R_{Right} P^T \quad (7),$$

where S_{Left} and S_{Right} are the whiten average covariance matrices of each class which can be produced by using matrix P .

Operation 6 - Performing the eigenvalue decomposition again on one of the whiten average covariance matrices :

$$S_{Left} = B \Gamma_{Left} B^T \text{ or } S_{Right} = B \Gamma_{Right} B^T \quad (8).$$

Here, B is the matrix of eigenvectors of S_{Left} and S_{Right} , Γ_{Left} and Γ_{Right} are the diagonal matrices of eigenvalues of S_{Left} and S_{Right} . Since $S_{Left} + S_{Right} = PR_cP = I$, S_{Left} and S_{Right} share the same eigenvectors, that is the reason for having same eigenvectors for both left and right equations.

Operation 7 - Computing the projection matrix (W):

$$W = B^T P \quad (9).$$

The elements of W are spatial filters. They increase the variance of the unique set of output signals and minimize the variance of the other set of output signals.

Operation 8 - Projecting the original signals (X_{Left} and X_{Right}) onto the spatial filters:

$$Z_{Left} = W X_{Left} \quad (10)$$

$$Z_{Right} = W X_{Right} \quad (11),$$

where Z_{Left} and Z_{Right} are the matrices that contain spatially filtered signals from the CSP filters. These Z matrices have the same shape ($N \times T \times C$) of the signal matrices that are prepared to build the CSP filters (see the shape of matrices in Figure 7(a) & Figure 7(b)). The output signals from a set of CSP components should exhibit higher amplitudes (indicating higher sample variances) for one class and lower amplitudes (indicating lower sample variances) for the other class. Conversely, the other set of CSP components should display the opposite pattern. Figure 7(b) demonstrates this graphically.

4.3.2 Spike Encoding Schemes

After the preprocessing steps (BPF and CSP), the signals were uniquely processed in each pipeline to generate spike patterns in order to feed the SNN classifiers. This section discusses these schemes. Figure 8 presents a summarized overview of each spike encoding scheme and Figure 9 visually demonstrates the process of each spike coding method.

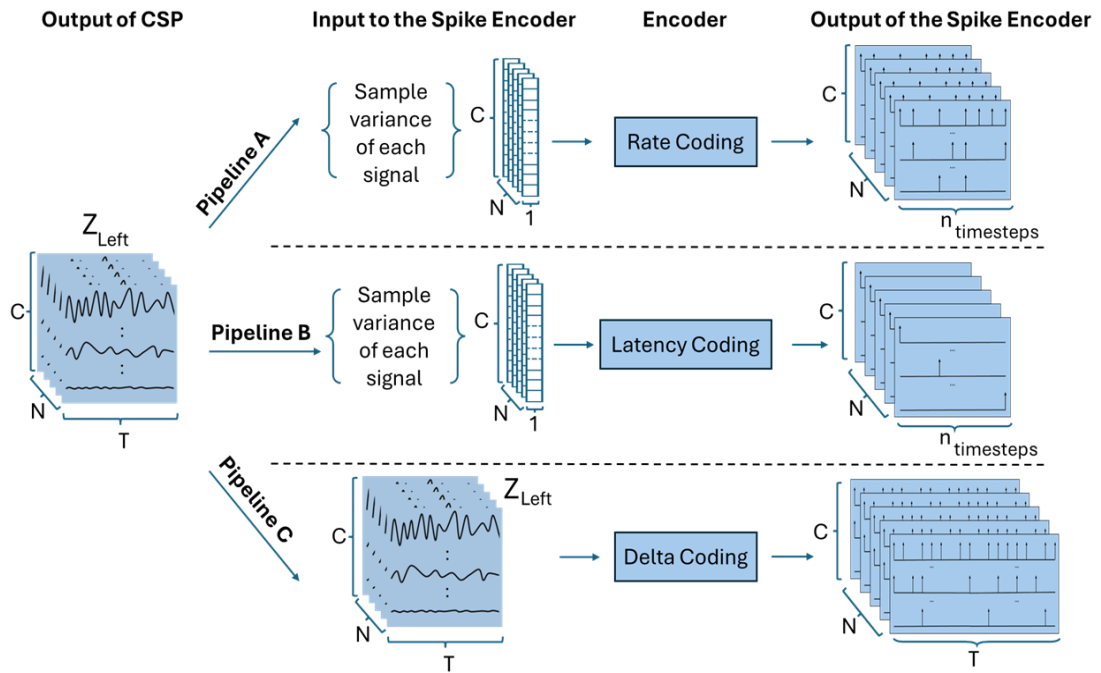


Figure 8. Spike encoding schemes intergrated in each pipeline. N , T , and C denote the number of trials, sample points per trial, and EEG channels, respectively. Number of timesteps ($n_{\text{timesteps}}$) is a manually defined integer for limiting the length of the outputs.

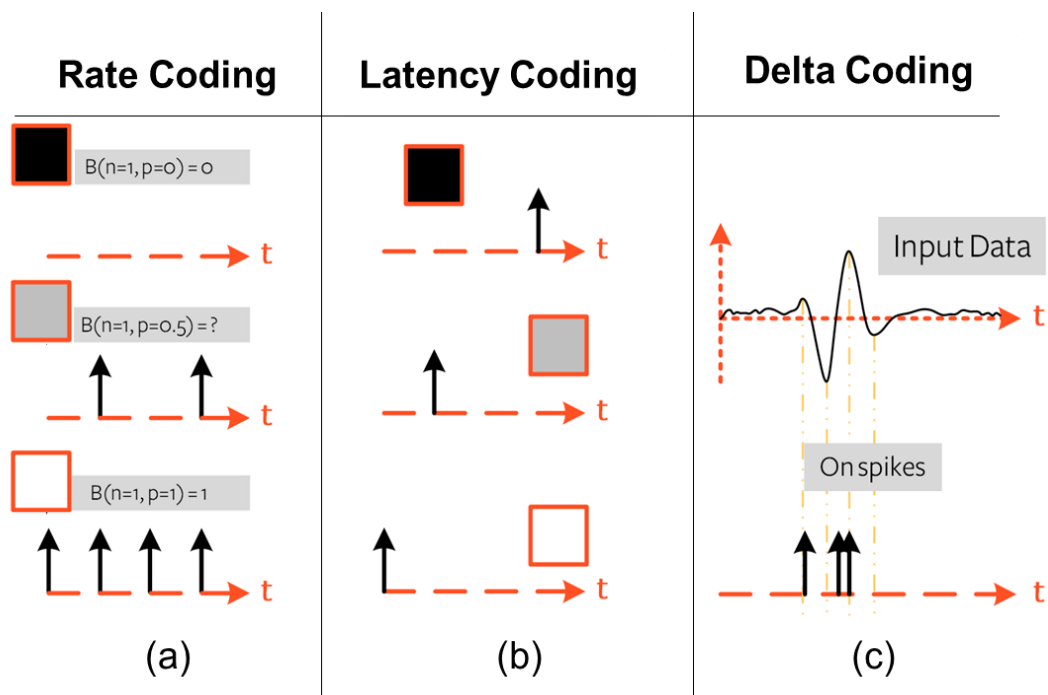


Figure 9. Visual demonstration of each spike encoding method. Figure is adapted from [14]. (a) Left column shows how the rate coding produces spike trains for different gray color values. (b) Middle column shows how latency coding embeds the data values into the spike times. (c) Right column shows how the delta coding method produces spikes according to the changes of input signals.

- **Rate Coding – Pipeline A**

Rate coding produces spike patterns over a defined number of steps (which is 14 ($n_{timesteps}$) in this work) based on a probability distribution influenced by input values. Figure 9(a) graphically explains the process of rate coding by showing the effects on the spike train according to the different input values (probability values).

In the rate coding pipeline, the spike encoder receives 59 (**C**) input values for generating spike patterns relating to a trial window. These input values are the sample variances of the 59 signals projected from the CSP spatial filters for that particular trial window (see Figure 8 which clearly illustrates this). To ensure these values can effectively represent probabilities, all sample variances are normalized within trials, limiting their range between 0 and 1 before starting the encoding process.

In this rate encoder, the Bernoulli distribution (which is a special case of the binomial distribution) was chosen to generate spike patterns because of its simplicity and its compatibility with the binary nature of spikes. The Bernoulli distribution outputs a 1 (spike) or 0 (no spike) based on a single probability value as expressed in Equation 12:

$$out_i \sim \text{Bernoulli}(p = x) \quad (12),$$

where out_i is spike information in the i^{th} position of the spike train and x is the input value which is assigned as the probability of succeeding (receiving 1) in Bernoulli distribution.

To generate a spike train with a length of $n_{timesteps}$ for a given input value, Equation 12 is executed in $n_{timesteps}$ times repeatedly with the same input value.

- **Latency Coding – Pipeline B**

In latency coding, input values are embedded into the time delay for occurring the first spike (see the middle column of Figure 9). The delay time is inversely proportional to the magnitude of the input value. In contrast to rate coding, only one spike is produced in the latency coding approach. This work used Equation 13 to generate spikes according to latency coding (see APPENDIX A for the derivation of Equation 13):

$$t_{delay} = \tau \ln \left(\frac{x}{x - V_{th}} \right)$$

where t_{delay} is the time to happen first spike of a leaky integrate and fire (LIF) neuron when providing a constant current which has an x (input value) magnitude. τ and V_{th}

are fixed parameters which represent the time constant and the membrane threshold voltage of the neuron. The achieved t_{delay} values from Equation 13 for the inputs are rounded to nearest integer values to determine the time steps for assigning spike occurrences. When encoding the data, the values 1 and 0.5 have been assigned to τ and V_{th} in this work (in SnnTorch these parameters are defined as unitless parameters for simplicity). In the case of x is smaller than V_{th} , it is recorded as an input without a spike. Hence, Equation 13 has not been used in such a case.

Similar to the rate encoding pipeline, the encoder in latency pipeline uses the normalized variances (within trials) of projected outputs from CSP filters as the input value x for determining the t_{delay} . Therefore, in the latency coding also, only 59 input values are used to produce output spike patterns for one trial window (see Figure 8).

The number of timesteps was limited to $n_{timesteps}$ (which is 14 in this work) to remove the inefficiency of the encoding method due to long delay times. However, this does not affect the overall performance of the pipeline since long-term delays represent the least significant information.

- **Delta Coding – Pipeline C**

Compared to the other two methods, spike encoding in delta coding pipeline is computationally less complex. It considers the difference between the adjacent time samples and if the change exceeds the defined threshold value, it produces a spike at that time step. The process of delta encoding can be stated as Equation 14:

$$out_i = \begin{cases} 1; & x[i] - x[i - 1] \geq V_{th} \\ 0; & x[i] - x[i - 1] < V_{th} \end{cases} \quad (14),$$

where out_i is the i^{th} position of the spike train and x is the preprocessed signal in the time domain. The left column of the Figure 9 illustrates of this process.

In contrast to the other two pipelines where the variances of the CSP signals were used as inputs, the delta coding pipeline uses the projected signals from CSP method as they are. Consequently, the output of the delta coding produces the spike trains which have the exact shape of the input which is 59×200 for a trial window in this work. Having such a long input and output dimension (200 time points), which correspond to each other sample by sample, gave us the opportunity to study the performance of an SNN-based classifier that runs online, in contrast to the other two batch classification methods. Figure 8, illustrates the differences in the inputs and outputs of the delta encoding process compared to the other two methods.

4.3.3 SNN based Classifiers

The primary classification process in the designed pipelines occurs within the SNNs. In these SNNs, the input spike trains are processed to indicate the class of the input signals. To elaborate how the SNNs act, this section covers key components of the SNN classifiers including the neuron model used for the networks' neurons, network architecture, training process, and testing method used to evaluate the classifiers' performance.

- **Neuron model**

For simplicity, a first-order LIF neuron model was used in every layer of SNN structures in all three pipelines. LIF is one of the simplest neuron models closely representing biological neuron activations. The membrane potential of the LIF neuron is updated according to Equation 15, and produces spike given a crossing of a membrane-potential threshold as expressed in Equation 16.

$$V[t + 1] = \beta V[t] + I_{in}[t + 1] - RV_{thr} \quad (15)$$

$$S = R = \begin{cases} 1 & \text{if } V \geq V_{thr} \\ 0 & \text{if } V < V_{thr} \end{cases} \quad (16).$$

Here, I_{in} is the sum of the input currents, V is the membrane potential of the Neuron, β represent the decay rate of the membrane potential (here β reflects the quality of time constant τ) and V_{thr} is the threshold membrane potential to mark a spike. R is there to represent the reset mechanism.

The spikes (from spike encoding schemes or previous layer neurons) that come toward the neurons in SNNs are converted to different current values with the multiplication of synaptic weights. Thereafter, at the neurons, these currents are summed together and given as input to the neuron as described in Equation 15. When $V > V_{thr}$, neurons produce a spike ($S = 1$) and assigns $R = 1$ to update the V subtracting it by V_{thr} . Otherwise, R and S remains as "0" as expressed in Equation 16.

- **Network architecture**

Each SNN architecture in this works' pipelines has three fully connected layers of neurons with learnable weights. Input layers of SNNs have 59 neurons to acquire the 59 spike trains which feed to the networks. Secondly, each SNN has a hidden layer. Number of neurons in these hidden layers were empirically adjusted to enable them to handle the complexities of the input data. In the output layers two-neurons are employed for representing "Left-MI" and "Right-MI". Ideally, one of these neurons will be dominantly activated at the inferencing stage to represent the predicted class by the SNN.

- **Training Process**

The employed training method for SNNs was very similar to the traditional deep learning algorithms. As in traditional methods, weights were updated with backpropagation to minimize the loss (a measure of error between models' predictions and actual values). In contrast to traditional neural networks, SNNs do not produce outputs with continuous values. Thus, gradients cannot be calculated for SNNs. Therefore, SNNs require an alternative method for calculating gradients to adjust the weight values according to the backward error propagation. An alternative solution for this is surrogate gradient method which introduces a differentiable approximation for the activation function of the spiking neurons. So, instead of Heaviside step function ($H(V)$), it introduces a differentiable function ($f(V)$) for gradient calculation in the backward propagation. In this work, the arc-tan function in Equation 17 was used as the approximated function. The first derivative of the arc-tan function ($f'(V)$) is shown in Equation 18.

$$H(V) \approx f(V) = \frac{1}{\pi} \arctan(\pi(V - V_{th})) \quad (17)$$

$$f'(V) = \frac{1}{\pi} \frac{1}{(1 + (\pi(V - V_{th}))^2)} \quad (18).$$

Figure 10 shows the behavior of the heavy sidestep function, surrogate function and its first derivative.

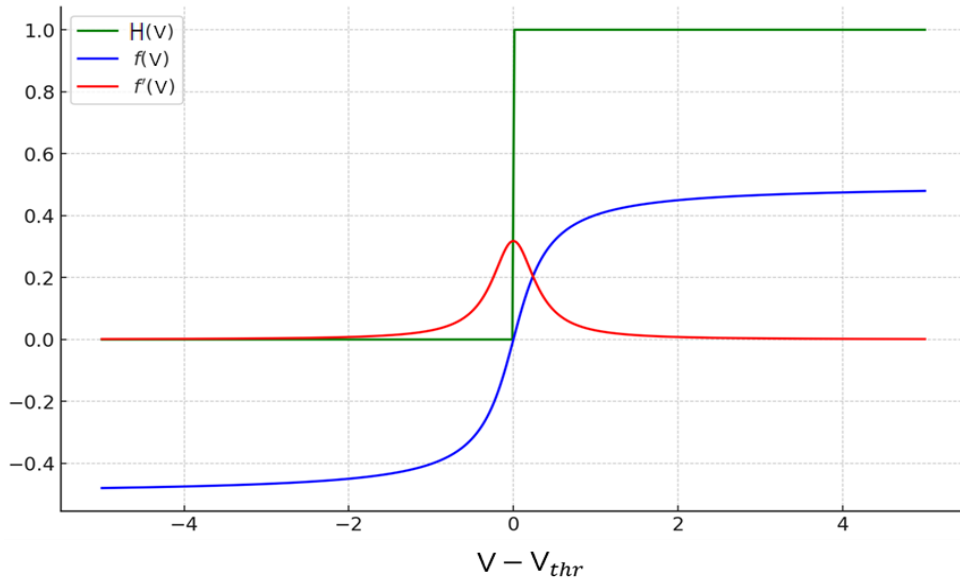


Figure 10. The graphical illustration of Heaviside step function $H(V)$ (green), surrogate function $f(V)$ (blue) and its first derivative $f'(V)$ (red). Note that the horizontal axis is shifted by V_{th} for achieving symmetry and the vertical axis has no units for values but represents numeric scale to express the behaviors of functions relative to the others.

- **Loss Functions**

In ML algorithms, loss refers to a measure of error between models' predicted outputs and target output values, indicating how well the model performs. A loss function is a mathematical formula used to compute this error for the optimization processes to minimize the total error during training. By evaluating the loss function's value, training algorithms adjust the models' weight to improve predictions.

In this work, two different loss functions provided by PyTorch were employed individually in each pipeline. The Cross Entropy (CE) Loss function was utilized in the rate coding (A) and latency coding (B) pipelines [85]. The Binary Cross-Entropy (BCE) with Logits Loss function was applied in the delta coding (C) pipeline [86]. Each of these loss functions was selected for pipelines based on their suitability for computing losses from the respective outputs, as evaluated by the overall accuracy levels achieved.

- **Hyperparameters**

Hyperparameters are configurations that control the training process and model architecture but are not learned from the data. Manually set hyperparameters in this work are mentioned in APPENDIX B. The hyperparameters not mentioned in APPENDIX B use the default values given by PyTorch.

- **Testing Method (Performance Measure)**

To measure the overall performance of the pipelines, we used the classifier models which were built by using the pipelines. In this process, we evaluated the classifiers' test accuracies using a 5-fold cross-validation method on each dataset. Each dataset was divided into five unique folds. For each of the five iterations, we trained a classifier using four folds for training and one fold for testing, rotating the test fold each time. By averaging the accuracy values obtained from these iterations, we ensured that the results are validated and not overfitted to a particular train-test split. This testing method provides a robust assessment of the pipelines' effectiveness in constructing classifier models across different subsets of the data.

80% of the encoded trials were used for training the weights of the classifier network based on the obtained values of the loss function, and the remaining 20% were used for testing the obtained classifier. Accuracy of the MI event predictions were considered as the performance measure. The process can be explained in brief with the following Equation 19, Equation 20 and Equation 21.

$$\hat{y}_i = \operatorname{argmax}(SC(O_L)_i, SC(O_R)_i) \quad (19)$$

$$Prediction_i = \begin{cases} 1, & \text{if } \hat{y}_i = \text{target class} \\ 0, & \text{if } \hat{y}_i \neq \text{target class} \end{cases} \quad (20)$$

$$Accuracy = \frac{\sum_{i=1}^N Prediction_i}{N} \times 100 \% \quad (21).$$

In Equation 19-21, i denotes the trial number, and N denotes the number of trials in the test datasets. $SC(O_L)_i$ and $SC(O_R)_i$ represent the spike counts of the left and right output neurons and the class of output neuron which produces the highest spike count identified as the classifier's prediction. $Prediction$ is a one-dimensional array. Its i^{th} element records whether the classifier's prediction for i^{th} trial is correct. The $Accuracy$ for test datasets are calculated after recording the prediction of each trial and expressed as a percentage of accurate predictions compared to the total number of trials in the considered test dataset.

4.4 Pipelines without the CSP algorithm

To evaluate the effectiveness of the CSP algorithm on these particular pipelines, we employed the same experimental procedures described in Section 4.3 but excluded the CSP step from the pipelines. This modification can be identified as a control experiment of the original pipelines. The aim of this control experiment is testing the ability of our SNN based pipelines to classify MI-EEG data in the absence of the CSP preprocessing step, which has previously been widely used in conventional EEG classification applications [87] without additional adjustments to our pipelines. Figure 11 illustrates the steps in the pipelines of these control experiments.

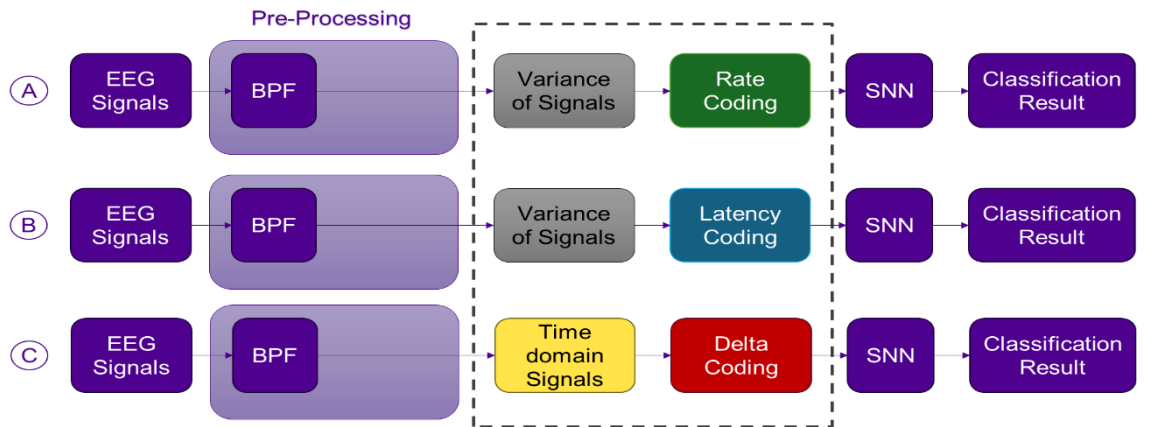


Figure 11. Block diagrams of the three implemented pipelines for the control experiments. Pipelines A, B, and C correspond to the rate coding, latency coding, and delta coding approaches, respectively. In contrast to the proposed pipelines, the CSP method has been removed from here.

5. RESULTS

The raw MI-EEG signals were transformed into spike trains for feeding the SNN classifiers using preprocessing methods and spike encoding schemes in the developed pipelines. This chapter presents their effects and the pipelines' overall performances with illustrations of the data at different stages of the pipelines. The insights gained from these result figures are included in the Discussion.

5.1 Preprocessing of Raw MI-EEG Signals

As discussed in the Materials and Methods section, BPF and CSP filters were employed to preprocess the raw MI-EEG signals (see Subsection 4.3.1). These two steps have been employed to increase the overall accuracy of the pipelines by enhancing the relevant features in the signals. The plots in Figure 12 illustrate the need for these preprocessing steps and the advantage gained through their use.

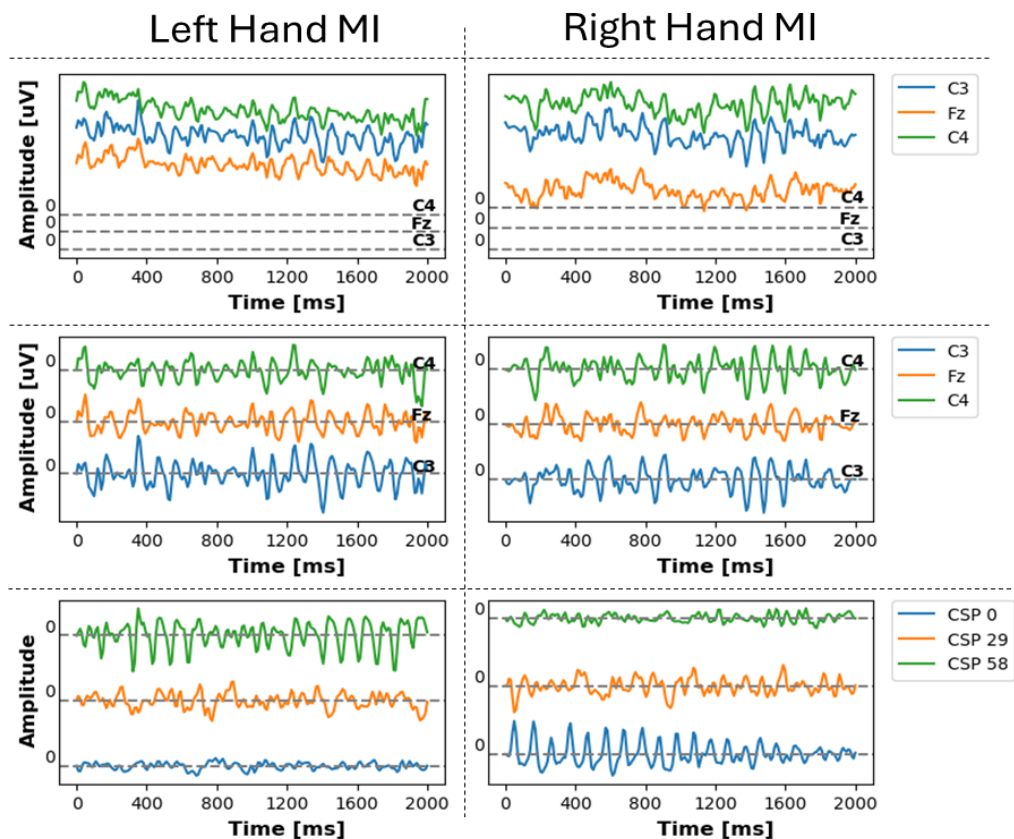


Figure 12. Transformation of raw EEG signals through preprocessing steps. The left and right columns show signals associated with left and right MI events respectively. The first row presents the raw signals, the second and third rows display signals after applying BPF and CSP. Legends represent the names of the electrodes or CSP components.

The significant displacements from the origin axes of the raw signals shown in Figure 12 indicate that the raw signals have large DC offsets. After applying the 2 Hz to 30 Hz BPF, those DC offsets are removed since BPF filters the 0th frequency component. Addition to that, the BPF has reduced the noise components which are beyond 30 Hz. In result, irrelevant fast fluctuations also have been mitigated from the signals as shown in below Figure 13.

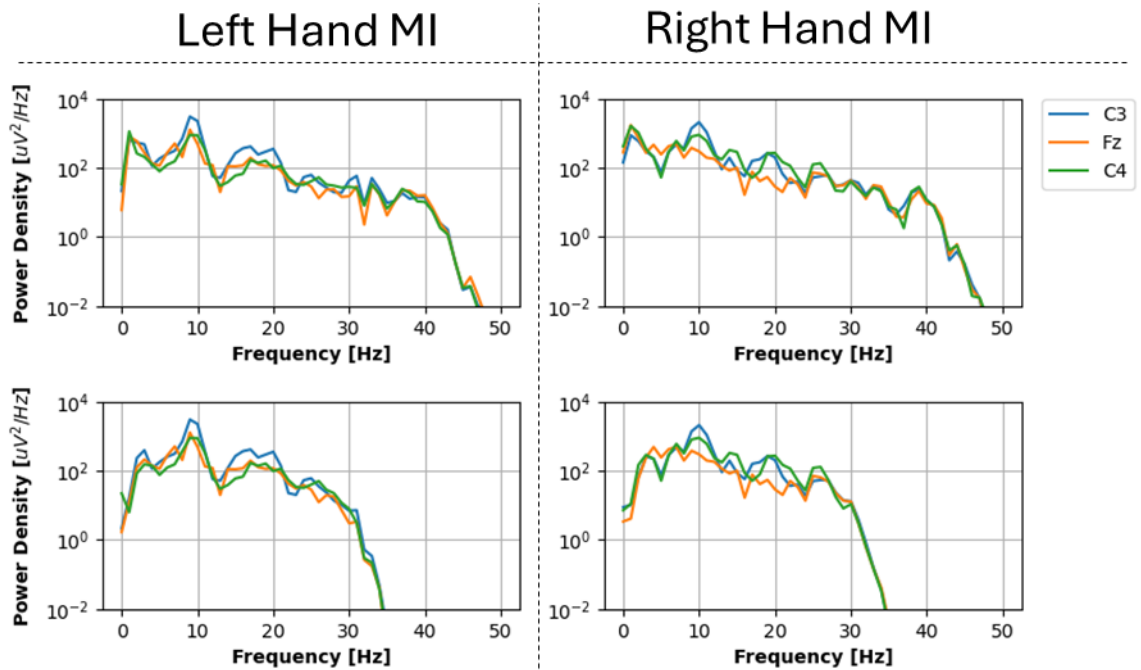


Figure 13. A comparison of power spectra before and after applying BPF. The left and right columns respectively represent the signals associated to left and right MI events while first and second rows represent power spectrum of signals before and after applying BPF. Legend indicates the name of the electrodes.

Usually, slower frequency components (less than 2 Hz) of MI-EEG signals contain the effect of heart activities, breathing artifacts, eyeblinks and other physical movements. Faster frequency components (greater than 30 Hz) of MI-EEG signals usually contain effects of powerline noises, muscle activities and ambient EM waves [88]. As shown in Figure 13, after the BPF, these irrelevant frequency components have been diminished from the MI-EEG signals. With the BPF, the SNR of the MI-EEG signals has been increased while preserving key information related to MI activities, such as mu (~8 Hz - 13 Hz) and beta (~14 Hz - 30 Hz) band powers.

The time domain signals in the third row of the Figure 12 illustrates the effect of CSP spatial filters (which were implemented by using Eq. 1-11). It can be observed that CSP filters have increased the amplitudes of certain signals while decreasing the amplitude of other signals corresponding to the class of MI-EEG signals. This increases the separability between classes, which is a core feature of the CSP algorithm [89]. The effect of

this can be clearly interpreted by using the mean variances of signals as shown in Figure 14.

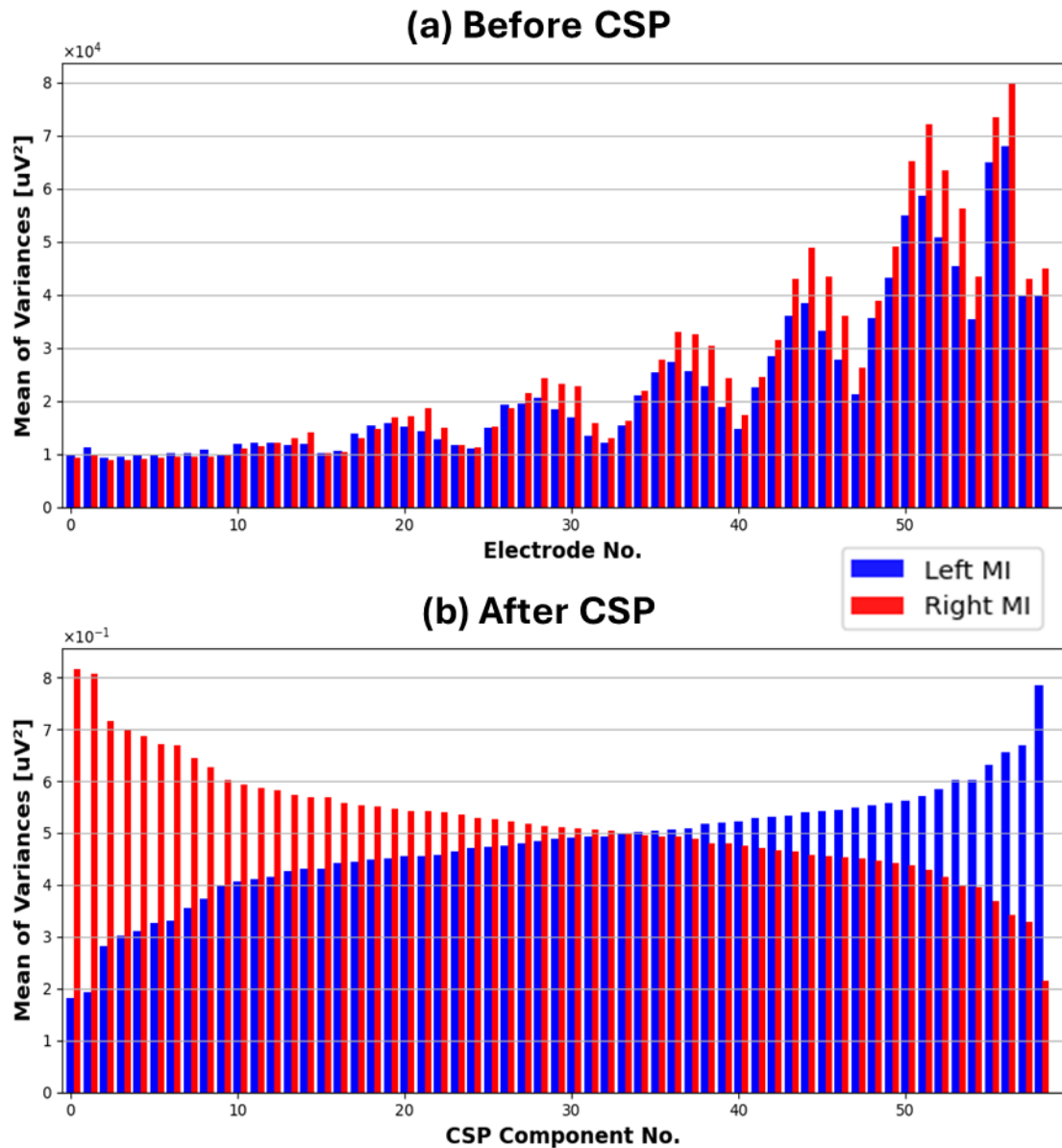


Figure 14. Difference of signal variances before and after the CSP filtering. (a) Average average variance of the EEG signal components after the BPF and before the CSP filtering and (b) average variance of the EEG signal components after the CSP filtering. Blue and red color bars correspond to left trials and right trials respectively.

The CSP components, acting as virtual electrodes, are constructed through linear combinations of each electrode. The variances of signals corresponding to these components show distinguishable differences between classes, as shown in Figure 14(b). For the left MI signals, lower variance values can be found in CSP components closer to 0, while higher variance values can be found in CSP components closer to 58. For the right MI signals, the output of the CSP has the exact opposite pattern.

5.2 Effectiveness of Spike Encoding Schemes

After the preprocessing steps, the signals were encoded into spike trains using the spike encoding schemes specific to each pipeline (see Figure 8). The encoded signals (as described in Eq. 12-14) also demonstrated class separability through distinguishable CSP components. Figure 15 shows sample raster plots after the CSP method.

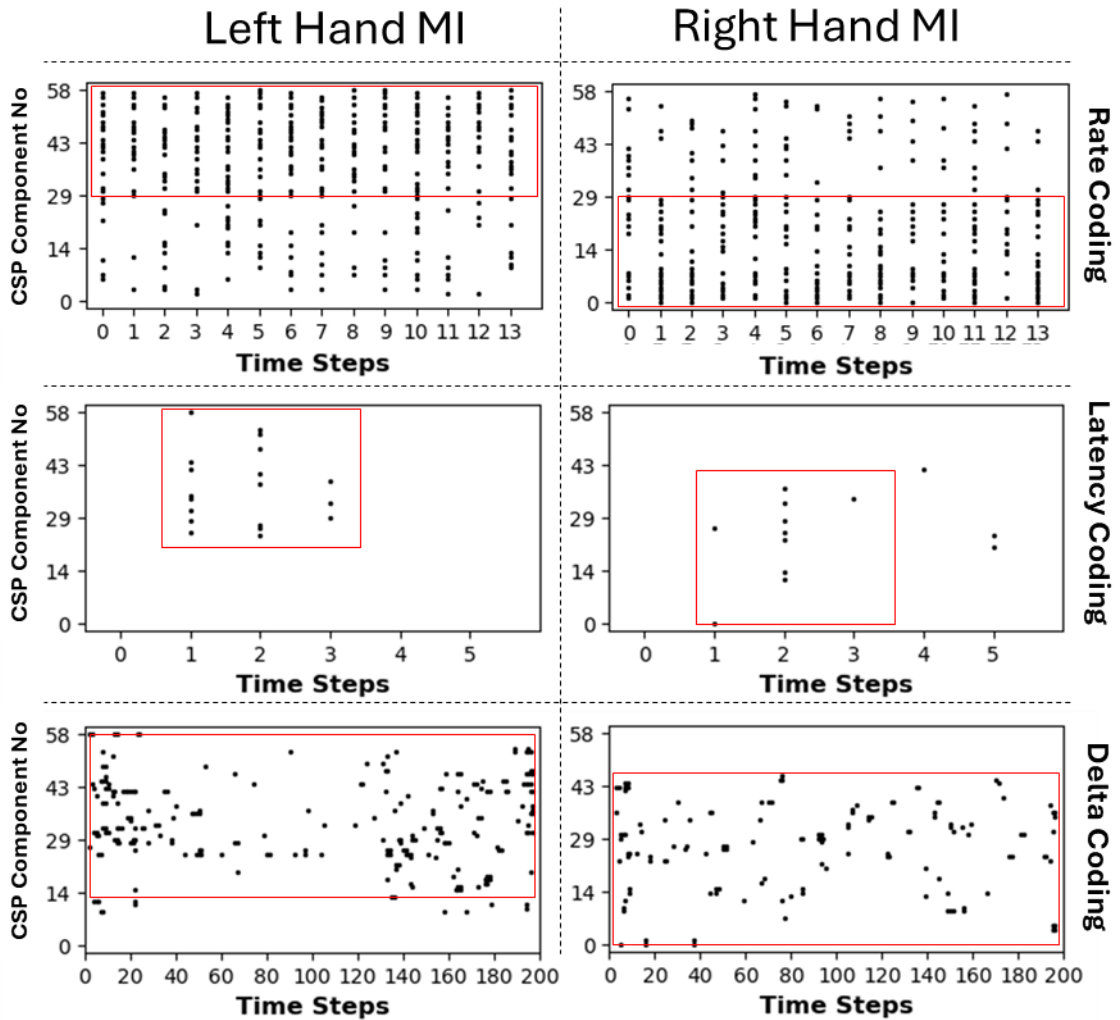


Figure 15. Examples of visually illustrated spike trains. These plots are related to two trials (left MI and Right MI) which are obtained from one subject. Employed coding methods have been mentioned at the right side of each row. Regions with comparatively high spike counts have been surrounded by red rectangles.

The raster plots of Figure 15, show a clear distinguishability between the CSP components for all encoding methods. The red rectangles highlight the areas with higher spikes, indicating that corresponding CSP components are crucial for predicting the associated class labels (mentioned at the top of the column). These qualitative observations prove that each employed spike encoding is capable of preserving the class distinguishing features. The quantitative interpretation of the distinguishability of the generated spike trains is shown with the illustrated histograms in Figure 16.

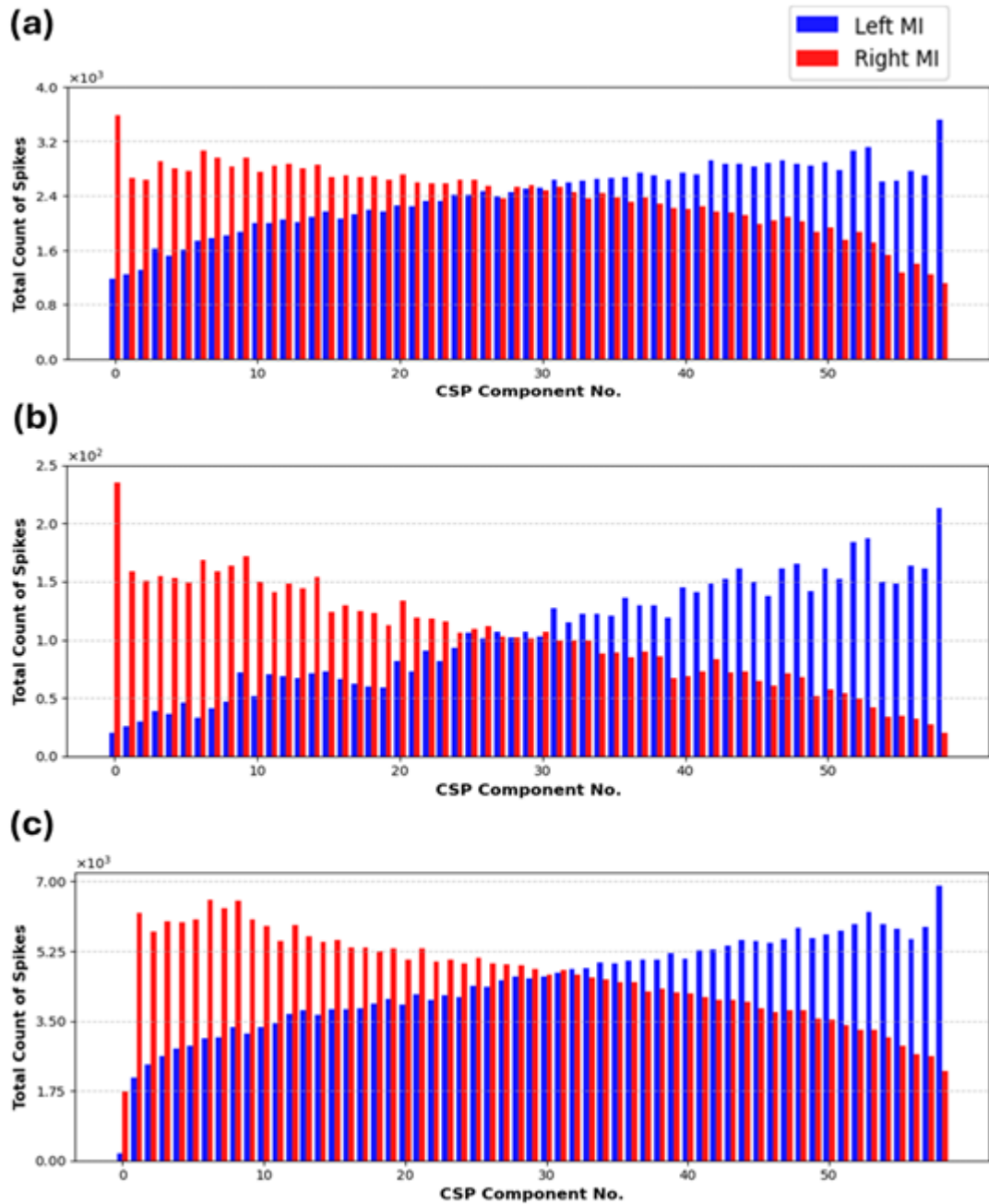


Figure 16. Histograms of spikes across multiple subjects' data. Subfigures are associated with each spiking method: (a) rate-coded spike trains, (b) latency-coded spike trains, and (c) delta-coded spike trains. Blue and red color bars represent the spike counts of CSP components corresponding to left trials and right trials respectively.

Figure 16 illustrates that the generated spike trains also exhibit class separability across subjects and demonstrates the capability of implementing generalized classifiers using all three pipelines to classify both single subject and multi subject data. The visually apparent results in these plots provide sufficient justification for effectively using SNN classifiers in classification tasks.

5.3 Learning Curves

As explained in Subsection 4.3.3, the training process of the SNN based classifier utilized 80% of the entire dataset considered for implementing the classifier. The remaining 20% of the dataset was used to test the model. During training, loss and accuracy values were recorded after each batch iteration for both the training and test datasets. The evolution of these values through the training iterations (learning curves) reflects the effectiveness of the training process.

Figure 17 shows how the learning curves from each type of classifier were reduced during the training phase when the models were trained on data from a single subject (similar behaviors were also observed with data from other individual subjects, which are not presented here).

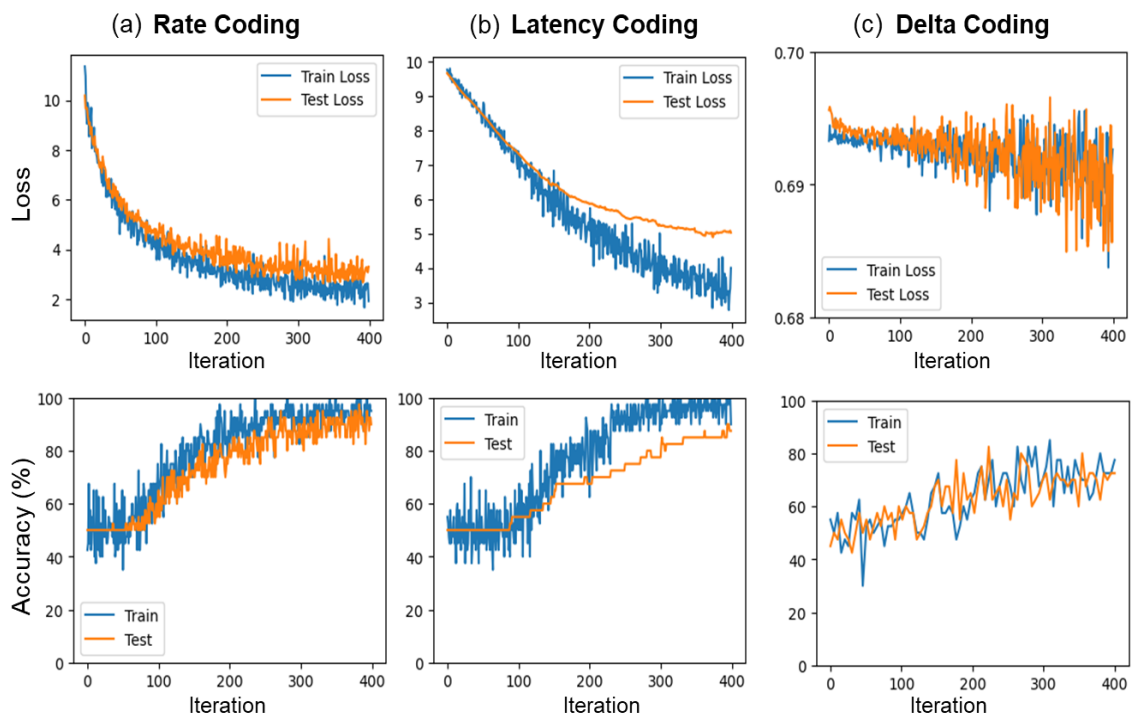


Figure 17. Example set of learning curves related to the classifier models which are trained with a single subject data. Columns (a), (b) and (c) correspond to the rate coding pipeline, latency coding pipeline and delta coding pipeline, respectively. The top row includes the loss curves, and the bottom row includes the accuracy curves.

While the loss curves quantify the model performance on classifying the data used for training, accuracy curves mark the final performance of the models. As shown in Figure 17, the loss curves for each classifier tend to decrease, while the accuracy curves tend to increase. These trends in the learning curves indicate that the implemented pipelines are capable of learning to predict single-subject MI EEG tasks without overfitting to the training data.

In this work, besides using only single-subject data, we implemented generalized models across subjects that can adapt to multiple subjects and predict the labels of trials from multiple subjects by using the same pipelines. Similar to the single subject models, learning curves have been obtained for these generalized models to evaluate the pipelines' applicability across multiple subjects. Figure 18 present the learning curves of the generalized models.

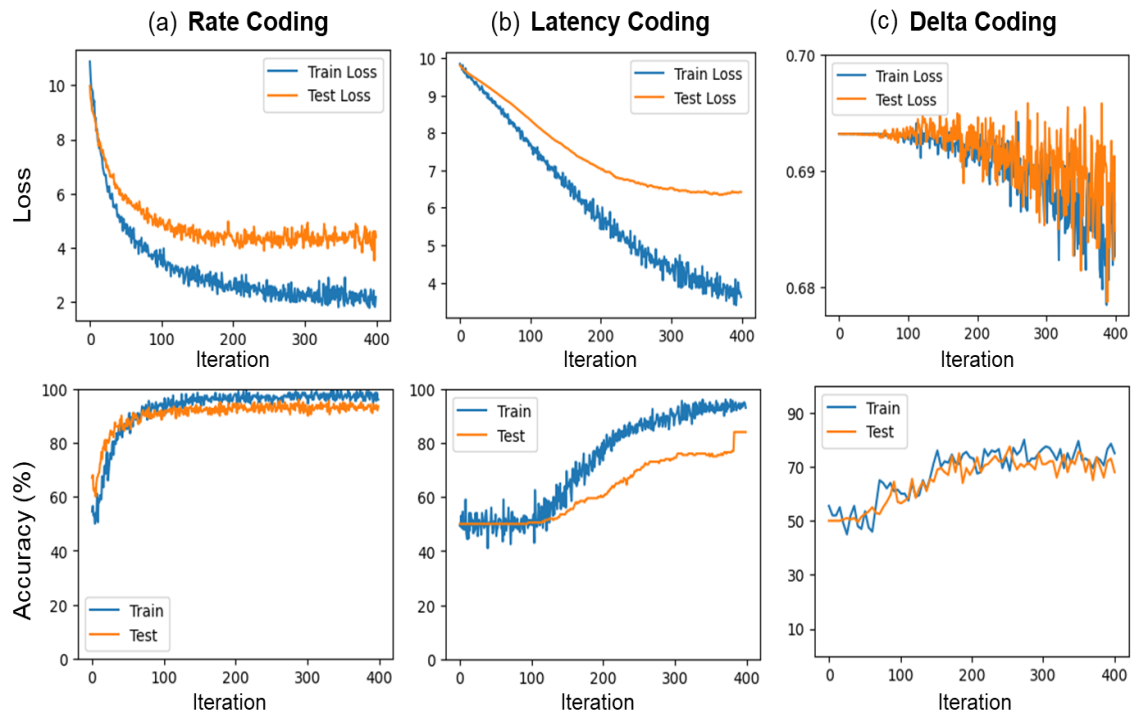


Figure 18. Example set of learning curves related to models which are trained with data from multiple subjects. (a), (b) and (c) columns respectively correspond to the rate coding pipeline, latency coding pipeline and delta coding pipeline. The top row includes the loss curves, and the bottom row includes the accuracy curves.

It can be seen that the learning curves of the generalized models are closely similar to the learning curves of the single-subject models. This suggests that the pipelines are capable of training the MI-EEG classifiers for multiple subject data as well as the single subject data.

5.4 Activations of SNN Layers in Testing Phase

It is important to evaluate the final performances of the models qualitatively. This subsection presents how the activations of neurons have been occurred in the each SNNs models. Figure 19 in below shows examples of the activations of neurons in each layer of a SNN model which is trained with a single subject data.

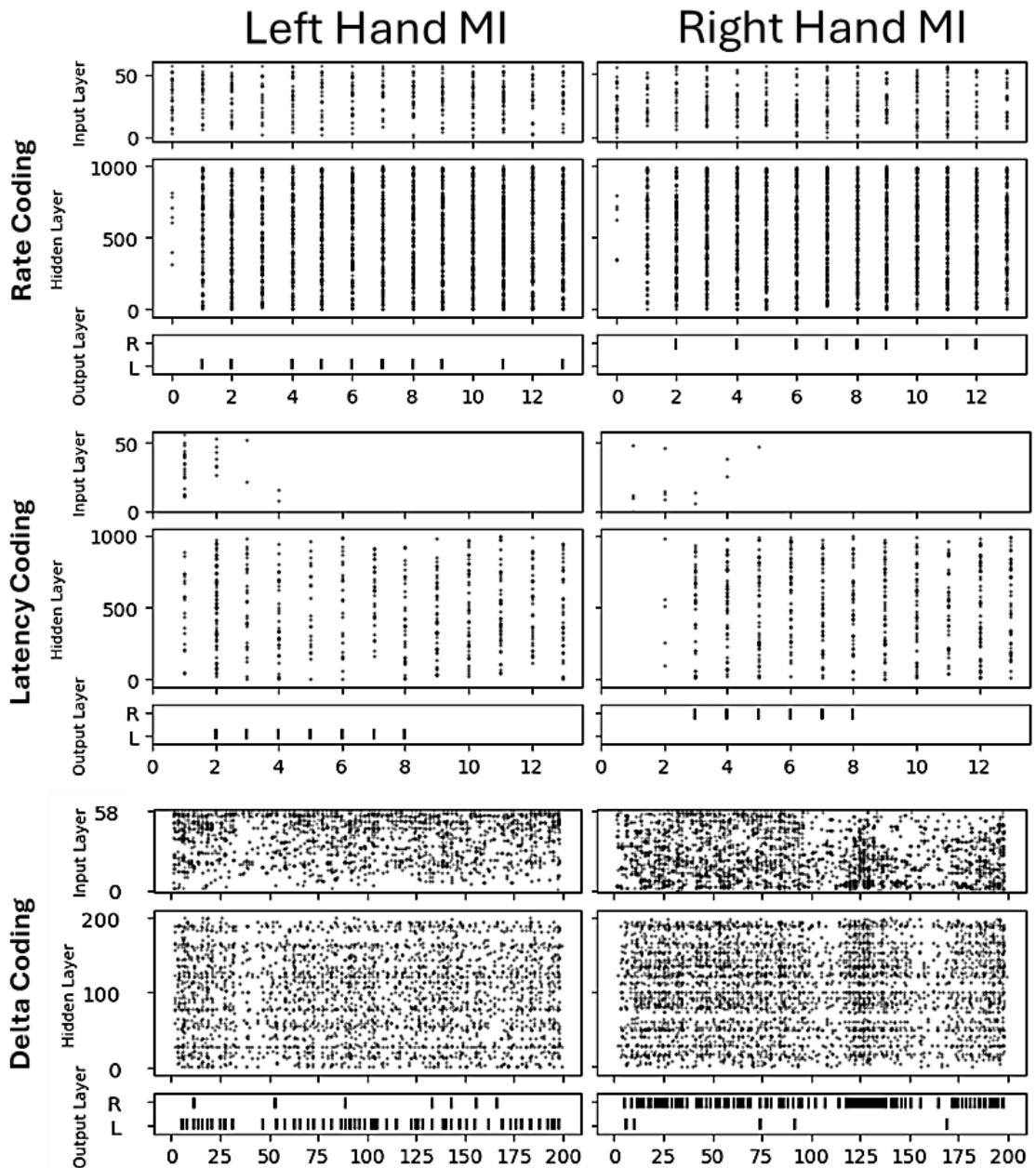


Figure 19. An example of neuron activations in each layer of a SNN trained with a single subject data. The left and right columns display samples from the left and right classes, respectively. The first, second, and third rows represent the types of models: rate coding, latency coding, and delta coding, respectively.

These figures visualize the overall spiking dynamics happening in each layer of SNNs for completing a classification task. It can be seen that the delta coding pipeline has shown the least number of spikes compared to the latency coding method. However, the delta coding pipeline has shown the highest number of spikes because it produced outputs for 200 time steps, while other two methods produced spikes for 14 timesteps. Hence, when comparing the performance of each, that also needs to be considered. As

seen in Figure 19, Figure 20 shows the same information on the SNNs that are trained with multiple subject data.

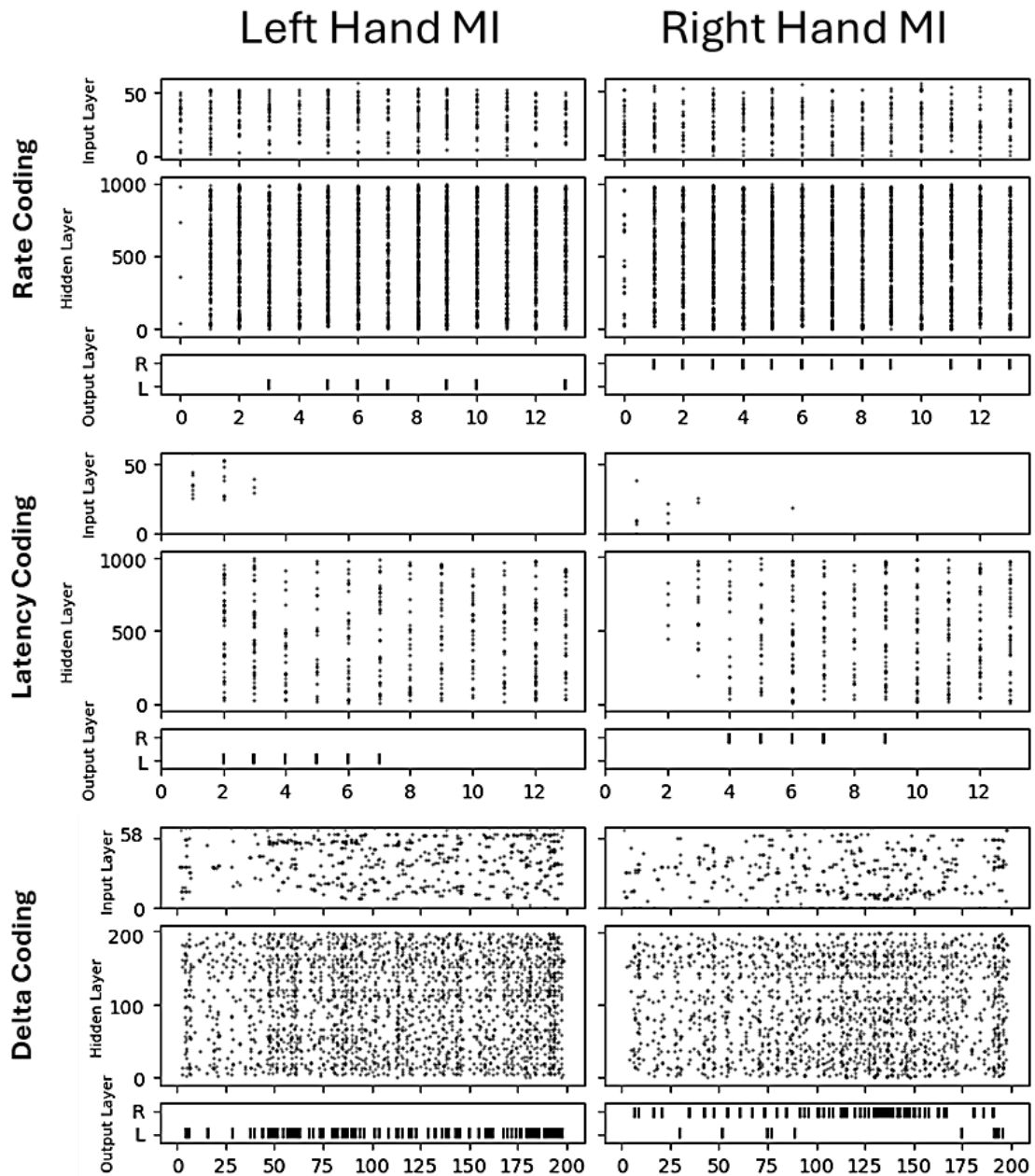


Figure 20. An example of neuron activations in each layer of a SNN trained with data from multiple subjects. The left and right columns display samples from the left and right classes, respectively. The first, second, and third rows represent the types of models: rate coding, latency coding, and delta coding, respectively.

The “L” and “R” output layer neurons of each subplot in Figure 19 and Figure 20 represent the left MI and right MI classes, respectively. The predicted class from the SNN classifiers are determined according to Equation 19 and Equation 20, which compare the total spike counts of “L” and “R” neurons.

5.5 Performance Measures

The classification performance of each train model was obtained by recording their accuracy performances on both the training data and test data for further analysis. Table 6 presents the result of all pipelines that use the CSP method.

Table 6. Accuracy performance of the pipelines (with CSP)

Pipeline Model	Rate Coding		Latency Coding		Delta Coding	
	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)
Subject 1	96.96	91.67	90.84	87.25	72.87	66.92
Subject 2	98.99	94.91	92.38	93.94	75.82	72.20
Subject 3	99.49	97.47	93.44	92.99	77.96	73.94
Subject 4	98.99	97.98	92.46	87.31	74.58	75.69
Subject 5	96.44	92.45	84.35	79.80	71.91	65.10
Multi-Subject	95.36	93.83	92.09	88.73	72.43	72.28
Average	97.69	94.69	90.87	88.21	74.23	70.92

Table 7 presents the accuracy performances recorded from the control experiment's pipelines which do not include the CSP algorithm for enhancing the spatial features prior to the spike encoding schemes.

Table 7. Accuracy performance of control experiments' pipelines (without the CSP)

Pipeline Model	Rate Coding		Latency Coding		Delta Coding	
	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)
Subject 1	53.30	47.79	49.24	49.97	50.91	50.00
Subject 2	51.92	47.18	48.41	50.00	49.53	50.00
Subject 3	50.00	48.76	55.94	50.00	46.27	50.00
Subject 4	49.49	50.88	51.12	50.00	48.48	50.00
Subject 5	54.35	56.80	55.50	56.88	46.88	50.00
Multi-Subject	51.57	51.42	49.44	50.20	47.38	50.00
Average	51.74	50.37	51.52	51.12	48.22	50.00

The lower accuracy performances presented in Table 7 compared to the values in Table 6, it is evident that all three pipelines rely on the CSP technique to achieve proper performance. The values reported in both Table 6 and Table 7 represent the average performance of multiple models, each trained using 5-fold cross-validation on the dataset

under consideration. The use of 5-fold cross-validation ensures that the results are robust and less prone to have erroneous results, as each model is evaluated on different subsets of the data. Figure 21 shows the comparison between the distributions of all the accuracy values which were obtained through the 5-fold method on both training and testing datasets by using the pipelines with and without the CSP method.

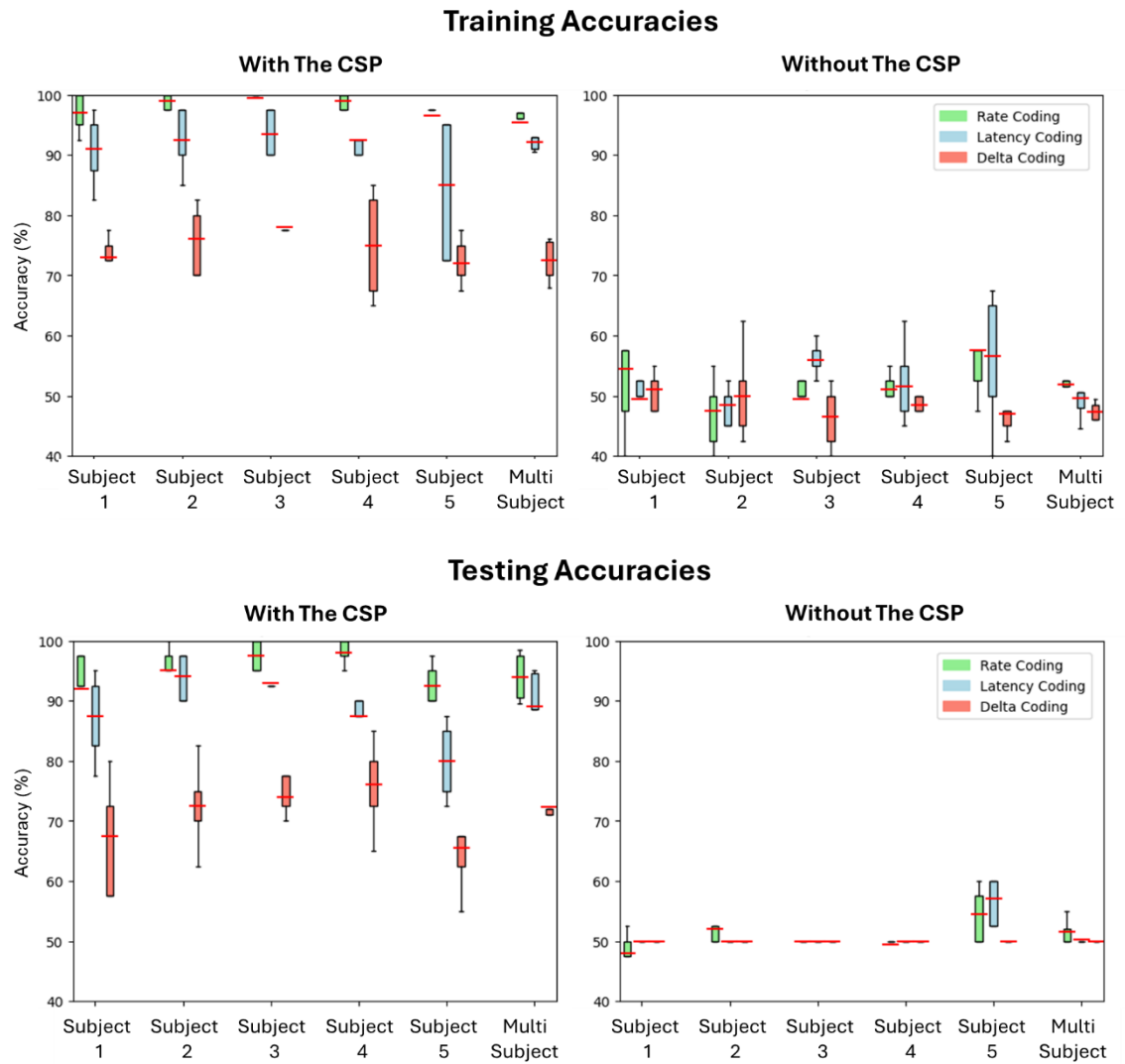


Figure 21. A comparative illustration of accuracy performances on testing datasets with and without the CSP method in pipeline. Green, blue, light red color box and whisker plots show accuracy value distribution related to rate coding, latency coding and delta coding respectively. Dark red horizontal lines represent the mean value of each distribution.

6. DISCUSSION

This chapter provides an in-depth analysis of the insights obtained from the results of this work. Beginning with the purpose of thesis, this chapter highlights key findings of the project and their impact on further advancement of SNN based approaches for implementing MI-EEG classifications. Additionally, this chapter addresses the limitations of the outcomes and discusses potential future directions for continuing the project work.

6.1 The Potential of CSP for SNN based MI-EEG classifiers

As detailed in Introduction, the use of the SNN based pipeline in MI-EEG classification can offer a significant advantage in terms of energy efficiency, particularly for portable devices. These energy efficient advancements can be used in various practical applications including developing exoskeletons, wheelchairs, and other assistive devices in healthcare applications. Nevertheless, implementations of SNN based pipelines for MI-EEG classification have to pose the same challenges as conventional approaches due to the properties of the EEG signals, such as low SNR and complex temporal dynamics [90].

The CSP algorithm has been widely used in MI-EEG classification to address the inherent issues in EEG signals [91],[92]. Since the CSP algorithm transforms the signals into a new set of components (a new set of signals) that maximize the signal variance for one class and minimize the signal variance for the other class, it can increase the separability of the MI-EEG classes. With this outcome, conventional ML approaches can efficiently train models to classify the EEG signals [45].

These differences in the sample variance of the CSP output signals can be seen in the time domain signals. For one class, one set of CSP components shows time domain signals with higher amplitude values, while the other set of CSP components shows time domain signals with lower amplitude values. And for the other class, time domain signals show the opposite of this behavior (see time domain signals in Figure 12). This unique feature shows the potential of CSP output signals to convert into spikes by using all three major spike encoding methods without distorting critical information. Inspired by this, this thesis work was focused on evaluating the suitability of using CSP spatial filters to boost the performance of SNN classifiers for MI-EEG task classifications.

However, the CSP method is not a spiking-friendly algorithm. Therefore, it is not ideal to implement a complete pipeline for deploying on a neuromorphic chip. The results of our

work emphasize the necessity of a preprocessing step resulting in a set of signals similar to the output of CSP signals.

6.2 Evaluation of each Pipeline

Based on the three major spike encoding methods (rate coding, latency coding and delta coding), this work has implemented three pipelines to train SNN based MI-EEG classifier models as elaborated in the Materials and Methods. Also, as mentioned there, the “BCI Competition IV—Dataset 1” was used to train the classifier models and assess the performance of the pipelines in terms of training efficiency, accuracy, and energy efficiency (the recorded performances of each pipeline are presented in the Results).

- **Training efficiency**

Both loss and accuracy curves of the single subjects’ models from all three pipelines (see Figure 17) have shown that they are trying to learn something from training data to predict test data accurately. However, when considering training efficiency, the rate coding pipelines have performed well by minimizing the loss and maximizing the accuracy with the minimum training iterations. Compared to the rate coding pipelines, the latency coding pipelines have underperformed by showing slow convergence. In addition to that, there is an offset between the test and training curves of latency coding, which means there is an overfitting issue when we use the latency coding pipelines for training the models. And the learning curves related to delta coding have shown a very chaotic oscillating behavior. However, those have still shown some positive changes (a decrease in losses and an increase in accuracy).

In multiple subject models (see Figure 18), learning curves related to each pipeline have shown similar performance like in single subject models with bit of an underperformance. One possible reason for that is the inability of the basic SNN architectures to adapt to the variability of the multiple subjects. This could be handled with advanced SNN architectures and using model optimizers with the regularization techniques.

However, the overall training efficiencies are depended on the used surrogate gradient based backpropagation algorithm. Hence, these behaviors may change when a biologically plausible learning algorithm that is more compatible with spiking neurons is employed for optimizing the weights. While backpropagation algorithms aim to optimize synaptic weights to minimize loss, biologically plausible algorithms may pursue different objectives in weight optimization.

- **Accuracy performance**

Even though the primary goals of using the spiking based algorithms are focused on the energy consumption of intelligence systems, maintaining an acceptable accuracy performance is also important. According to the accuracies presented in Table 6, all proposed pipelines have shown their potential to produce classifiers for classifying MI-EEG signals with good accuracy levels. When comparing with conventional works, which achieved average accuracy from ~70% to ~99 % (a set of examples are shown in Table 3) and considering the objectives and the scope of this work, the accuracy performances of this work can be considered as acceptable.

Not only conventional methods based studies, but there are also SNN based approaches (examples shown in Table 3) that have achieved similar performances in MI-EEG classifying with different methods on different datasets [19]-[24]. That also implies that the performances which have been achieved by this work are acceptable. In contrast to those SNNs, these CSP based pipelines produce spiking trains with higher separability, allowing SNNs to adapt with a small amount of data.

The best average accuracy has been recorded from the rate coding pipeline by achieving 94.69% average test accuracy from all the datasets that have been used. The latency coding pipelines have recorded slightly less accuracy performance compared to the rate coding pipelines. The delta coding pipelines have reported the lowest accuracy performance by showing the necessity for further improvements. It could be improved by addressing its associated high variability in input spiking patterns, which is highly dependent on the threshold level used to generate the spikes. This could be handled with a 2nd order input neuron layer as those neurons can adapt to the input signals/spikes accordingly and produce spikes only when they carry significant information.

- **Energy efficiency**

Energy efficiency associated with spiking based processing is the main reason for transitioning from conventional computing architectures to spiking neural architectures. Unlike traditional digital systems, spiking based processing units do not require a clock signal to process the information. Spiking based processing units process information asynchronously only when there is significant information to cause a spike. So, the spikes are the events that are responsible for energy consumption in spiking based processing units. In ideal situations, these units consume zero energy in no-spike conditions. Due to these behaviors, the spiking counts in operations are directly proportional to the energy consumption of those operations.

According to the outcomes of each spike encoding scheme (see Figure 16) in this work, the latency coding has shown the lowest number of spiking counts as the inputs to the SNNs. Thus, the latency code has shown the highest energy efficiency in producing the spike trains compared to the other two methods. In the same way, the rate coding has shown 2nd best energy efficiency associated with the spike encoding methods. The delta coding has shown the worst energy efficiency in encoding results. The main reason for this is that delta coding produces spikes for 200 time steps in a trial, while rate coding and latency coding methods only produce spikes for 14 time steps.

The outcome of the spike encoding method directly affected the number of neuronal action potentials (spikes) in the layers of SNN, because input spikes propagate along with the neurons. This can be observed in Figure 19 and Figure 20. As those spiking activations imply, the overall best energy performances also have been achieved by the latency coding pipeline models. The rate coding pipeline models have achieved the second best overall energy efficiency, making the delta coding pipeline models as the worst models for energy efficiency.

Despite delta coding exhibiting lower accuracy and energy efficiency performance in this work, its hardware circuitry is simpler and more efficient than those of latency coding and rate coding. Delta coding can capture data directly in spike form, which is more energy efficient than methods that convert captured signals into spikes [17]. In this study, delta coding has shown its capability of producing continuous spike trains effectively from continuous (although preprocessed) input signals. This is especially beneficial for building SNN classifier models to produce output spikes in real-time without having any operational level delay. Therefore, addressing the issues of delta coding to improve the accuracy level and energy efficiency is highly advantageous for making hardware systems.

6.3 Impact of the CSP method

In order to understand the contribution of the CSP method, within this work, we had done a set of control experiments by using the pipelines without incorporating the CSP method (see Section 4.4). As originally proposed pipelines, these modified pipelines were used to implement classifier models and to test them on different datasets. After training the classifiers (without the CSP), the obtained accuracy values on different datasets have been presented in Table 7.

For a bi-class classifier, achieving a 50% accuracy on a balanced dataset is the weakest performance. Even a bi-class classifier which is completely biased to one class can achieve 50%. According to the values in Table 6 and Table 7 and the distributions of the

obtained accuracy values illustrated in Figure 21, it is clear that without the CSP algorithm, the classifiers have completely lost their capability for adapting to MI-EEG signals and predicting the class of the MI tasks. Further, this necessitates a spiking based spatial feature enhancing method for our overall pipelines which can perform similarly to the CSP method.

6.4 Limitations

Although the implemented pipelines have shown promising results, their applicability is limited to this study due to the scope of the thesis. Therefore, it is important to understand the limitations of this work and consider the robustness and applicability of the pipelines before using and improving them in future work.

One significant limitation pertains related to the data set used for this study. For implementing and evaluating the pipelines, only the 'BCI Competition IV – dataset 1' was utilized in this thesis work. Since this dataset was collected from the same experimental setup, the results obtained from this experiment could be overfitted to this experimental setup. Although it can be presumed that these pipelines will work on different datasets due to the solid structure, it cannot be guaranteed. MI-EEG patterns can vary significantly between subjects according to their mental status, their familiarity with the paradigm and also the paradigm that is used to record the signals (e.g., some paradigms solely use visual cues - calibration dataset from [66], some paradigms use acoustic cues instead of visual cues - evaluation dataset from [66], some paradigms use acoustic alerts along with visual cues [67],[68]). Furthermore, technical aspects of the recording device (such as electrode placement, resolution, and signal precision) can impact the data quality and ultimately affect the accuracy and reliability of the results. Hence, the performance of pipelines should be validated across the datasets (a few sets of datasets are included in Table 2).

In addition to dataset limitations, the weight initializing of the CSP filters is another concern. The CSP method uses the samples of labelled trials to initialize the weights, and the number of these used trials positively affects the performances of the CSP filters [89]. Here in this work, considering the limited number of available trials in the datasets, all the trials were used to initialize the spatial filters to achieve high separability through the CSP filters. Consequently, newer signals to the pipelines might not show the same level of separability after passing through the CSP filters, which could also change the pipelines' overall performance.

Another limitation is related to our selections of methods for encoding schemes. When implementing encoding schemes, the rate and latency coding methods can also be applied directly to time-domain signals to generate spikes instead of using single-entry feature values. However, achieving effective encoding with these methods requires considering multiple time steps to generate spikes for a single input sample (see Subsection 4.3.2). This increases the amount of data to process and adds complexity to the pipeline, necessitating further modifications to improve classifier performance. Considering that and the scope of this work, the variances of single trials were used as single entry features to obtain spike trains. Nevertheless, these rate and latency coding methods may achieve greater accuracy and efficiency by directly dealing with time-domain signals compared to delta coding pipelines.

6.5 Future Directions

Ultimately, this project will lead to the implementation of a complete neuromorphic pipeline that can be used to build MI-EEG classifiers in neuromorphic hardware. So, as for future work, there are mainly three things that need to be worked on.

1. Replacing the CSP method with spiking friendly algorithm which mimics the transformation happening in the CSP method.
2. Changing the learning algorithm of SNNs to a neuromorphic (biologically plausible) learning algorithm.
3. Validate the complete pipeline in a neuromorphic simulation environment and deploy it on a neuromorphic hardware chip.

Through the spike based spatial filters, which can alternate the CSP method, the separability of the MI-EEG signals can be enhanced to allow the use of energy efficient spike based learning algorithms to train SNN based classifiers optimally. That will lead to the implementation of the complete spike neuromorphic pipeline for building the MI-EEG classifier models. Validating such a pipeline in a simulation environment will ensure the way for seamless deployment on neuromorphic hardware chips, demonstrating its practical capabilities as well as challenges.

7. CONCLUSIONS

This thesis work was initiated with the primary intention of developing an SNN based pipeline to implement an MI-EEG classifier, with a focus on harnessing the energy efficiency of spiking based algorithms. In our preliminary studies, we have understood that the CSP algorithm has the potential for transforming MI-EEG signals into a form that can be effectively encoded into spike trains and that facilitate the separability of the MI-. To validate this, three novel types of pipelines were designed based on the spike encoding method (rate coding, latency coding, and delta coding) by incorporating the CSP algorithm to analyze the performances (see Materials and Methods).

According to the designed pipelines, a set of MI-EEG classifier models were implemented and evaluated using the calibration dataset of the “BCI Competition IV—Dataset 1” (which has labelled trials). A surrogate gradient decent based backpropagation algorithm was used to update the (synaptic) weights in the SNN classifier models according to the training datasets. After that, the trials in the testing datasets were classified using the trained SNN models to evaluate the performances of the pipelines. Table 8 briefly presents outcome of these performance analysis (for comprehensive details see Results and Discussion).

Table 8. Performances of each pipeline in brief.

Pipeline	Training efficiency	Accuracy (%)	Energy Efficiency
Rate Coding	+ Minimizes loss + Maximizes accuracy + Fast convergence	+ Highest	+ Medium
Latency Coding	+ Minimizes loss + Maximizes accuracy - Slow convergence - Exhibits overfitting	+ Medium	+ Highest
Delta Coding	+ Decreasing loss + Increasing accuracy - Chaotic oscillating learning curves	- Lowest but acceptable (Requires improvements)	- Lowest (Requires improvements)

The performance results in this study show that the proposed pipelines are able to classify the MI-EEG signals. Additionally, a comparative study done in this work suggests that without the CSP method, all three pipelines were unable to achieve acceptable accuracy performance. This validates our presumption about the CSP algorithm. In these experiments, the CSP algorithm transformed the MI-EEG signals into a newer set of signals that can effectively encode into spike trains without losing the critical information to separate them. Nonetheless, the CSP algorithm is not a spiking friendly algorithm. Hence, it needs to be alternated in future work with a spiking friendly algorithm which can produce similar outcome to implement complete neuromorphic pipeline.

In summary, this work successfully demonstrates that integrating the CSP algorithm to enhance spatial features enables SNNs to achieve excellent accuracy in MI-EEG classification. Future efforts to develop comprehensive neuromorphic pipelines can build on this study as a valuable foundation.

8. REFERENCES

- [1] P. Sawangjai, S. Hompoonsup, P. Leelaarporn, S. Kongwudhikunakorn, and T. Wilaiprasitporn, "Consumer Grade EEG Measuring Sensors as Research Tools: A Review," *IEEE Sens. J.*, vol. 20, no. 8, pp. 3996–4024, Apr. 2020, doi: 10.1109/JSEN.2019.2962874.
- [2] A. M. Alvi, S. Siuly, and H. Wang, "Neurological abnormality detection from electroencephalography data: a review," *Artif. Intell. Rev.*, vol. 55, no. 3, pp. 2275–2312, Mar. 2022, doi: 10.1007/s10462-021-10062-8.
- [3] J. P. Lachaux, D. Rudrauf, and P. Kahane, "Intracranial EEG and human brain mapping," *J. Physiol.-Paris*, vol. 97, no. 4–6, pp. 613–628, Jul. 2003, doi: 10.1016/j.jphysparis.2004.01.018.
- [4] E. Ryding, J. Decety, H. Sjöholm, G. Stenberg, and D. H. Ingvar, "Motor imagery activates the cerebellum regionally. A SPECT rCBF study with 99mTc-HMPAO," *Cogn. Brain Res.*, vol. 1, no. 2, pp. 94–99, Apr. 1993, doi: 10.1016/0926-6410(93)90015-W.
- [5] I. Brunner, C. B. Lundquist, A. R. Pedersen, E. G. Spaich, S. Dosen, and A. Savic, "Brain computer interface training with motor imagery and functional electrical stimulation for patients with severe upper limb paresis after stroke: a randomized controlled pilot trial," *J. NeuroEngineering Rehabil.*, vol. 21, no. 1, p. 10, Jan. 2024, doi: 10.1186/s12984-024-01304-1.
- [6] G. Onose *et al.*, "On the feasibility of using motor imagery EEG-based brain–computer interface in chronic tetraplegics for assistive robotic arm control: a clinical test and long-term post-trial follow-up," *Spinal Cord*, vol. 50, no. 8, pp. 599–608, Aug. 2012, doi: 10.1038/sc.2012.14.
- [7] G. Prapas, K. Glavas, K. D. Tzimourta, A. T. Tzallas, and M. G. Tsipouras, "Mind the Move: Developing a Brain-Computer Interface Game with Left-Right Motor Imagery," *Information*, vol. 14, no. 7, p. 354, Jun. 2023, doi: 10.3390/info14070354.
- [8] Bin He, Lin Yang, C. Wilke, and Han Yuan, "Electrophysiological Imaging of Brain Activity and Connectivity—Challenges and Opportunities," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 7, pp. 1918–1931, Jul. 2011, doi: 10.1109/TBME.2011.2139210.
- [9] M. Li, X. Luo, J. Yang, and Y. Sun, "Applying a Locally Linear Embedding Algorithm for Feature Extraction and Visualization of MI-EEG," *J. Sens.*, vol. 2016, 2016, doi: 10.1155/2016/7481946.
- [10] P. Geethanjali, Y. K. Mohan, and J. Sen, "Time domain Feature extraction and classification of EEG data for Brain Computer Interface," in *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, Chongqing, Sichuan, China: IEEE, May 2012, pp. 1136–1139. doi: 10.1109/FSKD.2012.6234336.
- [11] R. T. Schirrmeyer *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, Nov. 2017, doi: 10.1002/hbm.23730.
- [12] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg, "Designing optimal spatial filters for single-trial EEG classification in a movement task," *Clin. Neurophysiol.*, vol. 110, no. 5, pp. 787–798, May 1999, doi: 10.1016/S1388-2457(98)00038-8.
- [13] P. Lu, D. Yuan, Y. Lou, C. Liu, and S. Huang, "Single-trial identification of motor imagery EEG based on HHT and SVM," *Lect. Notes Electr. Eng.*, vol. 256 LNEE, pp. 681–689, 2013, doi: 10.1007/978-3-642-38466-0_75.
- [14] L. Wang and Z. Li, "Eeg classification based on common spatial pattern and lda," presented at the Proceedings of International Conference on Artificial Life and Robotics, 2020, pp. 809–812. doi: 10.5954/ICAROB.2020.OS15-5.
- [15] T. Wang, E. Dong, S. Du, and C. Jia, "A Shallow Convolutional Neural Network for Classifying MI-EEG," presented at the Proceedings - 2019 Chinese Automation Congress, CAC 2019, 2019, pp. 5837–5841. doi: 10.1109/CAC48633.2019.8996981.
- [16] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997, doi: 10.1016/S0893-6080(97)00011-7.
- [17] J. K. Eshraghian *et al.*, "Training Spiking Neural Networks Using Lessons From Deep Learning," *Proc. IEEE*, vol. 111, no. 9, pp. 1016–1054, Sep. 2023, doi: 10.1109/JPROC.2023.3308088.

- [18] G. Zhan *et al.*, “Applications of Spiking Neural Network in Brain Computer Interface,” in *2021 9th International Winter Conference on Brain-Computer Interface (BCI)*, Gangwon, Korea (South): IEEE, Feb. 2021, pp. 1–6. doi: 10.1109/BCI51272.2021.9385361.
- [19] N. Kumar, G. Tang, R. Yoo, and K. P. Michmizos, “Decoding eeg with spiking neural networks on neuromorphic hardware,” *Trans. Mach. Learn. Res.*, 2022.
- [20] Z. Tayeb, E. Ercelik, and J. Conradt, “Decoding of motor imagery movements from EEG signals using SpiNNaker neuromorphic hardware,” in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, Shanghai, China: IEEE, May 2017, pp. 263–266. doi: 10.1109/NER.2017.8008341.
- [21] M. Pals, R. J. P. Belizon, N. Berberich, S. K. Ehrlich, J. Nassour, and G. Cheng, “Demonstrating the Viability of Mapping Deep Learning Based EEG Decoders to Spiking Networks on Low-powered Neuromorphic Chips,” in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Mexico: IEEE, Nov. 2021, pp. 6102–6105. doi: 10.1109/EMBC46164.2021.9629621.
- [22] C. D. Virgilio G, H. Sossa, J. M. Antelis, and L. E. Falcón, “Motor Imagery Task Classification in EEG Signals with Spiking Neural Network,” in *Pattern Recognition*, vol. 11524, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. A. Olvera-López, and J. Salas, Eds., in *Lecture Notes in Computer Science*, vol. 11524. , Cham: Springer International Publishing, 2019, pp. 14–24. doi: 10.1007/978-3-030-21077-9_2.
- [23] Z. Liu, L. Wang, and S. Xu, “Time-frequency analysis and spiking neural network for motor imagery EEG classification,” in *2022 China Automation Congress (CAC)*, Xiamen, China: IEEE, Nov. 2022, pp. 6458–6462. doi: 10.1109/CAC57257.2022.10054980.
- [24] C. Zhang, W. Pan, and C. Della Santina, “NiSNN-A: Non-iterative Spiking Neural Networks with Attention with Application to Motor Imagery EEG Classification,” 2023, *arXiv*. doi: 10.48550/ARXIV.2312.05643.
- [25] D. M. Lovinger, “Communication networks in the brain: neurons, receptors, neurotransmitters, and alcohol,” *Alcohol Res. Health J. Natl. Inst. Alcohol Abuse Alcohol.*, vol. 31, no. 3, pp. 196–214, 2008.
- [26] “VIII. A dynamical theory of the electromagnetic field,” *Philos. Trans. R. Soc. Lond.*, vol. 155, pp. 459–512, Dec. 1865, doi: 10.1098/rstl.1865.0008.
- [27] “Electrical Signaling in Neurons,” in *Functional Neuroanatomy and Clinical Neuroscience: Foundations for Understanding Disorders of Cognition and Behavior*, 2023, pp. 14–23. doi: 10.1093/oso/9780190943608.003.0002.
- [28] M. a. B. Brazier and J. U. Casby, “Cross-correlation and autocorrelation studies of electroencephalographic potentials,” *Electroencephalogr. Clin. Neurophysiol.*, vol. 4, no. 2, pp. 201–211, May 1952, doi: 10.1016/0013-4694(52)90010-2.
- [29] B. Zou, Y. Zheng, M. Shen, Y. Luo, L. Li, and L. Zhang, “BEATS: An Open-Source, High-Precision, Multi-Channel EEG Acquisition Tool System,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 16, no. 6, pp. 1287–1298, 2022, doi: 10.1109/TBCAS.2022.3230500.
- [30] F. Christidi *et al.*, “A Comprehensive Review on the Role of Resting-State Functional Magnetic Resonance Imaging in Predicting Post-Stroke Motor and Sensory Outcomes,” *Neurol. Int.*, vol. 16, no. 1, Art. no. 1, Feb. 2024, doi: 10.3390/neurolint16010012.
- [31] M. Yang, Z. Yang, T. Yuan, W. Feng, and P. Wang, “A Systemic Review of Functional Near-Infrared Spectroscopy for Stroke: Current Application and Future Directions,” *Front. Neurol.*, vol. 10, p. 58, Feb. 2019, doi: 10.3389/fneur.2019.00058.
- [32] P. Grover and P. Venkatesh, “An Information-Theoretic View of EEG Sensing,” *Proc. IEEE*, vol. 105, no. 2, pp. 367–384, Feb. 2017, doi: 10.1109/JPROC.2016.2615179.
- [33] Y. Qu *et al.*, “Methodological issues of the central mechanism of two classic acupuncture manipulations based on fNIRS: suggestions for a pilot study,” *Front. Hum. Neurosci.*, vol. 16, p. 1103872, Feb. 2023, doi: 10.3389/fnhum.2022.1103872.
- [34] J. P. Pijn, J. Van Neerven, A. Noest, and F. H. Lopes Da Silva, “Chaos or noise in EEG signals; dependence on state and brain site,” *Electroencephalogr. Clin. Neurophysiol.*, vol. 79, no. 5, pp. 371–381, Nov. 1991, doi: 10.1016/0013-4694(91)90202-F.
- [35] X. Jiang, G.-B. Bian, and Z. Tian, “Removal of Artifacts from EEG Signals: A Review,” *Sensors*, vol. 19, no. 5, p. 987, Feb. 2019, doi: 10.3390/s19050987.
- [36] C. Kaur, P. Singh, and S. Sahni, “EEG Artifact Removal System for Depression Using a Hybrid Denoising Approach,” *Basic Clin. Neurosci.*, vol. 12, no. 4, pp. 465–476, 2021, doi: 10.32598/bcn.2021.1388.2.

- [37] J. Decety and D. H. Ingvar, "Brain structures participating in mental simulation of motor behavior: A neuropsychological interpretation," *Acta Psychol. (Amst.)*, vol. 73, no. 1, pp. 13–34, 1990, doi: 10.1016/0001-6918(90)90056-L.
- [38] G. Pfurtscheller, C. Neuper, A. Schlogl, and K. Lugger, "Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters," *IEEE Trans. Rehabil. Eng.*, vol. 6, no. 3, pp. 316–325, 1998, doi: 10.1109/86.712230.
- [39] G. Pfurtscheller, Ch. Neuper, D. Flotzinger, and M. Pregenzer, "EEG-based discrimination between imagination of right and left hand movement," *Electroencephalogr. Clin. Neurophysiol.*, vol. 103, no. 6, pp. 642–651, Dec. 1997, doi: 10.1016/S0013-4694(97)00080-1.
- [40] G. Pfurtscheller and C. Neuper, "Motor imagery activates primary sensorimotor area in humans," *Neurosci. Lett.*, vol. 239, no. 2–3, pp. 65–68, 1997, doi: 10.1016/S0304-3940(97)00889-6.
- [41] T. N. Aflalo and M. S. A. Graziano, "Possible Origins of the Complex Topographic Organization of Motor Cortex: Reduction of a Multidimensional Space onto a Two-Dimensional Array," *J. Neurosci.*, vol. 26, no. 23, pp. 6288–6297, Jun. 2006, doi: 10.1523/JNEUROSCI.0768-06.2006.
- [42] M. D. Crutcher and G. E. Alexander, "Movement-related neuronal activity selectively coding either direction or muscle pattern in three motor areas of the monkey," *J. Neurophysiol.*, vol. 64, no. 1, pp. 151–163, Jul. 1990, doi: 10.1152/jn.1990.64.1.151.
- [43] A. Riehle and J. Requin, "Monkey primary motor and premotor cortex: single-cell activity related to prior information about direction and extent of an intended movement," *J. Neurophysiol.*, vol. 61, no. 3, pp. 534–549, Mar. 1989, doi: 10.1152/jn.1989.61.3.534.
- [44] Y. Jeon, C. S. Nam, Y.-J. Kim, and M. C. Whang, "Event-related (De)synchronization (ERD/ERS) during motor imagery tasks: Implications for brain–computer interfaces," *Int. J. Ind. Ergon.*, vol. 41, no. 5, pp. 428–436, Sep. 2011, doi: 10.1016/j.ergon.2011.03.005.
- [45] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K. Muller, "Optimizing Spatial filters for Robust EEG Single-Trial Analysis," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 41–56, 2008, doi: 10.1109/MSP.2008.4408441.
- [46] A. Javanshir, T. T. Nguyen, M. A. P. Mahmud, and A. Z. Kouzani, "Training Spiking Neural Networks with Metaheuristic Algorithms," *Appl. Sci.*, vol. 13, no. 8, Art. no. 8, Jan. 2023, doi: 10.3390/app13084809.
- [47] P. Pietrzak, S. Szczęsny, D. Huderek, and Ł. Przyborowski, "Overview of Spiking Neural Network Learning Approaches and Their Computational Complexities," *Sensors*, vol. 23, no. 6, Art. no. 6, Jan. 2023, doi: 10.3390/s23063037.
- [48] V. Sakellariou and V. Paliouras, "An FPGA accelerator for spiking neural network simulation and training," presented at the Proceedings - IEEE International Symposium on Circuits and Systems, 2021. doi: 10.1109/ISCAS51556.2021.9401317.
- [49] C. Gillet, A. F. Vincent, B. Le Gal, and S. Saïghi, "A High-Level Methodology to Evaluate and Optimize Digital Architectures Targeting Spike Encoding," *IEEE Access*, vol. 11, pp. 120654–120665, 2023, doi: 10.1109/ACCESS.2023.3324877.
- [50] H. Jang, O. Simeone, B. Gardner, and A. Gruning, "An Introduction to Probabilistic Spiking Neural Networks: Probabilistic Models, Learning Rules, and Applications," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 64–77, Nov. 2019, doi: 10.1109/MSP.2019.2935234.
- [51] E. Stomatias, D. Neil, M. Pfeiffer, F. Galluppi, S. B. Furber, and S.-C. Liu, "Robustness of spiking Deep Belief Networks to noise and reduced bit precision of neuro-inspired hardware platforms," *Front. Neurosci.*, vol. 9, Jul. 2015, doi: 10.3389/fnins.2015.00222.
- [52] A. Al-Saegh, S. A. Dawwd, and J. M. Abdul-Jabbar, "Deep learning for motor imagery EEG-based classification: A review," *Biomed. Signal Process. Control*, vol. 63, p. 102172, Jan. 2021, doi: 10.1016/j.bspc.2020.102172.
- [53] J. Jin, Y. Miao, I. Daly, C. Zuo, D. Hu, and A. Cichocki, "Correlation-based channel selection and regularized feature optimization for MI-based BCI," *Neural Netw.*, vol. 118, pp. 262–270, Oct. 2019, doi: 10.1016/j.neunet.2019.07.008.
- [54] F. Li, F. He, F. Wang, D. Zhang, Y. Xia, and X. Li, "A Novel Simplified Convolutional Neural Network Classification Algorithm of Motor Imagery EEG Signals Based on Deep Learning," *Appl. Sci.*, vol. 10, no. 5, p. 1605, Feb. 2020, doi: 10.3390/app10051605.
- [55] G. Dai, J. Zhou, J. Huang, and N. Wang, "HS-CNN: a CNN with hybrid convolution scale for EEG motor imagery classification," *J. Neural Eng.*, vol. 17, no. 1, p. 016025, Jan. 2020, doi: 10.1088/1741-2552/ab405f.

- [56] X.-L. Tang, W.-C. Ma, D.-S. Kong, and W. Li, "Semisupervised Deep Stacking Network with Adaptive Learning Rate Strategy for Motor Imagery EEG Recognition," *Neural Comput.*, vol. 31, no. 5, pp. 919–942, May 2019, doi: 10.1162/neco_a_01183.
- [57] P. Wang, A. Jiang, X. Liu, J. Shang, and L. Zhang, "LSTM-Based EEG Classification in Motor Imagery Tasks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 11, pp. 2086–2095, Nov. 2018, doi: 10.1109/TNSRE.2018.2876129.
- [58] S. Taheri, M. Ezoji, and S. M. Sakhaei, "Convolutional neural network based features for motor imagery EEG signals classification in brain–computer interface system," *SN Appl. Sci.*, vol. 2, no. 4, p. 555, Apr. 2020, doi: 10.1007/s42452-020-2378-z.
- [59] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *J. Neural Eng.*, vol. 14, no. 1, p. 016003, Feb. 2017, doi: 10.1088/1741-2560/14/1/016003.
- [60] D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang, and Y. Liu, "Making Sense of Spatio-Temporal Preserving Representations for EEG-Based Human Intention Recognition," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3033–3044, Jul. 2020, doi: 10.1109/TCYB.2019.2905157.
- [61] "torch.optim — PyTorch 2.5 documentation." Accessed: Nov. 14, 2024. [Online]. Available: <https://pytorch.org/docs/stable/optim.html#algorithms>
- [62] S. Paszkiel, "Review of solutions for the application of example of machine learning methods for Motor Imagery in correlation with Brain-Computer Interfaces," *PRZEGLĄD ELEKTROTECHNICZNY*, vol. 1, no. 11, pp. 113–118, Nov. 2021, doi: 10.15199/48.2021.11.20.
- [63] P. Gert and S. Alois, "BCI Competition II - Data set III." [Online]. Available: <https://www.bbci.de/competition/ii/>
- [64] P. Gert and S. Alois, "BCI Competition III - Data sets IIIa." [Online]. Available: <https://www.bbci.de/competition/iii/>
- [65] M. Klaus-Robert, B. Benjamin, and C. Gabriel, "BCI Competition III - Data set IVa." [Online]. Available: <https://www.bbci.de/competition/iii/>
- [66] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio, "The non-invasive Berlin Brain–Computer Interface: Fast acquisition of effective performance in untrained subjects," *NeuroImage*, vol. 37, no. 2, pp. 539–550, Aug. 2007, doi: 10.1016/j.neuroimage.2007.01.051.
- [67] B. Clemens, L. Robert, M.-P. Gernot, S. Alois, and P. Gert, "BCI Competition IV - Data sets 2a." [Online]. Available: <https://www.bbci.de/competition/iv/>
- [68] L. Robert, B. Clemens, M.-P. Gernot, S. Alois, and P. Gert, "BCI Competition IV - Data sets 2b." [Online]. Available: <https://www.bbci.de/competition/iv/>
- [69] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "EEG Motor Movement/Imagery Dataset." *physionet.org*, 2009. doi: 10.13026/C28G6P.
- [70] M.-H. Lee *et al.*, "EEG dataset and OpenBMI toolbox for three BCI paradigms: an investigation into BCI illiteracy," *GigaScience*, vol. 8, no. 5, p. giz002, May 2019, doi: 10.1093/gigascience/giz002.
- [71] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 51–63, Nov. 2019, doi: 10.1109/MSP.2019.2931595.
- [72] S. Bohte, J. Kok, and J. Poutré, "SpikeProp: backpropagation for networks of spiking neurons," in *ESANN*, Jan. 2000, pp. 419–424.
- [73] N. G. Pavlidis, O. K. Tasoulis, V. P. Plagianakos, G. Nikiforidis, and M. N. Vrahatis, "Spiking neural network training using evolutionary algorithms," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Montreal, Que., Canada: IEEE, 2005, pp. 2190–2194. doi: 10.1109/IJCNN.2005.1556240.
- [74] C. C. Law and L. N. Cooper, "Formation of receptive fields in realistic visual environments according to the Bienenstock, Cooper, and Munro (BCM) theory.," *Proc. Natl. Acad. Sci.*, vol. 91, no. 16, pp. 7797–7801, Aug. 1994, doi: 10.1073/pnas.91.16.7797.
- [75] W. Senn and J.-P. Pfister, "Spike-Timing-Dependent Plasticity, Learning Rules," in *Encyclopedia of Computational Neuroscience*, D. Jaeger and R. Jung, Eds., New York, NY: Springer New York, 2014, pp. 1–10. doi: 10.1007/978-1-4614-7320-6_683-1.
- [76] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. M. McGinnity, "A review of learning in biologically plausible spiking neural networks," *Neural Netw.*, vol. 122, pp. 253–272, Feb. 2020, doi: 10.1016/j.neunet.2019.09.036.

- [77] M. Davies *et al.*, “Loihi: A Neuromorphic Manycore Processor with On-Chip Learning,” *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018, doi: 10.1109/MM.2018.112130359.
- [78] F. Akopyan *et al.*, “TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015, doi: 10.1109/TCAD.2015.2474396.
- [79] H. A. Gonzalez *et al.*, “SpiNNaker2: A Large-Scale Neuromorphic System for Event-Based and Asynchronous Machine Learning,” Jan. 09, 2024, *arXiv*: arXiv:2401.04491. Accessed: Oct. 28, 2024. [Online]. Available: <http://arxiv.org/abs/2401.04491>
- [80] H. Bos and D. Muir, “Sub-mW Neuromorphic SNN audio processing applications with Rockpool and Xylo,” 2022, *arXiv*. doi: 10.48550/ARXIV.2208.12991.
- [81] Z. Qiu *et al.*, “Optimized Motor Imagery Paradigm Based on Imagining Chinese Characters Writing Movement,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 7, pp. 1009–1017, Jul. 2017, doi: 10.1109/TNSRE.2017.2655542.
- [82] M. Aljalal, R. Djemal, and S. Ibrahim, “Robot Navigation Using a Brain Computer Interface Based on Motor Imagery,” *J. Med. Biol. Eng.*, vol. 39, no. 4, pp. 508–522, Aug. 2019, doi: 10.1007/s40846-018-0431-9.
- [83] “Eigen Decomposition -- from Wolfram MathWorld.” Accessed: Nov. 29, 2024. [Online]. Available: <https://mathworld.wolfram.com/EigenDecomposition.html>
- [84] A. C. Koivunen and A. B. Kostinski, “The Feasibility of Data Whitening to Improve Performance of Weather Radar,” *J. Appl. Meteorol.*, vol. 38, no. 6, pp. 741–749, Jun. 1999, doi: 10.1175/1520-0450(1999)038<0741:TFODWT>2.0.CO;2.
- [85] “CrossEntropyLoss — PyTorch 2.5 documentation.” Accessed: May 12, 2024. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html#torch.nn.CrossEntropyLoss>
- [86] “BCEWithLogitsLoss — PyTorch 2.5 documentation.” Accessed: May 12, 2024. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html#torch.nn.BCEWithLogitsLoss>
- [87] S. Rasheed, “A Review of the Role of Machine Learning Techniques towards Brain–Computer Interface Applications,” *Mach. Learn. Knowl. Extr.*, vol. 3, no. 4, pp. 835–862, Oct. 2021, doi: 10.3390/make3040042.
- [88] R. Romo Vázquez, H. Vélez-Pérez, R. Ranta, V. Louis Dorr, D. Maquin, and L. Maillard, “Blind source separation, wavelet denoising and discriminant analysis for EEG artefacts and noise cancelling,” *Biomed. Signal Process. Control*, vol. 7, no. 4, pp. 389–400, Jul. 2012, doi: 10.1016/j.bspc.2011.06.005.
- [89] B. Reuderink and M. Poel, “Robustness of the common spatial patterns algorithm in the BCI-pipeline,” *IEEE Trans. Circuits Syst. -Regul. Pap. - IEEE TRANS CIRCUIT SYST-I*, Jan. 2008.
- [90] M. Rashid *et al.*, “Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface: A Comprehensive Review,” *Front. Neuroinformatics*, vol. 14, p. 25, Jun. 2020, doi: 10.3389/fninf.2020.00025.
- [91] L. F. Nicolas-Alonso and J. Gomez-Gil, “Brain Computer Interfaces, a Review,” *Sensors*, vol. 12, no. 2, pp. 1211–1279, Jan. 2012, doi: 10.3390/s120201211.
- [92] P. Wierzgała, D. Zapała, G. M. Wojcik, and J. Masiak, “Most Popular Signal Processing Methods in Motor-Imagery BCI: A Review and Meta-Analysis,” *Front. Neuroinformatics*, vol. 12, p. 78, Nov. 2018, doi: 10.3389/fninf.2018.00078.

APPENDIX A : Determining the time of the next spike in LIF neurons

1. The LIF neuron model equation :

$$\tau \frac{dV(t)}{dt} = -V(t) + x ,$$

Where,

t = Time

$V(t)$ = Membrane potential

τ = Time constant (the time that need to take for $V(t)$ to change by $1 - e^{-1}$)

x = The constant input current

2. Solution of the equation :

Rearranging the equation, we have

$$\frac{dV(t)}{dt} + \frac{1}{\tau}V(t) = \frac{1}{\tau}x .$$

Multiplying both sides of the equation by $e^{\frac{t}{\tau}}$, the equation becomes

$$e^{\frac{t}{\tau}} \frac{dV(t)}{dt} + e^{\frac{t}{\tau}} \frac{1}{\tau}V(t) = e^{\frac{t}{\tau}} \frac{1}{\tau}x .$$

Then the left side of the equation can be simplified to

$$\frac{dV\left(e^{\frac{t}{\tau}}V(t)\right)}{dt} = e^{\frac{t}{\tau}} \frac{1}{\tau}x .$$

Integrating both sides of the equation by t , gives the solution

$$\int \frac{dV\left(e^{\frac{t}{\tau}}V(t)\right)}{dt} dt = \int e^{\frac{t}{\tau}} \frac{1}{\tau}x dt$$

$$e^{\frac{t}{\tau}}V(t) = x e^{\frac{t}{\tau}} + C ,$$

where C is the integration constant.

Rearranging the solution, gives

$$V(t) = x + C e^{-\frac{t}{\tau}}.$$

Considering the initial condition at $t = 0$ and $V(0) = V_0$, C can be found as

$$V(0) = x + C \Rightarrow C = V_0 - x.$$

Thus, the solution becomes,

$$V(t) = x + (V_0 - x) e^{-\frac{t}{\tau}}.$$

3. Determining the time of the next spike in LIF neurons

When the LIF neurons are used for spike encoding, their initial membrane potential is always 0. So, we can consider $V_0 = 0$.

Hence the solution can be written as

$$V(t) = x - x e^{-\frac{t}{\tau}}.$$

Considering $V(t_{delay}) = V_{th}$, the solution can be modified to

$$V_{th} = x - x e^{-\frac{t_{delay}}{\tau}}$$

$$V_{th} = x (1 - e^{-\frac{t_{delay}}{\tau}})$$

$$e^{-\frac{t_{delay}}{\tau}} = \frac{x - V_{th}}{x}$$

$$-\frac{t_{delay}}{\tau} = \ln\left(\frac{x - V_{th}}{x}\right)$$

$$t_{delay} = -\tau \ln\left(\frac{x - V_{th}}{x}\right)$$

$$t_{delay} = \tau \ln\left(\frac{x}{x - V_{th}}\right).$$

APPENDIX B : Hyperparameters

Table 9. Hyper parameters related to the training phase of the SNN architectures.

	Rate Coding		Latency Coding		Delta Coding	
	Single subject models	Multi subject model	Single subject models	Multi subject model	Single subject models	Multi subject model
Number of input neuron	59	59	59	59	59	59
Number of hidden layers	1	1	1	1	1	1
Number of hidden Neurons	1000	1000	1000	1000	200	200
Number of output neuron	2	2	2	2	2	2
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Learning rate	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
β_1	0.9	0.9	0.9	0.9	0.9	0.9
β_2	0.99	0.99	0.99	0.99	0.999	0.999
Batch size	40	200	40	200	40	200
Number of epochs	400	400	400	400	400	400