university of groningen

faculty of science and engineering

# Motor Imagery EEG Classification with the SNN-based NeuCube Framework

Mariya Shumska

university of groningen / faculty of science and engineering

**University of Groningen**

# Motor Imagery EEG Classification with the SNN-based NeuCube Framework

**Bachelor Thesis**

To fulfill the requirements for the degree of
Bachelour of Science in Computing Science
at University of Groningen under the supervision of
Dr. Kerstin Bunte (Intelligent Systems, University of Groningen)
and
Prof. Dr. Michael Biehl (Intelligent Systems, University of Groningen)

**Mariya Shumska (s3544990)**

August 2, 2021

# Contents

# Abstract

A brain-computer interface (BCI) bridges the gap in communication between humans and computers by analysing brain activity and controlling external devices via motor imagery (imagined movements). BCIs typically use the Electroencephalogram (EEG) which represents the brain signals recorded from the scalp surface. Recently, the Deep Learning (DL) techniques proved to be very accurate classifiers of users' intents. Moreover, they shortened the pipeline of the signal classification in comparison to traditional techniques and can infer the motor imagery command based on minimally preprocessed EEG data. However, the bottleneck of DL frameworks for brain signals classification is high power and memory consumption which might be problematic for portable devices with BCI. The list of such devices includes various medical technologies for motor rehabilitation, prostheses control, cognitive disorders treatment, as well as other not healthcare-related gadgets for games or smart environment control. Inspired by the progress of Spiking Neural Network (SNN) models in pattern recognition, where they demonstrated excellent performance in speech recognition, visual processing, and medical diagnosis, we would like to investigate their potential for EEG motor imagery classification. SNNs very closely simulate the behaviour of a biological nervous system and are more hardware friendly. They have been shown to accurately preserve the temporal pattern of the data which is considered a key strategy in brain signal processing. Therefore, this study aims to evaluate the performance of an SNN-based classifier called NeuCube for motor imagery EEG classification.

We conduct an experiment, where we investigate the NeuCube performance on raw and preprocessed real-world data set. Preprocessing methods include filtering and Independent Component Analysis (ICA). We compare several ICA algorithms: Infomax, FastICA, and FastICA with prior dimensionality reduction with Principal Component Analysis (PCA). In summary, our experiment showed that preprocessing, especially FastICA-based methods, improves the results. However, the accuracy obtained with the NeuCube framework is lower in comparison to other SNN-based classifiers used on the same data set.

# 1 Introduction

We are our brains: our movements are the result of the commands from the brain. We see, hear, and feel because of the messages received by the brain from our sense organs. Unsurprisingly, humans have had the thirst for understanding of how such a complex organ works throughout entire history. Almost a century ago, Hans Berger was the first to record the electrical currents produced by the human brain [1]. Such a procedure is called Electroencephalography (EEG), it measures and records brain electrical activity with the use of electrodes placed on the scalp surface. Along with the EEG development, the ideas about using the brain signals as a channel of communication and information transmission medium emerged [2]. Now there are technologies that provide interaction with the human brain. They are usually called Brain-Computer Interface (BCI), however, one can meet other terms like Brain-Machine Interface (BMI), Mind-Machine Interface (MMI), or Neural Interface (NI) referring to the same concept. Although often these terms are used as synonyms, BMI is more applicable to the systems with implanted sensors, while BCIs use externally recorded signals (e.g. EEG) [3]. BCIs enable the direct control of devices without the physical motor output [4], but by motor imagery (imagined movements).

Neurointerface research was at first targeted to medical science, especially tackling the problems of severely disabled patients who could not interact with the world by any natural means. Today the application of BCIs goes far beyond healthcare and includes entertainment, communication, and leveraging the natural performance of humans [5]. Especially over the past decade, BCI keeps being a hot topic in research and among the general public. Within the realm of healthcare, BCIs often serve as "tools" to restore or improve the communication and motor skills of highly impaired patients and help with rehabilitation after a stroke or spinal cord injury [6]. Besides motor disabilities, BCIs are used to treat epilepsy [7], improve behaviors in autism spectrum disorder, and facilitate rehabilitation from the cognitive issues related to dementia [8]. Moreover, BCIs are widely applied to control prostheses and wheelchairs [9]. As for the areas beyond medicine, BCI technology can be viewed as the next hardware interface. Therefore, it gained the attention of various commercial companies. For instance, Facebook-supported researchers published a paper on real-time speech decoding with BCI [10]. Gaming firm Valve recently announced their project involving BCIs which aims to enhance the gaming experience along with VR headset [11]. Furthermore, BCI has been a tool for artistic expression, including painting, installations, and music composing since the 1960s [12]. Some creative applications provide a way for paralysed patients to participate in artistic activities [13, 14]. Seeing BCI as yet another hardware interface suggests that its applications do not have clear boundaries. Therefore, the development of this technology has the potential not only to improve the life quality of people with disabilities but also to enhance a large scope of routine activities. Although the BCI's applications fields are versatile, they all share common challenges and problems which slow down the promotion of even wider use of such a technology. These concerns are mostly related to a performance bottleneck caused by the gaps in understanding of the neural mechanisms of the EEG data encoding, and lack of efficiency in the signal processing approach [15].

Conventional Machine Learning techniques like Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and Multi-Layered Perceptron (MLP) are common in EEG classification [16, 17]. These supervised methods, however, do not consider the spatio-temporal relationship of the EEG signals. Therefore, a step from Machine Learning to DL draws the attention of the scientific community. Recently, due to the increasing availability of large EEG data sets and the improvements of graphics processing units (GPU), DL frameworks have been applied to the processing of EEG

signals [17]. Researchers believe that the DL approach may provide more robust motor imagery classification [18, 16, 19]. So far, no consensus on the most appropriate DL algorithm for motor imagery classification has been reached, however, Convolutional Neural Networks (CNN) and Deep Belief Networks (DBN) are the prevailing architectures [17, 19]. Although CNNs have great potential, they are also demanding in terms of power and memory resources [20]. Biologically inspired SNNs could be a reasonable solution to overcome such high requirements due to their compact information representation, energy efficiency, and fast processing. Motivated by the recent achievements of SNNs in pattern recognition [21, 22, 23] and EEG data classification [24, 25, 26, 27, 28], we would like to explore their performance for EEG motor imagery classification. Moreover, unlike widely used CNNs, SNNs employ spatial and temporal characteristics. Such an ability is advantageous since EEG data by itself is spatio-temporal. Therefore, we would like to use them during the classification step.

Particularly, we are interested in a SNN-based framework called NeuCube, as it allows fast, one-pass online learning of spatio-temporal data, early event prediction and is claimed to be robust against noise [29]. NeuCube allows to develop an SNN-based model for various types of spatio-temporal data using different spike encoding methods, neuron properties, learning characteristics, etc. Although the NeuCube architecture proved to be efficient for spatio-temporal data classification, some challenges still remain open, in particular:

1. How the NeuCube-based models will perform on various EEG data sets? [30]
2. How much noise can be tolerated in an evolving spatio-temporal data machines? [31]
3. Is transfer learning in terms of classification of the EEG data recorded on different sessions is possible?

Therefore, in this thesis, we are going to focus on the following research questions:

1. **Suitability**: What are the benefits and limitations of the NeuCube for the motor imagery classification?
2. **EEG preprocessing**: To what extent preprocessing and noise elimination influences the obtained accuracy?
3. **Transfer learning**: To what extent is a NeuCube model trained for a subject directly transferable to a new session with that subject?

This thesis is structured as follows: first, all the theory behind the brain signals, preprocessing techniques and classification will be explained in Section 2, next we describe our experimental design and implementation details in Section 3, finally, we discuss the obtained results in Section 4 and summarize our findings in Section 5. In Section 6 we also share our ideas on future work.

# 2 Background Information

## 2.1 The Brain organization

The brain is a complex and astonishing biological structure that consists of billions of neurons [32], each of which is a processing machine generating electrochemical pulse called action potential or spike. The neurons communicate with each other through these spikes at junctions called synapses (see Figure 1). An action potential firing in one (presynaptic) neuron leads to the transmission of a signal to another (postsynaptic) neuron, influencing the ability of the "receiving" neuron to fire its own spike. The action potential can be measured in the form of the electrical wave also known as a brain rhythm [33]. The brain constantly produces the signals, and their amplitudes and frequencies depend upon the perception and state of the human. Five main brain waves are identified based on the frequency ranges: alpha, theta, beta, delta, and gamma [34]. According to the International Federation of Clinical Neurophysiology [35] the rhythms have the ranges as described in Table 1. There exists another important brain rhythm called mu (also referred to as rolandic, sensorimotor, wicket, or arceau rhythm) rhythm. Similarly to alpha-waves, its frequency ranges 8-12Hz (although according to Hari et al. [36], the shape of the mu rhythm implies that it has two or three frequency components). However, it is different in topographical and physiological terms. They are found over the sensorimotor cortex while the individual is being at rest. Unlike alpha waves, mu rhythms are not influenced by eye movements but attenuated by the actual movement or imaginary movement [37], which makes mu rhythm important in BCI research.

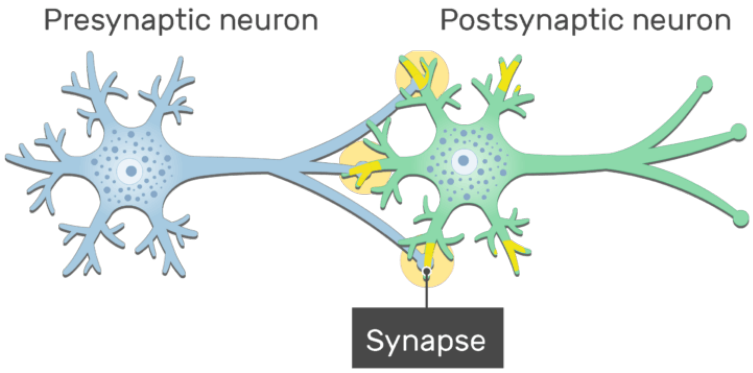


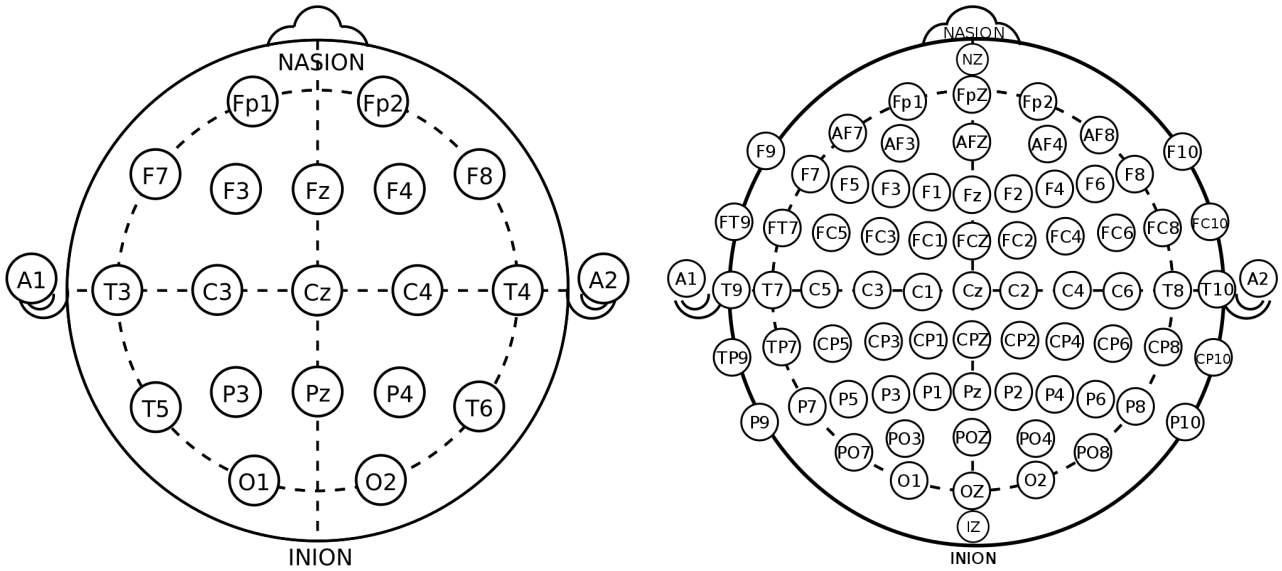

Figure 1: Illustration of a pre-synaptic neuron (blue) and a post-synaptic neuron (green) connected through a synapse (Image Source: [38]).

## 2.2 Electroencephalography analysis

Electroencephalography is a technique to record brain electrical activity with the use of electrodes placed on the scalp surface [39]. The EEG can record changes in brain activity almost instantaneously and reflect the duration of the response stimuli. However, the spatial resolution of EEG is poor and there is no information about the anatomy of the brain and regions that produced the response [40]. The International Federation of Societies for Electroencephalography and Clinical Neurophysiology has developed the recommended system of electrodes (channels) placement [41]. Such a setup is called 10-20, which indicates that the distances between the adjacent electrodes are 10% or 20% of the total front-back or right-left distance of the skull (see Figure 2 (a)). Every channel's position is assigned a letter that represents the lobe or area: pre-frontal (Fp), frontal (F), temporal (T), parietal

Table 1: Subdivision of the scalp-recorded resting state EEG rhythms in fixed frequency bands according to the Guidelines of International Federation of Clinical Neurophysiology (IFCN Glossary of terms most commonly used by clinical electroencephalographers) [35].

| Rythm | Frequency (Hz) | Associations |
|---|---|---|
| Delta | 0.1 -< 4 | Deep sleep |
| Theta | 4 -< 8 | Deep meditation, drowsiness |
| Alpha | 8-13 | Relaxed awareness |
| Beta | 14-30 | Active thinking |
| Gamma | 30-80 | Alertness |




(a) Electrode locations of International 10-20 system for EEG recording (Image Source: [43]).

(b) Electrode locations of International 10-10 system for EEG recording (Image Source: [44]).

Figure 2: Variations of the electrode locations of International 10-20 system for EEG recording.

(P), occipital (O), and central (C). Although traditional anatomical terms were used to design the positions of the channels according to the brain lobes, the central region is an exception, since it consists of parts of frontal and parietal lobes [41]. This region is referred to as the sensory-motor area and the signals recorded from it can be used in BCI systems to infer the intent [42]. In order to record more detailed EEG, extra electrodes can be added using the 10% division, filling intermediate sites in between those of the 10–20 system (Figure 2 (b)).

The acquired EEG is contaminated with various other non-motor imagery signals called artifacts. Such artifacts include eye blinks, muscle and electrode movements, environment noise, etc., and lead to a poor signal-to-noise ratio (SNR), as they interfere in signals of interest [33]. Therefore, a preprocessing step might be needed to simplify and speed up the process of a signal translation. According to [45], there are three main approaches for dealing with eye artifacts: avoidance, rejection, and removal. Since avoidance is practically not possible (eye movements and blinks are inevitable for the trials of such a length), and rejection will result in the set size reduction, the option of artifacts removal is the most suitable. This approach aims to remove the artifacts but keep the relevant brain

signals intact [45]. Band-pass filtering is one of the classical methods for artifacts removal (see Section 2.2.1 for the details). There exist other, potentially more robust techniques, such as adaptive filtering, regression, blind source separation methods, such as Independent Component Analysis (discussed in Section 2.2.2), empirical mode decomposition (EMD) and nonlinear mode decomposition (NMD) [46]. In our study, we focus on filtering and blind source separation methods and discuss them in the following subsections.

### 2.2.1 Filtering

Filtering exploits the difference between noise spectra and spectra of interest to improve SNR, by attenuating the data more in the spectral regions dominated by noise, and less in the ones where target frequencies dominate [47]. There are four approaches to use filters: high-pass and low-pass filtering refer to retaining high or low frequencies respectively, while band-pass and band-stop keep or remove the activity between the specified frequencies. Band-pass filtering in principle can not only remove artifacts but extract the relevant frequencies according to the task. In the case of motor imagery, the target range lies within 8-32Hz [48] and includes mu and beta rhythms. Below we explain some terms related to filtering (see Figure 3 for a graphical representation):

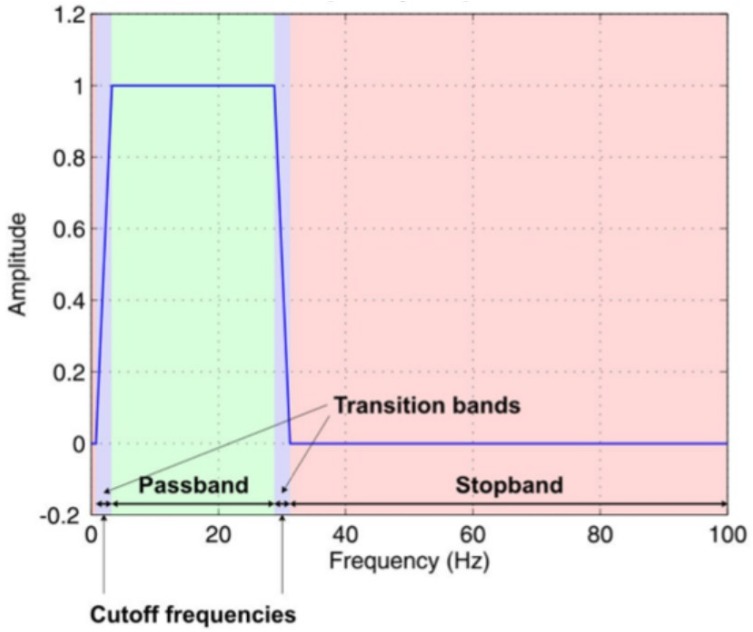



Figure 3: A plot of the frequency response with graphical representations of cutoff frequencies, pass-band, stop-band and transition band. The cutoff frequencies in the center of the transition bands separate pass-band and stop-band (Image Source: [49]).

- Cutoff frequency: the value which lies in the transition band and separates pass-band and stop-band;
- Transition band: the region between pass-band and stop-band containing the cutoff frequency;
- Order: defines a filter complexity. Increasing the filter's order allows more flexible design, e.g. the sharpness of the transition between preserved and attenuated frequencies;

- Roll-off: slope of the filter response in the transition region from the pass-band to the stop-band. The roll-off depends on the order of the filter and is given in dB/octave (a doubling of frequency) or dB/decade (ten times the frequency). For the $n$th order filter a roll-off can be computed as $20n$ dB/decade or $6n$ dB/octave. Steep roll-off results from narrow transition bands, while wide transition bands lead to shallow roll-off;
- Pass-band ripple: variation in the frequency magnitude response within the pass-band of a filter.

Filters can be classified to have finite (FIR) or infinite (IIR) impulse response. The impulse response of IIR filters is realized through a functional form defined by the filter coefficients and state variables. FIR filtering is implemented as the convolution of the input with the impulse response. Although FIR filters are more stable and produce less nonlinear phase distortions, they are more computationally expensive than IIR filters [50]. Another important characteristic of a filter is the direction of the signal used as input. Causal filters consider only past and present information, while non-causal depend on the past and future input. Non-causal filtering helps to reduce phase delays in the signal, however, they cannot be used online since they rely on the fully acquired data. Butterworth IIR filter [51] is common in electrophysiology, as it has no pass-band and stop-band ripple (i.e. the frequency response is maximally flat) and its roll-off near the cutoff frequency is shallower compared to the other filters like Chebyshev or elliptic IIR filters [52]. Therefore, we will apply this type of filter in our experiment.

### 2.2.2 Independent Component Analysis

ICA [53] is a blind source separation unsupervised technique, which linearly unmixes the data into mathematically independent components. As Makeig et al. [54] concluded, ICA is efficient at performing source separation if

1. The mixing medium is linear and propagation delays are negligible;
2. The time courses of the sources are independent;
3. The number of sources is the same as the number of sensors.

These assumptions hold for the EEG, although (3) is questionable, since the effective number of statistically independent brain signals contributing to the recorded signals is not known [54].

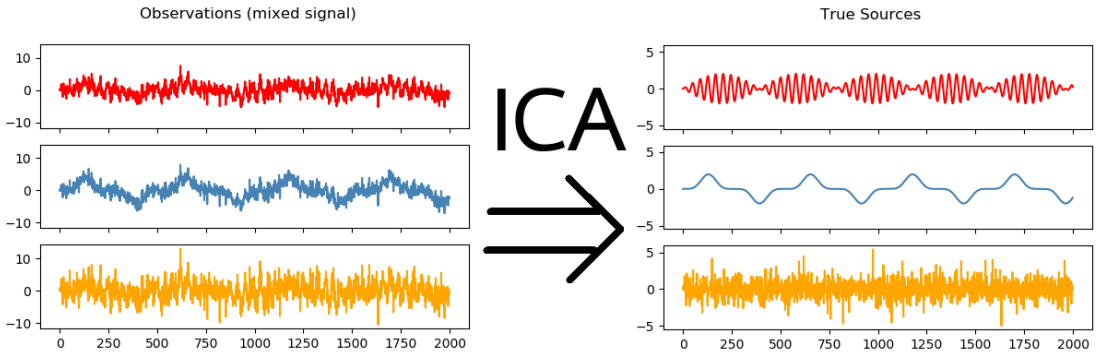



Figure 4: Schematic representation of signals decomposition to sources by ICA (Image Source: [55]).

According to Cohen [50], ICA can successfully separate neural activity from muscle and eye artifacts. It can be observed that some of the obtained components are artifactual since noise is usually uncorrelated and independent from EEG sources. Decomposing data by ICA results in a linear change of

basis from the data collected from electrodes into a spatially transformed basis. This means that the data are transformed to simultaneously recorded outputs of spatial filters which were applied to the entire multi-channel data, rather than to a collection of simultaneously recorded single-channel data records. The observed EEG data recorded from $n$ channels can be represented in matrix form as:

$$X = WS \ . \tag{1}$$

$W$ is a so-called $n \times n$ mixing matrix, and $S$ are independent components or source signals. The goal of ICA is to estimate $W$, and source distribution provided only $X$. The following linear transform is used to recover the best possible approximation of the original signals using the assumptions mentioned above:

$$S = W^{-1}X \ . \tag{2}$$

After the EEG signals decomposition, the resulting components can be examined, and based on the topology, power spectrum, and other criteria, one can conclude which ICs can be considered as noise. Moreover, there exist automated EEG independent components classifier, e. g. ICLabel [56], which estimates the signal origin. Such artifactual components can be removed, and the remaining ones can be considered as spatial filters and applied to the original EEG signal to eliminate noise.

Certain preprocessing is recommended to achieve a good quality of resulting ICs. In general, providing ICA with relatively clean EEG data (without noisy time segments) is the best solution [57]. Although using continuous EEG might be beneficial due to the larger amount of data, egregious movements that might be present during the breaks between trials might prevent the algorithm from isolation of the typical artifacts like eye blinks [58]. Therefore, one should carefully decide whether ICA is to be performed on raw or epoched data. Winkler et al. [59] showed that High-pass filtering around 1-2Hz improves ICA decomposition in terms of SNR, single-trial classification accuracy, and the percentage of "near-dipolar" ICA components. Performing Principal Component Analysis (PCA) [60] in combination with ICA is also a good strategy. Its application on the EEG data prior to ICA may improve the results [61], while subsequent PCA application may decrease the quality of ICA [62]. That is due to the so-called whitening or sphering transformation (which can be achieved with PCA), which transforms $X$ to $\widetilde{X}$ in such a way, that sources are uncorrelated and their variances equal unity. Essentially, this process transforms the mixing matrix into an orthogonal one:

$$\widetilde{X} = \widetilde{W}S \ . \tag{3}$$

Due to such an orthogonal rotation, the original channels are transformed into an equal number of linearly-uncorrelated variables or Principal Components (PCs), each of which accounts for the largest possible portion of remaining data variance. The orthogonality reduces the number of parameters to be estimated: instead of estimating $n^2$ elements of the original $W$ we need to consider only $n(n-1)/2$ [63]. Furthermore, PCA can be utilised to reduce the number of the resulting components while doing whitening, which may prevent overfitting occurring when ICA is performed in high dimensions with an insufficient amount of data [64]. This is achieved by considering and performing ICA only on the subset of largest PCs, e. g. components which account for the pre-defined variance threshold [62].

There exist various ICA algorithms, some commonly used for brain signals processing are logistic Infomax, (Infomax) [65], Joint Approximation Diagonalization of Eigenmatrices (JADE) [66], Second-Order Blind Identification (SOBI) [67], Hyvarinen's fixed-point algorithm (FastICA) [63]. Brunner et al. [61] examined the classification performance of LDA on the same data set to be used

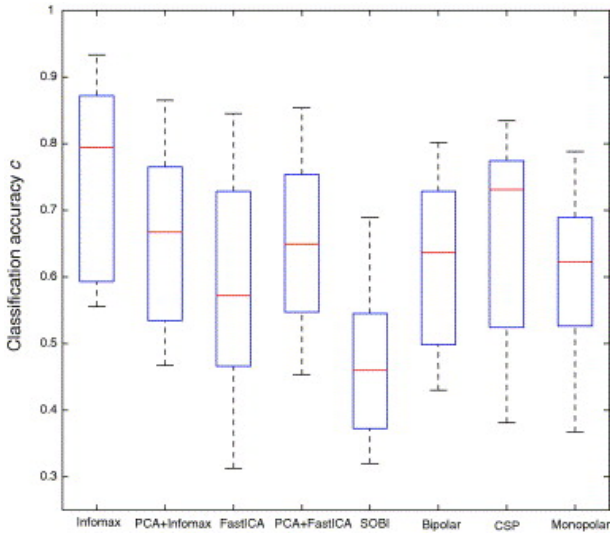



Figure 5: Boxplots showing the cross-validated performance of each method. The upper and lower lines of a box show the upper and lower quartile, respectively, while the median corresponds to the line within the box. The whiskers are lines that indicate the range of the rest of the data [61].

in our experiment with Infomax, SOBI, and FastICA processing. Combinations of PCA and ICA were also evaluated. Performing Infomax on PCA-filtered data did not improve the accuracy, while FastICA results were greatly improved by prior PCA application, where only 10 principal components were considered (see Figure 5). However, in this paper, we could not find a clear indication that the data has been whitened prior to FastICA, which is required by this algorithm. Therefore, we cannot be sure whether the improvement was achieved due to dimensionality reduction or transformation of the basis. Considering these results, we will use Infomax and FastICA methods in our study.

To maximize the statistical independence, FastICA seeks for an orthogonal rotation that maximizes a measure of non-Gaussianity of the rotated components. The reason is that according to the Central Limit Theorem, the signal is more Gaussian than the source since it is the linear combination of these sources. Infomax relies on a different strategy and aims for mutual information maximization by maximizing entropy. In our experiment, we will be using fast and robust versions of these algorithms referred to as Preconditioned ICA for Real Data under Orthogonal constraint (Picard-O) [68] and Preconditioned ICA for Real Data (Picard) [69], respectively. The main benefit of the ICA method is that it does not require any reference channel or previous information about the noise. The disadvantages include the need for a large amount of data and the availability of a full EEG matrix to derive the components, meaning that it is not suitable for online analysis. However, Grandchamp et al. [70] investigated the stability of ICA decompositions across sessions from a single subject, comparing ICs from 11 sessions. They observed that ICs form clusters of similar scalp topographies, equivalent dipole locations, and spectral activities. The ability of ICA to identify recurring brain and artefactual components has important implications for the BCI research, as it allows precomputing the unmixing matrix once and apply it to the online signal processing.

## 2.3 Classification

SNNs are a third generation neural networks [71]. Their data representation and learning approach are designed to be very close to the human brain [40]. The communication in SNN is realized

through spikes and spike sequences, providing compact information representation. Moreover, they are hardware-friendly and have low energy consumption [72], which make them suitable for portable devices with BCI. Such classifiers as SNNs could also improve BCI performance since they consider the evolving over time input, meaning that the spatio-temporal nature of EEG data is taken into account. Brain-inspired SNN became a foundation for the NeuCube framework proposed by Prof. Nikola Kasabov [73]. It was designed for spatio-temporal data mapping, learning, and understanding. Studies have shown its successful application with respect to EEG, including classification of motor [74, 75] and cognitive [30] tasks, sleep stages [76] and emotions [77]. NeuCube was also used for motor imagery classification. The Taylor's et al. [78] paper discovered the feasibility analysis of using the NeuCube architecture for classifying the real and imagined movements. The subjects had to either perform the specified movements or imagining the movements (resting, flexing the wrist, or extending the wrist). The NeuCube performed significantly better (average accuracy 76% on 50/50 split) than the other machine learning techniques (MLR, SVM, MLP and ECM). The closest competitor was SVM with the average accuracy of 62%. Behrenbeck et al. [79] classified motor imagery of left and right hand movements obtained from three bipolar recordings (C3, Cz, and C4) and achieved an average of 75% accuracy. Promising accuracy results, efficiency, and potential to be implemented in neuromorphic architecture inspired us to investigate further the applicability of the NeuCube for motor imagery classification. In the next Section 2.4, we will explain the details of the framework architecture.

## 2.4 NeuCube

The NeuCube is an SNN-based framework, which consists of three main modules (see Figure 6):

- Input encoding module: conversion of multivariable input stream data into spike sequences (see Subsection 2.4.1);
- SNN reservoir (SNNr) module: unsupervised learning of spatio-temporal patterns from data in a SNN reservoir (see Subsection 2.4.2);
- Output Classification/regression module: supervised learning of classification/regression output system classification/regression problems (see Subsection 2.4.3).

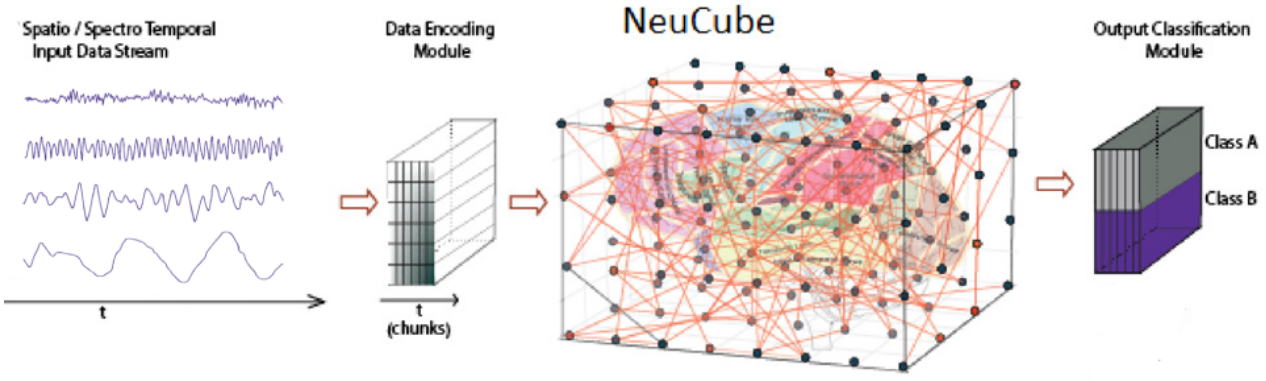



Figure 6: The general NeuCube architecture, consisting of: input data encoding module; 3D SNN Cube module; output classification module. (Image is adapted from [73]).

### 2.4.1 Input Encoding

In order to reduce the amount of time series data but preserve relevant information for pattern recognition and classification, temporal compression is performed through spike-time encoding [40], which results in binary value series. Different encoding techniques have different approaches to extract information from the input signal, e. g. firing rate, population rank coding, or temporal coding. The latter uses the exact timing of individual spikes and is suitable for fast processing of electrophysiological signals [80]. There are four temporal-based encoding methods available in the NeuCube framework, namely threshold-based representation (TBR), Step Forward (SF) encoding, Moving Window (MV) encoding, and Ben's Spiker algorithm (BSA). Next, we will elaborate more on these algorithms.

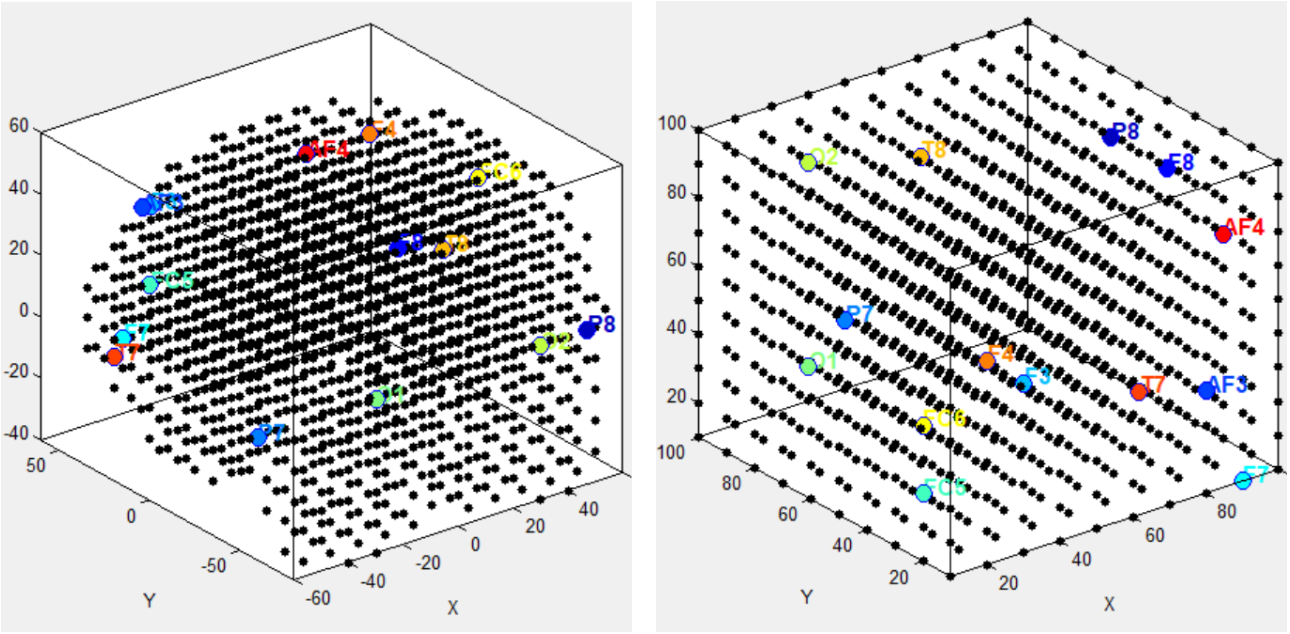
BSA utilises a linear filter to find a unipolar (positive one or zero) spike train. This has certain implications, e.g. of falsely encoded end of the signal due to convolution or appearance of low-frequency artifacts. MW is similar to a moving average filter and uses a moving baseline that equals the mean of the previous signal values in a time window and a set threshold value. A positive (negative) spike is generated if the value of the signal is above (below) the sum (difference) of the baseline and a threshold. Similarly to BSA, the beginning of the signal is not encoded well until the window size is reached. Moreover, the square root of the window size is proportional to the noise elimination. This leads to tradeoffs between reduction of noise and the range of the spectrum after encoding, which may not be suitable for not sufficiently oversampled EEG data [80].

Contrary to above mentioned methods, TBR and SF do not falsely encode the start or/and end of the signal. TBR is the simplest implementation of temporal encoding which compares the difference between consecutive signal values and a threshold. A spike is emitted if the difference passes the threshold. TBR is able to register relatively large changes in a signal and underrepresents smooth changes. Moreover, it may introduce strong low-frequency artifacts after the reconstruction. It is recommended to consider all possible events such that the threshold can capture all of them [80], which does not sound like a scalable approach for a broad range of events like motor imagery tasks. Likewise TBR, SF represents only signal changes, however, unlike TBR, it does not bring artifactual low-frequency components. SF employs an interval around a moving baseline with a set threshold, which is amplitude-dependent. The baseline is initialised with the initial signal value. If the following signal value is above (below) sum (difference) of baseline and threshold, the positive (negative) spike is produced and the baseline is moved to the upper (lower) limit of the threshold interval. Despite the step-wise reconstruction, most types of signals can be reconstructed well in time and frequency domains due to multiple steps per a single change possibility [80]. The threshold value can be adjusted such that large and small amplitude events are well represented, which is relevant for different motor imagery tasks classification.

Careful attention should be paid to the encoding method selection, as the proper encoding might be crucial for the SNN classifier performance. Petró et al. [80] provided a paper with an analytical approach for encoding method selection based on the signal characteristics and different error metrics for encoding parameters tuning, including SNR, Root Mean Square Error (RMSE), and coefficient of regression (R-squared). They concluded that SF is efficient for all kinds of tested signals, such as smooth signals with sine components continuously ranging from 2 to 20 Hz with random power, EEG signals during perturbation-evoked potential events, trended and step-wise signals. Additionally, SF has only one parameter which is easy to optimise. Therefore, we will consider Step Forward encoding for our experiment.

### 2.4.2 SNN Reservoir

A 3D Reservoir module (SNNr) is also referred to as the ”Cube” and represents a group of spiking neurons with configurable locations (see Figure 7). The framework uses Leaky Integrate and Fire Model of neuron (LIFM) [81]. The membrane potential increases with every input spike, multiplied by the synaptic strength until it reaches a certain threshold. Then an output spike is generated and the membrane potential is reset to an initial state. The membrane potential can have leakage between spikes, which is defined by a temporal parameter. The encoded spike trains are mapped into the SNNr. Initially, the connections between neurons are set using the Small World Connectivity method (i. e. the closer the neurons the more likely they will be connected), where the radius can be parametrised. Next, the Cube is trained in an unsupervised manner to modify the initial connection weights, which are adjusted based on the Spike-timing-dependent plasticity (STDP) rule [82]. This rule suggests that the strength of a synaptic connection's weight depends on the time of spiking in the pre- and post-synaptic neurons. Due to the assumption of causality of two neurons, the synaptic weight of connection from neuron $j$ to neuron $i$ increases if a pre-synaptic neuron ($j$) spikes just before the post-synaptic neuron ($i$) and decreases otherwise. After the connections adjustment, the Cube establishes specific trajectories of spiking activities when a particular pattern is encountered.




(a) A Cube with 1471 spiking neurons. The locations of these neurons correspond to the Talairach template coordinates [83] with a resolution of 1 $cm^3$.

(b) A Cube with 1000 spiking neurons. The locations of these neurons are configured automatically using graph matching.

Figure 7: Examples of different reservoir configurations. Black dots represent spiking neurons; the features (electrodes in that case) are bold coloured dots.

### 2.4.3 Output Classification

Following the unsupervised training stage, a supervised model is trained to classify patterns by learning to recognize the corresponding Cube states according to predefined classes of these patterns. Here a dynamic evolving SNN (deSNN) algorithm (see Figure 8) is used since it is fast and efficient for spatio-temporal pattern recognition and enables one-pass learning [84]. In the training phase of

deSNN, new output neurons are first created for each training sample and then connected to the neurons in SNNr (feature neurons). Next, their spike trains are input to the Cube. The initial weights of the connections between Cube's and output neurons are set based on the rank order (RO) rule [85]. Its core idea is the assumption that the most important information of a pattern is contained in spikes that arrive earlier. Therefore, it prioritises the inputs based on the order of the spike arrival. The computation of the synaptic weight $w_{(j,i)}$ between presynaptic ($j$) and postsynaptic ($i$) neurons with RO rule can be mathematically described as:

$$w_{(j,i)} = \alpha \cdot \text{mod}^{\text{order}(j,i)} \quad , \tag{4}$$

where $\alpha$ is a learning parameter, *mod* is a modulation factor, which defines the importance level of the order of the first spike; $order(j,i)$ is the rank of the first spike at synapse $(j,i)$ ranked among all spikes arriving from all synapses to the neuron $i$. After the initialisation, the weights are further tuned according to an STDP rule upon arrival of the spikes. The weights increase if a new spike arrives at the corresponding synapse and decrease if there is no spike at this time. This change is defined by a $drift$ parameter. The weights change their values in parallel every time unit.

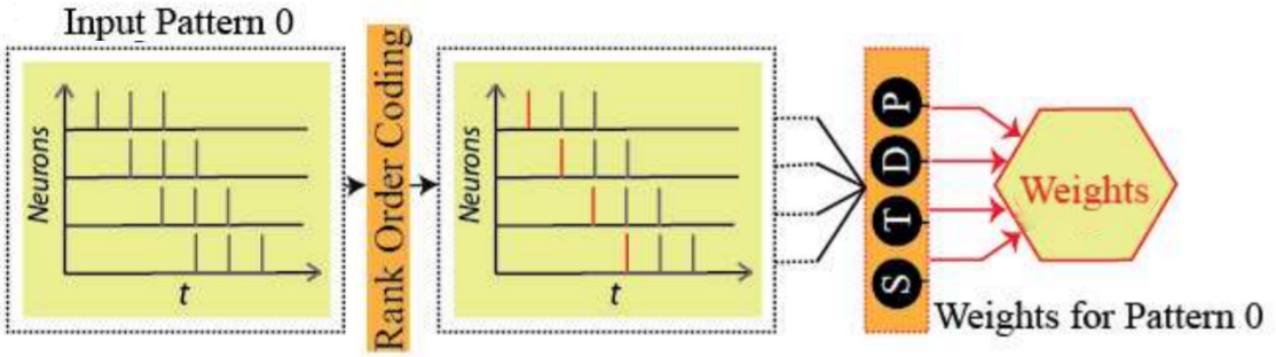



Figure 8: An illustration of the main idea of deSNN algorithm. A pattern consisting of 4 input spike trains is learned into a single output neuron. RO rule is then used to compute the initial weights based on the order of the first spike (marked in red). (Image Source: adapted from [84]).

While the training sample is presented, the spiking threshold of the neuron is calculated to trigger that neuron to spike whenever a similar pattern is encountered during the recall. If the weight vector of the new neuron is similar to the one of an already trained neuron (according to the Euclidean distance), they will be merged, and the connection weights and the thresholds will be averaged. During the recall operation (i. e. a new sample is presented), the input pattern is presented to all neurons created during learning. Similar to the learning phase, a new output neuron is created for each recall pattern, and its weight vector is compared against the ones of existing neurons using Euclidean distance. Then again, the neuron with the most similar connection weights is the "winner". In order to compute the degree of similarity of two neurons, the transductive reasoning [86] and nearest neighbour classification (kNN) are utilised.

# 3 Experiment

## 3.1 Data

For this experiment, the public data set 2a [87] from BCI competition IV is used [88]. Our data choice is motivated by the fact that it has been extensively used in other papers including a study with an SNN-model [28] which achieved 74.54% classification accuracy of left and right hand imagined movements. We stress, however, that the NeuCube framework has not been tested on that data set before. The details of the data can be found in Table 2.

Table 2: Quantitative data description of set 2a from BCI competition IV.

| **Characteristic** | **Quantity** |
|---|---|
| #subjects | 9 |
| #classes | 4: left hand, right hand, both feet, tongue |
| #sessions per subject | 2 |
| #runs per session | 6 |
| #trials per run | 48 (12 per each class) |

The EEG was recorded using 22 (10-20 system) electrodes. Additionally, 3 electrooculography (EOG, a technique for measuring the eye movements) channels were provided (see Figure 9). The signals were sampled with 250 Hz and band-pass filtered between 0.5 Hz and 100 Hz. An additional 50 Hz notch filter (a band-stop filter) was enabled to suppress line noise.

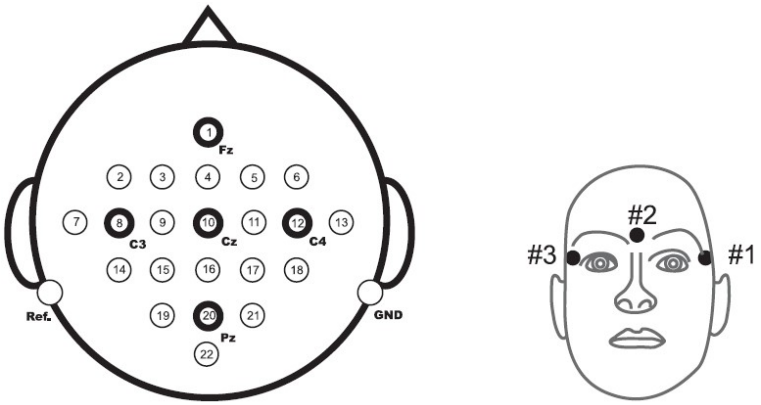



Figure 9: Left: Electrode montage corresponding to the international 10-20 system. Right: Electrode montage of the three monopolar EOG channels. [87]

At the beginning of each session, a recording of approximately 5 minutes was performed to estimate the EOG influence. The subjects were sitting in front of a screen on which at $t = 0$s, a fixation cross appeared and at $t = 2$s, a cue corresponding to one of the four classes appeared and stayed on the screen for $\Delta t = 1.25$s. The subjects were asked to perform the motor imagery task until the fixation cross disappeared from the screen at $t = 6$s. The data is available at .gdf format and contains the information about the boundaries of each trial.

## 3.2 Experimental setup

In order to answer our research questions on suitability, EEG preprocessing, and transfer learning in context of the NeuCube framework and motor imagery, we design the experiment, where we

1. Test the NeuCube framework with the EEG data it has not been tested before;
2. Compare the classification results obtained from raw, filtered, and ICA-cleaned data;
3. Evaluate the classifier's performance on recalling the data recorded on a different session from one the training was performed on.

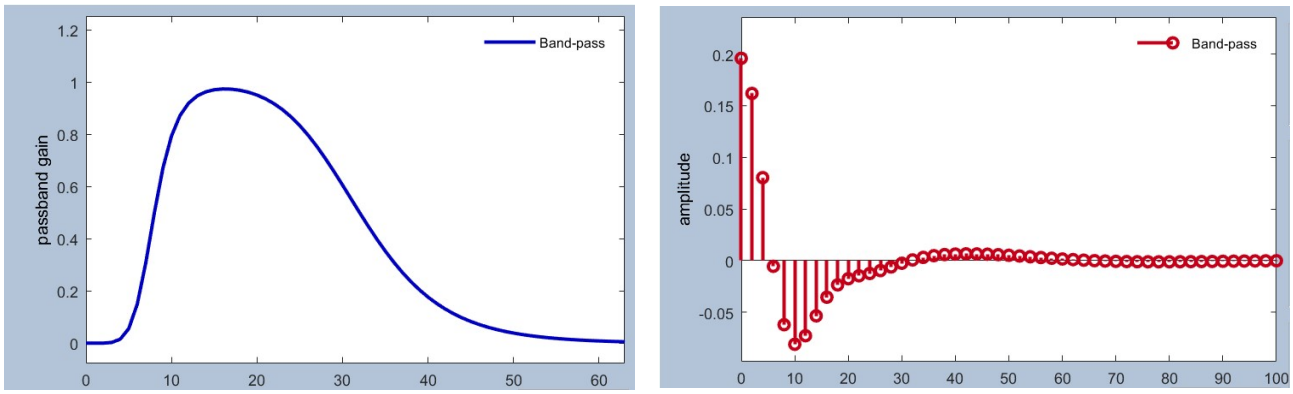
We consider the within-subject analysis of motor imagery of right and left hands. As we aim to minimize the human intervention, we will use the simplest EEG montage and will not consider EOG channels in our analysis. We note that no new features are inferred from the continuous data of the testing session, only application of transformations or filters designed using training data (session 1) are used. Regardless of the method of preprocessing (if any), the EEG data has to undergo certain manipulations to meet the requirements and limitations of the NeuCube framework free distribution used in our study [1]. The available module (M1) is meant for generic prototyping and testing and cannot be tested for online classification, as the input data has to be in a format of individual *samples*, i.e. EEG patterns. Therefore, similarly to other offline NeuCube studies [79, 30, 76, 77], we make an assumption that all data, including the testing one (recorded on session 2), is available in separate EEG pieces containing motor imagery tasks. To achieve this, we first *epoch* the continuous EEG recorded from 22 electrodes. In the context of EEG manipulation, *epoching* means extracting and concatenating portions of the data of the defined duration w.r.t. available events. In our case, the "event" is a class of a motor imagery task (e.g. left or right hand). As for the time interval for the classification, we consider 3-second pieces from $t = 2.5$s to $t = 5.5$s to ensure that they contain only motor imagery, excluding the waiting for the visual cue. Finally, to obtain the individual *samples*, we slice the epoched data, such that every 3-second pattern is a separate file. Therefore, all NeuCube input files are .csv files, containing a 750 (250Hz $\cdot$ 3s) x 22 (electrodes) matrix. For the experimenting on raw data, only epoching and slicing have been done. In the following Section 3.3 we will elaborate on each preprocessing technique's implementation.

## 3.3 Preprocessing

### 3.3.1 Filtered

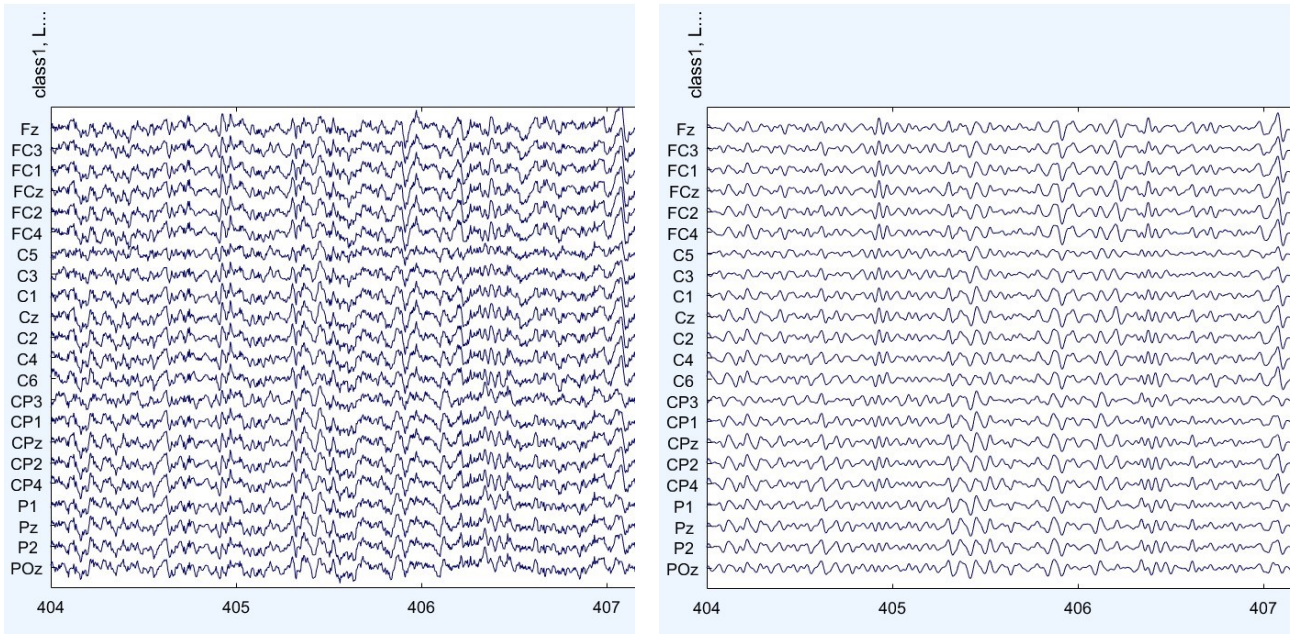
The training data (recorded on session 1) is first filtered using 6th order Butterworth band-pass filter (roll-off 36dB/octave or 120dB/decade) with cutoff frequencies of 8Hz and 32Hz (see Figure 10 for frequency and impulse response plots). The filter is applied to whole (unepoched) data to avoid edge artifacts. It can be seen in Figure 11, that the filtered signal is cleaner and smoother. As for the testing data (recorded on session 2), the same filter is applied to each 3-second sample. In both cases, the underlying MATLAB `filtfilt` routine is used (a zero-phase-shift filter), meaning that after the data is filtered in forward direction, the filtered sequence is reversed and is ran back through the filter. It avoids phase distortion, however, it leads to non-causality, as the whole piece of data should be available. According to our assumption mentioned in Section 3.2, we consider 3-second EEG windows to be fully available. We believe that this approach might still be suitable for the online processing of the signals, however, these considerations are in scope of future work.

[1] All EEG manipulations are done using MATLAB R2020a [89], EEGLAB environment [57], BioSig toolbox [90], ERPLAB [91] plugin, ICLabel [56] tool.




(a) (b)

Figure 10: Frequency (a) and impulse (b) response of the designed 6th order Butterworth band-pass filter.




(a) Raw data. (b) 8-32Hz bandpass filtered data.

Figure 11: Raw (a) and filtered (b) EEG of the motor imagery of the left hand movement of subject A08.

### 3.3.2 ICA

For ICA preprocessing, we consider 3 options, namely

1. Infomax
2. FastICA
3. Dimensionality reduction with PCA followed by FastICA

The ICA is applied to each subject individually to data from session 1 only. For all cases, the data is first pre-processed with the high pass 2Hz 2nd order Butterworth filter, as it improves the quality of

ICA decomposition (see Chapter 2 Section 2.2.2). We would like to have a balance between sufficient amount of data and quality of data presented to ICA, therefore, we segment the continuous EEG signals into samples containing motor imagery of length $\Delta t = 5.55$s (from $t = 0$s to $t = 5.55$s) and concatenate the samples of all tasks in order to achieve the same ICA decomposition for all classes. This would help to avoid too noisy segments occurring during the breaks between trials. Before ICA the data is scaled to unit variance and whitened with PCA. In case of the dimensionality reduction, we keep only first 10 components, similarly to another study on the same data [61]. The resulting PCs are passed to the ICA algorithm. The outcome of the ICA is the so-called weight matrix, which we save for future usage on raw EEG signals from both sessions. The corresponding whitening matrix obtained after PCA is saved as well in order to operate in the same space.

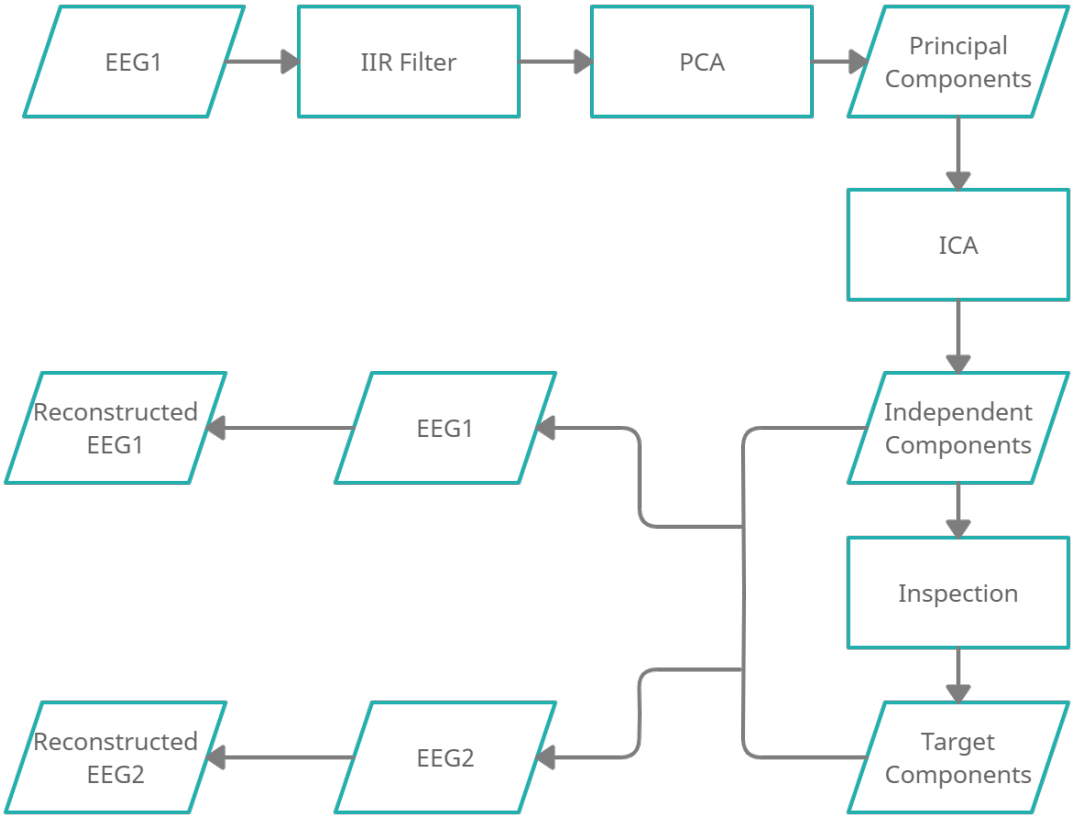
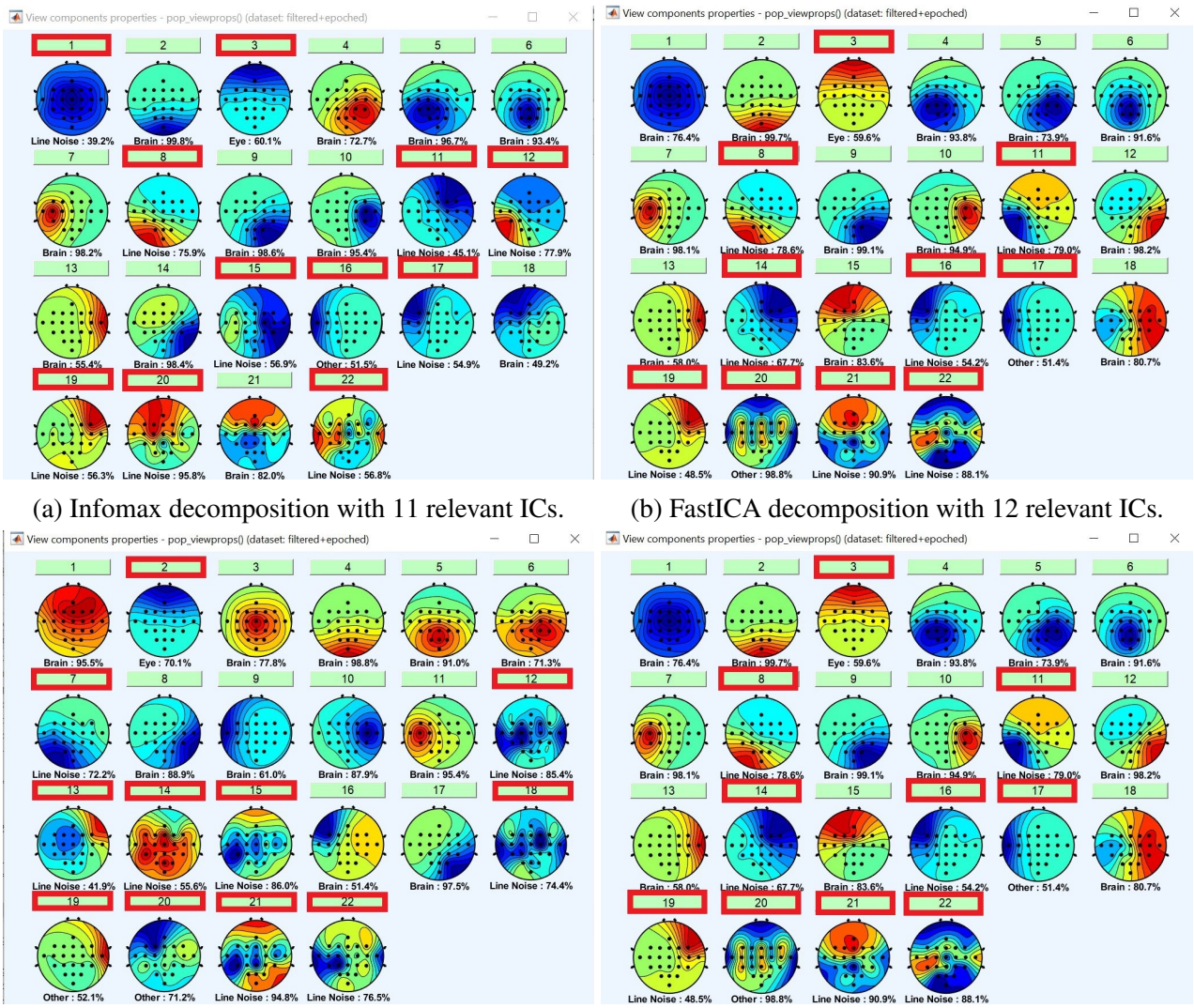



Figure 12: A generalised flowchart for ICA experiments. Input/Output is enclosed in a parallelogram, Processes are shown as rectangles. EEG1 and EEG2 correspond to data recorded during sessions 1 and 2 correspondingly. The resulting ICA weight matrix is derived using high-pass filtered signals from session 1 only and is applied to raw data from any session of the same subject.

We automate the ICA pipeline completely such that no human intervention is needed by rejecting the components based on the ICLabel (default version) prediction. We prune all components which are classified as "non-brain" (see Appendix A Figure 24 for the example). The indices of rejected components are saved, such that they can be subtracted from other data to which ICA weights are applied. Finally, since we can treat ICA components as spatial filters, which tend to be stable across sessions (see Chapter 2 Section 2.2.2), we apply the weight matrix to raw EEG from both sessions and obtain reconstructed cleaner signals. For the flowchart overview refer to Figure 12. We also measure the time needed for the analysis (which includes whitening and ICA) to see which ICA algorithm is

the fastest. Time performance and the number of retained components can be found in Appendix A Table 5. In most cases, we do not see a big difference in the number of retained components between Infomax and FastICA, however, this is not the case for subject A07 (see Figure 13). As for the time performance, FastICA with prior dimensionality reduction expectedly requires the least time, while in the case of comparable Infomax and FastICA, the latter is more than 1.5 times as fast as Infomax.




(a) Infomax decomposition with 11 relevant ICs.

(b) FastICA decomposition with 12 relevant ICs.

(c) Infomax decomposition with 11 relevant ICs.

(d) FastICA decomposition with 17 relevant ICs.

Figure 13: ICA decomposition of subject A09 ((a), (b)), and A07 ((c), (d)). The components are sorted in decreasing order of the EEG variance accounted for by each component (i. e. the lower the order of a component, the more data it accounts for). In cases (a) and (b) the number of retained components is similar, while in cases (c) and (d) the number of retained components differs substantially.

## 3.4 Classification

### 3.4.1 Encoding

The first step of the NeuCube classifier is encoding. As discussed in Section 2.4, the SF algorithm is used due to its suitability for motor imagery and simplicity of parameter optimisation. Since we

are considering prior EEG processing to eliminate the artifacts, SF's poor ability to reduce noise is therefore not relevant for most parts of our experiment. The only parameter for the SF method is the threshold. In order to achieve the best results, the threshold should be optimised per subject and per preprocessing technique. However, since we want to see clearly how different preprocessing techniques influence the accuracy, we use the same threshold which we try to optimise for all subjects and preprocessing methods. To do so, we use the optimisation tool Spiker [2], which seeks the best threshold value for a single signal according to SNR, and thus provides its best reconstruction. We consider the data from all subjects from session 1 only, and compute optimal thresholds for every channel for every sample class-wise, i. e. we obtain the mean ($\mu$) and standard deviation (std) value for a class by averaging 72 samples x 22 electrodes. We do the same operation for every preprocessing method and obtain tables which can be found in the Appendix B Table 6. Each subject's threshold is then computed as

$$t_{\text{opt}} = \frac{(\sum_{i=1}^{n} \mu_i - \sum_{i=1}^{n} \text{std}_i)}{n} \ , \tag{5}$$

where $n$ is the number of motor imagery tasks (in our case $n = 2$). We "round down" the average threshold to ensure that the resulting spike rate is higher. Finally, we can compute the threshold to use for all subjects by simply averaging all values received per subject per preprocessing method. In our case, the resulting value used for the encoding threshold is $T = 1.38$.

### 3.4.2 Cube Initialisation

After encoding we need to provide the spatial configuration of the SNN reservoir. We designed the configuration consisting of 65 neurons located according to 10-10 cortical projection [92] with the coordinates rounded to the nearest integer. The Small World Connectivity radius is set to $r = 3$. Our map is different from the one proposed by the framework authors, which consists of 1471 neurons representing the entire brain geometry. Although our model does not provide a lot of connectivity information (which is not in the scope of motor imagery classification), it reduces training time significantly due to a fewer number of neurons. Moreover, we performed 5-fold cross-validation on the demo data supplied with NeuCube with our configuration and obtained 77% test accuracy as opposed to 60% obtained with the model provided by authors, indicating the suitability of the setup. Next, we map all 22 electrodes used in our data to the reservoir (see Figure 14).

### 3.4.3 SNN model design

The proper hyperparameters choice is crucial for the model performance. Moreover, the values optimal for one subject or certain preprocessing might be suboptimal for other ones. However, since we do not want to introduce extra variability and would like to clearly see if there is a trend in accuracy w.r.t preprocessing condition, we use the same parameters for all cases. The selection of a total of 11 learning hyperparameter values is not a trivial task, even though the NeuCube framework provides certain optimisation tools based on a grid search or genetic algorithm. After the analysis of other studies on EEG where NeuCube was used (see Appendix C Table 7) and our own testing, we concluded that the parameters which are the most influential are *mod* and $drift$. Therefore, we focus on their optimisation.

We tested the values used in other EEG classification studies, including the ones proposed by the framework authors. Additionally, we ran a grid searches within observed ranges and explored how

[2] Available at https://github.com/KEDRI-AUT/snn-encoder-tools.

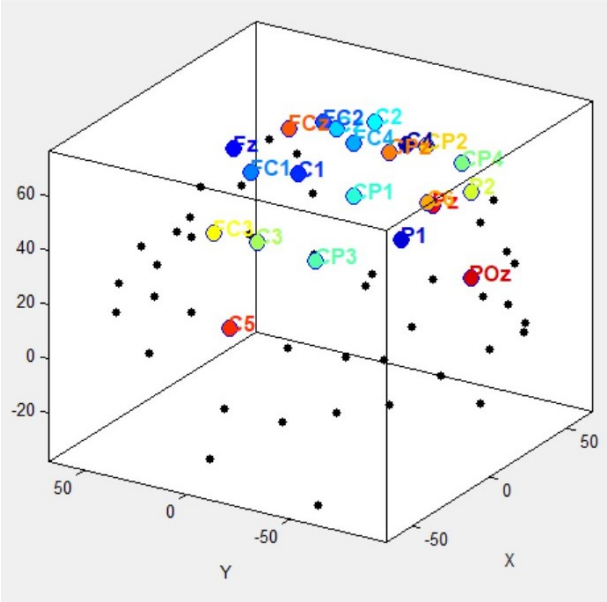



Figure 14: A 3D configuration of 10-10 cortical projection consisting of 65 neurons with spatially mapped EEG features (electrodes).

different parameters work across different subjects. We performed 5-fold cross-validation on session 1 data for all subjects and all conditions using 3 sets of parameters which seemed to be promising (see Appendix C Table 8 for the details). We decided to use $mod = 0.4$ and $drift = 0.25$ [3], since they provided the highest accuracy in most cases. The full list of parameters can be found in the appendix C Table 9. Finally, we perform the recall operation on session 2 data.

[3]These values appeared to be the same as authors of the framework suggested for their demo data set.

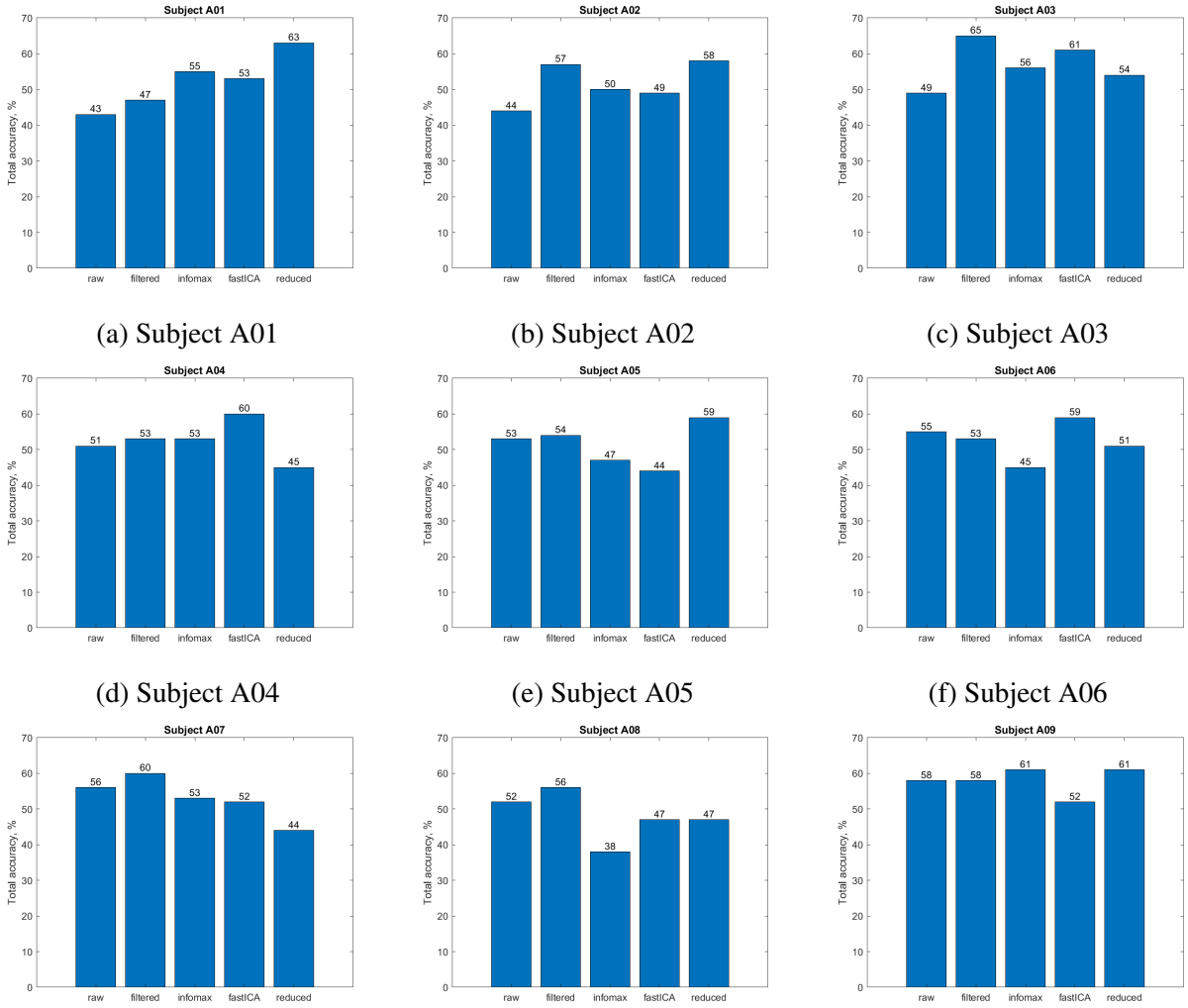
# 4 Results

We performed the binary within-subject classification using motor imagery of left and right hands recorded on session 1. The per class and average accuracy obtained after 5-fold cross-validation on session 1 data can be found in Table 3. All results are rounded to the nearest integer.

Table 3: The average accuracy and accuracy per class obtained after 5-fold cross-validation on session 1 data. The highest result within a subject is in bold.

| Subject | Raw | | | Filtered | | | Infomax | | | FastICA | | | Reduced | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | lh | rh | avg | lh | rh | avg | lh | rh | avg | lh | rh | avg | lh | rh | avg |
| **A01** | 53 | 33 | 43 | 44 | 50 | 47 | 60 | 50 | 55 | 54 | 51 | 53 | 61 | 65 | **63** |
| **A02** | 39 | 49 | 44 | 51 | 63 | 57 | 43 | 57 | 50 | 38 | 60 | 49 | 57 | 58 | **58** |
| **A03** | 31 | 66 | 49 | 65 | 65 | **65** | 60 | 53 | 56 | 60 | 62 | 61 | 46 | 63 | 54 |
| **A04** | 40 | 63 | 51 | 58 | 49 | 53 | 51 | 56 | 53 | 69 | 51 | **60** | 42 | 49 | 45 |
| **A05** | 39 | 67 | 53 | 49 | 60 | 54 | 40 | 54 | 47 | 50 | 39 | 44 | 53 | 65 | **59** |
| **A06** | 53 | 57 | 55 | 58 | 47 | 53 | 51 | 39 | 45 | 58 | 60 | **59** | 57 | 44 | 51 |
| **A07** | 38 | 74 | 56 | 51 | 68 | **60** | 50 | 56 | 53 | 42 | 63 | 52 | 44 | 43 | 44 |
| **A08** | 53 | 51 | 52 | 49 | 63 | **56** | 33 | 42 | 38 | 51 | 43 | 47 | 35 | 58 | 47 |
| **A09** | 46 | 71 | 58 | 46 | 71 | 58 | 54 | 68 | **61** | 49 | 56 | 52 | 50 | 72 | **61** |

We observe that our results are lower than reported in other motor imagery studies with NeuCube on different data [78, 79], and only 5 (A01, A03, A04, A07, A09) out of 9 subjects were able to score 60% or more of overall accuracy at least with one preprocessing condition. We cannot observe a clear trend that a certain preprocessing provides the best results, however, in all cases, it yields better accuracy than raw EEG. A bar plots representing overall accuracy for every method for each subject can be found in Figure 15. It suggests that for all subjects, except A06, filtering improves the results. However, if we take a closer look and compare the accuracy of raw and filtered methods, we see that in many cases the improvement gained with filtering is rather small (see Figure 16). As for ICA methods, we observe that it is less stable than filtering, and for certain subjects, a particular ICA approach may improve the classification accuracy, while for others this does not hold. This is expectable since the quality of ICA decomposition depends on each subject's data. We also note that Infomax provides the worst and most inconsistent between-subject results. Figure 17 shows the performance comparison of raw and FastICA-based methods. For 2 subjects (A07, A08) raw data outperforms ICA-cleaned data, however, in other cases ICA seems to improve results even to a larger extent than filtering. As for mentioned exceptional subjects (A07, A08), for which raw outperforms ICA-processed data, we cannot find a definitive factor that causes that. Both subjects have a high number of retained components for FastICA (17 and 18, respectively), however, that would suggest that the processed EEG is even more similar to the raw one, therefore, we cannot claim that this was the reason for performance drop. We suspect that incorrect automatic component labeling could lead to the rejection of the relevant ICs. We also compare all ICA methods in Figure 18 and specify the number of components retained on top of each bar to see if there is a dependency. We cannot conclude that number of preserved components influence the outcome. However, we take a look into actual ICA decompositions, specifically in cases where ICA shows low accuracy, to inspect which components have been rejected with respect to their order (i. e. the lower the




(a) Subject A01 (b) Subject A02 (c) Subject A03

(d) Subject A04 (e) Subject A05 (f) Subject A06

(g) Subject A07 (h) Subject A08 (i) Subject A09

Figure 15: The accuracy obtained after 5-fold cross-validation on session 1 data.

order the more data the component accounts for). In some cases e. g. subject A05, the component ordered as 1 was removed after Infomax and FastICA methods, leading to 47% and 44% accuracy respectively, while Reduced FastICA preserved the component ordered as 1 and scored 59%. In other cases e. g. subject A09, both, Infomax and Reduced FastICA scored 61% with component ordered 1 removed, while FastICA alone kept component 1 and achieved 52% only. Therefore, we note the results are potentially influenced by the accuracy of the automated labeling: misclassified components ranked with low order can potentially be erroneously rejected or retained, thus affecting the accuracy. Additionally, we investigate the per-class accuracy of left and right hands motor imagery (see Figure 19). We observe a certain imbalance between classes recognition with the left hand scoring lower than the right hand. This is true 7 out of 9 subjects (except A01 and A08) for raw, filtered (except A04 and A06) and Reduced FastICA (except A06 and A07) approaches. In the case of Infomax and FastICA, we observe that left hand recognition has a lower rate for 6 and 5 subjects out of 9, respectively.

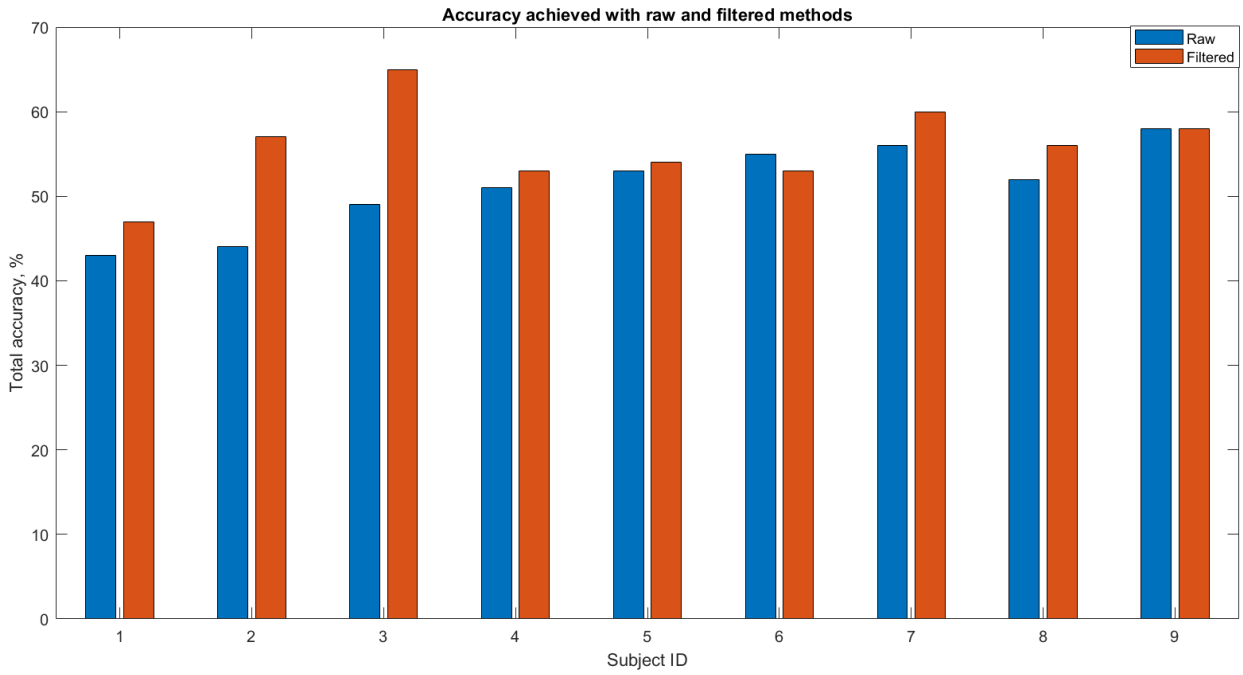



Figure 16: The accuracy obtained after 5-fold cross-validation on session 1 data. Comparison between raw (blue) and filtered (orange) data.

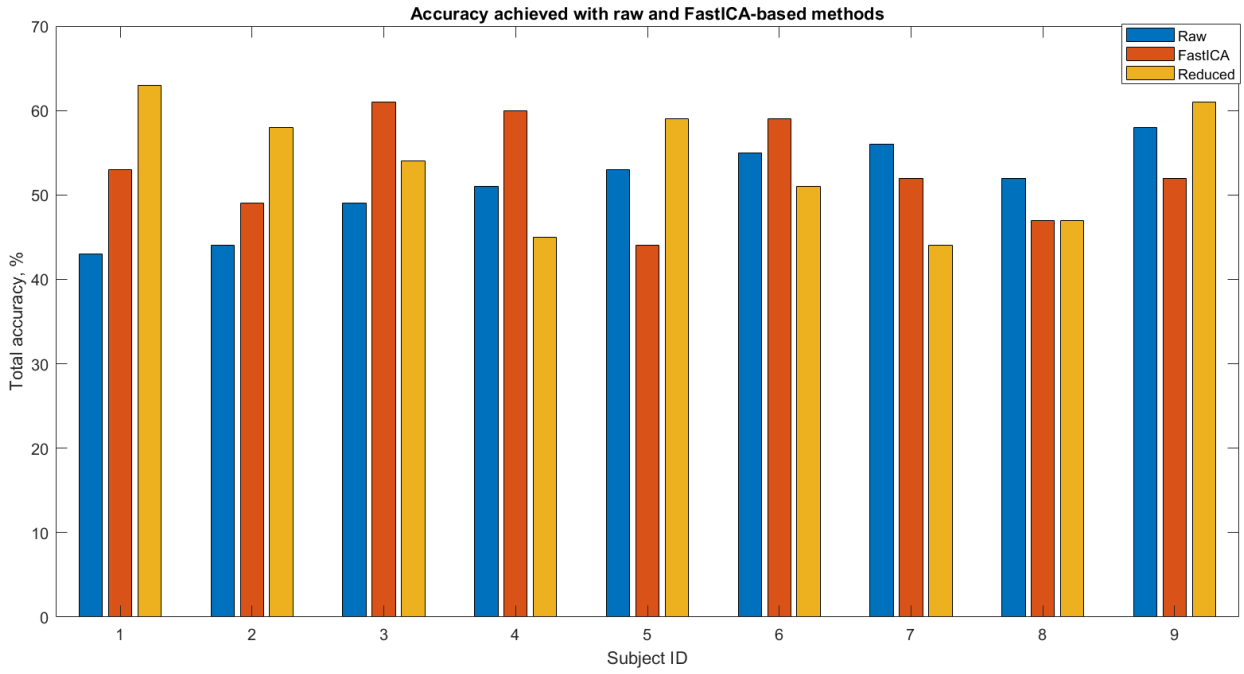



Figure 17: The accuracy obtained after 5-fold cross-validation on session 1 data. Comparison between raw (blue) and FastICA-based results (FastICA in orange, Dimentionality Reduction + FastICA in yellow).

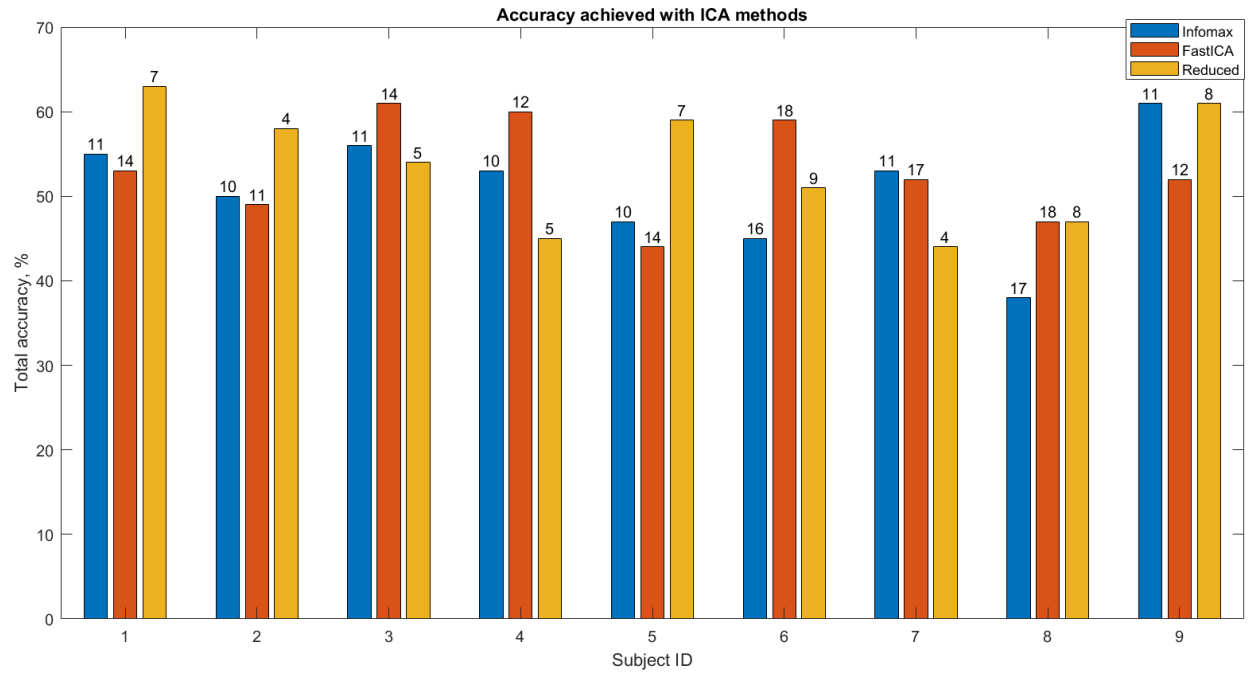



Figure 18: The accuracy obtained after 5-fold cross-validation on session 1 data. Comparison between Infomax (blue) and FastICA-based results (FastICA in orange, Dimentionality Reduction + FastICA in yellow). The number of retained components is specified on top of each bar.

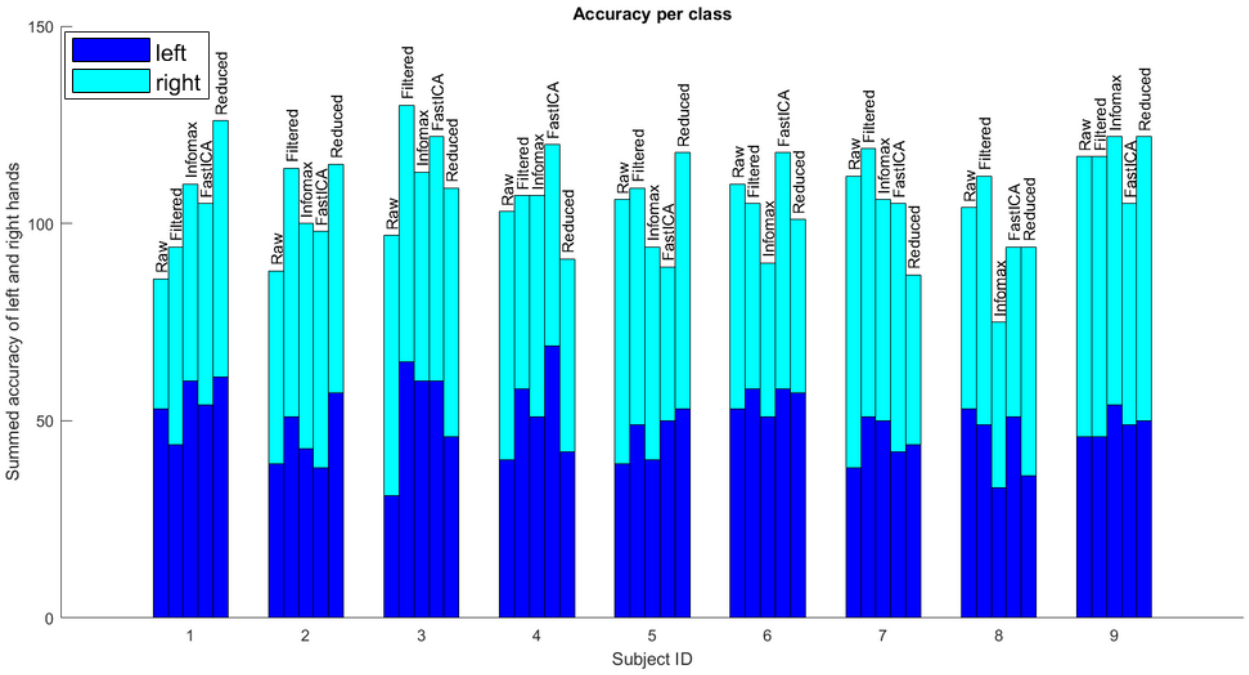


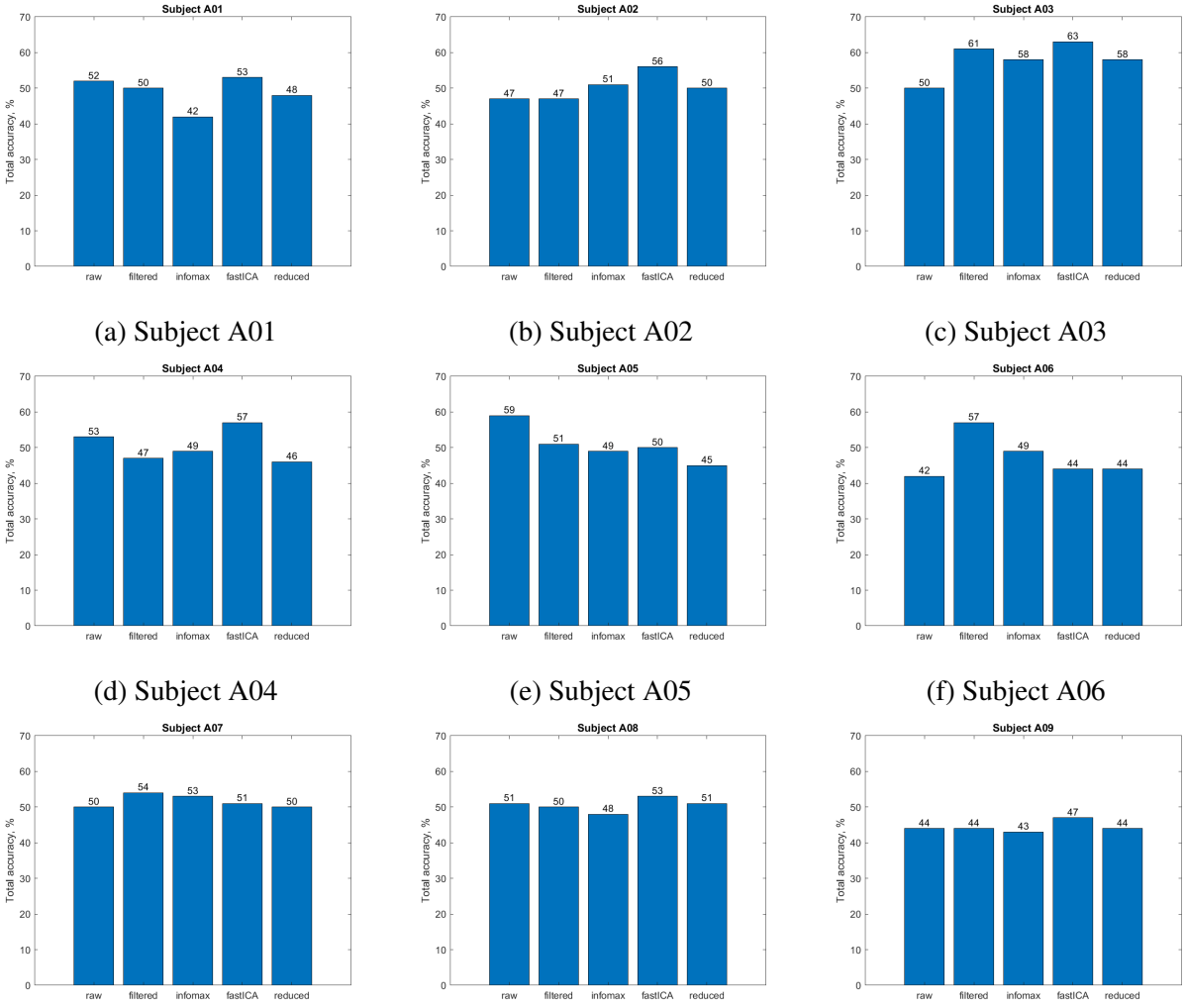

Figure 19: Per class (left hand - blue, right hand - cyan) accuracy obtained after 5-fold cross-validation on session 1 data.

We also used the model trained on the entire data of session 1 and performed a recall operation on session 2 EEG. The per class and average accuracy obtained can be found in Table 4. We see that the accuracy drops in comparison to session 1 evaluation, and only one subject (A03) was able to surpass 60% accuracy score. Figure 20 shows overall accuracy for every subject for each preprocessing technique. In that case, it is even harder to see a trend w. r. t. preprocessing method. Similar to session 1, Infomax does not provide good results, but unlike session 1, FastICA with prior dimensionality reduction leads to lower performance. Filtering outperforms the raw approach in only 3 cases (A03, A07, A06). Comparison between raw, filtered, and FastICA results can be seen in Figure 21. Although in general accuracy is low, FastICA provides the best results in 6 out of 9 cases.

Table 4: The average accuracy and accuracy per class obtained after recall operation on session 2 data. The highest result within a subject is in bold.

| **Subject** | **Raw** | | | **Filtered** | | | **Infomax** | | | **FastICA** | | | **Reduced** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | lh | rh | avg | lh | rh | avg | lh | rh | avg | lh | rh | avg | lh | rh | avg |
| **A01** | 42 | 63 | 52 | 60 | 40 | 50 | 44 | 40 | 42 | 51 | 54 | **53** | 53 | 33 | 48 |
| **A02** | 36 | 58 | 47 | 76 | 19 | 47 | 40 | 63 | 51 | 36 | 76 | **56** | 28 | 72 | 50 |
| **A03** | 50 | 50 | 50 | 60 | 63 | 61 | 54 | 61 | 58 | 56 | 69 | **63** | 58 | 58 | 58 |
| **A04** | 44 | 61 | 53 | 43 | 50 | 47 | 47 | 51 | 49 | 60 | 54 | **57** | 42 | 50 | 46 |
| **A05** | 49 | 69 | **59** | 49 | 54 | 51 | 79 | 18 | 49 | 99 | 1 | 50 | 46 | 44 | 45 |
| **A06** | 43 | 42 | 42 | 56 | 58 | **57** | 50 | 49 | 49 | 43 | 44 | 44 | 44 | 44 | 44 |
| **A07** | 17 | 83 | 50 | 46 | 63 | **54** | 18 | 88 | 53 | 21 | 82 | 51 | 26 | 74 | 50 |
| **A08** | 54 | 47 | 51 | 100 | 0 | 50 | 43 | 53 | 48 | 64 | 42 | **53** | 60 | 43 | 51 |
| **A09** | 49 | 40 | 44 | 44 | 43 | 44 | 60 | 26 | 43 | 50 | 43 | **47** | 53 | 35 | 44 |

As for the per-class accuracy of session 2, we can still observe an imbalance between left and right hand recognition rates. In some cases, this is most likely due to the relatively significant changes in electrode montage (e.g. subject A07), as we see a big difference for all preprocessing methods. If we look at subject A05, we notice that in cases of Infomax and FastICA, the classification accuracy of the left hand is significantly higher, which suggests that ICA decomposition obtained after analysis of session 1 data does not fit for session 2 recording. Finally, we also observe that filtering leads to a big imbalance for subjects A02 and A08. As we do not see similar behaviour for other preprocessing methods for these subjects, nor for filtering of session 1 data, we cannot be sure what causes such results. Finally, we examine how performance varies across sessions for each subject method-vice (see Figure 23). We cannot observe a clear tendency, however, we notice that in cases where FastICA with prior dimensionality reduction provided the best results when applied to session 1 data, it does not work as well for session 2 data. FastICA without dimensionality reduction and filtered approaches have comparable across-session performance.




(a) Subject A01 (b) Subject A02 (c) Subject A03

(d) Subject A04 (e) Subject A05 (f) Subject A06

(g) Subject A07 (h) Subject A08 (i) Subject A09

Figure 20: The accuracy obtained after recall operation on session 2 data.

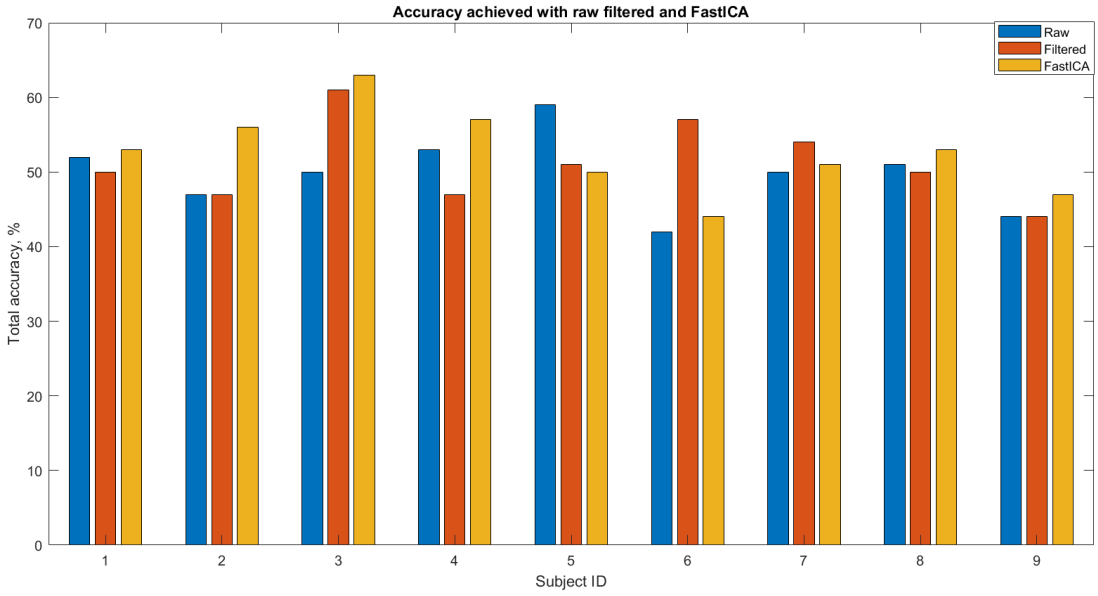



Figure 21: The accuracy obtained after recall operation on session 2 data. Comparison between raw (blue), filtered (orange), and FastICA (yellow).

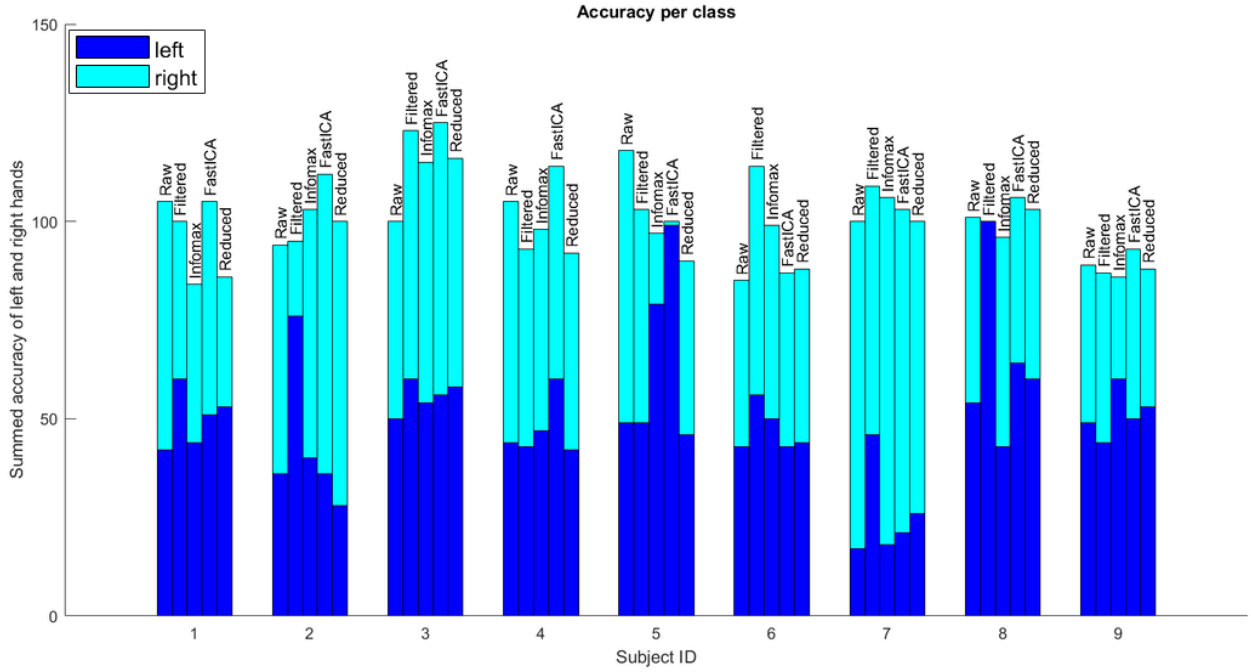
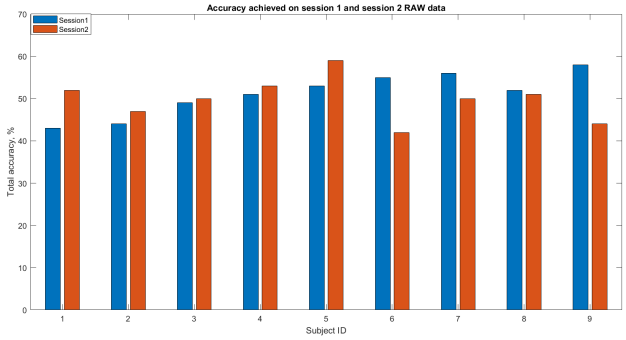
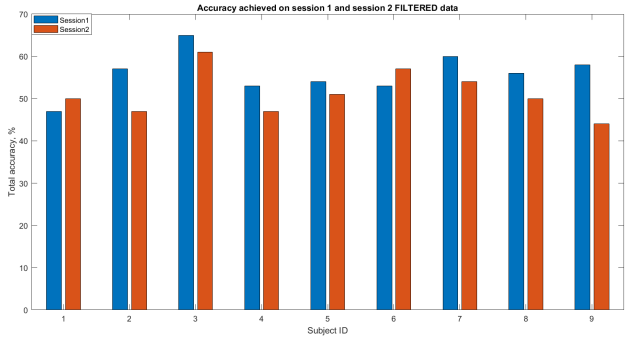


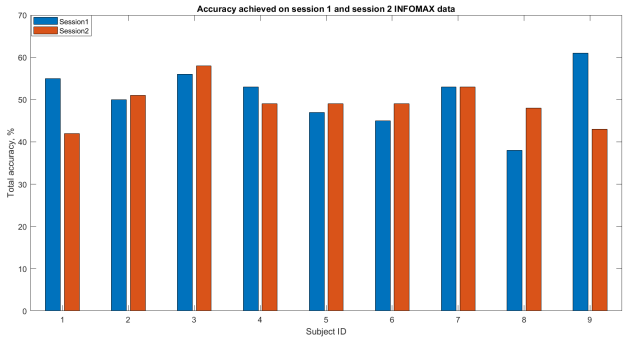

Figure 22: Per class (left hand - blue, right hand - cyan) accuracy obtained after recall operation on session 2 data.



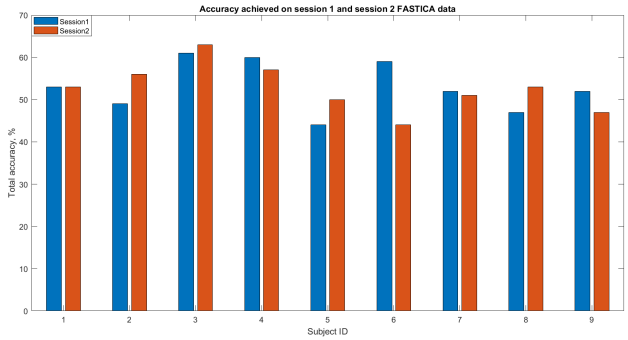

(a) Raw



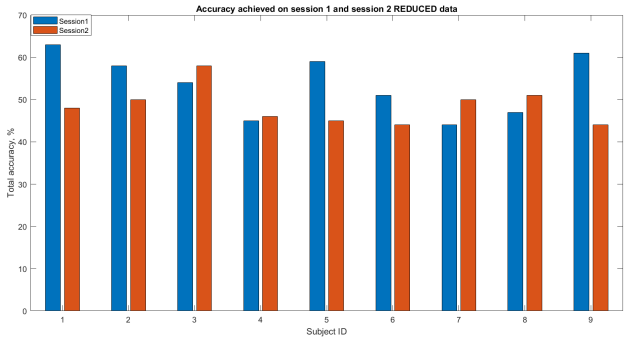

(b) Filtered




(c) Infomax




(d) FastICA




(e) Reduced

Figure 23: Comparison between accuracy obtained after 5-fold cross-validation of session 1 data (blue) and recall operation on session 2 data (orange).

# 5 Conclusion

The goal of our project was to study the suitability of the NeuCube Framework for Motor Imagery EEG classification, investigate the possibility of transfer learning and the influence of EEG preprocessing on classification accuracy. To answer these questions we chose a well-known data set, that so far has not been investigated with Neucube. It consists of 2 sessions, left and right hand imagined movements were considered in our study. We developed different approaches for EEG preprocessing and ended up with 5 conditions, namely raw, 8-32Hz filtered, and denoised with the aid of ICA data. Infomax, FastICA, and FastICA with prior dimensionality reduction were used. As for classification with NeuCube, we set the same parameters for all subjects and conditions. A threshold for EEG-to-spikes conversion with the Step Forward algorithm was derived using data from session 1 by trying to maximize SNR. A 3D Cube reservoir was designed according to cortical projection and consisted of 65 spiking neurons. We tested our mapping on the demo data supplied with the framework and observed that it yields a better 5-fold cross-validation result, as opposed to the map proposed by the framework authors (77% vs 60%). Considering hyperparameters, most of them were the same as used in other EEG studies, as we could not see that they influence the results to a large extent, and therefore concluded that the proposed values are suitable for most EEG data. We optimised only parameters which varied across studies and have an observable influence on the classification accuracy (*mod* and $drift$). We first performed 5-fold cross-validation to validate our results on data from session 1 and then did a recall operation on session 2 to investigate the transfer learning possibility.

The results suggest that the current model does not perform sufficiently well on the motor imagery data set used in the study. The maximum accuracy reached was 65% (subject A03, filtered) on session 1 and 63% (subject A03, FastICA) on session 2. This is lower than in other motor imagery studies with NeuCube, which reported 76% and 75% accuracy (see Section 2). We note, however, that the electrode montage was different in those studies. It had fewer sensors i.e. fewer dimensions (14 or 3 bipolar as opposed to 22 in our case). Moreover, in a study that scored 76%, only 3 subjects participated and wrist motor imagery was performed with closed eyes. Also, the results were validated with a 50/50 split, therefore, cannot be compared directly to 5-fold cross-validation results. Still, we believe that more thorough parameter optimisation with more extensive grid search or genetic algorithm may improve the results obtained with our data set. With regard to preprocessing techniques, we could observe that in general, it helps to improve the results, with filtering and FastICA-based methods performing best on session 1. Infomax leads to worse results, but its direct comparison to FastICA w.r.t. number of retained components or the order of components removed does not clarify the difference in the performance. It should be also noted that we highly rely on automated component labeling, and misclassification and further rejection may cause ICA-based methods to be unstable across subjects. Yet, we could not see a clear trend and the ultimate preprocessing technique for all subjects, but rather we can conclude that the accuracy depends on the individual subject. For example, A03 classification was most successful for most conditions on both sessions. Regarding transfer learning and recall on session 2, we see the expected drop in the performance. In some cases, performance across sessions within the same condition is comparable, while it is not true for other cases. This can be attributed to the change of the electrode montage or ICA decomposition derived from session 1 data not being suitable for session 2 EEG. We also note that FastICA with dimensionality reduction and filtering show worse results than in session 1, while just FastICA seems to be the most stable across sessions. We suggest that more suitable NeuCube parameters, which would improve the classification accuracy on session 1 data may, in turn, increase the results of other sessions. Furthermore, since filtering is rather a crude technique, and together with Infomax and Reduced FastICA has less stable across-

session behaviour, more attention should be put to the FastICA method, which in fact is faster than the Infomax algorithm.

In conclusion, the problem of EEG time series classification without feature extraction is a difficult task. However, the potential of fast online motor imagery recognition is still to be investigated. Since spiking neural networks learn considering temporal dimension, and have a possibility to be implemented on neuromorphic architecture, more research is needed to investigate NeuCube or other SNN-based models for robust imagined movements classification with minimal human intervention. That would not only bring closer the integration of a new interface into the daily life of healthy people but also help those for whom Brain-Computer Interface is the only way to communicate with the world.

# 6 Future Work

Our results showed accuracy that is lower than reported in other motor imagery classification with the NeuCube framework. One of the most likely reasons is that the parameters used are suboptimal. Therefore, optimisation tools provided by the framework like genetic algorithm or more extensive grid search should be used. Potentially other approaches for computing the threshold for encoding EEG into spikes should be investigated. Although it might be the case that the data set used with its electrode setup is indeed hard for NeuCube, the question of the influence of preprocessing is still relevant. Therefore, we are planning to apply our preprocessing pipelines to the data used in other NeuCube motor imagery classification study which reached 75% accuracy with raw EEG, and see if it has any effect on the accuracy. Moreover, further investigation of FastICA-based methods is needed. First of all, since we rely on automated component labeling, the rejection criteria should take into account the order of the component classified as non-brain originated. If it is low and the component accounts for most of the variance, the likelihood of it being a brain component should be treated more carefully. As for prior dimensionality reduction, performing Kernel Principal Component Analysis rather than just PCA, in case the data are confined to nonlinear subspace might be a valid approach.In case we discover that the NeuCube framework is indeed not suitable for motor imagery classification with various electrode setups and data dimensionality, we still believe that other deep learning, SNN-based methods, in particular, should be designed and tested. Various spiking neuron models and their configuration should be explored such that no human intervention is required to engineer and extract features, and temporal EEG patterns can be recognised in an efficient online manner.

# Bibliography

[1] D. S. Cantor, "1 - an overview of quantitative eeg and its applications to neurofeedback," in *Introduction to Quantitative EEG and Neurofeedback* (J. R. Evans and A. Abarbanel, eds.), pp. 3–27, San Diego: Academic Press, 1999.

[2] F. Lotte, C. Nam, and A. Nijholt, *Introduction: Evolution of Brain-Computer Interfaces*, pp. 1–8. United Kingdom: CRC Press, Jan. 2018.

[3] J. R. Wolpaw, J. del R. Millán, and N. F. Ramsey, "Chapter 2 - brain-computer interfaces: Definitions and principles," in *Brain-Computer Interfaces* (N. F. Ramsey and J. del R. Millán, eds.), vol. 168 of *Handbook of Clinical Neurology*, pp. 15–23, Elsevier, 2020.

[4] R. Leeb, R. Chavarriaga, and J. d. R Millán, *10. Brain-Machine Symbiosis*, pp. 175–197. De Gruyter Open Poland, 2016.

[5] C. Clément, *Dreamers*, pp. 255–260. Cham: Springer International Publishing, 2019.

[6] J. Kögel, R. J. Jox, and O. Friedrich, "What is it like to use a bci? – insights from an interview study with brain-computer interface users," *BMC Medical Ethics*, vol. 21, no. 1, 2020.

[7] V. A. Maksimenko, S. V. Heukelum, V. V. Makarov, J. Kelderhuis, A. Lüttjohann, A. A. Koronovskii, A. E. Hramov, and G. V. Luijtelaar, "Absence seizure control by a brain computer interface," *Scientific Reports*, vol. 7, no. 1, 2017.

[8] C. Guger, V. Prabhakaran, R. Spataro, D. J. Krusienski, and A. O. Hebb, "Editorial: Breakthrough bci applications in medicine," *Frontiers in Neuroscience*, vol. 14, p. 1128, 2020.

[9] A. Fernandez-Rodriguez, F. Velasco-Alvarez, M. Bonnet-Save, and R. Ron-Angevin, "Evaluation of switch and continuous navigation paradigms to command a brain-controlled wheelchair," *Frontiers in Neuroscience*, vol. 12, p. 438, 2018.

[10] D. A. Moses, M. K. Leonard, J. G. Makin, and E. F. Chang, "Real-time decoding of question-and-answer speech dialogue using human cortical activity," *Nature Communications*, vol. 10, no. 1, 2019.

[11] "Valve is working on brain-computer interface gaming, president reveals," Jan 2021. Available at `https://www.independent.co.uk/life-style/gadgets-and-tech/valve-brain-computer-interface-video-game-b1792225.html`.

[12] A. Nijholt, *Introduction: Brain-Computer Interfaces for Artistic Expression*, pp. 1–29. Cham: Springer International Publishing, 2019.

[13] B. Laar, I. Brugman, F. Nijboer, M. Poel, and A. Nijholt, "Brainbrush, a multimodal application for creative expressivity," 02 2013.

[14] J. I. Münßinger, S. Halder, S. C. Kleih, A. Furdea, V. Raco, A. Hösle, and A. Kübler, "Brain painting: First evaluation of a new brain–computer interface application with als-patients and healthy volunteers," *Frontiers in Neuroscience*, vol. 4, 2010.

[15] Y. Wang, M. Nakanishi, and D. Zhang, *EEG-Based Brain-Computer Interfaces*, pp. 41–65. Singapore: Springer Singapore, 2019.

[16] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update," *Journal of Neural Engineering*, vol. 15, p. 031005, apr 2018.

[17] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: a review," *Journal of Neural Engineering*, vol. 16, p. 031001, apr 2019.

[18] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. Mcalpine, and Y. Zhang, "A survey on deep learning-based non-invasive brain signals:recent advances and new frontiers," 2020.

[19] Z. Gao and X. Wang, *Deep Learning*, pp. 325–333. Singapore: Springer Singapore, 2019.

[20] A. Samadzadeh, F. S. T. Far, A. Javadi, A. Nickabadi, and M. H. Chehreghani, "Convolutional spiking neural networks for spatio-temporal feature extraction," 2021.

[21] A. Gupta and L. N. Long, "Character recognition using spiking neural networks," in *2007 International Joint Conference on Neural Networks*, pp. 53–58, 2007.

[22] M.-J. Escobar, G. Masson, T. Viéville, and P. Kornprobst, "Action recognition using a bio-inspired feedforward spiking network," *International Journal of Computer Vision*, vol. 82, pp. 284–301, 05 2009.

[23] A. Tavanaei and A. Maida, "Bio-inspired multi-layer spiking neural network extracts discriminative features from speech signals," *Lecture Notes in Computer Science*, p. 899–908, 2017.

[24] J. Liu, G. Wu, Y. Luo, S. Qiu, S. Yang, W. Li, and Y. Bi, "Eeg-based emotion classification using a deep neural network and sparse autoencoder," *Frontiers in Systems Neuroscience*, vol. 14, p. 43, 2020.

[25] C. Tan, M. Šarlija, and N. Kasabov, "Neurosense: Short-term emotion recognition and understanding based on spiking neural network modelling of spatio-temporal eeg patterns," *Neurocomputing*, vol. 434, pp. 137–148, 2021.

[26] Z. G. Doborjeh, M. Doborjeh, and N. Kasabov, "Eeg pattern recognition using brain-inspired spiking neural networks for modelling human decision processes," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2018.

[27] C. Tang, L. Xu, P. Chen, Y. He, A. Bezerianos, and H. Wang, "A novel multiple motor imagery experimental paradigm design and neural decoding," in *2020 Chinese Automation Congress (CAC)*, pp. 4024–4028, 2020.

[28] C. Virgilio González, H. Sossa, J. Antelis, and L. Falcon, *Motor Imagery Task Classification in EEG Signals with Spiking Neural Network*, pp. 14–24. 01 2019.

[29] N. Kasabov, "Neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data," *Neural Networks*, vol. 52, 04 2014.

[30] N. Kasabov and E. Capecci, "Spiking neural network methodology for modelling, classification and understanding of eeg spatio-temporal data measuring cognitive processes," *Information Sciences*, vol. 294, p. 565–575, 2015.

[31] N. Kasabov, N. M. Scott, E. Tu, S. Marks, N. Sengupta, E. Capecci, M. Othman, M. G. Doborjeh, N. Murli, R. Hartono, and et al., "Evolving spatio-temporal data machines based on the neucube neuromorphic framework: Design methodology and selected applications," *Neural Networks*, vol. 78, p. 1–14, 2016.

[32] S. Herculano-Houzel, "The human brain in numbers: a linearly scaled-up primate brain," *Frontiers in Human Neuroscience*, vol. 3, p. 31, 2009.

[33] P. Chaudhary and R. Agrawal, *Brain Computer Interface: A New Pathway to Human Brain*, pp. 99–125. 06 2020.

[34] S. Sanei, p. 40–47. John Wiley &; Sons Inc., 2013.

[35] "International federation of clinical neurophysiology (ifcn) – eeg research workgroup: Recommendations on frequency and topographic analysis of resting state eeg rhythms. part 1: Applications in clinical research studies," *Clinical Neurophysiology*, vol. 131, no. 1, pp. 285–307, 2020.

[36] R. Hari and R. Salmelin, "Human cortical oscillations: a neuromagnetic view through the skull," *Trends in Neurosciences*, vol. 20, no. 1, pp. 44–49, 1997.

[37] J. A. Pineda, "The functional significance of mu rhythms: Translating "seeing" and "hearing" into "doing"," *Brain Research Reviews*, vol. 50, no. 1, pp. 57–68, 2005.

[38] "General structure of a neuron synapse," Apr 2020. Available at `https://www.getbodysmart.com/nervous-system/neuron-synapse-structure`.

[39] X. Xia and L. Hu, *EEG: Neural Basis and Measurement*, pp. 7–21. Singapore: Springer Singapore, 2019.

[40] N. K. Kasabov, *Deep Learning and Deep Knowledge Representation of EEG Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019.

[41] H. Jasper, "Report of the committee on methods of clinical examination in electroencephalography," *Electroencephalography and Clinical Neurophysiology*, vol. 10, pp. 370–375, 1958.

[42] B. Garcia, *Motor imagery: Emerging practices, role in physical therapy and clinical implications*. 01 2015.

[43] "10–20 system (eeg)," Jan 2021. Available at `https://en.wikipedia.org/wiki/10%E2%80%9320_system_(EEG)`.

[44] "International 10-20 system for eeg-mcn.svg." Available at `https://commons.wikimedia.org/wiki/File:International_10-20_system_for_EEG-MCN.svg`.

[45] M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch, "Emg and eog artifacts in brain computer interface systems: A survey," *Clinical Neurophysiology*, vol. 118, no. 3, pp. 480–494, 2007.

[46] J. A. Urigüen and B. Garcia-Zapirain, "EEG artifact removal—state-of-the-art and guidelines," *Journal of Neural Engineering*, vol. 12, p. 031001, apr 2015.

[47] A. de Cheveigné and I. Nelken, "Filters: When, why, and how (not) to use them," *Neuron*, vol. 102, no. 2, pp. 280–293, 2019.

[48] K. J. Miller, G. Schalk, E. E. Fetz, M. den Nijs, J. G. Ojemann, and R. P. N. Rao, "Cortical activity during motor execution, motor imagery, and imagery-based online feedback," *Proceedings of the National Academy of Sciences*, vol. 107, no. 9, pp. 4430–4435, 2010.

[49] "Eeglab wiki: a.filtering." Available at `https://eeglab.org/tutorials/05_Preprocess/Filtering.html`.

[50] M. X. Cohen, *Analyzing neural time series data: theory and practice*. The MIT Press, 2014.

[51] S. Butterworth *et al.*, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.

[52] A. Widmann, E. Schröger, and B. Maess, "Digital filter design for electrophysiological data – a practical approach," *Journal of Neuroscience Methods*, vol. 250, pp. 34–46, 2015. Cutting-edge EEG Methods.

[53] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[54] S. Makeig, A. Bell, T.-P. Jung, and T. Sejnowski, "Independent component analysis of electroencephalographic data," vol. 8, 08 1996.

[55] "Faster independent component analysis for real data." Available at `https://team.inria.fr/parietal/research/statistical-and-machine-learning-methods-for-large-scale-data/faster-independent-component-analysis-for-real-data/`.

[56] L. Pion-Tonachini, K. Kreutz-Delgado, and S. Makeig, "Iclabel: An automated electroencephalographic independent component classifier, dataset, and website," *NeuroImage*, vol. 198, pp. 181–197, 2019.

[57] A. Delorme and S. Makeig, "Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis," *Journal of Neuroscience Methods*, vol. 134, no. 1, p. 9–21, 2004.

[58] M. Nunez, P. Nunez, and R. Srinivasan, *Electroencephalography (EEG): neurophysics, experimental methods, and signal processing*, pp. 175–197. 01 2016.

[59] I. Winkler, S. Debener, K.-R. Müller, and M. Tangermann, "On the influence of high-pass filtering on ica-based artifact reduction in eeg-erp," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4101–4105, 2015.

[60] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[61] C. Brunner, M. Naeem, R. Leeb, B. Graimann, and G. Pfurtscheller, "Spatial filtering and selection of optimized components in four class motor imagery eeg data using independent components analysis," *Pattern Recognition Letters*, vol. 28, no. 8, pp. 957–964, 2007.

[62] F. Artoni, A. Delorme, and S. Makeig, "Applying dimension reduction to eeg data by principal component analysis reduces the quality of its subsequent independent component decomposition," *NeuroImage*, vol. 175, pp. 176–187, 2018.

[63] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks : the official journal of the International Neural Network Society*, vol. 13 4-5, pp. 411–30, 2000.

[64] A. Hyvsrinen, J. Ssrels, and R. Vigário, "Spikes and bumps: Artefacts generated by independent component analysis with insufficient sample size," 03 2000.

[65] S.-i. Amari, A. Cichocki, and H. Yang, "A new learning algorithm for blind signal separation.," vol. 8, pp. 757–763, 01 1995.

[66] J. Cardoso and A. Souloumiac, "Blind beamforming for non gaussian signals," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, pp. 362 – 370, 01 1994.

[67] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[68] P. Ablin, J.-F. Cardoso, and A. Gramfort, "Faster independent component analysis by preconditioning with hessian approximations," *IEEE Transactions on Signal Processing*, vol. 66, p. 4040–4049, Aug 2018.

[69] P. Ablin, J.-F. Cardoso, and A. Gramfort, "Faster ica under orthogonal constraint," 2017.

[70] R. Grandchamp, C. Braboszcz, S. Makeig, and A. Delorme, "Stability of ica decomposition across within-subject eeg datasets," vol. 2012, pp. 6735–9, 08 2012.

[71] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[72] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, p. 47–63, Mar 2019.

[73] N. K. Kasabov, "Neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data," *Neural Networks*, vol. 52, pp. 62–76, 2014.

[74] J. Hu, Z.-G. Hou, Y.-X. Chen, N. Kasabov, and N. Scott, "Eeg-based classification of upper-limb adl using snn for active robotic rehabilitation," in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 409–414, 2014.

[75] K. Kumarasinghe, M. Owen, D. Taylor, N. Kasabov, and C. Kit, "Faneurobot: A framework for robot and prosthetics control using the neucube spiking neural network architecture and finite automata theory," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4465–4472, 2018.

[76] S. Budhraja, B. Sen Bhattacharya, S. Durrant, Z. Doborjeh, M. Doborjeh, and N. Kasabov, "Sleep stage classification using neucube on spinnaker: a preliminary study," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

[77] A. Al-Nafjan, K. Alharthi, and H. Kurdi, "Lightweight building of an electroencephalogram-based emotion detection system," *Brain Sciences*, vol. 10, no. 11, 2020.

[78] D. Taylor, N. Scott, N. Kasabov, E. Capecci, E. Tu, N. Saywell, Y. Chen, J. Hu, and Z. Hou, "Feasibility of neucube snn architecture for detecting motor execution and motor intention for use in bciapplications," *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 3221–3225, 2014.

[79] J. Behrenbeck, Z. Tayeb, C. Bhiri, C. Richter, O. Rhodes, N. Kasabov, J. Espinosa Ramos, S. Furber, G. Cheng, and J. Conradt, "Classification and regression of spatio-temporal signals using neucube and its realization on spinnaker neuromorphic hardware," *Journal of Neural Engineering*, vol. 16, 12 2018.

[80] B. Petró, N. Kasabov, and R. Kiss, "Selection and optimization of temporal spike encoding methods for spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, 04 2019.

[81] L. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain Research Bulletin*, vol. 50, no. 5, pp. 303–304, 1999.

[82] S. Song, K. Miller, and L. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, pp. 919–926, 2000.

[83] J. Lancaster, M. Woldorff, L. Parsons, M. Liotti, C. Freitas, L. Rainey, P. Kochunov, D. Nickerson, S. Mikiten, and P. Fox, "Automated talairach atlas labels for functional brain mapping," *Human Brain Mapping*, vol. 10, pp. 120–131, 07 2000.

[84] N. Kasabov, K. Dhoble, N. Nuntalid, and G. Indiveri, "Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition," *Neural Networks*, vol. 41, pp. 188–201, 2013. Special Issue on Autonomous Learning.

[85] S. Thorpe and J. Gautrais, "Rank order coding," pp. 113–118, 12 1998.

[86] V. Vapnik, *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Berlin, Heidelberg: Springer-Verlag, 1982.

[87] R. Leeb and C. Brunner, "Bci competition 2008 graz data set a," 2008.

[88] "Bci competition iv," 2008. Available at `http://www.bbci.de/competition/iv/`.

[89] MATLAB, *9.8.0.1451342 (R2020a)*. Natick, Massachusetts: The MathWorks Inc., 2020. Available at `https://nl.mathworks.com/products/matlab.html`.

[90] C. Vidaurre, T. Sander, and A. Schlögl, "Biosig: The free and open source software library for biomedical signal processing," *Computational intelligence and neuroscience*, vol. 2011, p. 935364, 03 2011.

[91] J. Lopez-Calderon and S. J. Luck, "Erplab: an open-source toolbox for the analysis of event-related potentials," Mar 2014.

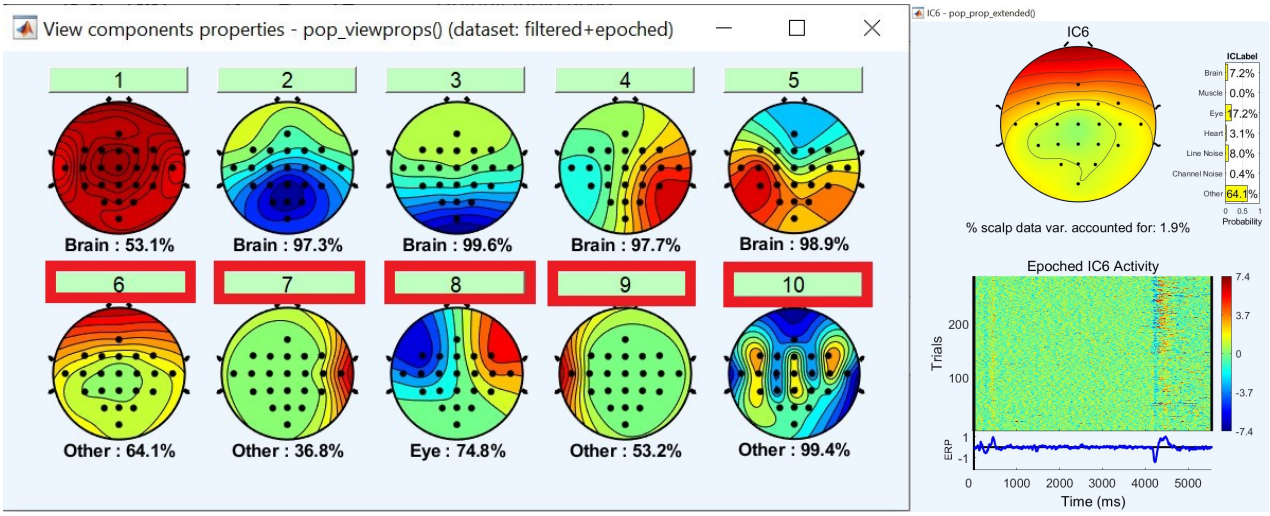
[92] L. Koessler, L. Maillard, A. Benhadid, J. Vignal, J. Felblinger, H. Vespignani, and M. Braun, "Automated cortical projection of eeg sensors: Anatomical correlation via the international 10-10 system," *NeuroImage*, vol. 46, pp. 64–72, 03 2009.

[93] K. Kumarasinghe, N. Kasabov, and D. Taylor, "Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements," *Scientific Reports*, vol. 11, p. 2486, 01 2021.

# Appendices

## A Independent Component Analysis

In this appendix section we provide more details on automated component labelling and quantitative outcome of the ICA decomposition.




(a) (b)

Figure 24: The example of ICA decomposition of subject A03 with prior dimensionality reduction with component classification by IClabel. The components for which the likelihood of being "non-brain" is higher than being "brain" are marked for rejection and highlighted in red (a). The details and the likelihood of the origin of each component can be examined (b).

Table 5: The number of retained components after performing ICA with subsequent automated artifact rejection and the average time and standard deviation needed to perform different ICA approaches. For each subject, the analysis has been run 5 times.

| **ICA algorithm** | **Subject ID** | | | | | | | | | **Running Time** | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | A01 | A02 | A03 | A04 | A05 | A06 | A07 | A08 | A09 | Avg, s | Std, s |
| Infomax | 11 | 10 | 11 | 10 | 10 | 16 | 11 | 17 | 11 | 67.24 | 18.48 |
| FastICA | 14 | 11 | 14 | 12 | 14 | 18 | 17 | 18 | 12 | 42.75 | 16.07 |
| Reduced | 7 | 4 | 5 | 5 | 7 | 9 | 4 | 8 | 8 | 12.03 | 3.14 |

## B Encoding

In this section we present optimal thresholds for every subject and preprocessing method. Negative values indicate the large standard deviation. The thresholds **t** are then used to compute the final threshold for encoding with SF algorithm.

Table 6: Optimal per-subject thresholds (in bold) for each preprocessing method.

| Subject | lh | | rh | | t |
|---|---|---|---|---|---|
| | Avg | Std | Avg | Std | |
| A01 | 2.98 | 0.73 | 2.74 | 0.66 | **2.16** |
| A02 | 2.49 | 1.47 | 2.80 | 1.45 | **1.18** |
| A03 | 3.26 | 1.90 | 3.64 | 1.92 | **1.54** |
| A04 | 2.72 | 1.48 | 2.82 | 1.22 | **1.42** |
| A05 | 2.72 | 0.74 | 2.76 | 0.70 | **2.02** |
| A06 | 3.42 | 1.63 | 3.70 | 2.11 | **1.69** |
| A07 | 2.86 | 0.97 | 2.82 | 0.97 | **1.87** |
| A08 | 4.63 | 1.61 | 4.85 | 1.58 | **3.15** |
| A09 | 4.01 | 3.10 | 4.03 | 1.78 | **1.58** |

(a) Raw

| Subject | lh | | rh | | t |
|---|---|---|---|---|---|
| | Avg | Std | Avg | Std | |
| A01 | 1.77 | 0.56 | 1.89 | 0.45 | **1.33** |
| A02 | 1.55 | 0.62 | 1.58 | 0.60 | **0.95** |
| A03 | 1.81 | 0.89 | 1.87 | 0.94 | **0.93** |
| A04 | 1.67 | 0.61 | 1.69 | 0.55 | **1.10** |
| A05 | 1.69 | 0.44 | 1.58 | 0.39 | **1.22** |
| A06 | 2.01 | 0.84 | 2.53 | 1.29 | **1.21** |
| A07 | 1.67 | 0.69 | 1.49 | 0.73 | **0.87** |
| A08 | 2.82 | 0.93 | 2.76 | 1.00 | **1.82** |
| A09 | 2.85 | 1.37 | 2.51 | 1.11 | **1.44** |

(b) Filtered

| Subject | lh | | rh | | t |
|---|---|---|---|---|---|
| | Avg | Std | Avg | Std | |
| A01 | 1.93 | 0.68 | 1.83 | 0.60 | **1.24** |
| A02 | 1.31 | 0.65 | 1.24 | 0.63 | **0.64** |
| A03 | 1.01 | 1.08 | 0.97 | 1.09 | **-0.10** |
| A04 | 1.75 | 0.96 | 1.77 | 0.81 | **0.87** |
| A05 | 1.47 | 0.53 | 1.67 | 0.50 | **1.05** |
| A06 | 3.48 | 1.63 | 3.66 | 2.07 | **1.72** |
| A07 | 2.23 | 1.14 | 2.37 | 1.01 | **1.22** |
| A08 | 4.96 | 1.49 | 4.69 | 1.56 | **3.31** |
| A09 | 1.42 | 2.87 | 1.60 | 1.75 | **-0.80** |

(c) Infomax

| Subject | lh | | rh | | t |
|---|---|---|---|---|---|
| | Avg | Std | Avg | Std | |
| A01 | 2.03 | 0.70 | 2.09 | 0.59 | **1.41** |
| A02 | 2.40 | 1.38 | 2.59 | 1.24 | **1.19** |
| A03 | 1.36 | 1.00 | 1.29 | 1.02 | **0.31** |
| A04 | 1.97 | 1.05 | 1.82 | 0.82 | **0.96** |
| A05 | 1.55 | 0.52 | 1.64 | 0.49 | **1.10** |
| A06 | 3.71 | 1.65 | 3.88 | 2.14 | **1.90** |
| A07 | 2.63 | 1.12 | 2.64 | 1.00 | **1.58** |
| A08 | 4.38 | 1.57 | 4.61 | 1.53 | **2.94** |
| A09 | 2.84 | 2.33 | 2.85 | 1.60 | **0.88** |

(d) FastICA

| Subject | lh | | rh | | t |
|---|---|---|---|---|---|
| | Avg | Std | Avg | Std | |
| A01 | 2.43 | 0.69 | 2.67 | 0.60 | **1.90** |
| A02 | 2.21 | 1.40 | 1.80 | 1.28 | **0.66** |
| A03 | 3.10 | 1.81 | 3.53 | 1.84 | **1.49** |
| A04 | 1.91 | 0.85 | 1.95 | 0.85 | **1.08** |
| A05 | 2.70 | 0.74 | 2.62 | 0.72 | **1.93** |
| A06 | 3.27 | 1.62 | 3.87 | 2.10 | **1.71** |
| A07 | 1.04 | 0.96 | 1.08 | 0.93 | **0.12** |
| A08 | 4.36 | 1.56 | 4.57 | 1.51 | **2.93** |
| A09 | 3.15 | 2.34 | 3.14 | 1.57 | **1.18** |

(e) Reduced

# C Classification

Table 7: Parameters used in other EEG studies. If the value was not reported, it is marked as "-".

| Parameter | Study | | | | | | |
|---|---|---|---|---|---|---|---|
| | Demo data | [79] | [93] | [77] | [30] | [25] | [24] |
| Potential Leak Rate | 0.002 | 0.002 | 0.002 | - | 0.02 | 0.002 | - |
| STDP rate | 0.01 | 0.01 | 0.1 | 0.01 | 0.01 | 0.001 | 0.01 |
| Threshold of firing | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | - |
| Refractory time | 6 | 6 | 6 | - | 6 | 6 | 6 |
| Mod | 0.4 | 0.9 | - | 0.8 | 0.4 | - | 1.1 |
| Drift | 0.25 | 0.01 | - | 0.005 | 0.25 | - | 0.5 |

Table 8: Total accuracy obtained after 5-fold cross-validation using different parameters for *mod* and $drift$. A = (0.85, 0.1), B = (0.4, 0.25), C = (0.9, 0.01). The highest accuracy among the 3 sets of parameters is in bold. The best set for each subject is highlighted.

| Method | Subject | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | A01 | | | A02 | | | A03 | | |
| | A | B | C | A | B | C | A | B | C |
| Raw | 44 | 43 | **56** | **44** | **44** | 43 | **49** | **49** | 48 |
| Filtered | 46 | **47** | 42 | **58** | 57 | 45 | **68** | 65 | 62 |
| Infomax | 47 | **55** | 47 | 45 | 50 | **54** | 56 | 56 | **62** |
| FastICA | 49 | **53** | 44 | 42 | 49 | **52** | **61** | **61** | 60 |
| Reduced | 56 | **63** | 56 | 48 | **58** | 48 | 53 | **54** | 50 |
| | A04 | | | A05 | | | A06 | | |
| | A | B | C | A | B | C | A | B | C |
| Raw | 44 | **51** | **51** | 41 | 53 | **61** | 49 | **55** | 42 |
| Filtered | **54** | 53 | 45 | 50 | **54** | 50 | 49 | **53** | 47 |
| Infomax | 50 | **53** | 50 | **56** | 47 | 52 | 47 | 45 | **49** |
| FastICA | 53 | **60** | 39 | 40 | 43 | **54** | 51 | **59** | 49 |
| Reduced | 45 | 45 | **50** | 58 | **59** | 53 | 44 | **51** | 50 |
| | A07 | | | A08 | | | A09 | | |
| | A | B | C | A | B | C | A | B | C |
| Raw | 49 | **56** | 47 | 43 | **52** | 51 | 51 | **58** | 52 |
| Filtered | 49 | **60** | 41 | 52 | **56** | 53 | 56 | **59** | 51 |
| Infomax | **55** | 53 | 48 | 38 | 38 | **44** | 50 | **61** | 55 |
| FastICA | 44 | **52** | 49 | 42 | **47** | 43 | 49 | **52** | 50 |
| Reduced | 44 | 44 | **50** | **47** | **47** | 46 | 54 | **61** | 53 |

Table 9: Parameters used for NeuCube classification.

| Parameter | Value |
|---|---|
| Spike theshold | 1.38 |
| Small World Radius | 3 |
| Potential leak rate | 0.002 |
| STDP rate | 0.01 |
| Threshold of firing | 0.5 |
| Training round | 1 |
| Refractory time | 6 |
| LDC probability | 0 |
| Mod | 0.4 |
| Drift | 0.25 |
| K | 3 |
| Sigma | 1 |
| C | 1 |

In this section, we provide some more details about the NeuCube learning parameters. We investigated the parameters used in other EEG studies, and this supported our conclusion that *mod* and $drift$ are the most influential and have the largest variation across studies. It can be seen that we do not include all parameters in Table 8. We have not listed Spike threshold and Small World Radius, as these parameters are optimised for our particular data set, encoding algorithm, and designed mapping. LDC probability, K, Sigma, and C were reported in very few papers. After testing these values and running our own experiments, we concluded that the values suggested by the authors of the framework suit our data as well.