UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DEPARTMENT OF INFORMATION ENGINEERING

MASTER COURSE IN BIOENGINEERING

Dimensionality Reduction techniques for latent space visualization on Motor Imagery BCI data

Supervisor:
Prof. Luca Tonin
Cosupervisor:
Dr. Tommaso Cortecchia

Student:
Morena Truglia
ID Number:
2072640

Academic Year 2023-2024

Graduation Date 10/10/2024

Un ringraziamento speciale a tutti coloro che hanno reso possibile realizzare questo lavoro e a chi mi ha sostenuta anche con un semplice consiglio o una parola di incoraggiamento. Grazie alla mia famiglia che mi ha permesso di arrivare fin qui. Grazie a Padova, la città che mi ha regalato emozioni e momenti indimenticabili e grazie a chi li ha resi tali: una dedica non sarebbe abbastanza per descrivere i due anni trascorsi. Grazie ai vecchi e ai nuovi amici e a tutte le persone importanti della mia vita, spero di avervi sempre accanto anche in futuro.

Abstract

The Brain-computer interface (BCI) is a neurotechnology capable of acquiring, processing and decoding brain signals through invasive or non-invasive acquisition techniques and translating them into commands for the external environment. This command depends on the final objective of the BCI, which can be used to control a robotic device, a rehabilitation technology, or a game on the computer screen. The type of BCI used in this study is non-invasive and based on motor imagery. Data from a single subject who participated in the Cybathlon competition for two consecutive years were used, and on this dataset comprising various offline, online, and control sessions, a deep neural network was trained. The advantage of using a deep learning approach in this context is to minimize preprocessing of the raw data and the manual feature extraction process. At the same time, a major disadvantage of this system is explainability. Therefore, the aim of this study is not so much to find the best classifier, but rather to understand how the model makes predictions and how this process changes over the sessions, in relation to user adaptation. The goal is to demonstrate that the user has learned to use the BCI over the months through the visualization of the neural network's latent space. To enable visualization in two-dimensional space, a data dimensionality reduction technique based on UMAP (Uniform Manifold Approximation and Projection) was adopted, and it emerged that there was indeed a shift in the data distribution across the sessions, particularly evident when the subject became skilled in performing the required motor imagery tasks. To obtain a quantitative correspondence, several metrics were used, in addition to the classic accuracy, which together with the visualization allowed for an in-depth study of short-term and long-term user learning.

Abstract

L’interfaccia uomo-macchina (Brain-computer interface, BCI) è una neurotecnologia capace di leggere i segnali cerebrali attraverso registrazioni invasive o non invasive e tradurli in comandi per l’ambiente esterno. Questo comando dipende dall’obiettivo finale della BCI, che può essere utilizzata per controllare un dispositivo robotico, una tecnologia di riabilitazione o un gioco sullo schermo del computer. La tipologia di BCI discussa in questo studio è non invasiva e basata sull’immaginazione motoria. Sono state utilizzate le registrazioni di un singolo soggetto che ha partecipato alla competizione Cybathlon per due anni consecutivi, e su questo set di dati, comprendente molte sessioni offline, online e di controllo, è stata addestrata una rete neurale profonda. Il vantaggio di utilizzare un approccio di deep learning in questo contesto è quello di ridurre al minimo la fase di pre-elaborazione dei dati grezzi e il processo manuale di estrazione delle caratteristiche. Allo stesso tempo, un grande svantaggio di questo sistema è la spiegabilità. Pertanto, l’obiettivo di questo studio non è tanto quello di trovare il miglior classificatore, ma piuttosto comprendere come il modello effettui le predizioni e come questo processo cambi nel corso delle sessioni, in relazione all’adattamento dell’utente. Si vuole quindi dimostrare che l’utente ha imparato nei mesi a usare la BCI attraverso la visualizzazione dello spazio latente della rete neurale. Per permettere la visualizzazione nello spazio bidimensionale è stata adottata una tecnica di riduzione della dimensionalità dei dati basata su UMAP (Uniform Manifold Approximation and Projection), e ne è emerso che c’è stato effettivamente uno spostamento nella distribuzione dei dati tra le varie sessioni, particolarmente evidente quando il soggetto è diventato abile nell’eseguire i compiti di immaginazione motoria richiesti. Per ottenere una corrispondenza quantitativa, sono state utilizzate diverse metriche, oltre alla classica accuratezza, che insieme alla visualizzazione hanno permesso uno studio approfondito dell’apprendimento dell’utente a breve e a lungo termine.

Contents

List of Figures

Chapter 1

Introduction

The Brain-computer interface (BCI) constitutes an alternative communication channel between humans and the external world. This technology interprets brain signals and decodes the user's intentions without the need for the user to move. For this reason, it can be utilized in different scenarios, such as helping impaired individuals recover motor functions or as an assistive device following permanent injuries and traumas. They can also be used to control neuroprosthetics [1] or robotic devices [2]. Unlike other types of technologies, the main characteristic of BCI is the closed loop. The loop is closed when a command is given, and the subject receives feedback of any kind, such as visual or auditory, on the issued command. For the entire system to function, reciprocal interaction between the user, the decoder, and the device is necessary, following the principle of mutual learning where the three entities learn one from each other.

1.1 A quick look at neurophysiology

Neurophysiology is a discipline that studies the functioning of the nervous system, including the interactions between neurons and how they generate electrical and chemical signals to communicate with each other. The neuron is the primary cell of the nervous system responsible for receiving and transmitting impulses, but there are also glial cells, which have an equally important function, providing structural and functional support. The neuron is essentially composed of three parts:

- **Soma**: the cell body that contains the nucleus.
- **Axon**: the extension that transmits the signal from the soma to the axon terminals.
- **Dendrites**: extensions that surround the soma and receive signals from other neurons.

Neurons communicate with each other through synapses, which can be electrical or chemical. The most common type, chemical synapses, use neurotransmitters, which are substances synthesized within the neuron that can exert an excitatory or inhibitory action on another neuron or an effector organ.

The transmission mechanism of nerve signals (*Figure 1.1*) is an electrochemical process, which can be measured and recorded from the scalp using techniques such as electroencephalography (EEG). It can be divided into four main phases: resting state, depolarization, repolarization, and hyperpolarization. In the first phase, the neuron has a resting membrane potential of about -70 mV, maintained by passive channels and ion pumps, which regulate the concentration of sodium (Na^+) and potassium (K^+) ions inside and outside the cell. When a stimulus reaches the neuron, it causes the voltage-gated sodium channels to open, and the membrane potential reverses and becomes positive due to the influx of sodium ions into the cell. When the membrane potential reaches about +30 mV, the repolarization phase begins, in which the sodium channels close and the potassium channels open, bringing the membrane potential back to negative values. During the hyperpolarization phase, the ion pumps restore the normal balance, because the outflow of potassium ions from the cell can make the membrane potential more negative than the resting potential. Important characteristics of transmission include the unidirectional propagation of the signal, which propagates along the axon forward and without attenuation, thanks to the sequential opening of ion channels [3]. Only when the action potential reaches the axon terminal, the influx of calcium ions (Ca^{2+}) causes the release of neurotransmitters in the synapses.

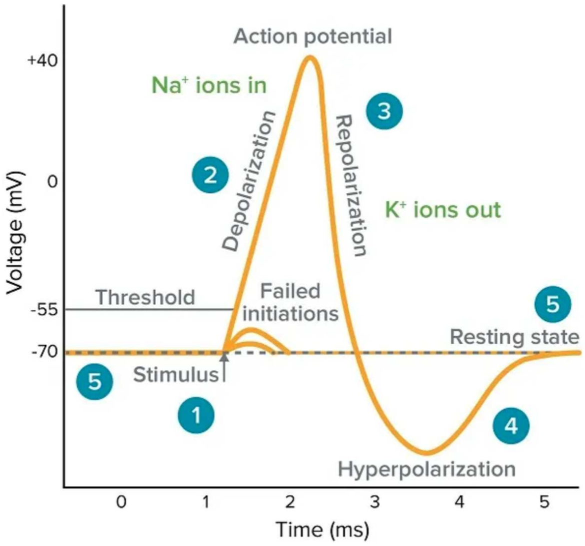


Figure 1.1: Action potential

The nervous system is composed of the Central Nervous System (CNS) and the Peripheral Nervous System (PNS). The CNS consists of the brain, protected by the skull, and the spinal cord, contained within the vertebral canal, and it serves as the main center for processing and integrating information. The brain is divided into two hemispheres, and in each of them can be distinguished: the frontal lobe, the temporal lobe, the parietal lobe, and the occipital lobe (*Figure 1.2*). The PNS includes the nerves that extend from the CNS, and its function is to transmit signals from the centers to the periphery of the body.

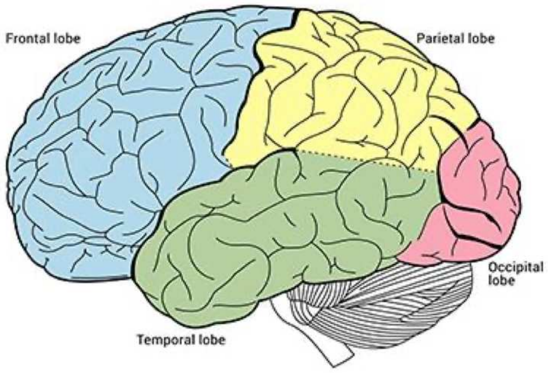


Figure 1.2: Cerebral lobes

The CNS, particularly the brain, can be studied with various tools depending on the final objective of the study. The main tools are:

- **Electroencephalogram (EEG)**: measures the electrical activity of the brain in a non-invasive way.
- **Magnetic Resonance Imaging (MRI)**: provides detailed images of the brain's structure.
- **Emission Tomography (PET)**: provides an indirect measure of brain activity.
- **Electrocorticography (ECoG)**: measures the electrical activity of the brain in a invasive way.
- **Transcranial Magnetic Stimulation (TMS)**: used to stimulate specific areas of the brain.

1.2 Brain-Computer Interface (BCI)

Many conditions such as spinal cord injury, cerebral palsy, and stroke can partially or completely damage an individual's communication channels with the external world. BCI can intervene in all these situations to restore a mode of interaction or contact through the intermediary of a machine. In this sense, BCI helps to improve the quality of life for people with disabilities and reduces the need for a caregiver [4]. There are essentially two types of BCI: assistive BCI, which aim to create a permanent communication channel with an external device, and rehabilitative BCI, which seek to restore the normal physiological communication channel by leveraging neuronal plasticity. The latter approach is preferred, especially in post-stroke rehabilitation during the preliminary phases, when the patient's recovery potential is highest [5]. The most frequent clinical applications [6] of BCI presented in the literature are:

- **Motion rehabilitation**: to recover and improve the movement of patients' upper and lower limbs.
- **Virtual reality control**: the user moves a virtual object on the monitor, and the computer transmits this movement to an external device.
- **Speech rehabilitation**: the user selects the desired letters from the monitor and composes words.

A complete BCI system is very complex because it involves multiple steps that require different skills to manage (*Figure 1.3*). It is essential to coordinate the hardware and software components as well as the human element, which is an integral part of the system. First, the brain signal is acquired through electrodes, and this signal requires preprocessing to remove noise and artifacts that may obscure the useful signal. Then follows the feature extraction phase, either manually or automatically, and classification, for which several promising strategies currently exist. At this point, the classifier output is provided as a command to the external device through an interface, and the user can see the issued command as feedback. Generally, in real applications, a command is not produced for every classifier output; instead, the output is integrated over multiple trials until there is sufficient evidence that the predicted class is correct. To achieve this, a so-called control framework is applied, allowing for a good level of precision and accuracy [7].

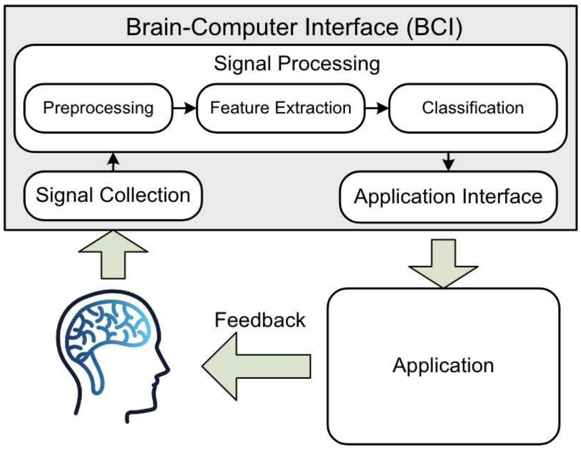


Figure 1.3: BCI operating loop

A primary distinction of BCIs can be made between invasive and non-invasive. Invasive BCI involves the implantation of electrodes directly into the brain tissue, either within the cortex or deeper. The acquired signal comes from a small population of closely situated neurons or even a single neuron. The positive aspects of invasive BCI include very high spatial and temporal resolution, allowing precise reading of neural activity, and reliability, as the recorded signal is less influenced by external noise [8]. Despite several experiments conducted on both primates and humans over time [9, 10, 11], there are aspects that make the invasive BCI challenging for real-world applications. These include the risks associated with surgical intervention (infections, brain damage, scar formation, etc.), the risk of rejection of a foreign body with possible inflammation, and the cost, as this technology is expensive both in terms of installation and maintenance. Non-invasive BCI uses electrodes applied to the scalp to record signals. The advantages are associated with safety, accessibility, and comfort. However, this type of BCI is characterized by low spatial resolution, making it more difficult to read neural activity, and it is also more susceptible to external noise and interference. Comparing the two, invasive BCIs are common in research on patients with severe motor disabilities. They are mostly used to control prostheses or assistive devices. Non-invasive BCIs, on the other hand, are used in commercial applications and are intended for therapeutic, rehabilitative, or even as an alternative interface for entertainment activities.

From the perspective of signal analysis, invasive and non-invasive BCIs also differ. The former can be studied through spike detection and sorting, using a high-pass filter to locate the activity of individual neurons, or through Local Field Potential (LFP) [12, 13], applying a low-pass filter to locate the activity of a small population of nearby neurons that fire synchronously. The second type of BCI will be discussed extensively in the next section.

1.2.1 EEG signals for non-invasive BCI

Different methods for acquiring brain signals for non-invasive BCI include EEG, MEG, fNIRS, fMRI, or PET. Each of these techniques has advantages and disadvantages. Considering a compromise between accessibility and good temporal resolution, which is essential since the purpose of BCI is to be used in real-time, EEG is preferred [14].

The EEG represents the sum of the electrical activity from multiple neurons that activate simultaneously over a certain period, recorded through surface electrodes. It mainly reflects the activity of pyramidal neurons located in the outermost layer of the cerebral cortex, and due to geometric considerations, only the current dipoles oriented vertically to the scalp contribute. The deeper areas of the brain also contribute to the surface field potentials, but indirectly. It has been estimated that each EEG electrode covers the summed activity of roughly 6 cm² of the underlying cortex (thousands of neurons) [3]. For this reason, the spatial resolution of the EEG signal is not good, and the signal requires extensive preprocessing to be cleaned from external noise. Human EEG shows activity from 1 to 30 Hz, with amplitudes ranging from 20 to 300 µV.

In some cases, the frequency of the signal is more important than its succession over time; therefore, by convention, frequency bands associated with different brain rhythms are defined:

- δ rhythms [<4 Hz]: slow waves typical of deep sleep.
- θ rhythms [4-8 Hz]: waves typical of drowsiness.
- α rhythms [8-13 Hz]: waves associated with a state of relaxation.
- β rhythms [13-30 Hz]: waves associated with concentration.
- γ rhythms [>30 Hz]: waves associated with sensory processing.

Each of these rhythms appears in the EEG signal based on the activity the subject is performing. For example, if the subject is asked to close the eyes and relax, alpha waves will appear on the occipital channels; but they will disappear as soon as he opens the eyes due to the desynchronization of the neurons involved.

The electrodes used for EEG are small metal disks, typically made of silver, covered with a layer of conductive material. They are applied with conductive paste or gel directly onto the scalp. An alternative is a cap with pre-fixed electrodes that make preparation much easier.

The placement of EEG electrodes follows the 10-20 international system (*Figure 1.4*), ensuring consistency across recordings. This system is called this way because the distances between electrodes are either 10% or 20% of the total distance between specific points on the skull, using the following important landmarks: nasion in the frontal region, vertex in the central region, and inion in the occipital region. The electrodes record the difference in potential generated by neuronal electrical activity relative to a reference electrode. The choice of reference is important for waveform analysis, while it is completely irrelevant for topographic analysis.

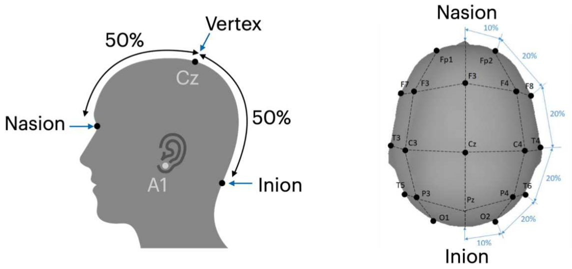


Figure 1.4: 10-20 placement system

Brain signal oscillations can be induced by an external stimulus and are called **Evoked-Related Potentials (ERPs)**, or by internal stimulation and are called **Spontaneous Potentials (SPs)**. ERPs are time-locked and phase-locked activities because they are synchronized to the onset of the stimulus and coherent in phase relative to the stimulus itself. Examples of ERPs include:

- Visual Evoked Potential (VEP) and Steady-State Visual Evoked Potential (SSVEP)
- P300 Evoked Potential (P300)
- Error Potential (ErrP)

SPs, on the other hand, are generated by voluntary mental activities that the subject decides to perform. Therefore, they are not time-locked or phase-locked, as they can have different phases and frequencies. Examples of SPs include:

- Slow Cortical Potentials (SCPs)
- Sensorymotor rhythms (Motor Imagery)

VEPs are electrophysiological responses to visual stimuli, usually presented as flashes or pattern. These are often used to assess the integrity of the visual system. SSVEPs are specific types of

VEPs recorded when the visual stimulus is presented continuously and periodically, at fixed frequencies greater than 6 Hz. The neuronal response manifests as a continuous oscillation at the same frequency as the stimulus. The P300 is a component of the brain wave that occurs approximately 300 ms after the presentation of a stimulus, which can be visual, auditory, or somatosensory. The P300 is extensively studied as it is linked to cognitive processes such as attention and memory, and BCIs based on this evoked potential are widespread. The ErrP occurs approximately 50-100 ms after the perception of an error or unexpected outcome in a task. SCPs reflect gradual changes in neuronal polarization that occur over longer time scales compared to rapid and brief action potentials. They are used in BCIs as they anticipate intentions and motor actions of the subject. Finally, motor and sensorimotor rhythms are patterns of brain activity associated respectively with movement and imagined movement.

The subsequent signal analysis varies based on the type of potential. For ERPs, the response to repeated stimuli is typically averaged to enhance the useful signal and reduce noise. Additionally, components are analyzed in the time domain considering latency and amplitude of the response. SPs do not have a precise onset, change between successive trials, and do not provide comprehensive information if analyzed over time. For this reason, they require frequency domain analysis.

1.2.2 Motor Imagery BCI

Motor Imagery (MI) is a paradigm that, as the name suggests, involves imagining a movement rather than actually performing it. This allows controlling wheelchairs, robotic arms, and prosthetic devices purely through thought [15]. The type of imagination referred to is called kinesthetic; that's the ability to mentally perceive the movement of one's own body. Therefore, the user of the BCI is asked to imagine not the movement of any hand or foot, but of their own hand and foot.

The main areas involved in the mental process of movements are:

- **Primary Motor cortex (M1)**: this area is located in the frontal lobe and plays a fundamental role in the execution and planning of voluntary movements. During motor imagery, the M1 is activated in a manner similar to when the movement is actually performed, although to a lesser extent.
- **Primary Somatosensory cortex (S1)**: this area is located in the parietal lobe, immediately behind the central sulcus. It is responsible for receiving and processing sensory

information from all over the body.

- **Premotor cortex**: This area is located anterior to the M1 and plays a role in planning complex and coordinated movements. During motor imagery, the premotor cortex is active in preparing motor plans without the physical execution of the movement.

The M1, like the S1, is specific: the topographic representation of the distribution of motor areas in the human brain is called the motor homunculus (*Figure 1.5*). This is organized in such a way that different body parts are represented proportionally to their importance and motor complexity. For example, body parts that require greater precision and control, such as the hands and face, occupy more space on the map compared to less complex body parts, such as the arms and legs. A fundamental characteristic of the motor homunculus is contralaterality, which means that the areas controlling the right side of the body are located on the left side of the motor cortex, and vice versa.

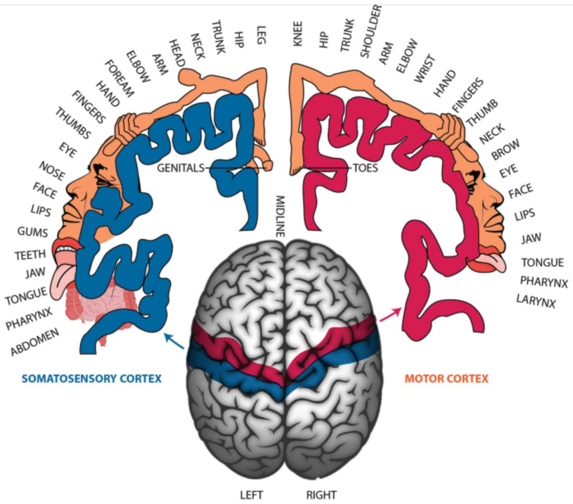


Figure 1.5: Motor homunculus

The sensorimotor rhythms used for motor imagery BCI (MI BCI) are:

- μ rhythms [8-13 Hz]
- β rhythms [13-30 Hz]

μ rhythms have the same frequency content as α rhythms but are generated in the sensorimotor cortex. During a MI task, there is desynchronization of some neurons, which can be seen on the EEG as a decrease in the amplitude of oscillations on the central electrodes. Immediately after the task is performed, there is synchronization of the involved neurons, observed as an increase in the amplitude of oscillations. Analyzing what happens during the task in the frequency domain, it is noted that desynchronization occurs in the μ band, while subsequent synchronization occurs in the β band, referred to as the β rebound. This phenomena are specifically known as **Event-Related Desynchronization (ERD)** and **Event-Related Synchronization (ERS)**. The electrodes involved are: Fz, FC3, FC1, FCz, FC2, FC4, C3, C1, Cz, C2, C4, CP1, CPz and CP2, although activation varies based on the subject and the type of task. For example, if a right-hand movement is imagined, ERD and ERS will be observed on the left electrodes (FC3, C3, CP1) and vice versa for the left hand (FC4, C4, CP2) (*Figure 1.6*). If the feet are imagined to move, the central areas of the sensorimotor cortex (Fz, Cz, CPz) will be activated. The μ band ERD starts 2.5 s before movement-onset, reaches maximal values shortly after, and recovers to baseline level within a few seconds. The central β activity, in contrast, displays a short-lasting ERD during initiation of movement followed by a synchronization with a maximum after movement execution. It is of interest that the β ERS occurs while the μ rhythm is still attenuated [14].

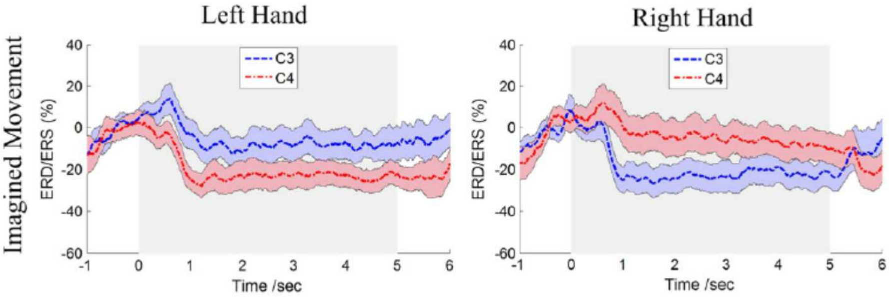


Figure 1.6: ERD/ERS

MI experiments usually take place over several sessions carried out on different days. Each session is divided into parts referred to as ‘runs’, and a run consists of repeating different tasks. Typically, the sequence of tasks is randomized while the number of trials per each task is fixed. Two to four MI tasks are currently considered among imagining the movement of a hand, both hands, a foot, both feet, the tongue, a wrist, an elbow, a forearm, or fingers [14]. The classic experimental procedure for MI BCI consists of dividing a single trial into four main parts (*Figure 1.7*):

1. **Fixation cross**: this initial relax phase is typically triggered by an auditory or a visual signal. The user must fixate on a cross at the center of the screen, which helps limit involuntary eye movements.

2. **Cue**: during this phase, the user is informed about the MI task to perform immediately

afterward.

3. **Motor Imagery**: a time interval (from 3 to 5 s) during which the assigned task must be performed and continous feedback is generated.
4. **Break**: final relax phase with random duration.

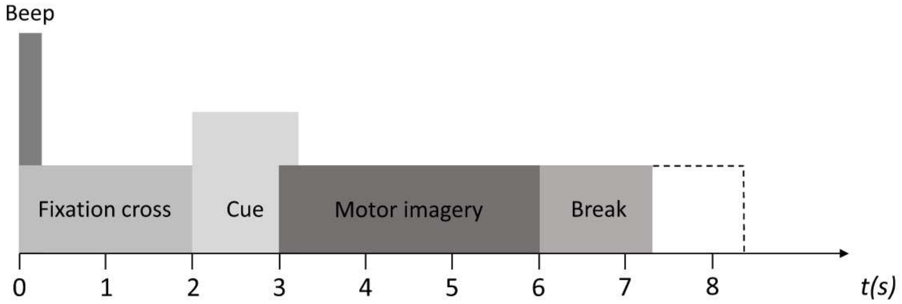


Figure 1.7: An example of MI-BCI experimental procedure

1.3 The classification algorithm

A fundamental step in the operating loop of a BCI is the choice of the most suitable algorithm for the classification of the task. The goal is to find the algorithm that performs best; however, there are issues that often hinder achieving high accuracy and are intrinsic in the acquired signal: non-stationarity, noise and interference, significant variability inter-subjects and inter-sessions. Therefore, a model that is robust and stable over time is necessary. The classification problem is closely linked to the feature extraction of the signal. In fact, the more discriminative the chosen features are for the performed task, the better the classification improves. The extraction process is done exclusively during the calibration sessions of the decoder. Once the features are selected and the model is trained offline, the decoder can be used online. The features represent those channel-frequency pairs that make it possible to discriminate between two or more classes, and it often happens that over sessions on different days, the significant pairs change, for example, due to different modulation of brain rhythms by the subject or due to changes in some experimental conditions. This makes it difficult to use the same model on subsequent days without the decoder's performance dropping drastically. Therefore, frequent calibration of the decoder is necessary, and this process can be long, as well as tiring for the subject using the BCI.

Ideally, one would want a model that, with minimal calibration, can work well for every subject and for every type of task. Unfortunately, this is not possible today, but efforts are ongoing to find something that achieves maximum prediction accuracy with minimal preprocessing work. The most commonly used algorithms in the field of BCI are supervised Machine Learning (ML) algorithms, which use labeled data to learn the characteristics of the brain signal from the specific subject. These data are used to train a model that then predicts the user's intentions on new, unseen data [16]. One type of supervised learning is classification, used for MI tasks. The simplest classification problem involves two classes, such as both hands or both feet, right hand or left hand, but more complex multiclass paradigms can also be implemented. In addition to supervised classification techniques, such as classification and regression, there are also unsupervised learning and reinforcement learning techniques (*Figure 1.8*).

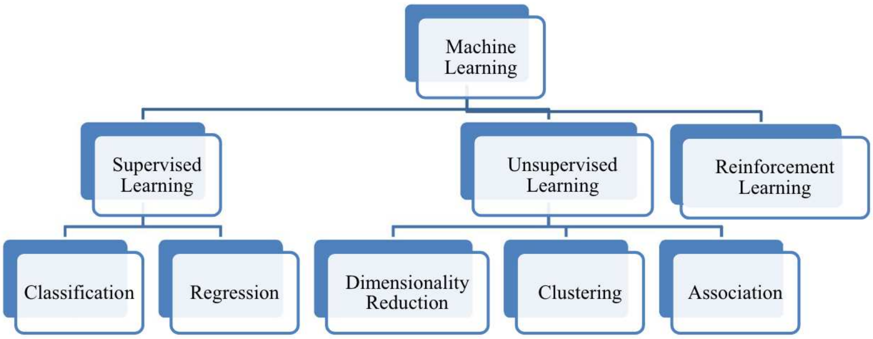


Figure 1.8: Machine learning techniques

Among the classification methods for MI BCI most frequently found in the literature, the most popular are Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM). K-Nearest Neighbor (K-NN) and Bayesian analysis are also used. A distinction must be made between linear and nonlinear classification algorithms. For example, LDA is a simple, strong algorithm that presumes the Gaussian distribution of data. However, due to the characteristics of the EEG signal, it is often impossible to find a solution with linear methods, necessitating a switch to nonlinear solutions such as Artificial Neural Networks (ANNs), K-NN, and SVMs [17]. Moreover, since the goal is always to use the BCI for real-time predictions, it is essential to find a compromise between accuracy and speed, and often the most robust algorithms are also those that take so much time and resources that they cannot be used online.

1.3.1 Deep learning for BCI

Recently, the use of Deep Neural Networks (DNNs) for BCI has been extensively explored, having previously been set aside due to two main issues: lack of data and difficulty in the neurophysiological interpretation of the algorithm. Deep Learning (DL) is a subfield of ML that involves the use of a network with a series of hidden layers between the input and output layers. These hidden layers process the data by extracting nonlinear relationships and then combine the features without any external instructions. This makes learning automatic and fast, but the processing done to reach the output is unknown, which is why these networks are considered as ”black boxes”. Moreover, DNNs, rather than simple NN, historically require much more data to achieve good results. Unfortunately, in the BCI field, collecting such large amounts of data is very difficult, if not impossible. However, it has been shown that despite the lack of data, some particular DNNs structures outperform the traditional algorithms mentioned above, making them an excellent alternative with great potential.

The most commonly used network architecture in the field of BCI is Convolutional Neural Network (CNN). Following this are Recurrent Neural Network (RNN), particularly the Long Short-Term Memory (LSTM), autoencoders (AE) and finally hybrid networks. Even with DNNs, it is necessary to find an architecture with a low computational load, that can be used in short time frames; for example, RNN takes more than 25 times as long to train as CNN for a sequence of 2500 items [17]. The advantages of using DNNs are numerous, with the most important being the limited data preprocessing required. With traditional algorithms, preprocessing is a crucial step that determines the success of all subsequent phases. Using DL, data manipulation is minimized, though still necessary to achieve high levels of prediction accuracy. Essential steps that should not be skipped include signal filtering and artifact removal, either manually or automatically, such as through Independent Component Analysis (ICA). Another significant advantage is related to feature extraction, which, in the case of DNNs, is done automatically by the network itself, simplifying the pipeline considerably since discriminative features are often not known a priori. Through DL, the so-called end-to-end learning is utilized, which consists of obtaining an output from the network directly from raw or minimally manipulated data.

1.3.2 What is a CNN

Convolutional Neural Networks (CNNs) constitute an architecture specifically designed for image analysis and processing. The first to discuss CNNs was Yann LeCun in the 1990s, and since then, these networks have revolutionized not only image recognition but are also used in the fields of computer vision, speech recognition, and natural language processing. CNNs are composed of layers, some of which are also present in traditional DNNs, such as activation layers and fully-connected layers, while others are characteristic of this type of network. The latter are:

- **Convolutional layers**: they apply filters through a moving window that slides over the input image, producing activation maps that highlight different features of the image. Each filter consists of a set of weights that are updated during the training of the network.

- **Pooling layers**: they reduce the spatial dimension of the activation maps, decreasing the number of parameters and the computational complexity of the network. Pooling can be either max pooling, where the maximum value in a window is selected, or average pooling, where the average of the values in a window is computed.

A numerical example illustrating how these layers work is shown in the *Figure 1.9*

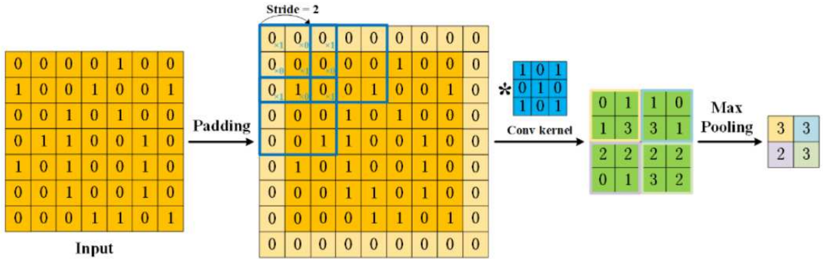


Figure 1.9: Procedure of a 2D CNN

One of the largest limitations of a traditional DNN is that it tends to struggle with the computational complexity required to compute image data [18]. The main advantages of using CNNs over traditional architectures are [19]:

- Local connections: each neuron is no longer connected to all neurons of the previous layer, but only to a small number of neurons, which is effective in reducing parameters and speed up convergence.

- Weight sharing: a group of connections can share the same weights, which reduces parameters further.

- Downsampling dimension reduction: a pooling layer harnesses the principle of image local correlation to reduce the amount of data while retaining useful information.

When using CNNs on multichannel EEG signals, the latter is treated as if it were an image, achieving excellent classification performance with far fewer parameters compared to, for example, LSTMs, which are also used for the analysis of biological signals. The input EEG signal can either be left in the time domain, allowing the network to learn the temporal characteristics of the signal, or be in the form of a spectrogram, allowing the learning of the frequency characteristics of the signal. CNNs are used not only for MI classification but also, for instance, for the prediction and monitoring of epilepsy and for the detection of visual-evoked responses [20, 20, 21, 22].

1.3.3 EEGNet

EEGNet is a particular and very compact CNN that incorporates spatial and temporal separation techniques. The network is structured to efficiently handle EEG data, which typically have high temporal dimensions and a relatively low number of spatial channels, limiting the learnable parameters of the network. The two main limitations in using deep learning for BCI, as mentioned earlier, are the large amount of data needed to train the network and the difficulty in the physiological interpretation of the features extracted by the network for classification. EEGNet overcomes both limitations, allowing training on a limited amount of data and producing largely interpretable features [23]. The network can be outlined as: input, 2D convolution, depthwise convolution, separable convolution, and classification (*Figure 1.10*). Going into more detail, three blocks can be identified within which different operations are performed:

- **Block 1**: two convolution operations are performed in sequence. First, a certain number of 2D convolutional filters are applied, followed by a Depthwise Convolution. After the first convolution, Batch Normalization is performed, which is repeated after the second convolution as well. Then, the following operations are performed: activation function, Average Pooling, and dropout technique is applied to prevent overfitting.
- **Block 2**: in this block, a Separable Convolution is used. As in the previous block, after the convolution, Batch Normalization is applied, followed by an activation layer, an Average Pooling layer, and dropout is applied.
- **Classification Block**: a Softmax function is used to predict one among the N output classes.

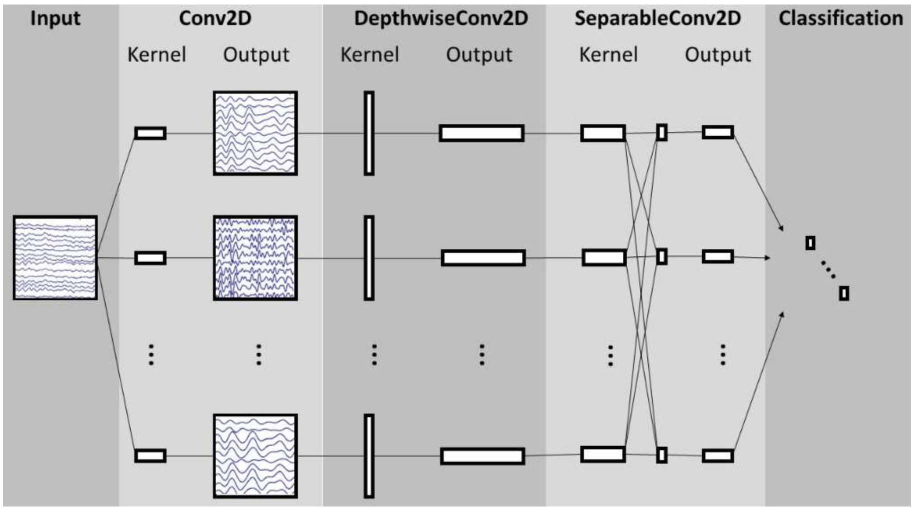


Figure 1.10: EEGNet architecture

If the EEG signal is given as input to the network in the time domain, with C channels and T time windows, the first operation involves a temporal convolution to learn frequency filters. Typically, a number of temporal filters F1 is chosen to be half of the signal's sampling frequency, allowing the capture of information from subbands of 2 Hz each. The second convolution (Depthwise) is used to learn a certain number D of spatial filters for each frequency subband extracted previously. Finally, the third convolution (Separable) is a combination of a Depthwise Convolution, which acts individually on each feature map produced by the previous convolution, and a number F2 of Pointwise Convolutions, which serve to mix the feature maps together in a optimal way. The main advantage of using a separable convolution is the reduction in the number of parameters to be fit.

1.4 The importance of the closed loop

The success of BCI relies on its architecture, specifically on the closed-loop control, which allows the user to see what is happening in real-time through feedback. This way, the user understands if the produced classification is correct or if there was an error and therefore needs correction. Depending on the purposes for which the BCI was designed, it is a device that must be used many times by a patient, both for assistance and recovery. It is thus necessary to become familiar with the technology. Often, the main problem is the difficulty of use, especially for certain categories such as the elderly or patients in particularly severe conditions. Although there are still few studies on the prolonged use of BCI by the same subject, it has recently been demonstrated that the user improves session after session, and this learning brings multiple benefits both from the subject's and the performance's point of view [24, 25, 26]. Therefore, summarizing the multiple advantages offered by this continuous interaction loop between user and machine:

- real-time feedback allows the brain to adapt and improve control. The user can immediately see the effect of their mental actions, facilitating learning and optimizing control strategies through the modulation of their brain rhythms.
- Effective feedback can reduce the cognitive effort needed to control the device. Indeed, when the user sees an improvement in performance, the interaction becomes less tiring and more intuitive.
- The closed-loop promotes neuroplasticity, that is the brain's ability to reorganize and create new neural connections. This is particularly useful in rehabilitation processes, where BCI can help patients recover lost functionalities.
- With this continuous feedback system, the device can better adapt to changes in the user's intention or mental state, making the interaction smoother and more natural.

1.4.1 Mutual learning

The concepts of closed-loop and feedback are closely linked to that of mutual learning, which refers to the reciprocal learning that occurs through the sharing of intelligence among the three entities that make up the system, that is at the machine, subject and the application level [26]. Regarding the user, they generally adapt to the decoder as they begin to use it, and this happens thanks to feedback, as previously mentioned. The decoder can self-adapt to the user's mental changes [27], or more commonly, it is recalibrated after a few sessions by selecting new features to use in the classification process, namely the features effectively modulated by the user. This brings up the issue of calibration: if it occurs too often, the subject cannot adapt to the decoder, but if too much time passes, the decoder is no longer focused on the discriminative features. Currently, there is no universally accepted method for determining how often recalibration should occur, so it is generally done when the system's accuracy starts to decline. Finally, to properly control a robotic device that needs to move, it is necessary to integrate information from the user with information extrapolated from the surrounding environment through sensors and cameras, resulting in a shared control framework [2, 28]. This way, the device becomes intelligent and capable of moving in a real environment where obstacles are normally present. The continuous control of robotic devices through BCI is one of the most interesting but also most complex challenges today.

1.5 Current limitations of BCI

Despite the multiple possibilities offered by the BCI system to disabled subjects, there are still many aspects that need improvement before it can be widely adopted in clinical practice. Some systems are already on the market, others are in advanced experimental stages, and still others are used only in laboratories for clinical research. The main limitations to overcome concern both the patient and the technology itself, including hardware and software aspects [29].

Firstly, learning to use BCI requires time and effort. The system is often not simple, especially for certain categories of subjects, and for this reason, more user-friendly technology should be developed. Using BCI requires a lot of concentration, making it incompatible with prolonged use. Additionally, there could be discomfort from the cup and electrodes used for EEG signal recording. During use, the user must remain still and generally seated. This can lead to fatigue and boredom, preventing the performance of common actions such as walking or using the smartphone. User's emotional and mental state can influence the system, creating significant variability both for the same subject and between different subjects, as can lifestyle, gender, and age. From a technical standpoint, the non-linearity and non-stationarity of the EEG make signal analysis difficult and lead to constant shifts over time. Often, the signal-to-noise ratio is low, making classification extremely difficult. To control an external device that moves in real-time, such as a prosthesis, much more accurate, fast algorithms that are less sensitive to external variations would need to be developed. The time required to collect data is long, providing very few data points to create a valid model. There is a lack of universally recognized metrics to evaluate performance and make the results of multiple studies comparable. Standardization in the methods used to quantify a particular BCI application is necessary. As mentioned earlier, the BCI system is often used only in the laboratory, but all the requirements the system should meet if used in a real environment must be considered. Finally, there are also ethical challenges that concern physical, physiological, and social factors.

1.6 Related works

To justify the choice of the model and methods applied in this work, the technique of dimensionality reduction used for the visualization of the latent space, and the selection of the metrics, several studies, along with their respective results and conclusions, are presented below. These studies comprehensively explain the current state of the art on the topics discussed.

In [30], three convolutional networks (ConvNets) with a different number of convolutional layers are proposed. The architectures range from two layers in the shallow ConvNet, to five in the deep ConvNet, up to 31 layers in the residual network (ResNet). Additionally, a hybrid ConvNet was created, combining elements of both the deep and shallow ConvNet. The performance of these networks was compared with the best-performing method for the BCI competition IV dataset 2a, namely Filter Bank Common Spatial Patterns (FBCSP). In all cases, the ConvNets achieved performance comparable to that of FBCSP, with the added advantage of end-to-end learning, meaning no need for preprocessing or manual feature extraction. When comparing the architectures, it was found that the shallow ConvNet performed just as well as the deep ConvNet, unlike the hybrid and residual architectures. Furthermore, to study the interpretability of the network, an advanced visualization technique for spatially mapping the learned features was used, which demonstrated that the ConvNets independently learned to utilize spectral power modulations in the alpha, beta, and high gamma frequencies.

In [23], the EEGNet architecture is proposed for the first time for the interpretation of EEG signals in the context of BCI. The study performs both within-subject and cross-subject classification on various BCI paradigms: P300-evoked potentials, error-related negativity responses (ERN), movement-related cortical potentials (MRCP), and sensory motor rhythms (SMR). Additionally, the performance of this architecture is compared with other convolutional networks already proposed in the literature, such as deep ConvNet and shallow ConvNet, as well as traditional models: One-Versus-Rest (OVR) and filter-bank common spatial patterns (FBCSP). In the case of SMR classification, for within-subject classification, the performance of shallow ConvNet, FBCSP, and EEGNet is very similar, while that of the deep ConvNet is significantly lower. In cross-subject classification, no significant performance differences were found among all CNN-based models, which still slightly outperformed FBCSP. It can be concluded that the deep ConvNet is more data-intensive compared to EEGNet, which, on the other hand, performs well on all tested datasets without the need for data augmentation, making the model simpler to use in practice.

In [31], a comparison was made between the performance of various deep learning models recently proposed for MI-EEG classification, including: EEGNet, shallow and deep ConvNet, Multi-branch 3D CNN (MB3D), and parallel self-attention network (ParaAtt). These models were tested on two large public datasets. The first dataset, MBT-42, was recorded during MI-BCI training with a total of 42 subjects, who had to perform a cursor movement task on the screen to the right and left. In the second dataset, MBT-62, 62 subjects performed the cursor movement task on the screen to the right, left, up, and down, and finally in all four directions. Exponential moving standardization was applied to the raw data during preprocessing to also investigate the impact of preprocessing on the results. In MBT-42, all models achieved high accuracy. Additionally, they performed relatively similarly, and no model significantly outperformed the others. In MBT-62, EEGNet performed significantly better than the other models. Regarding computational cost, for training on 30 epochs for each subject, all models took a similar amount of time, except for MB3D, which took much longer. However, all five models were able to decode a single sample in less than 40 ms, which means they all have the potential to be used in real-world applications. Finally, the model trained on preprocessed data achieved better performance in all cases compared to the model trained on raw data, indicating that a minimum amount of preprocessing, even when using neural networks, is necessary to achieve higher performance.

In [32], the EEGNet network was tested on data from 10 hemiparetic stroke patients while performing left and right MI tasks. Compared to the original EEGNet study [23], where the network achieved 67.25% accuracy on healthy subjects with 4 classes, here an accuracy of 70.25% was achieved in within-subject classification and 67% accuracy in cross-subject classification on unhealthy subjects with 2 classes. The results demonstrate that EEGNet achieves good accuracy in this type of analysis and has great potential for feature calibration-free BCI systems.

Regarding UMAP (Uniform Manifold Approximation and Projection), in addition to the standard technique proposed in [33], there are several variants that have been tested on data from BCI systems and compared. In [34], an approximate UMAP technique called aUMAP is proposed, aimed at generating fast projections for real-time introspections. Therefore, it is a technique specifically designed for use in online BCIs. From the results on three different datasets, it emerges that aUMAP preserves the standard UMAP clustering, suggesting that aUMAP's accuracy is suitable for online projection, although it is more prone to outliers. Regarding projection time, aUMAP consistently achieves the lowest projection times compared to standard UMAP and pUMAP (parametric UMAP), which is the slowest when run on a CPU. pUMAP is a parametric optimization of UMAP over neural network weights, learning a parametric relationship between data and embedding [35].

In [36], three individuals with disabilities were trained to use an MI-BCI by performing tasks involving both hands and both feet. The aim was to demonstrate the effectiveness of mutual learning, which is the reciprocal learning between the user and the machine. This is achieved through infrequent decoder calibration, necessary only when there is a persistent decline in performance, along with intensive training by the subject. In this context, three sessions per week were conducted over several months, and it was found that Subject 1 learned the most during the longitudinal training phase and achieved a significant improvement in BCI performance, both in terms of classifier accuracy and wheelchair control. Additionally, learning was also investigated at a neurophysiological level by visualizing the discriminability of features, which is an excellent indicator of how well the user can modulate sensorimotor rhythms.

In [26], clear evidence is provided showing how user learning and adaptation to the classifier are possible through the adoption of a specific experimental paradigm, which in the study involved calibrating the decoder only twice for each subject during the longitudinal training phase. The research was conducted on two tetraplegic subjects who participated in the Cybathlon competition. Evidence of the effectiveness of mutual learning was first shown in race performance, including the race competition time and the "pad crossing time", which assesses BCI command delivery accuracy and speed in a single metric. Both of these parameters showed excellent results. Additionally, it was observed that the decoder accuracy increased as the sessions progressed, stabilizing at a very high level. Finally, as a decisive proof of user learning, the spatial representation of feature discrimination over time was provided, which gradually becomes larger. This study is particularly interesting, especially for the conclusions that can be drawn from the presented results. First, learning cannot be evaluated solely based on the increase in classification accuracy or application performance, which do not necessarily imply modulation of brain signals. In fact, better performance can be due to recalibration of the decoder, re-parameterization of the BCI, or the adoption of a better mental strategy. Another aspect is that feature discriminability does not stabilize when classification accuracy saturates at high levels, but continues to increase. This improvement manifests, for example, in delivery speed, which had further margins of improvement. Finally, the research concludes that infrequent model recalibration leads to more unstable results in the short term, but is preferable because it produces better long-term results, as the subject is able to adapt to the BCI system.

In [24], a study on long-term learning was conducted with a Paralympic athlete who participated in the Cybathlon competition twice, in 2019 and 2020, winning the gold medal both times. It involved an MI-BCI with two tasks, both hands and both feet. The experimental paradigm was characterized by infrequent recalibration of the decoder, which favored mutual learning between the subject and the machine, proving to be the winning strategy. The data were collected during a longitudinal training phase of 8 months, followed by a long one-year break after the first competition. The results showed, first of all, a significant improvement in terms of classifier accuracy and game execution time. To provide evidence of user learning, the topographical representation of the discriminability of the two tasks over the months was shown, which demonstrates a gradual increase up until the first competition, and maintenance during the training for the second competition. Additionally, two metrics were presented, calculated both in the channel and in the Riemannian domain. The between-class distance is defined as the distance between the two classes, while the within-class distance is defined as the distance of subsequent runs from the first run. The results show that only in the Riemannian domain there is a stability in both metrics, especially visible after the break, making them more effective in tracking the acquisition and stabilization of BCI skills, compared to the same metrics calculated in the channel domain. In conclusion, it is stated that the analysis conducted could be used to monitor user learning during training and provide a marker guiding decoder re-calibration.

In [25], a study on an invasive MI-BCI is presented, conducted over 43 ECoG sessions spanning more than 200 days on a single tetraplegic patient, allowing for long-term analysis. Computational experiments were performed to investigate various factors that can influence performance, such as subject adaptation and the length of the training set. Several models were compared: multilayer perceptron (MLP), a combination of CNN and LSTM (CNN+LSTM+MT), and a multilinear model. The results showed that DL-based methods provide similar or higher performance in almost all cases compared to the multilinear model, while using the same amount of data. Through the computational experiments, it was observed that the patient adapted to the model and the data quality improved over the sessions, suggesting improvements in patient BCI skills, not specific to a single model. UMAP was used to visualize the raw data and eliminate artifacts, as well as to visualize the distribution of model embeddings, revealing interesting data manifold structures. Finally, a metric was calculated on the raw data, namely intrinsic dimensionality (ID), which was found to be positively correlated with cosine distance, used in the study to evaluate model performance. This led to the conclusion that, despite some concerns about the reliability of this index, data with higher ID allows for better predictive performance, as it reflects the complexity of motor imagery patterns. This is a positive aspect when using a network with high capacity, capable of learning these patterns, but it can be detrimental for other less complex models.

Here are two studies that utilized Riemannian geometry, which was not covered in this work but are very interesting from the perspective of long-term learning and the indices used to evaluate user learning. In [27], a research is presented on a single tetraplegic subject who participated in the 2019 Cybathlon competition. The dataset consists of the subject's training sessions, as this is a long-term study, comprising 20 recorded sessions over 3 months of training, all of which included closed-loop training runs with online BCI feedback. The classifier used changed over the months, transitioning from linear classifiers to an adaptive Riemannian classifier, which aims to reduce between-session variability. To visualize in 2D the different distribution of data during sessions close in time, a dimensionality reduction technique, namely t-SNE, was used. To evaluate the subject's learning and adaptation to the BCI system, two metrics were proposed: Class distinctiveness (ClassDist), which measures how distinct and stable the EEG patterns produced by the user are, independently of any classifier, and Test-Train Adaptation (TTA), which quantifies how much the user's online EEG test data distribution becomes similar to the EEG data used to train the classifier. The results showed a 30% increase in accuracy at the end of training compared to the first sessions. The metrics demonstrated a significant increase in TTA, while ClassDist (a metric generally studied to quantify user learning) remained more or less unchanged, concluding that although class separation remained the same, the user still adapted their EEG signal to the BCI classifier, opening the door to a new form of learning.

In [37], a new way to study user learning is proposed, not only by looking at classification accuracy, which can be influenced by various factors, but also by considering different metrics that are independent of any model and directly reflect the self-modulation of EEG patterns by the user. These metrics are calculated using Riemannian geometry. Class distinctiveness (classDis) measures how distinct the EEG patterns produced during two MI tasks are from each other. Rest distinctiveness (restDist) measures how distinct the EEG patterns produced during an MI task are from the rest state. Class stability (classStab) measures how stable an EEG pattern remains, without changing drastically between trials. The study was conducted on two different datasets, comprising 1 session and 6 sessions, respectively. In a first analysis, performed by averaging across all subjects, the results show that although accuracy remains more or less stable, the metrics undergo changes. In a second analysis, conducted on individual subjects, the metrics change in very different ways for each user. Therefore, the conclusion is that accuracy alone is not able to reflect all the different aspects of MI-BCI control skills, which are referred to in this research as ”subskills” and may be learned differently by different subjects.

1.7 Thesis' aims and structure

The general objective of this dissertation is to study the learning mechanisms in MI-BCI in all its forms, experimenting with various techniques and metrics, some of which have already been applied to other datasets but are not universally accepted as direct indicators of user learning.

To date, there is no standard method for evaluating this aspect, both because it is not always easy to acquire enough data to conduct a consistent study, and because there can be significant variability in results from subject to subject. Therefore, beyond understanding how the subject learns, one of the goals is to invalidate or, conversely, validate the tested methods, so that they can potentially be used in future research and even in online experiments to improve the BCI system. Focusing solely on the algorithm often overlooks the subject. The user of the BCI is an integral part of the closed-loop cycle, and deeper investigation is needed into the shift of neural patterns during training, which is now understood to play a fundamental role in the successful execution of motor imagery tasks and in optimizing the entire system. Therefore, more specifically, the aim is to study this rearrangement at the brain level, which is reflected in changes in the distribution of the motor imagery classes, both in the short term, where short term refers to what occurs in the early recorded sessions, and in the long term, considering the period from the first session to the last. This is useful both for understanding which training paradigm to adopt in order to achieve excellent results in a short time and as a warning to update the decoder only when necessary.

The structure of the thesis is divided into four chapters, with the first being the introduction. Chapter 2 presents the adopted methodology. It covers the dataset used and how the recorded data are preprocessed. Next, the selection of the hyperparameters and the training of the neural network are discussed. This is followed by an explanation of the first offline experiment, the Translation Experiment, and of the latent space analysis through data dimensionality reduction using UMAP. Lastly, a section is dedicated to the statistical analysis performed, which is essential to validate the obtained results. Chapter 3 presents the results and interpretations of the previously described experiments, with various graphs to facilitate understanding. The chapter concludes with a discussion of these results and the study's limitations. Finally, Chapter 4 is the conclusion, which includes a brief summary of the entire work carried out and potential improvements for future research on the same topic.

Chapter 2

Methodology

The dataset comes from a single male subject, a Paralympic swimmer, who participated in the Cybathlon competition for two consecutive years, in 2019 and 2020, winning the gold medal both times [24]. The dataset is particularly interesting because it encompasses the entire preparation of the participant, during which he underwent a longitudinal learning phase lasting 8 months, followed by a long break of 1 year between the two competitions. Unlike many other studies in the literature, where experiments were conducted over a few sessions [37, 38, 39], this research conducted offline experiments on a large number of sessions recorded over an extended period, allowing for the exploration of the effects of short-term and long-term user learning.

2.1 Cybathlon competition

The Cybathlon BCI race consists of controlling a virtual game called ”BrainDriver”, where the pilot must drive a vehicle on a track by performing MI tasks. The avatar passes through 4 sections evenly distributed along the track: ”right”, ”headlight”, ”left”, and ”noninput”. The first three sections allow the pilot to accelerate by turning or going straight, while in the last section, if no command is given, the avatar decelerates. The BCI implemented by the WHI Team, as described in the original research published after the competitions [24], is a non-invasive system based on MI, where the pilot had to perform two mental tasks: both hands versus both feet. In the initial recorded sessions, only two classes were used: both feet to send the ”right” command, and both hands to send the ”left” command, subsequently referred to as class 0 and class 1, respectively. In the sessions leading up to the race, a third class was added to send the ”headlight” command. In this case, the pilot had to perform both hands and both feet tasks in sequence, not in a predetermined order (*Figure 2.1*). However, this third class is not considered in this study, so all trials consisting of this MI task were discarded, resulting in a binary classification approach.

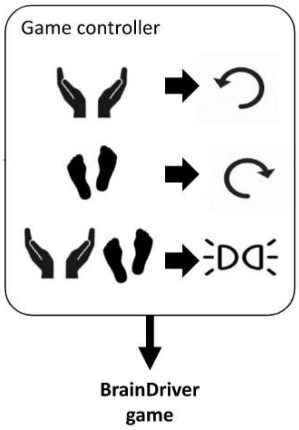


Figure 2.1: BrainDriver game controller

The initial recorded sessions are offline, as they were originally used for open-loop calibration of the model. These are followed by online sessions and control sessions, which were used by the pilot to effectively prepare for the race. In the cue-guided offline and online sessions, the pilot received continuous visual feedback during task execution, while during the game, the pilot had no feedback other than the movement of the avatar on the screen. The labels assigned to each trial correspond to the cue in the online and offline sessions, whereas for the control sessions, the labels were derived from the logs provided by the BrainDriver application, and thus may not all be accurate. The sessions (also referred to as runs with the same meaning) considered in this study range from 07/09/2019 to 11/11/2020, totaling 55 sessions recorded in 2019, including 7 offline, 4 online, and 44 control sessions, and 85 sessions recorded in 2020, including 16 offline, 3 online, and 66 control sessions.

2.2 Raw data and preprocessing

The EEG signal was recorded at a frequency of 512 Hz and hardware filtered between 0.1 and 100 Hz. The signal was acquired using a 16-channel g.USBamp amplifier, with 14 electrodes placed on the sensorimotor cortex. The channels are as follows: Fz, FC3, FC1, FC2, FC4, C3, C1, Cz, C2, C4, CP1, CPz, CP2, positioned according to the 10-20 EEG standard, with the ground location on the AFz electrode and referenced to the right earlobe. The other two electrodes, CP3 and CP4, were moved to the F1 and F3 locations (i.e., on each eye) near and during the competition, for electrooculography (EOG) detection, in order to implement the artifact control scheme and prevent any outgoing commands towards the game while such artifacts are detected. To avoid discrepancies in the number of channels between runs, CP3 and CP4 are not considered in this study.

For preprocessing, Matlab software was used. Since a deep learning algorithm was employed for classification, the preprocessing was minimal, as the deep neural network autonomously extracts the relevant features. Initially, all the files were merged into a single $Samples \times Channels$ matrix containing the signal recorded during the sessions, while another structure was created to store information about the sessions and events such as: file number, run number, run type (offline, online, or control), year (2019 or 2020), type of event within the trial (both hands or both feet), start and end of the event of interest. A Laplacian reference filter was applied to the signal using the adjacent electrodes. Defective runs were discarded, and the log band power was then computed over multiple frequency sub-bands, creating temporal windows. Since the frequency bands modulated during MI tasks are generally variable from subject to subject, narrow bands were selected to obtain more precise information. Sub-bands of 3 Hz in the range 4-50 Hz were chosen, with a shift of 1 Hz, resulting in 44 sub-bands. Data were epoched using a sliding window of length 1 second and shifted by 0.5 seconds. A 0.5 s shift is not suitable for online application, where the value should be much smaller (around 62.5 ms), but as mentioned earlier, the objective of this study is not to find an algorithm that works perfectly in real-time, but rather to study user learning over time through offline experiments. Since the trials included in the dataset are of variable length, all trials shorter than the window duration (1 s) were discarded. The log band power was computed following the approach described in a related study [40]. The portion of the trial corresponding to the actual execution of the mental task was extracted, with dimensions $S \times C$, where S represents the temporal samples and C the channels. The signal was filtered using a 5th order digital Butterworth bandpass filter in the previously described frequency sub-bands, resulting in a signal with dimensions $S \times C \times B$, where B represents the sub-bands. Then, the energy of the signal was computed as:

$$p = log(var(x))$$

where $var(x)$ represents the variance over a time window within the trial considered, and log represents the subsequent logarithmic transformation in base 10. In the end, a signal with dimensions $W \times C \times B$ was obtained, where W represents the time windows. After repeating the same procedure for each trial in the dataset, EEG energy normalization was performed as follows:

$$P^r_{i,j} = \frac{P^r_{i,j} - m_{i,j}}{\delta_{i,j}}$$

where P is the energy matrix of each sample, r is the sample index, $m_{i,j}$ is the mean calculated for the i-th channel and the j-th frequency band, and $\delta_{i,j}$ represents the standard deviation computed at the same position.

2.3 Network training

Python and the open-source library PyTorch for neural networks (*torch.nn*) were used for model training and all subsequent experiments. The log band power of the signal was provided as input to the EEGNet network, the general architecture of which has already been described. EEGNet was designed to be trained with raw data in the time domain, and as explained in the original paper, the first operation should be a temporal convolution to learn frequency filters [23]. However, in this study, it was chosen to move to the frequency domain, and indeed, the third dimension of the signal corresponds to the frequency bands. For this reason, all filters that were supposed to be applied along the temporal axis are actually applied to the filtered signal. The first 13 sessions were selected for the training set, while the following 3 sessions were used for the validation set. The parameters to be passed as input to the network are:

- *num_classes*: number of outputs of the network.
- *num_channels*: spatial dimension of the input.
- *sub-bands*: number of frequency bands in which the signal was filtered.
- *dropout_rate*: probability with which some activations are zeroed during training to prevent overfitting.
- *kernel_length*: size of the convolutional filter along the frequency band axis.
- *F1*: number of filters in the first convolution applied to the input data.
- *D*: multiplier that determines how many spatial filters are applied in the depthwise convolution.
- *F2*: Number of filters in the pointwise convolution ($F1 \times D$).

The model has been trained by minimizing the cross-entropy loss between the predicted and the true (expected) label. The cross-entropy loss for a binary classification problem is defined as:

$$Loss = -\left[y \cdot \log(p) + (1 - y) \cdot \log(1 - p)\right]$$

where y is the true label (0 or 1) and p is the probability expected by the model for class 1. The weights are then fixed and used for validation. If the current performance is better than the previous one (lower validation loss), these weights are saved as the best model. The choice of hyperparameters listed in *Table* 2.1 has allowed us to obtain the best model in terms of accuracy and F1-Score on the validation set, computed as:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} 1(\hat{y}_i = y_i)}{N}$$

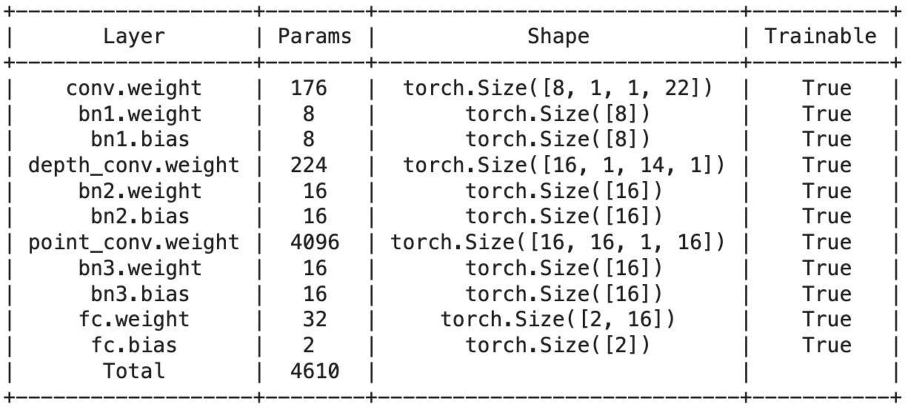
$$\text{F1-score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

where for accuracy: N denotes the total number of samples, $\hat{y}_i$ is the predicted class for sample i, y_i is the true class for sample i, and $1(\hat{y}_i = y_i)$ is a function that returns 1 if the prediction is correct, otherwise 0. For the F1-score: $Precision$ is the proportion of true positives among all instances classified as positive by the model, and $Recall$ is the proportion of true positives among all instances that are actually positive. For the latter, the parameter *average = 'macro'* was used, meaning that the metric was computed separately for each class and then the arithmetic mean was taken. This parameter is especially used when the two classes are imbalanced. The confusion matrix was also employed as an additional tool to see where the model is making the most errors and on what type of data (i.e., on which class). The best model has been tested for each session contained in the dataset.

Parameters	**Values**
num_classes	2
num_channels	14
sub-bands	44
dropout_rate	0.5
kernel_length	22
F1	8
D	2
F2	16
epochs	20
optimizer	Adam
learning_rate	0.001
loss_function	cross entropy loss
train_batch_size	4
eval_batch_size	128

Table 2.1: EEGNet hyperparameters

Below are reported the output size for each layer and the number of internal parameters of the model (i.e. weights and biases), further demonstrating that EEGNet is a lightweight and computationally non-intensive network compared to other types of architectures, making it optimal for potential use in online experiments as well (*Figure 2.2*).



```
+-------------------+--------+-----------------------------+-----------+
|       Layer       | Params |            Shape            | Trainable |
+-------------------+--------+-----------------------------+-----------+
|    conv.weight    |  176   |  torch.Size([8, 1, 1, 22])  |    True   |
|     bn1.weight    |   8    |       torch.Size([8])       |    True   |
|      bn1.bias     |   8    |       torch.Size([8])       |    True   |
| depth_conv.weight |  224   |  torch.Size([16, 1, 14, 1]) |    True   |
|     bn2.weight    |   16   |       torch.Size([16])      |    True   |
|      bn2.bias     |   16   |       torch.Size([16])      |    True   |
| point_conv.weight |  4096  | torch.Size([16, 16, 1, 16]) |    True   |
|     bn3.weight    |   16   |       torch.Size([16])      |    True   |
|      bn3.bias     |   16   |       torch.Size([16])      |    True   |
|     fc.weight     |   32   |     torch.Size([2, 16])     |    True   |
|      fc.bias      |   2    |       torch.Size([2])       |    True   |
|       Total       |  4610  |                             |           |
+-------------------+--------+-----------------------------+-----------+
```

Figure 2.2: Model trainable parameters and output size

2.4 Translation Experiment

User learning can be demonstrated in terms of improved data quality. The training should indeed lead to better performance on mental tasks day by day, resulting in greater class separation and thus more accurate classification. A study on user learning with an invasive BCI [25] has proposed what we will call a ”Translation Experiment”, which consists in translating the training set across the entire dataset, using 6 sessions for training and the following 6 sessions for testing, with a translation of 3 sessions (*Figure 2.3*). The results of the two studies will be compared subsequently. The total number of iterations, considering the 140 available sessions, is 43 (the last two sessions of 2020, i.e. 139 and 140, are not considered to always have an available test set of 6 sessions). Additionally, the difference in session length is ignored. Metrics on the test set were computed for each iteration to evaluate the classifier. To evaluate the model overall, but also the classes separately, both global accuracy and F1-score, as well as the F1-score for individual classes, were calculated. The hyperparameters used for each training and validation cycle are those listed in *Table 2.1* and remain unchanged throughout the experiment.

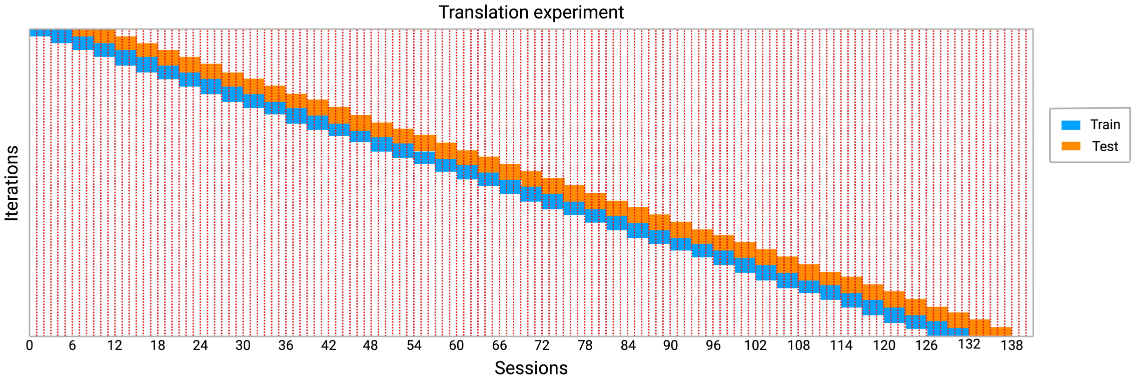


Figure 2.3: Example of Translation Experiment

Intrinsic dimensionality (ID), or effective dimensionality, is a parameter that reflects the complexity of the data being used, specifically indicating the minimum number of variables required to describe the data without losing important information [41]. A more formal definition states that a dataset $\Omega \subseteq R^N$ have an ID equal to M if its elements lie entirely within a M-dimensional manifold of R^N, where M < N [42]. Furthermore, it is important to underline that the definition of dimension in mathematics is not univocal. As a result of the subject's adaptation and improvement in using the BCI, due to visual feedback conditioning, the data distribution may change. The ID should reflect this change in distribution, and therefore it was calculated on the dataset to observe how this value evolves over the sessions. The global or local ID can be computed using various methods [41]; in any case, a slightly broader definition based on manifold dimension is generally used, where a representation in the ID needs to exist only locally. Such methods can handle different ID values in different parts of the same dataset, and this is referred to as local ID. In this case, the Expected Simplex Skewness (ESS) method was used, employing the scikit-dimension package. [25, 43].

2.5 UMAP

UMAP (Uniform Manifold Approximation and Projection) is an unsupervised, graph-based dimensionality reduction algorithm first described in 2018, and improved in 2020 [33]. This technique is currently used as an alternative to other, less recent methods such as t-SNE (t-Distributed Stochastic Neighbor Embedding) and PCA (Principal Component Analysis). For example, PCA is simple to implement and highly efficient computationally, but it assumes only linear relationships between variables. t-SNE is excellent at preserving local structures and clusters in the data; however, it is computationally intensive on large datasets and the results are sensitive to parameter settings. UMAP is optimized for large datasets, significantly reducing computational cost compared to t-SNE, though it is still relatively slow compared to PCA. It aims to preserve both the local and global structure of the original data, producing results that are fairly stable and reproducible. Nonetheless, careful selection of hyperparameters is crucial as they can greatly influence the results, potentially creating groupings in the data that do not actually exist in the original dataset, especially when the initial dimensionality is high [43]. This algorithm operates in two phases:

1. A particular weighted k-neighbour graph is constructed (fuzzy simplicial complex).
2. Throw stochastic gradient descent, a low dimensional layout of this graph is computed.

2.5.1 Model embeddings analysis

Everything that studies the decomposition of deep neural networks to examine their internal functioning is commonly referred to as interpretability [44, 45, 46]. Many studies on user learning in the context of BCI use reverse engineering methods to provide a neurophysiological interpretation of the output [30, 23, 47]. These usually include techniques that relate the input or hidden layers to the output or techniques for visualizing the latent space. One of the main focuses of this study is precisely to understand how the network makes decisions and which features most influence the network's output.

In the context of BCI, the patterns that emerge when summarizing datasets using visualization methods such as UMAP may reveal underlying structures in the data [48]. For a given input, we consider the network activations immediately after the flattening stage, i.e. after the convolutions and before the last fully connected layer. We refer to this latent representation as the model embedding of the input. To obtain the clearest possible results, the best model was first selected and saved, as described earlier. Using this model, all embeddings were computed (i.e., one vector for each trial in input). These high-dimensional embeddings (16×1) were then transformed with UMAP into 2D to allow visualization in the Cartesian space. The main parameters input to UMAP are:

- *n_neighbors*: the number of neighbors that UMAP considers when constructing the high-dimensional graphical representation of the data. A low value preserves the local structure of the data better, while a higher value captures the global structure more effectively.

- *n_components*: the number of dimensions in the reduced space. Typically set to 2 or 3 for visualization.
- *metric*: the distance measure used to determine the proximity between data points. It influences how distances between data points are computed and, therefore, how the initial graph is constructed.
- *min_dist*: the minimum distance between points in the new low-dimensional space. A low value allows points to be closer together in the new space, better preserving the local structure, while a higher value distributes the points more evenly.

The values were left at their defaults, as they provide a good visualization. (*Table* 2.2).

Parameters	**Values**
n_neighbors	15
n_components	2
metric	euclidean
min_dist	0.1

Table 2.2: UMAP hyperparameters

After reducing the model embeddings to 2-dimensional vectors, several significant scatterplots were created. First, the embeddings were plotted separately for each session and for the two classes. These subplots serve the dual purpose of a first visual inspection of how the distribution of the classes changes and of identifying sessions with few trials, which could make experiments results inconsistent. So, sessions with few trials of one class, another, or both, will be discarded in subsequent experiments. Then, the centroids of the embeddings for each session and class were computed and plotted. Finally, some plots were created showing the overall distribution of all the embeddings.

As a final step, to investigate changes in brain activity during the training period, in line with what was found in [24], two geometrical assessment metrics are computed:

1. **Within-class Distance (WcDist)**: it describes the variation in brain activity for each run relative to the first one, associated with the MI tasks (i.e., both hands and both feet). It was calculated separately for each class c as the distance of the mean of the embeddings distribution in the r-th day of training from the ”initial training period”, computed as the

average of the training runs (0-12):

$$WcDist = \frac{\delta(\mu_1^c, \mu_r^c)}{(\sigma_1^c + \sigma_r^c)}$$

2. **Between-class Distance (BcDist)**: it describes the discriminability of brain activity between the two MI tasks. It was calculated as the distance between the means of the embedding distributions for the two classes:

$$BcDist = \frac{\delta(\mu_{bh}, \mu_{bf})}{(\sigma_{bh} + \sigma_{bf})}$$

For calculating the distance between the arithmetic means of the two distributions, the Euclidean distance was used. These metrics are considered useful not so much for evaluating the model, but rather for measuring how the classes shift during training period, thereby quantifying the user learning.

2.6 Statistical analysis

For each metric reported in the study, the Pearson correlation coefficient (r) and the associated statistical significance (p_{value}) were calculated using functions and modules from Python's SciPy library for statistical analysis (*scipy.stats*). For accuracy and F1-score in the Translation Experiment, the linear relationship between these metrics and the iterations was computed, and the same was done for the ID. For WcDist and BcDist, the linear relationship between these metrics and the chronological sessions was computed to investigate the trend over time. Additionally, correlations between some of these metrics were estimated, such as between F1-score and ID in the Translation Experiment, between the accuracy of the best model and WcDist for both hands and both feet, and between accuracy and BcDist. For variables showing some linear relationship, with a statistically significant difference at the 5% level ($p_{value} < 0.05$), an Ordinary Least Squares (OLS) regression was performed to find the coefficients of the linear regression and report the estimated model. This was done using modules from Python's statsmodels library, which provides a range of tools for creating statistical models in a simplified manner.

For WcDist and BcDist, was also investigated the evolution of these metrics from the beginning to the end of 2019 and from the beginning to the end of 2020, considering four groups: the first 15 runs of 2019 (excluding runs 0 to 12 used for training the network), the last 15 runs of 2019, the first 15 runs of 2020, and the last 15 runs of 2020. An ANOVA (Analysis of Variance) was first performed between these groups to determine if there are significant differences between the means of at least one pair of groups. However, ANOVA alone does not indicate which specific groups differ from each other. Where ANOVA was statistically significant, a Tukey-Kramer post-hoc test was conducted. This test is designed to handle multiple comparisons between all possible pairs of groups while controlling the Type I error rate (false positives) for the entire experiment. The assumption for this test is that the data are normally distributed and that variances are equal (homoscedasticity) between groups. The result of this test provides the difference between the means of each pair of groups and a p_{value} associated with these differences, indicating whether they are statistically significant at the 5% level ($p_{value} < 0.05$). This test was chosen primarily because it reduces the risk of obtaining false positives when making multiple comparisons and because it can be used with any number of groups.

Chapter 3

Evidence of User learning

In this chapter, the results of the computational experiments conducted are presented, aiming to identify changes in the data distribution over the course of the sessions. These changes are intended to visually and quantitatively confirm that the neural patterns associated with each mental task have somehow altered or shifted. The chapter begins with the results of the Translation Experiment and their relationship with the Intrinsic Dimensionality (ID) metric. It then presents the performance of the best model, chosen accordingly to accuracy metric, on which the embedding analysis was conducted. Following this, the 2D embeddings plots are shown after dimensionality reduction using UMAP. Finally, the WcDist and BcDist metrics are reported. The chapter concludes with a general discussion on the results obtained and the limitations of this study.

3.1 Translation Experiment results

In this experiment, the F1-Score values are first reported separately for the classes Both Feet (BF) and Both Hands (BH) across all 43 iterations, where iteration means training on 6 sessions and validating on the following 6 sessions, shifting the training set across the entire available dataset. The date when the sessions were recorded is not taken into account, and thus, the long pause of the pilot between 2019 and 2020 is not shown in the graphs, as the focus is on analyzing the overall trend from the beginning to the end of the training. For the F1-Score related to the class BF (*Figure* 3.1), a qualitative upward trend can already be observed, suggesting an improvement in the BF task over the course of the user's training. For the F1-Score related to the class BH (*Figure* 3.2), the trend is less clear, showing that it starts from higher values compared to the other class (excluding iteration 11, which is an outlier due to the low number of trials) and decreases in the later iterations. Regarding accuracy and F1-Score computed across both classes, a generally stable trend can be observed, with occasional iterations showing much lower or higher values, and a decline is also seen in the final iterations (*Figure* 3.3, *Figure* 3.4).

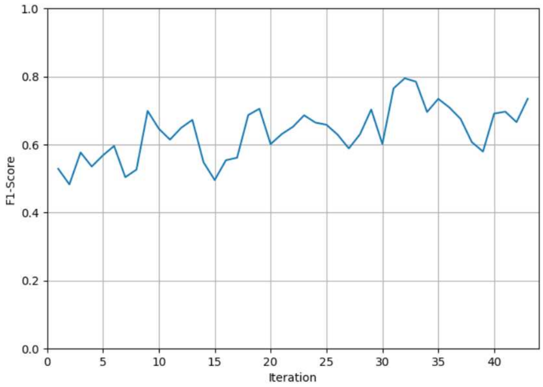


Figure 3.1: F1-Score Both Feet per iteration

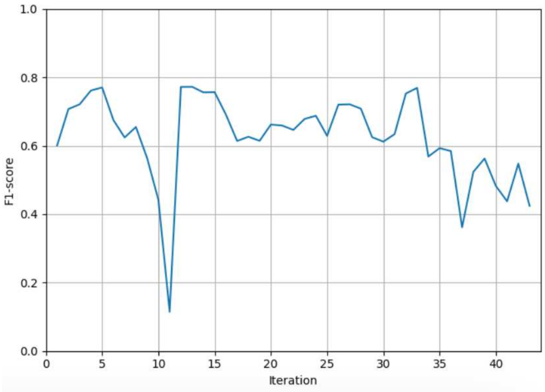


Figure 3.2: F1-Score Both Hands per iteration

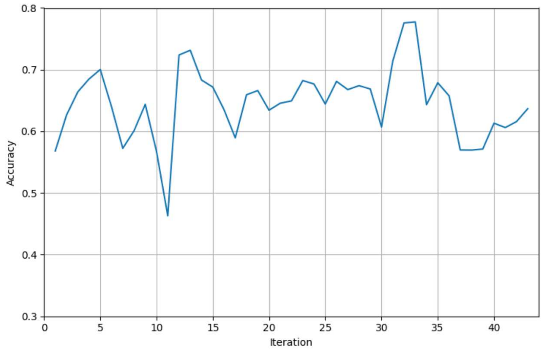


Figure 3.3: Global accuracy per iteration

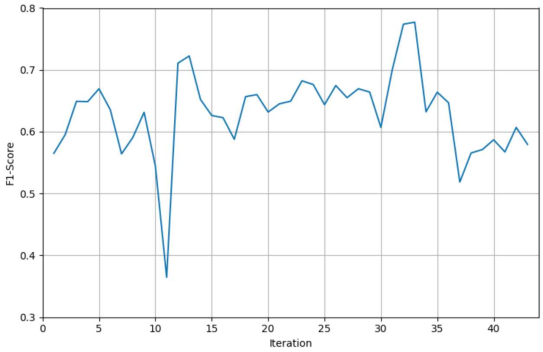


Figure 3.4: Global F1-Score per iteration

Looking at these results in the form of scatterplots allows for a more quantitative assessment. It can be confirmed that there is a linearly increasing trend for the F1-Score related to BF ($r = 0.63$), with high statistical significance ($p_{value} < 0.001$) (*Figure* 3.5). On the other hand, for the F1-Score related to BH, there is a less pronounced linear decreasing trend ($r = -0.31$), with lower statistical significance ($p_{value} < 0.05$) (*Figure* 3.6). Regarding the overall trend, there is no linear relationship between accuracy and iterations ($r = 0.05$), nor between F1-Score and iterations ($r = 0.06$), suggesting that, the performance do not increase or decrease, but the individual classes undergo variations.

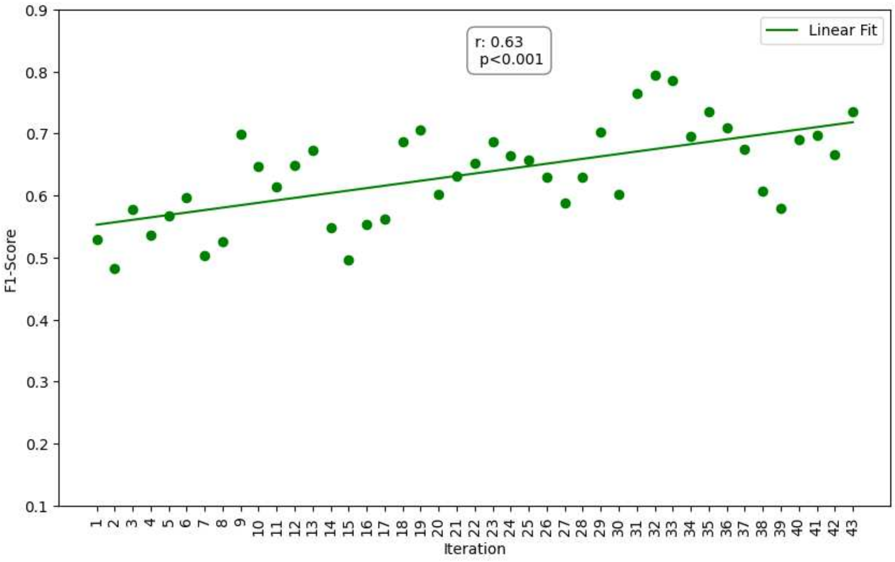


Figure 3.5: Scatterplot F1-Score Both Feet per iteration

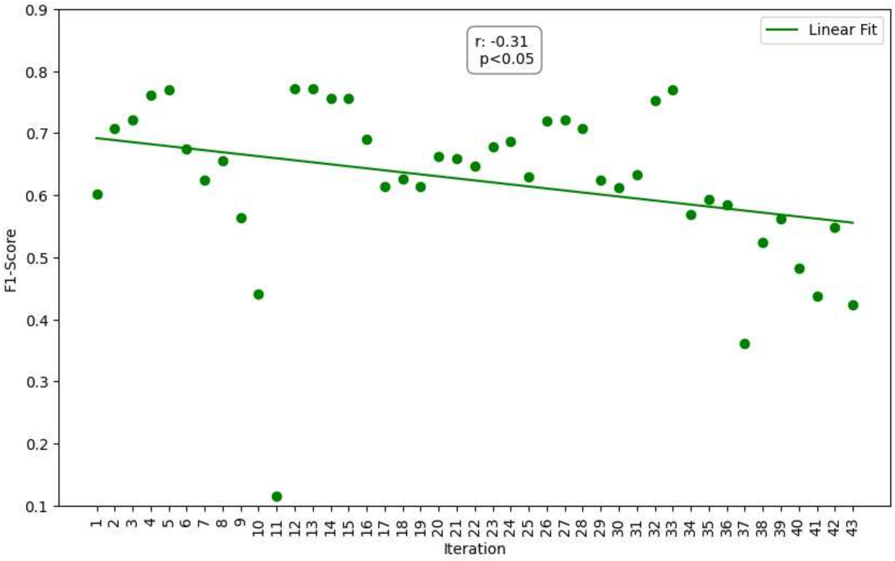


Figure 3.6: Scatterplot F1-Score Both Hands per iteration

These results, which show an improvement in data quality for the BF class during training, correlate well with the findings in the research [24], where it was stated that the subject initially exhibited strong activation of EEG features associated with the BH task, and that with longitudinal training, features associated with the BF task also began to emerge. The reason could be that the pilot already felt comfortable performing the BH mental task from the first sessions, focusing on improving the BF task, where he were initially weaker. Indeed, looking at the linear regression model in *Figure 3.5* and in *Figure 3.6*, the performance metric for BF starts at 0.55

and reaches 0.72, showing a 17% improvement. For BH, the predicted value starts at 0.69 and drops to 0.56, indicating a 13% decline. Therefore, the improvement in BF and the decline in BH (especially visible in the last 7 iterations, which all fall below the regression line), result in a stability of overall performance that accounts for both tasks.

In the research [25], from which the Translation Experiment is derived, the results report an improvement in only one mental task, in that case, the left hand rather than the right hand, similar to what was found in this study. The hypothesis formulated suggests that the MI patterns may be easier to adapt in one case rather than the other thanks to the remaining residual control resulting in better cortex preservation. A similar explanation can be hypothesized regarding the improvement of BF rather than BH, also in relation to the neural plasticity which allowed for greater modulation of brain activity for one task over the other. It has been demonstrated that intensive use of BCI stimulates neural plasticity [49]; however, further studies and experiments would be necessary to confirm these hypotheses at the neurophysiological level.

3.1.1 Intrinsic Dimensionality estimation

The ID value was computed for each session, and then, to compare it with the results of the Translation Experiment, an average was taken over a window of 6 sessions, sliding it across all sessions in the dataset with a shift of 3 sessions. The ID is a measure independent of the model being used; it depends solely on the data. It expresses the complexity of the dataset, so theoretically, the higher the ID, the more challenging the classification should be, and the worse the model should perform. This has been widely demonstrated in computer vision research (”curse of dimensionality” problem) [50, 51, 52, 53]. Conversely, in the study presenting the Translation Experiment, it was found that the local ID and the metric used to evaluate the classifier were positively correlated: as the ID increased, the model’s performance improved. Instead, in this study, it was found that the ID does not follow a linear trend, and no correlation was found between the ID and the metrics used to evaluate the model at each iteration. From the graph, it appears that there is an initial increase followed by a sort of plateau, which then decreases in the final iterations. (*Figure* 3.7). This difference in results could be related to the different experimental conditions, including the different data acquisition methods. In one case, ECoG was used, with very high ID values found (ranging from 250 to 330), while in this study, EEG was used, with lower ID values found (ranging from 59 to 112, excluding the outlier in iteration 1). In any case, the ID values are still very high also in this research, indicating a high complexity of the dataset. Further considerations on the ID will be made in the following sections.

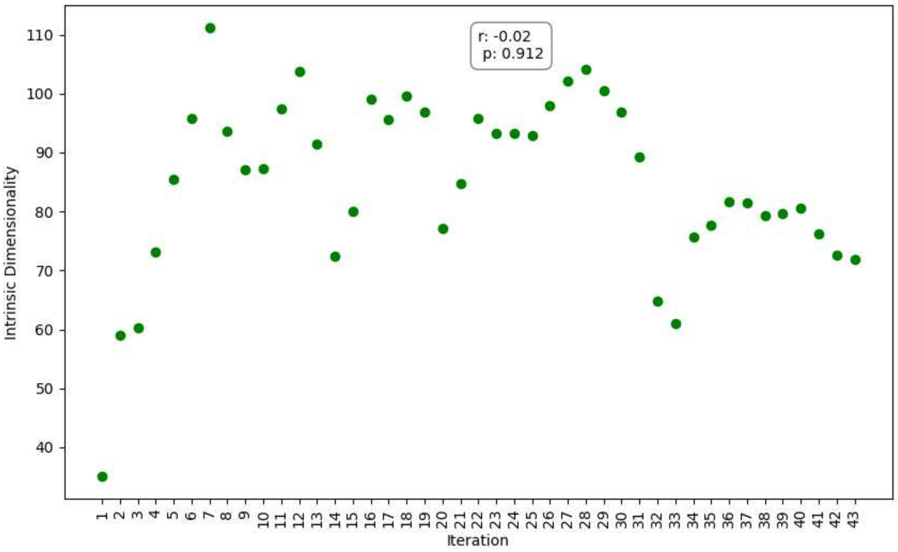


Figure 3.7: Scatterplot ID per iteration

3.2 Best model performance

After training multiple models with different combinations of hyperparameters, they were evaluated based on their accuracy and F1-Score on the validation set. Another criterion for model selection was the stability of accuracy across runs from 2019 to 2020. The goal was not merely to find the model with the best performance but to identify one that could predict reasonably well over the long term without a significant drop in performance. In the original study, the decoder was calibrated 5 times in 2019 and once in 2020, allowing the performance to remain high by selecting spatio-spectral relevant features. In this case, however, only the early runs of 2019 were used for training, which posed the risk of very low performance, especially in the later runs. Therefore, the model chosen for the subsequent analysis was selected not only for achieving the best validation accuracy but also based on a visual evaluation of the accuracy and F1-Score plots.

Here are the confusion matrices for the training set (runs 0-12) and for the validation set (runs 13-15). Additionally, the metrics have been computed on a small test set (runs 16-18) to evaluate the model on previously unseen data. In *Table* 3.1 the values for the F1-Score, accuracy, and loss function are reported for the training, validation, and test phases, respectively.

	F1-Score	**Accuracy**	**Loss**
Train	79.15%	79.45%	0.45
Validation	67.22%	67.63%	0.61
Test	64.67%	65.19%	0.61

Table 3.1: Best model metric values for training, validation and test

In *Figure* 3.8 the confusion matrix of the model for the training set is shown, where label 0 represents the BF class, and label 1 represents the BH class. Calculating the percentage error made by the model on each class on the training set, relative to the total number of samples for each class, reveals that the classification error for BF is approximately 24%, while the error for BH is 18%. However, it is important to note that the number of examples for BH in the training set (1166) is higher than for BF (922), meaning the classes are not perfectly balanced, so the network may have learned to classify one class slightly better than the other. In *Figure* 3.9 the confusion matrix for the validation set is shown. In this case, the percentage error for the BF class is 33%, while for the BH class, it is 32%. Additionally, here too, the number of samples for BF (393) is slightly lower than for BH (537).

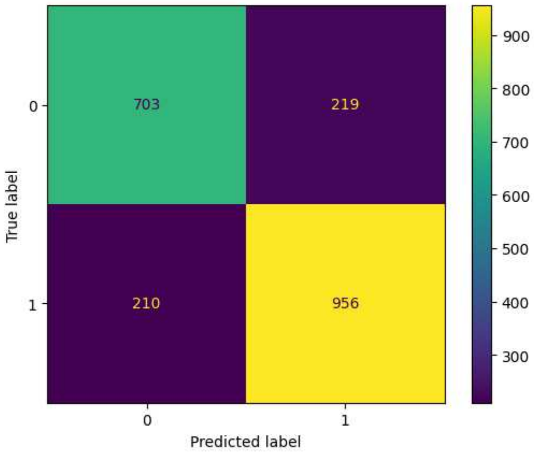


Figure 3.8: Confusion Matrix on the training set

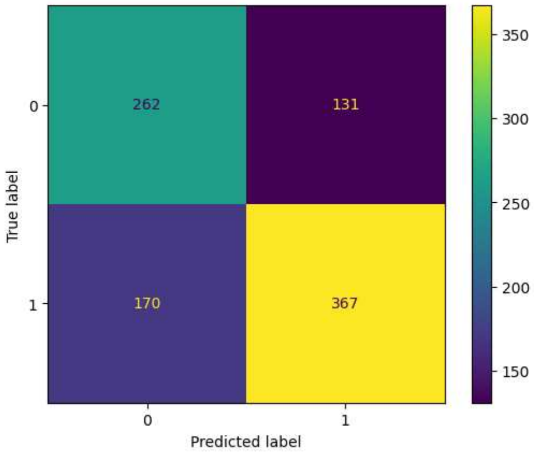


Figure 3.9: Confusion Matrix on the validation set

Finally, in *Figure* 3.10, the confusion matrix for the test set is shown. Here, the error for the BF class is 40%, while for the BH class, it is 30%. As with the training and validation sets, the number of samples for the BH class (638) is higher than for the BF class (511) in the test set as well.

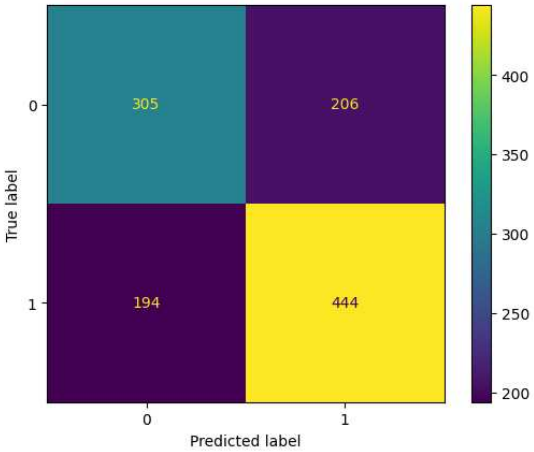


Figure 3.10: Confusion Matrix on the test set

What can be concluded from these results is that in all cases, the error percentage for BF is higher than for BH. This may be due to both the smaller number of examples for this class and, more likely, the fact that the network has more difficulty classifying this class compared to the other. Referring back to the results of the Translation Experiment, it was observed that in the early iterations, which correspond to the runs used here for training the model, the performance for BF was very low, while for BH, it started from higher values. This indicates poor data quality in the former case and good data quality in the latter. This may explain the higher error for BF in the training set, which then ”propagates” to the validation and test sets as well.

In *Figures* *3.11* and 3.12, the dates of all sessions and the type of run (i.e., offline, online, and control) are shown to provide as broad an overview as possible. It can be seen that the accuracy on the training runs (0-12) is high, with an average value of 0.82 ± 0.07. On the 2019 data, accuracy decreases but remains stable, almost always above 0.5, with an average value of 0.61 ± 0.07. For the initial 2020 data, the model's accuracy remains fairly stable but then decreases on the later 2020 data, with increasingly fluctuating values. The average accuracy on the 2020 data is 0.58 ± 0.08 (with a higher standard deviation compared to 2019). A similar pattern can be observed in the F1-Score graph. The average F1-Score for the training data is 0.80 ± 0.09, for the 2019 data it is 0.60 ± 0.06, and for the 2020 data it is 0.56 ± 0.09. This metric, which provides information on the model's quality, also confirms worse performance on the 2020 data, with greater variability relative to the mean value.

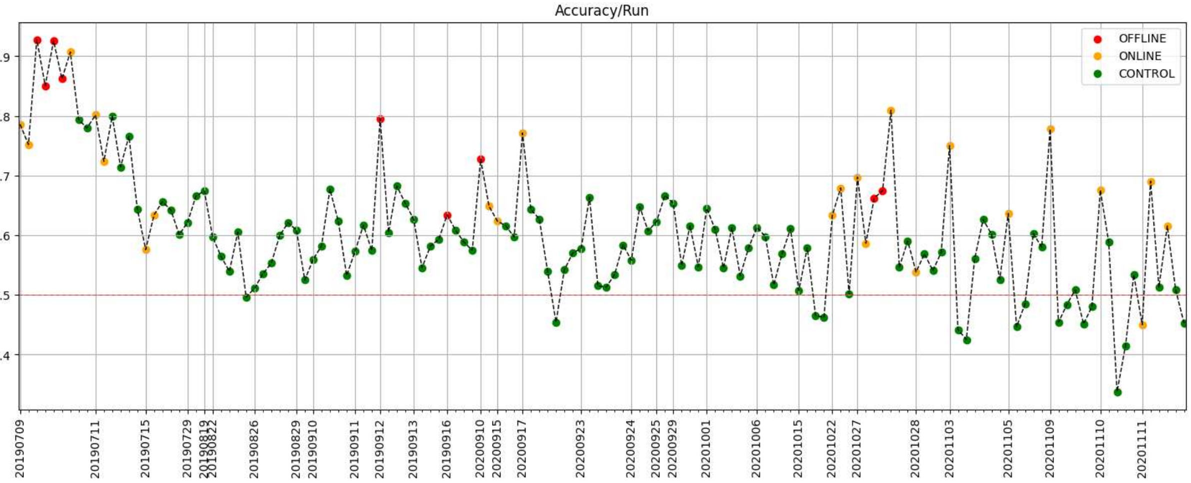


Figure 3.11: Accuracy trend over runs 2019-2020

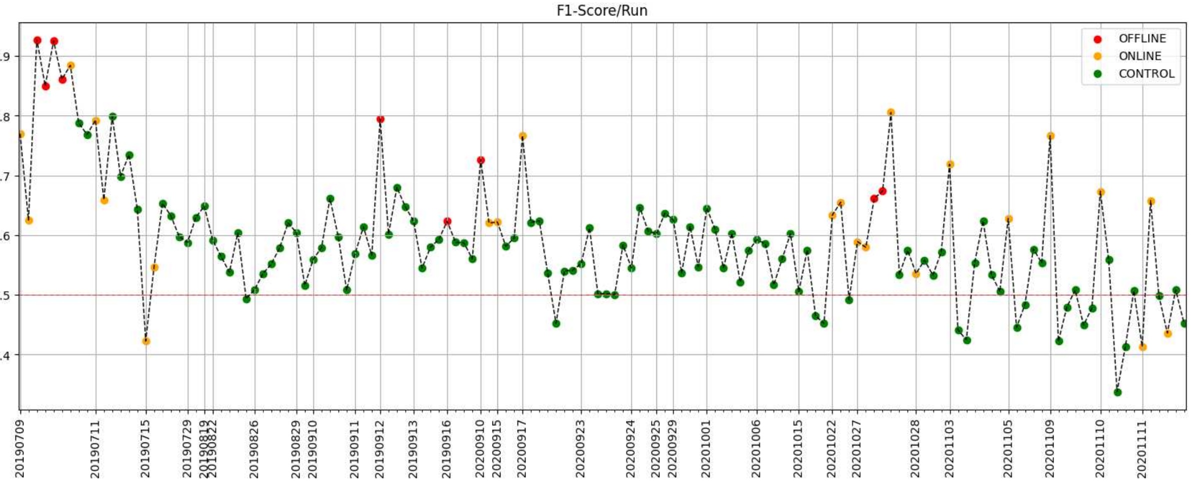


Figure 3.12: F1-Score trend over runs 2019-2020

3.3 Data dimensionality reduction

The embedding vectors were extracted by removing the classification head and taking the model's output. The embeddings, as they are, have a dimensionality of 16×1, which makes them difficult to analyze. Therefore, a single UMAP transformation was applied to all the embeddings, reducing their dimensionality to 2×1. This data transformation allowed for the creation of various plots for visualization and analysis in a reduced space. From here on, they will still be referred to as embeddings, but they refer to the UMAP projection of the data into two dimensions.

3.3.1 Embeddings distribution

The distributions for each run (0-139) were plotted separately for BH and BF (*Figures* 3.13, 3.14). The prevalent shape in most runs, for both tasks, is a sort of asymmetric parabola with its concavity facing upwards, although some runs deviate significantly, taking on different shapes. All subplots share the same scale, making it noticeable that the distribution of the BF class is denser in the lower-right part of the space, forming a kind of peak (more evident in runs 7, 8, 11, 14, etc.), while the BH class embeddings are more concentrated in the upper-left part (runs 8, 12, 13, 18, etc.), forming a peak on the opposite side. Apart from visual inspection, which allows us to observe how the class distributions shift across sessions, these plots clearly show that in some runs, there are too few trials of one class, the other, or both. Therefore, these plots were primarily used to decide which runs to eliminate in the subsequent analysis to ensure consistent results. Consequently, sessions 0 to 12 were removed, as they were used for training the model, along with runs 15, 16, 43, 60, 98, 100, 104, 107, 111, 118, 123, 129, 134, 135, and 137.

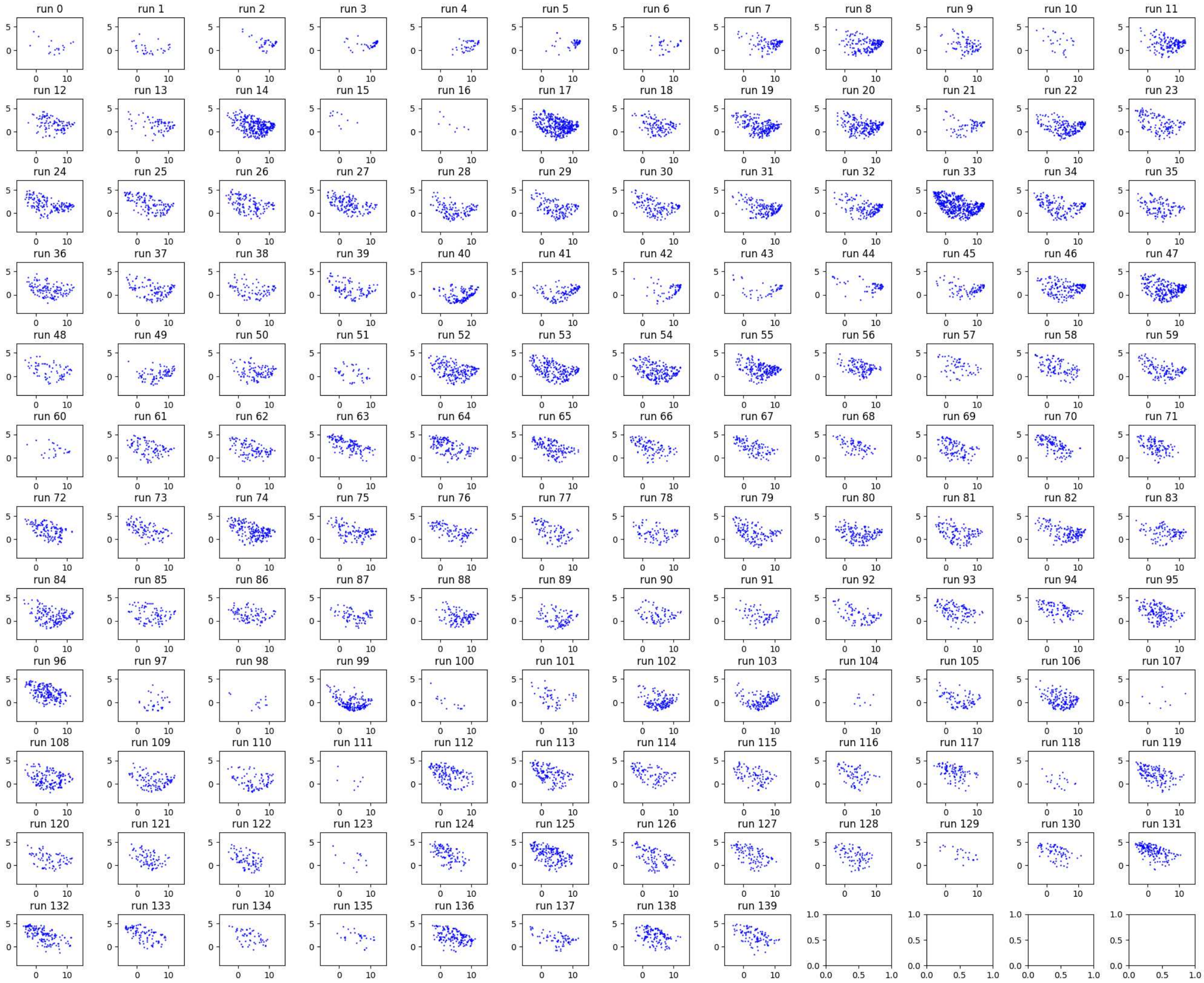


Figure 3.13: UMAP projection of embeddings distribution for Both Feet per run

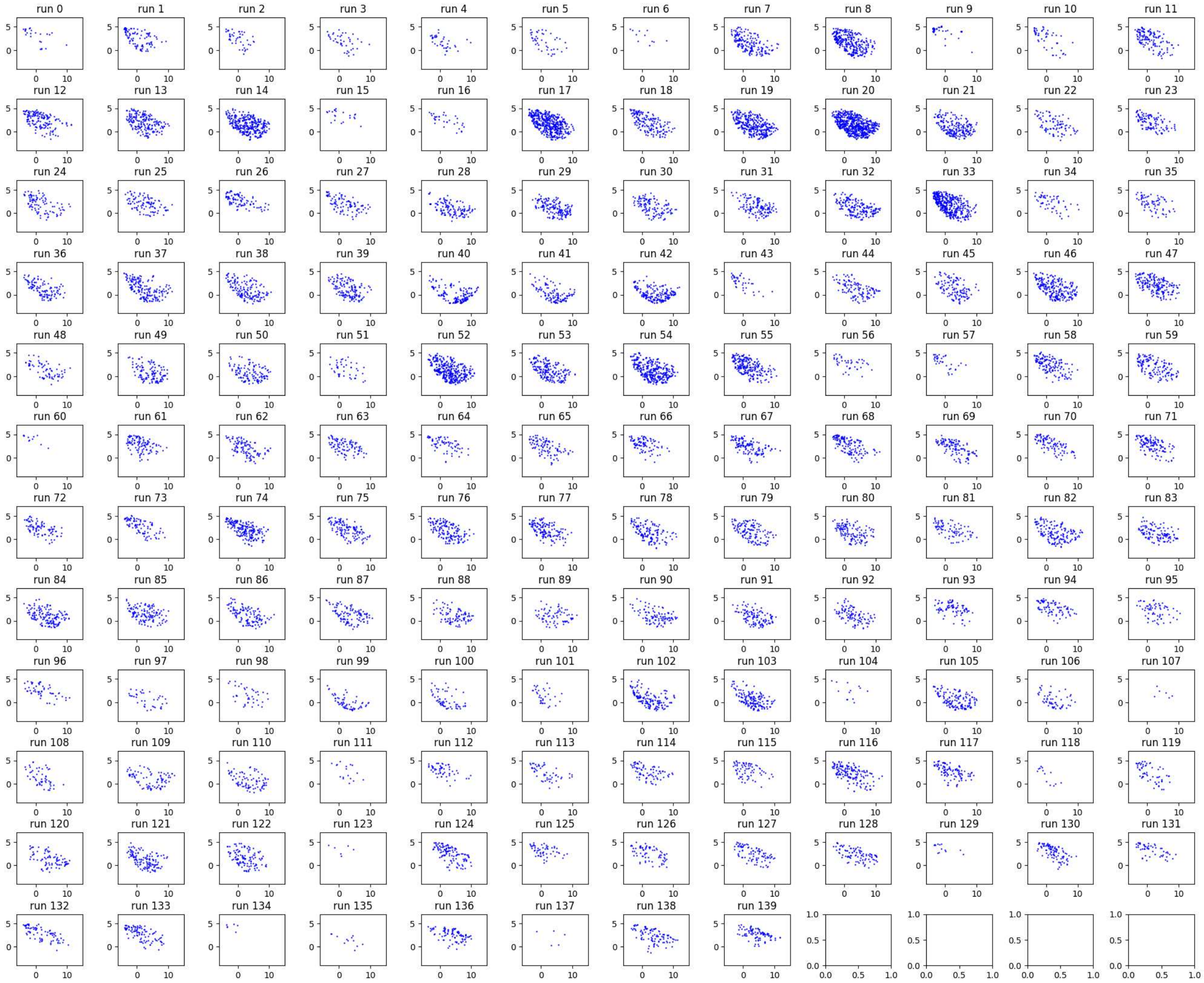


Figure 3.14: UMAP projection of embeddings distribution for Both Hands per run

3.3.2 Embeddings centroids visualization

The centroids of the distributions for each run were computed, maintaining the labels, to provide a clearer view of the shapes of the two classes. *Figure* 3.15 and *Figure* 3.16 display all the centroids for the 2019 and 2020 runs, respectively, to check for any shifts. Firstly, it can be observed that in 2019 the classes are much more separated compared to 2020. This statement aligns with the earlier observations regarding the classifier's performance, which deteriorates on the 2020 data. In fact, better classification is achieved with a greater distance between classes and a smaller distance within the class. Additionally, the BH class maintains a more or less consistent distribution between 2019 and 2020, while the BF class changes, showing smaller values along the x-axis and larger values along the y-axis in 2020. This can be noted by referencing the × symbol in the two graphs, which represents the average of the centroids calculated from the training runs (0-12) and is used to indicate the ”initial session”, that is, the starting point of the user’s training for BH and BF. In this case, it can be seen how the centroids for BF in 2020 not only shift back from the corresponding training centroid but also change shape, compared to 2019.

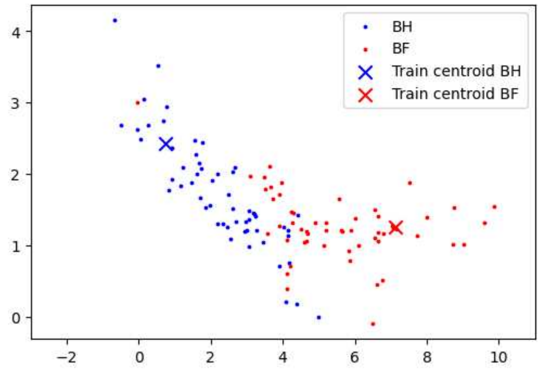


Figure 3.15: UMAP projection of embeddings centroids by labels on 2019 data

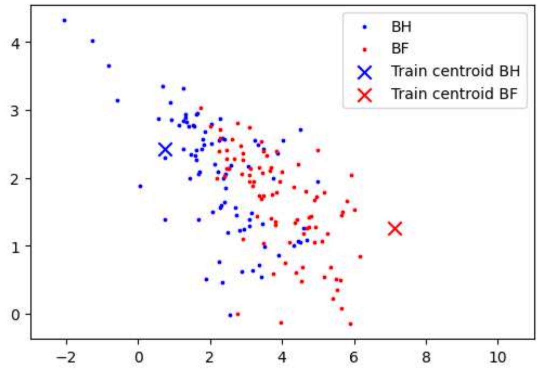


Figure 3.16: UMAP projection of embeddings centroids by labels on 2020 data

This result partially confirms what was observed in the Translation Experiment, namely that the BF class experienced the most significant variation, which is positive as there was an increase in the F1-Score. However, since the distribution of the BH class remained relatively unchanged between 2019 and 2020 and did not shift along with BF, there was no overall improvement in the classifier’s performance. In fact, there was a deterioration, visually confirmed by the mixing of the centroids of the two classes especially in the central area.

3.3.3 UMAP projection of all embeddings

Following are a series of plots displaying all the embeddings from all runs, to provide a broader overview that cannot be seen with the centroids alone, as they represent only a few values.

In the two figures below, the embeddings with reduced dimensions are shown without distinguishing between the sessions. In the plot in *Figure 3.17*, the true labels (i.e., correct labels) for the two classes are used, with black associated with BF and yellow with BH. The distribution of the two classes takes on a full asymmetric parabolic shape with two ends. On the right end, there is a higher concentration of samples belonging to BF, while on the left, there is a higher density of samples belonging to BH, without a clear boundary. In the plot in *Figure 3.18*, the labels predicted by the classifier are used instead. Here, a clear decision boundary is visible, separating the two classes. By comparing the two plots, it is possible to observe the model’s errors, that is instances where BH is predicted as BF and vice versa.

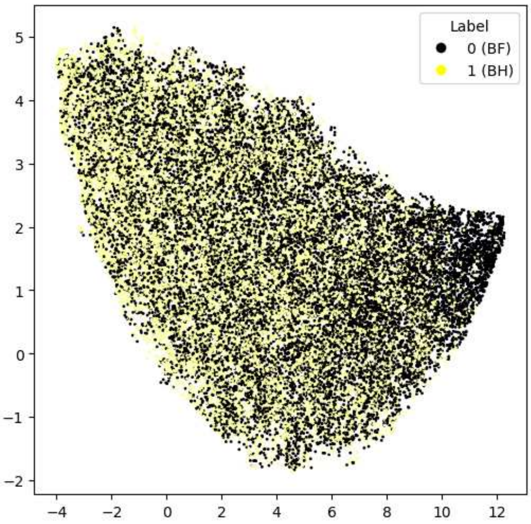


Figure 3.17: UMAP projection of embeddings by true labels

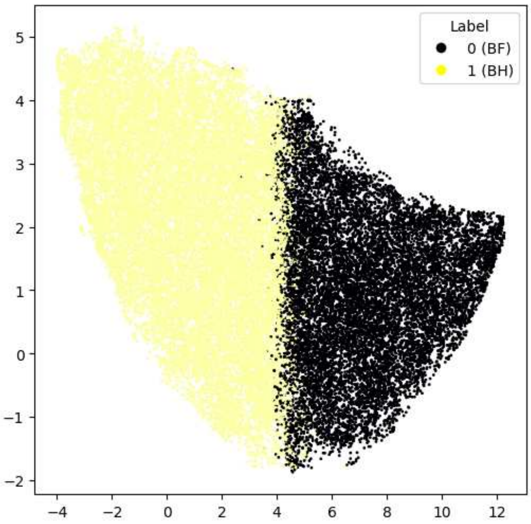


Figure 3.18: UMAP projection of embeddings by predictions

In *Figure 3.19*, each color corresponds to a time period, and therefore a group of runs. Brown is associated with early 2019, yellow with late 2019, green with early 2020, and blue with late 2020. In this plot, the labels are not preserved, so it is not possible to know which samples belong to the BH class or the BF class, thus only an observation on the overall distribution can be made. Paying attention to the color gradient, it can be seen that yellow is more densely distributed in the lower part (still in an upward-facing C-shape), while as you move upward and inward, the colors shift toward green and blue, ending with this last color, which is more densely distributed on the top and on the left tip. The brown color, associated with the earliest runs, seems to be the most evenly distributed throughout the figure.

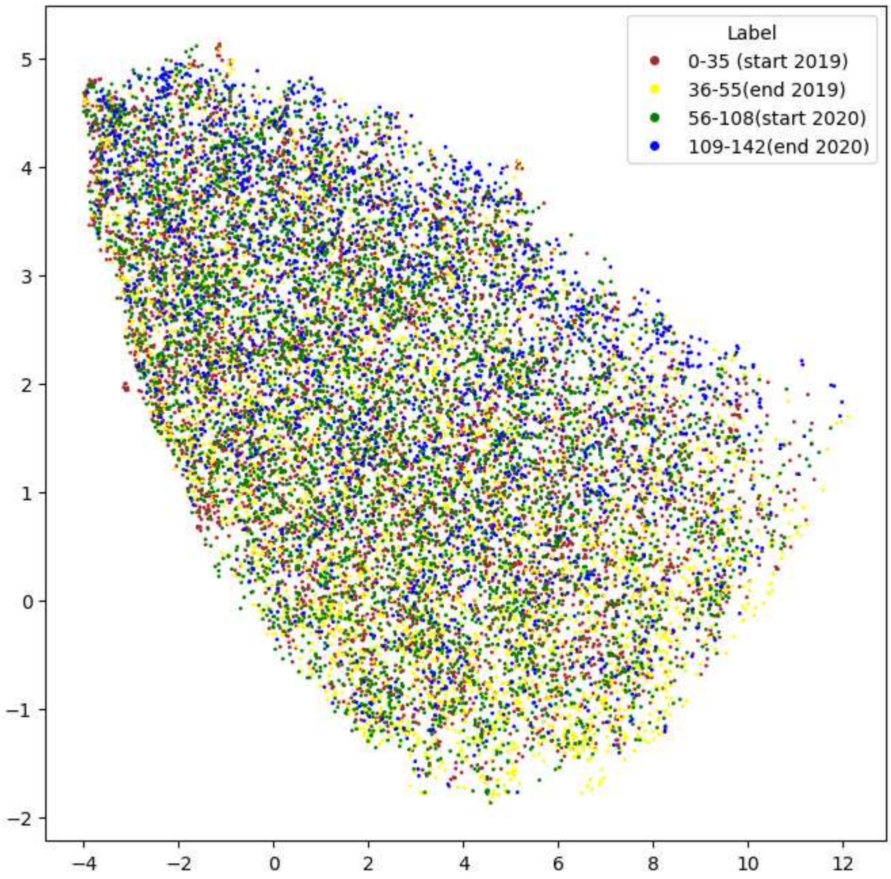


Figure 3.19: UMAP projection of embeddings by runs

It can be said that at the beginning of the training, the embeddings are much more distributed across the entire surface, later becoming more localized in one part of the figure. In the last sessions of 2019, the distribution concentrates more in the lower part; in 2020, it starts again from the bottom, gradually shifting upward. This consistency in the distribution in the lower part of the figure, both before and after the year-long break, could be an important indication of the overall stability of the two classes in the long term. Moreover, looking at the decoder's performance over time, particularly at the *Figure* 3.11, it can be seen that from 2019 to early 2020 (i.e., before and after the break), accuracy remains fairly stable, before fluctuating and decreasing only in the later sessions of 2020. This could explain what we observe with UMAP. What can be concluded from the representations of the embeddings in the reduced space is that there is definitely an internal shift in the distribution, and therefore what was found at the quantitative level can also be confirmed at the visual level. It is certainly not possible to see well-defined clusters or clear separation lines between one period and another, but this is due to the fact that the user's learning is gradual, and therefore the shift in the features used by the network for prediction is also gradual.

3.4 Evaluation metrics

Here are presented the results regarding metrics that should be related to the user's neural pattern modifications, namely Within-class distance (WcDist) and Between-class distance (BcDist). Subsequently, further considerations on ID and accuracy are reported. Finally, interesting and significant correlations between some of these measures are shown.

3.4.1 Within-class Distance

The WcDist is the measure of the distance of the network embeddings from the reference run, defined as the average over all the embeddings of the runs in the training set. It is computed for each class separately. All sessions belonging to the training set and those with an insufficient number of trials were discarded. Additionally, the data refers to non-reduced embeddings, as UMAP does not preserve the distances during dimensionality reduction. The WcDist is model-dependent and should reflect the shift in neural patterns as the subject acquires BCI skills. Therefore, compared to traditional metrics used to directly evaluate the classifier, such as accuracy or precision, it represents a measure related to changes in class distribution and learning.

In the two following figures, the WcDist values are represented in the form of boxplots, highlighting not only the median of the distribution but also the dispersion, symmetry, and the presence of any outliers. In *Figure* 3.20, the boxplot for the BH class is shown. Each group contains 15 runs, remembering that ”Start 2019” begins from run 13 of 2019. Looking at the trend of the average value, it can be seen that the distance from the reference session at the beginning of 2019 is slightly greater than 0.7, then rises to 0.8 at the end of 2019, which is the maximum distance reached. In other words, this means that at the end of the first year, the average distribution of the embeddings was quite distant from that characteristic of the initial training period. The WcDist decreases at the beginning of 2020, reaching a value slightly above 0.6, then rises again at the end of 2020. This latter increase is less pronounced than that of 2019. What immediately stands out is the ”jump” between the end of 2019 and the beginning of 2020, which results in a reduction of the distance from the reference session. Additionally, it can be seen that there is only one outlier at the beginning of 2019 (marked with the symbol of an empty circle), and the distributions are fairly close to the average value, except in the ”End 2019” group, where there are very long whiskers, especially towards positive values, indicating a large dispersion of values compared to the median. ANOVA revealed at least one significant difference between the groups, so a subsequent Tukey-Kramer test was performed. This test showed a statistically significant difference between the start of 2019 and the start of 2020 ($p\_value < 0.05$), between the start and the end of 2019 ($p\_value < 0.05$), between the end of 2019 and the start of 2020 ($p\_value < 0.001$), and between the end of 2019 and the end of 2020 ($p\_value < 0.001$).

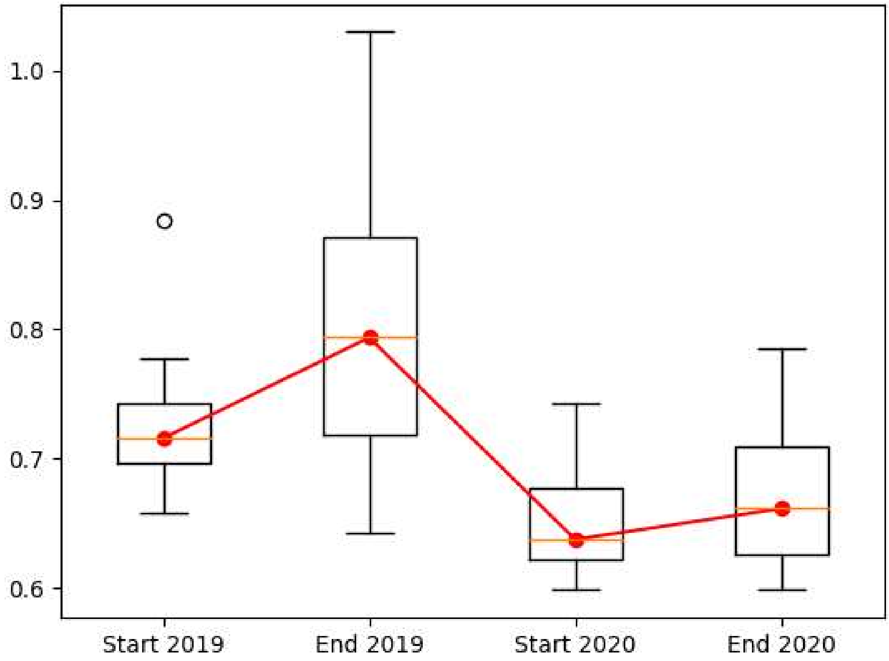


Figure 3.20: Boxplot Within-class Distance for Both Hands

In *Figure 3.21*, the boxplot for the BF class is shown, which exhibits a very different trend compared to that of BH. In fact, the WcDist starts at an average value of 0.8 at the beginning of 2019 and remains stable until the end of the year. There is then an increase to a value of around 0.95, the highest reached, before it drops back down to about 0.85. Here, the "jump" between the end of 2019 and the beginning of 2020 stands out, indicating that, once again, something changes significantly from one year to the next. For BF, there are no outliers, although the distribution appears asymmetric and very dispersed in several cases, particularly in the "Start 2019," "Start 2020," and "End 2020" groups. ANOVA revealed at least one significant difference between the groups. The Tukey-Kramer test indicated a difference between the end of 2019 and the start of 2020 ($p\_value < 0.01$) and between the end of 2019 and the end of 2020 ($p\_value < 0.05$). Comparing this graph with the previous one, it can be seen that the trend from the end of 2019 onward is opposite: in the first case, the distance from the reference session decreases and then increases, while in the second case, it increases and then decreases. Additionally, WcDist values were visualized as scatterplots to observe their evolution over the runs, but no linear pattern emerged, so the graphs are not shown.

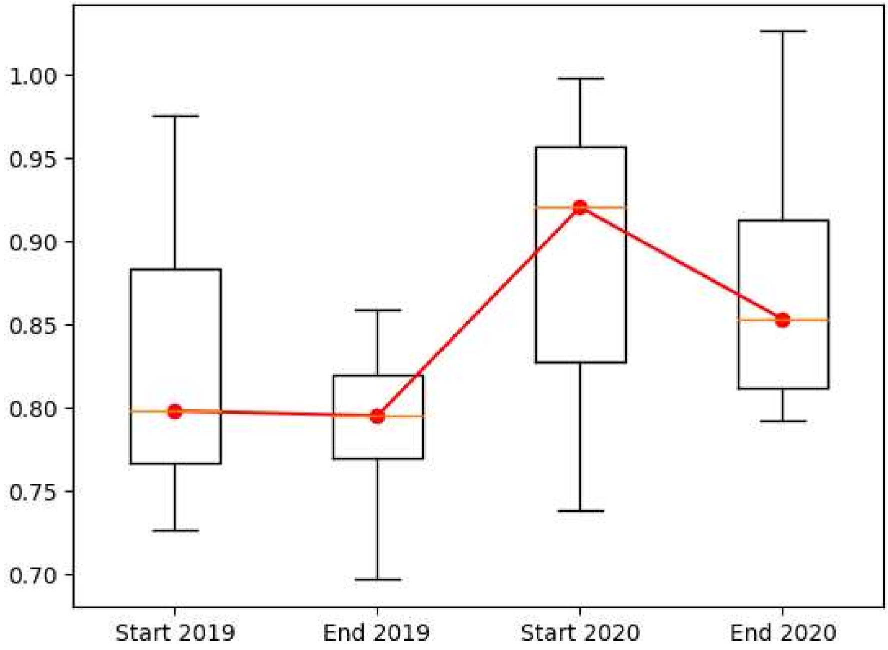


Figure 3.21: Boxplot Within-class Distance for Both Feet

While changes in distribution within the same year reflect short-term adaptations, those between different years indicate long-term adaptations, particularly given the long break in training between the end of 2019 and the beginning of 2020. In the study where these two metrics for evaluating user learning were presented [24], the WcDist was computed by averaging BH and BF, without considering the differences between classes. Additionally, metrics were calculated in both the channel domain and the Riemann domain, separately for the μ and β bands. In their research, the decoder was recalibrated using only features in the β band, while features in the μ band were not used. To compare their results with those of the present study, it is more appropriate to consider only the μ band, as this model was not retrained on new data, making the two trends comparable. For class BH, the trends are consistent. In both cases, there is a regression in the pilot's mental activity after the break between years, as indicated by the ”jump” in the boxplot. For class BF, however, the trend differs. In this study, compared to the previous one, there is an increase in distance from the initial period between 2019 and 2020, indicating a substantial modification in neural patterns from the start of training for this class. These results are once again consistent with those found in the Translation Experiment and are corroborated by the centroid visualization, confirming that the dimensionality reduction performed with UMAP is reliable.

3.4.2 Between-class Distance

In *Figure 3.22*, the boxplot related to the BcDist is shown. The trend of the median starts from higher values at the beginning of 2019, around 0.4, and gradually decreases, reaching about 0.1 at the end of 2020. Intuitively, these results align with the classifier's accuracy, which starts with higher values on the runs close to those used for training and gradually decreases over time. In fact, the smaller the distance between the two classes, the harder it will be to distinguish whether a sample belongs to one class or the other. There are outliers at the beginning and at the end of 2020, and there is a dispersion of values in all groups except for "End 2019," where they are distributed quite close to the median. ANOVA reports the presence of at least one statistically significant difference between groups, which, according to the Tukey-Kramer test, is between the start of 2019 and the end of 2020 ($p\_value < 0.01$), between the end of 2019 and the end of 2020 ($p\_value < 0.001$), and finally, between the start and the end of 2020 ($p\_value < 0.05$). If we compare these results with those from the original study, previously mentioned, the trend is very similar to that observed for the BcDist in the channel domain within the μ band.

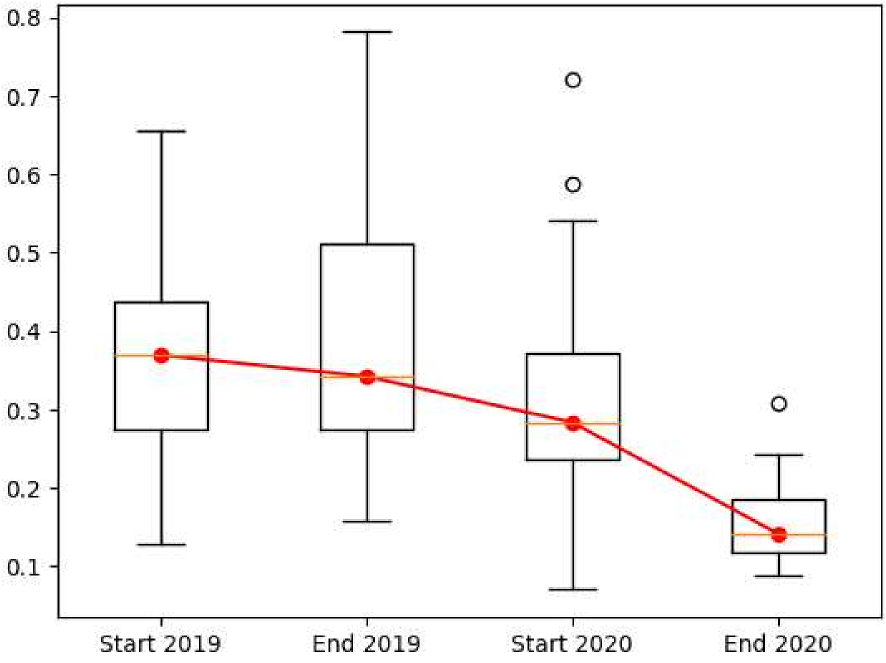


Figure 3.22: Boxplot Between-class Distance

The BcDist results were also represented in scatterplot form to visualize all the runs, not just those of the groups in the boxplot. As seen in *Figure 3.23*, there is no clearly defined linear pattern in 2019, whereas in 2020, a decreasing trend can be observed. In this case, the 2019 and 2020 embeddings were treated separately.

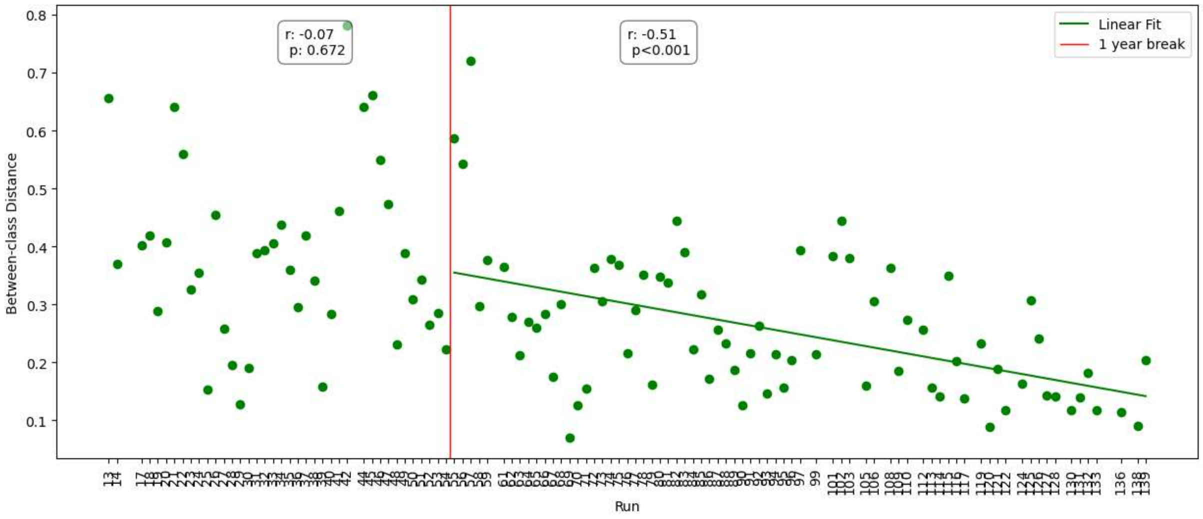


Figure 3.23: Scatterplot Between-class Distance per run

3.4.3 Other metrics

The accuracy and ID were also represented as scatterplots to provide a clear view of the trend of all metrics over the course of the runs, with the analysis again conducted separately for the 2019 and 2020 data. In *Figure* 3.24, it can be seen that ID does not show any particular trend in 2019, but in 2020 it decreases linearly in a statistically significant manner ($p\_value < 0.01$).

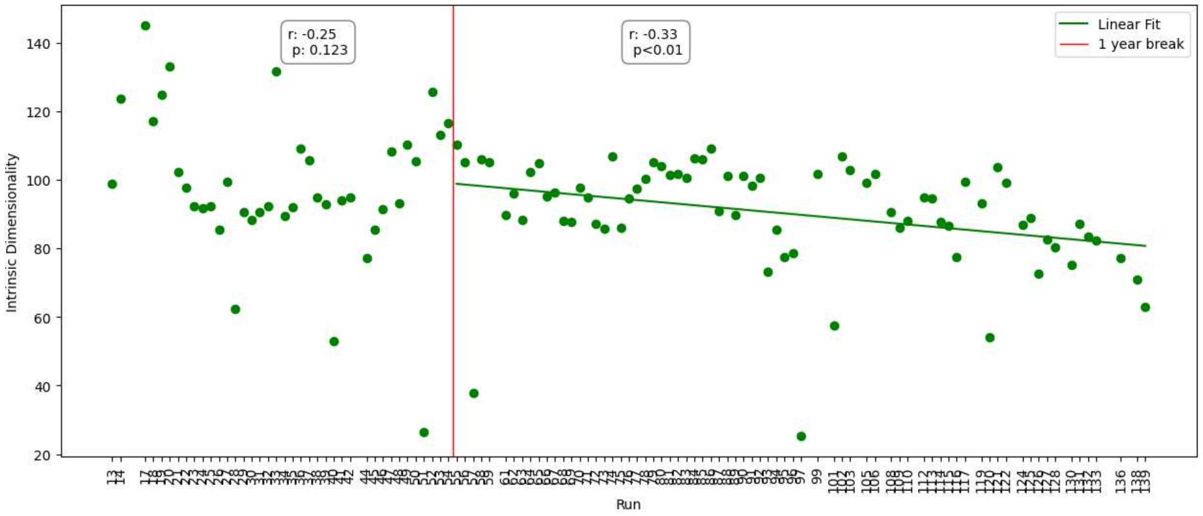


Figure 3.24: Scatterplot Intrinsic Dimensionality per run

In Figure 3.25, the scatterplot of accuracy is shown. Here too, no pattern emerges in 2019, while in 2020 a significant linear decrease can be observed ($p\_value < 0.001$).

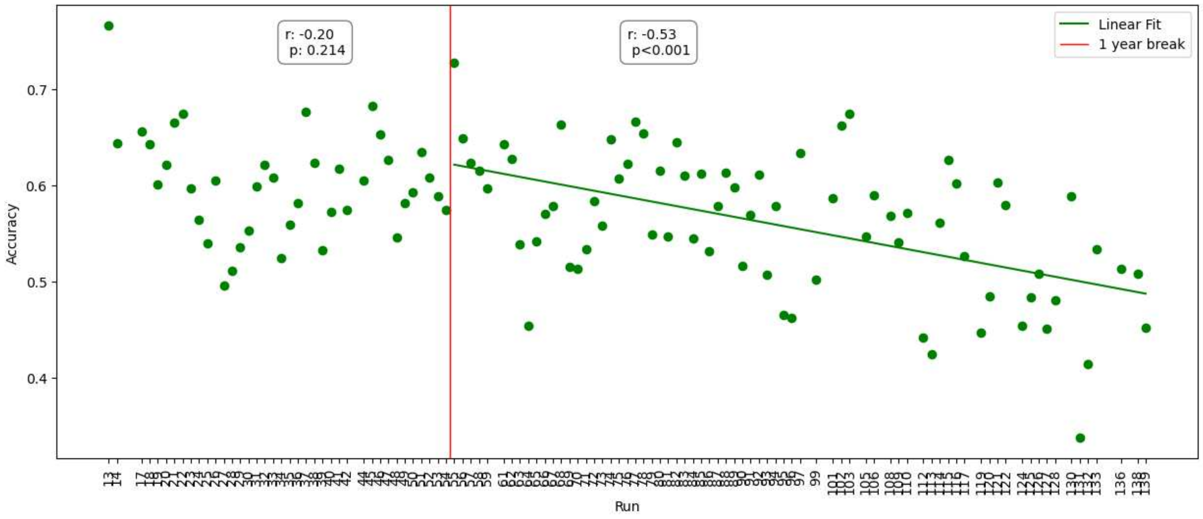


Figure 3.25: Scatterplot Accuracy per run

It stands out that for the BcDist, ID, and accuracy, there is the same type of evolution over time, which suggests a potential relationship between these three metrics, at least for the year 2020. The results on the correlations are reported later.

3.4.4 Correlation measures

First, the correlation between accuracy and WcDist was calculated for both BH and BF, considering the data from 2019 and 2020 together. In *Figure* 3.26, a significant relationship can be seen between accuracy and WcDist for the BH class: as WcDist increases, accuracy also increases. The positive correlation between the two is weak, but the p_value suggests that it is not due to chance. In *Figure* 3.26, the relationship between accuracy and WcDist for BF can be seen, which is a weak negative correlation with a significant p_value. In this case, high values of WcDist are associated with low accuracy, and vice versa. However, the performance degradation may not necessarily be directly due to a greater shift of the BF class from the distribution of the reference run. In fact, as previously mentioned regarding the visualization of the 2020 centroids, this may be because the BH class did not shift as much, which caused the two classes to come closer together, resulting in a decrease in accuracy. Therefore, it cannot be said that a larger shift of a single class always negatively impacts classifier performance; it depends on how the classes shift relative to each other. If there had been an increase in the WcDist for BH as well between 2019 and 2020, rather than the proximity observed in the corresponding boxplot (*Figure* 3.20), probably a positive correlation between accuracy and WcDist for BF might have been observed.

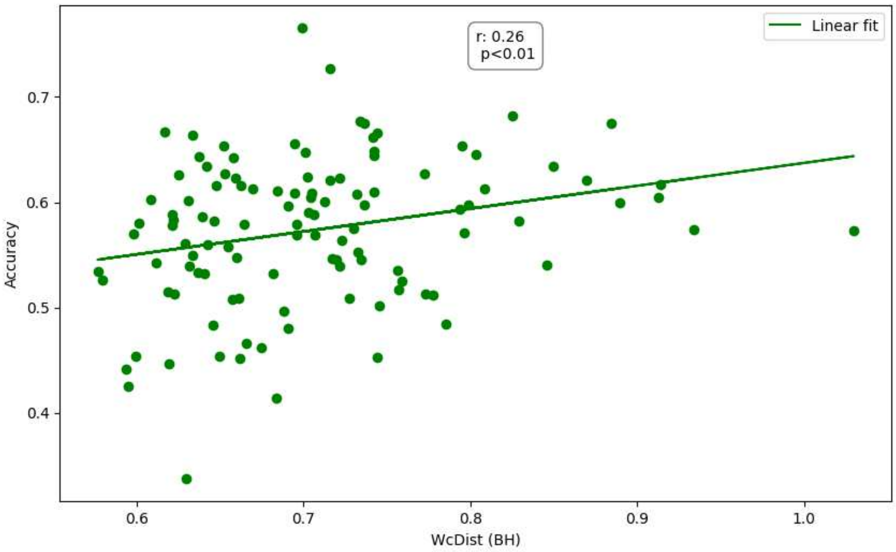


Figure 3.26: Correlation between accuracy and WcDist for BH on 2019-2020 data

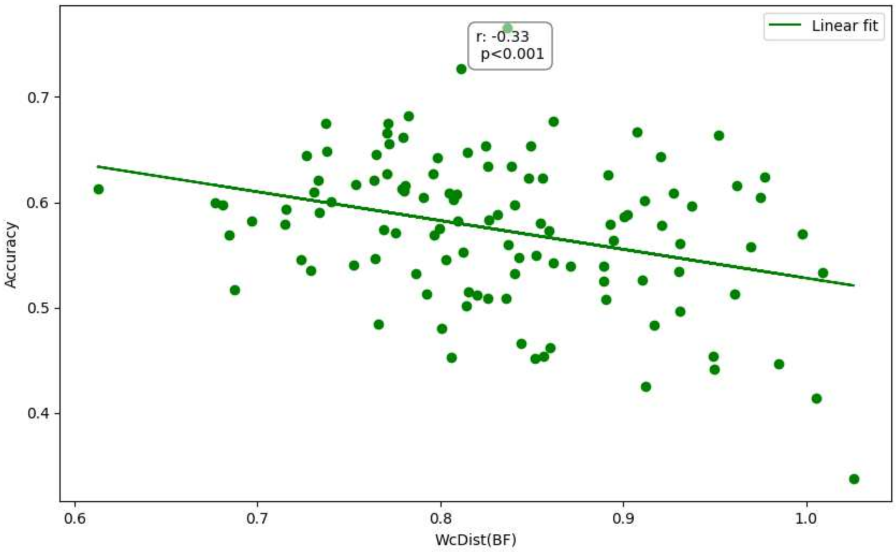


Figure 3.27: Correlation between accuracy and WcDist for BF on 2019-2020 data

An interesting relationship is that between accuracy and BcDist. As expected, the distance between the classes is directly correlated with classifier performance. In the graph below, it can be seen that as the distance between the two classes increases, accuracy improves, in a statistically significant manner (*Figure 3.28*).

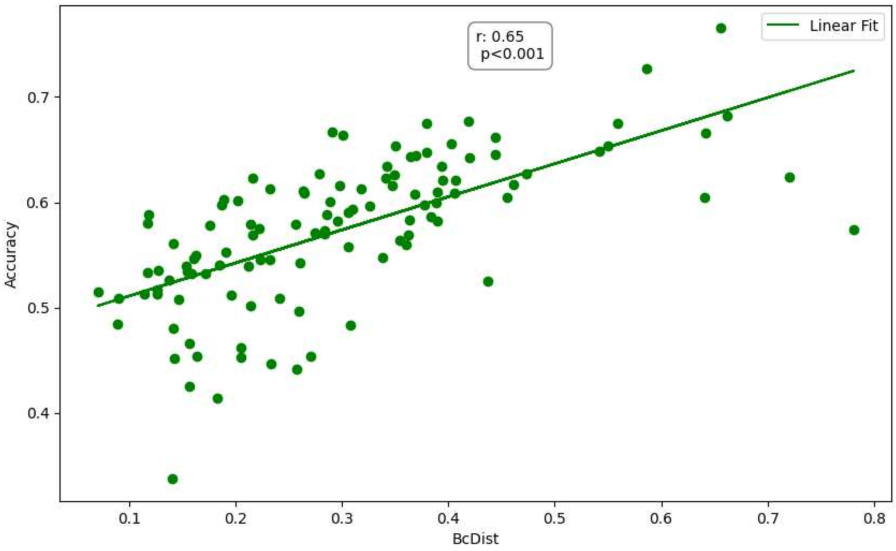


Figure 3.28: Correlation between accuracy and BcDist on 2019-2020 data

A weak but statistically significant positive linear correlation emerged between accuracy and ID (*Figure 3.29*). However, the trend is not as expected. In fact, with higher dimensionality and, consequently, greater data complexity, it would be more challenging to make accurate predictions. The correlation found, instead, indicates that higher ID values correspond to higher accuracy values. A similar and counterintuitive result was also found in the research where the Translation Experiment is presented, and ID is proposed as a metric. One hypothesis made in the article is that greater complexity may be helpful to some extent, as it creates a more diverse dataset that better reflects the real variety of all motor imagery patterns. One might also think that a higher ID could be due to more noise in the data; however, in that case, lower accuracy would have been observed. This leads to the conclusion that the higher number of variables needed to describe the dataset (i.e., higher ID) could provides useful information for prediction.

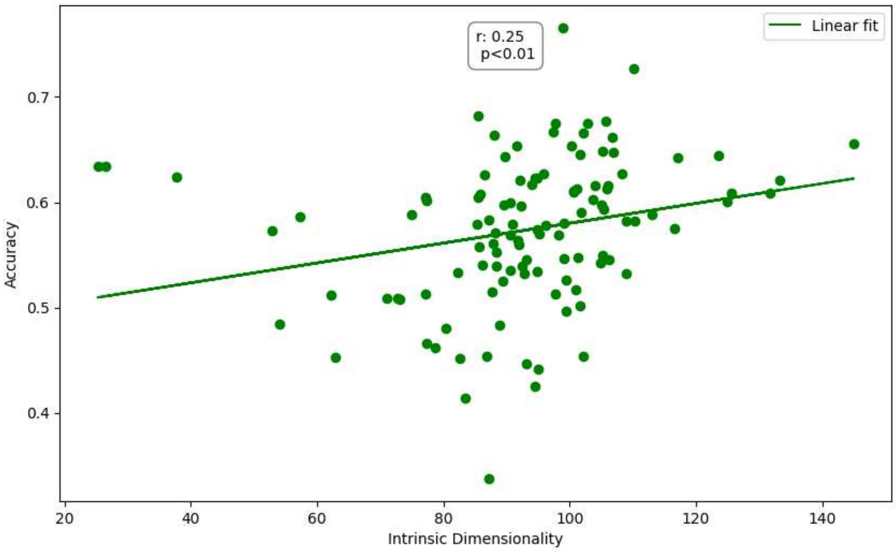


Figure 3.29: Correlation between accuracy and ID on 2019-2020 data

Finally, the correlation between ID and WcDist, as well as between ID and BcDist, was calculated, but neither of them was significant. Therefore, the results will not be reported.

3.5 Discussion

In this study, various approaches were followed, some already proposed in the literature on other datasets or using different experimental paradigms, all aimed at studying user learning. The dataset used lends itself well to this type of analysis, partly due to the long break between training sessions, which allowed for the study of neural patterns during the longitudinal training phase and the potential maintenance or change of those patterns after the year-long break. The initial hypothesis was that once BCI skills were acquired during the early stages, thanks to conditioning feedback, the user would remain stable over time or continue to improve, despite the break from training. This was expected to occur due to neural plasticity, which develops during intensive training and should enable the subject to adapt and continuously improve.

In the original research conducted on the same dataset, a change in the modulated features was observed, shifting from the lateral channels (associated with the BH task) to the central channels (associated with the BF task). This allowed the conclusion that the pilot had learned to perform the latter motor imagery task well over time. In this research, a more detailed analysis was conducted, focusing not only on the overall model performance but also on the performance of individual tasks, to verify whether the user learning differed between classes and how this change occurred over the sessions. Thanks to the Translation Experiment, it was found that there was indeed an improvement in the data quality for BF, which, however, was not accompanied by a similar improvement in data quality for BH. Why this growth happens for only one class and not the other cannot be determined with certainty, and the explanation likely lies at the neurophysiological level. Considering both classes, the overall performance trend, in terms of accuracy and F1-Score, remains stable for a while and then decreases towards the end of 2020. This indicates that, despite the model being continuously trained on new data, those from the later sessions are more difficult to interpret.

Regarding ID in relation to the Translation Experiment, it turned out to be a metric that, in this study, did not prove useful in explaining the classifier's performance trends over the iterations and instead seemed to be related to the dimensionality of the trial on which it was calculated each time. In the subsequent analysis, where the ID value is reported for each run, it was found to be positively correlated with the accuracy of the best model, which could indicate that more complex data provides useful information in the classification process. However, ID could be influenced by noise in the data or other hidden factors. Given the complexity of brain signals, this remains a potentially useful metric, but one that is difficult to interpret.

UMAP has proven to be a very useful technique for investigating hidden structures in the data. It is important to always consider that dimensionality reduction methods may not faithfully represent the original data and are highly dependent on the input parameters [54, 55]. Additionally, the higher the starting dimensionality, the more complex it becomes to achieve a reduction that preserves the data structure characteristics. The EEG signal is also quite noisy, and this noise could be misinterpreted by UMAP, creating structures that do not exist in the original dataset. In the conducted study, no anomalous results were found in the visualization of the embeddings; there are no clusters separated from the main group, and there do not appear to be clear separations suggesting artifacts created by the technique used. Moreover, as mentioned earlier, it seems that the actual distances between the classes are respected, as what is seen in the model embeddings visualization is consistent with the classifier's performance and with the other metrics (calculated on the non-reduced embeddings).

The metrics related to the shifting of neural patterns, namely WcDist and BcDist, seem to be good indicators for monitoring the user learning. Both correlate with the model's performance, especially BcDist, which measures the distance between BH and BF. Compared to the original study, where WcDist was calculated by averaging across the two classes, here, in line with the study's objectives, the metric was analyzed separately for the two motor imagery tasks. This allowed us to see that for BH, there was a regression between the end of 2019 and the beginning of 2020, with the distribution moving closer to that of the early training days, while for BF, there was a significant increase in distance between the two years. The fact that in both cases the most significant shifts, whether positive or negative, occurred between one year and the next suggests long-term cerebral adaptations, which, however, do not show the stability needed to maintain high decoder performance without requiring recalibration.

It is important to emphasize that the connection found between the results of the Translation Experiment and those of the subsequent analysis is not straightforward, as a different model was trained for each iteration in the former, while the latter was conducted on a single model. Obtaining results that do not conflict, especially regarding the significant improvement of the user in the BF task, means validating the methods used to evaluate learning and confirming both qualitatively and quantitatively that an actual shift in neural patterns occurred, which were two of the main objectives of this study.

Chapter 4

Conclusion

This work aims to demonstrate the neural changes that occur during the prolonged use of a MI-BCI, which are still not sufficiently considered when designing this type of technology. It therefore emphasizes the importance of placing the user at the center in order to fully exploit the system's potential and to enable its use in future real-world contexts, not just in laboratory settings. In recent times, significant progress has been made, but not enough for BCIs to be used in routine clinical practice. An important factor in the successful implementation of the system is the user's adaptation, which often does not depend on the use of a specific model, but on how frequently the decoder is recalibrated. Long-term studies have shown that it is crucial to recalibrate the BCI only when performance becomes too low, which happens due to a substantial change in the subject's neural patterns, reflected in the altered characteristics of the EEG signal, which the classifier can no longer decode. In this sense, the use of deep learning, rather than classical algorithms found in the literature for encoding sensorimotor rhythms, has many advantages, especially with regard to the autonomous selection of the most suitable features. Specific architectures optimized for BCI, such as EEGNet, open the door to a potential future where the system is calibration-free.

In this research, methods were tested to monitor the training process through the visualization of the neural network's latent space, that proves useful in observing classes distribution changes, i.e., MI tasks, over time. Additionally, UMAP, with appropriate modifications to reduce computational cost and increase execution speed, is a technique that could potentially be used in real-time to observe how the network's embeddings are positioned in space from one session to the next. In this case, UMAP was used solely for visualizing the neural network's latent space, but in future research, it could also be applied to raw data to visualize and remove potential noise, thus improving subsequent classification. This could be particularly useful because, when using deep learning, preprocessing is sometimes skipped, and therefore noise or artifacts in the data can have a very negative impact on the results. Furthermore, the importance of using appropriate metrics is once again emphasized, as these metrics can clearly and directly quantify how much the neural patterns have shifted or remained stable from the beginning to the end of the training. The aim was to demonstrate that classifier accuracy alone is not a metric that fully reflects the subject's developments or improvements. In fact, it has been observed that there can be an improvement in data quality for one class rather than the other, which, however, is not reflected in the overall accuracy.

Future developments of this work may involve applying the proposed techniques to a dataset consisting of multiple subjects, to validate the findings and, if necessary, to identify statistical significance where only low evidence was found. Additionally, these techniques will need to be applied online, and possibly a classifier should be developed that adapts when significant distribution changes occur, based on the computed values. Finally, the learning process could be further understood through an in-depth study of the neurophysiological changes that occur, which were not addressed in this case but are important for understanding, for example, how the discriminability of features or brain connectivity changes during training.

References

[1] Iñaki Iturrate et al. “Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control”. en. In: *Scientific Reports* 5.1 (Sept. 2015), p. 13893. issn: 2045-2322. doi: 10.1038/srep13893. url: https://www.nature.com/articles/srep13893 (visited on 07/09/2024).

[2] Xavier Perrin et al. “Brain-coupled interaction for semi-autonomous navigation of an assistive robot”. en. In: *Robotics and Autonomous Systems* 58.12 (Dec. 2010), pp. 1246–1255. issn: 09218890. doi: 10.1016/j.robot.2010.05.010. url: https://linkinghub.elsevier.com/retrieve/pii/S0921889010001144 (visited on 07/09/2024).

[3] Gregory L. Holmes and Roustem Khazipov. “Basic Neurophysiology and the Cortical Basis of EEG”. en. In: *The Clinical Neurophysiology Primer*. Ed. by Andrew S. Blum and Seward B. Rutkove. Totowa, NJ: Humana Press, 2007, pp. 19–33. isbn: 978-0-89603-996-4. doi: 10.1007/978-1-59745-271-7_2. url: http://link.springer.com/10.1007/978-1-59745-271-7_2 (visited on 07/08/2024).

[4] Robert Leeb et al. “Transferring brain–computer interfaces beyond the laboratory: Successful application control for motor-disabled users”. en. In: *Artificial Intelligence in Medicine* 59.2 (Oct. 2013), pp. 121–132. issn: 09333657. doi: 10.1016/j.artmed.2013.08.004. url: https://linkinghub.elsevier.com/retrieve/pii/S0933365713001218 (visited on 07/09/2024).

[5] Ravikiran Mane, Tushar Chouhan, and Cuntai Guan. “BCI for stroke rehabilitation: motor and beyond”. en. In: *Journal of Neural Engineering* 17.4 (Aug. 2020), p. 041001. issn: 1741-2560, 1741-2552. doi: 10.1088/1741-2552/aba162. url: https://iopscience.iop.org/article/10.1088/1741-2552/aba162 (visited on 07/07/2024).

[6] Mahdi Bamdad, Homayoon Zarshenas, and Mohammad A. Auais. “Application of BCI systems in neurorehabilitation: a scoping review”. en. In: *Disability and Rehabilitation: Assistive Technology* 10.5 (Sept. 2015), pp. 355–364. issn: 1748-3107, 1748-3115. doi:

10.3109/17483107.2014.961569. url: http://www.tandfonline.com/doi/full/10.3109/17483107.2014.961569 (visited on 07/07/2024).

[7] Luca Tonin, Felix Christian Bauer, and Jose Del R. Millan. “The Role of the Control Framework for Continuous Teleoperation of a Brain–Machine Interface-Driven Mobile Robot”. en. In: *IEEE Transactions on Robotics* 36.1 (Feb. 2020). Publisher: Institute of Electrical and Electronics Engineers (IEEE), pp. 78–91. issn: 1552-3098, 1941-0468. doi: 10.1109/tro.2019.2943072. url: https://ieeexplore.ieee.org/document/8879618/ (visited on 09/17/2024).

[8] Christian Klaes. “Chapter 28 - Invasive Brain-Computer Interfaces and Neural Recordings From Humans”. In: *Handbook of Behavioral Neuroscience*. Ed. by Denise Manahan-Vaughan. Vol. 28. Handbook of Neural Plasticity Techniques. Elsevier, Jan. 2018, pp. 527–539. doi: 10.1016/B978-0-12-812028-6.00028-8. url: https://www.sciencedirect.com/science/article/pii/B9780128120286000288 (visited on 09/16/2024).

[9] Jose M Carmena et al. “Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates”. en. In: *PLoS Biology* 1.2 (Oct. 2003). Ed. by Idan Segev, e42. issn: 1545-7885. doi: 10.1371/journal.pbio.0000042. url: https://dx.plos.org/10.1371/journal.pbio.0000042 (visited on 07/09/2024).

[10] Jordan J. Levett et al. “Invasive Brain Computer Interface for Motor Restoration in Spinal Cord Injury: A Systematic Review”. In: *Neuromodulation: Technology at the Neural Interface* 27.4 (June 2024), pp. 597–603. issn: 1094-7159. doi: 10.1016/j.neurom.2023.10.006. url: https://www.sciencedirect.com/science/article/pii/S1094715923007547 (visited on 09/16/2024).

[11] W. Wang et al. “Human motor cortical activity recorded with Micro-ECoG electrodes, during individual finger movements”. en. In: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 15. Minneapolis, MN: IEEE, Sept. 2009, pp. 586–589. doi: 10.1109/iembs.2009.5333704. url: http://ieeexplore.ieee.org/document/5333704/ (visited on 09/17/2024).

[12] Jörn Rickert et al. “Encoding of Movement Direction in Different Frequency Ranges of Motor Cortical Local Field Potentials”. en. In: *The Journal of Neuroscience* 25.39 (Sept. 2005), pp. 8815–8824. issn: 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0816-05.2005. url: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0816-05.2005 (visited on 07/09/2024).

[13] Rodrigo Quian Quiroga and Stefano Panzeri. “Extracting information from neuronal populations: information theory and decoding approaches”. en. In: *Nature Reviews Neuroscience* 10.3 (Mar. 2009), pp. 173–185. issn: 1471-003X, 1471-0048. doi: 10.1038/nrn2578. url: https://www.nature.com/articles/nrn2578 (visited on 07/09/2024).

[14] Pasquale Arpaia et al. “How to successfully classify EEG in motor imagery BCI: a metrological analysis of the state of the art”. en. In: *Journal of Neural Engineering* 19.3 (June 2022). Publisher: IOP Publishing, p. 031002. issn: 1741-2552. doi: 10.1088/1741-2552/ac74e0. url: https://dx.doi.org/10.1088/1741-2552/ac74e0 (visited on 06/20/2024).

[15] Janis Peksa and Dmytro Mamchur. “State-of-the-Art on Brain-Computer Interface Technology”. eng. In: *Sensors (Basel, Switzerland)* 23.13 (June 2023), p. 6001. issn: 1424-8220. doi: 10.3390/s23136001.

[16] Swati Aggarwal and Nupur Chugh. “Review of Machine Learning Techniques for EEG Based Brain Computer Interface”. en. In: *Archives of Computational Methods in Engineering* 29.5 (Aug. 2022), pp. 3001–3020. issn: 1886-1784. doi: 10.1007/s11831-021-09684-6. url: https://doi.org/10.1007/s11831-021-09684-6 (visited on 06/20/2024).

[17] Khondoker Murad Hossain et al. “Status of deep learning for EEG-based brain–computer interface applications”. en. In: *Frontiers in Computational Neuroscience* 16 (Jan. 2023), p. 1006763. issn: 1662-5188. doi: 10.3389/fncom.2022.1006763. url: https://www.frontiersin.org/articles/10.3389/fncom.2022.1006763/full (visited on 04/18/2024).

[18] Keiron O’Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. en. arXiv:1511.08458 [cs]. Dec. 2015. url: http://arxiv.org/abs/1511.08458 (visited on 07/25/2024).

[19] Zewen Li et al. “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects”. en. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (Dec. 2022), pp. 6999–7019. issn: 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2021.3084827. url: https://ieeexplore.ieee.org/document/9451544/ (visited on 07/25/2024).

[20] No-Sang Kwak, Klaus-Robert Müller, and Seong-Whan Lee. “A convolutional neural network for steady state visual evoked potential classification under ambulatory environment”. en. In: *PLOS ONE* 12.2 (Feb. 2017). Ed. by Friedhelm Schwenker. Publisher: Public Library of Science (PLoS), e0172578. issn: 1932-6203. doi: 10.1371/journal.

pone.0172578. url: https://dx.plos.org/10.1371/journal.pone.0172578 (visited on 09/17/2024).

[21] Mustafa Sameer and Bharat Gupta. “CNN based framework for detection of epileptic seizures”. en. In: *Multimedia Tools and Applications* 81.12 (May 2022), pp. 17057–17070. issn: 1573-7721. doi: 10.1007/s11042-022-12702-9. url: https://doi.org/10.1007/s11042-022-12702-9 (visited on 09/16/2024).

[22] Mengni Zhou et al. “Epileptic Seizure Detection Based on EEG Signals and CNN”. English. In: *Frontiers in Neuroinformatics* 12 (Dec. 2018). Publisher: Frontiers. issn: 1662-5196. doi: 10.3389/fninf.2018.00095. url: https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2018.00095/full (visited on 09/16/2024).

[23] Vernon J. Lawhern et al. “EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces”. en. In: *Journal of Neural Engineering* 15.5 (Oct. 2018). arXiv:1611.08024 [cs, q-bio, stat], p. 056013. issn: 1741-2560, 1741-2552. doi: 10.1088/1741-2552/aace8c. url: http://arxiv.org/abs/1611.08024 (visited on 04/18/2024).

[24] Stefano Tortora et al. “Neural correlates of user learning during long-term BCI training for the Cybathlon competition”. en. In: *Journal of NeuroEngineering and Rehabilitation* 19.1 (July 2022), p. 69. issn: 1743-0003. doi: 10.1186/s12984-022-01047-x. url: https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-022-01047-x (visited on 04/18/2024).

[25] Maciej Śliwowski et al. “Impact of dataset size and long-term ECoG-based BCI usage on deep learning decoders performance”. English. In: *Frontiers in Human Neuroscience* 17 (Mar. 2023). Publisher: Frontiers. issn: 1662-5161. doi: 10.3389/fnhum.2023.1111645. url: https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2023.1111645/full (visited on 06/24/2024).

[26] Serafeim Perdikis et al. “The Cybathlon BCI race: Successful longitudinal mutual learning with two tetraplegic users”. en. In: *PLOS Biology* 16.5 (2018). Publisher: Public Library of Science, e2003787. issn: 1545-7885. doi: 10.1371/journal.pbio.2003787. url: https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.2003787 (visited on 06/25/2024).

[27] Camille Benaroch et al. “Long-Term BCI Training of a Tetraplegic User: Adaptive Riemannian Classifiers and User Training”. English. In: *Frontiers in Human Neuroscience* 15 (Mar. 2021). Publisher: Frontiers. issn: 1662-5161. doi: 10.3389/fnhum.2021.

635653. url: https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2021.635653/full (visited on 06/25/2024).

[28] Robert Leeb et al. “Towards Independence: A BCI Telepresence Robot for People With Severe Motor Disabilities”. en. In: *Proceedings of the IEEE* 103.6 (June 2015). Publisher: Institute of Electrical and Electronics Engineers (IEEE), pp. 969–982. issn: 0018-9219, 1558-2256. doi: 10.1109/jproc.2015.2419736. url: http://ieeexplore.ieee.org/document/7109829/ (visited on 09/17/2024).

[29] M. F. Mridha et al. “Brain-Computer Interface: Advancement and Challenges”. en. In: *Sensors* 21.17 (Jan. 2021). Number: 17 Publisher: Multidisciplinary Digital Publishing Institute, p. 5746. issn: 1424-8220. doi: 10.3390/s21175746. url: https://www.mdpi.com/1424-8220/21/17/5746 (visited on 07/12/2024).

[30] Robin Tibor Schirrmeister et al. “Deep learning with convolutional neural networks for EEG decoding and visualization”. en. In: *Human Brain Mapping* 38.11 (Nov. 2017). arXiv:1703.05051 [cs], pp. 5391–5420. issn: 1065-9471, 1097-0193. doi: 10.1002/hbm.23730. url: http://arxiv.org/abs/1703.05051 (visited on 04/18/2024).

[31] Hao Zhu, Dylan Forenzo, and Bin He. “On the Deep Learning Models for EEG-Based Brain-Computer Interface Using Motor Imagery”. en. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 30 (2022), pp. 2283–2291. issn: 1534-4320, 1558-0210. doi: 10.1109/TNSRE.2022.3198041. url: https://ieeexplore.ieee.org/document/9854857/ (visited on 04/18/2024).

[32] Haider Raza, Anirban Chowdhury, and Saugat Bhattacharyya. “Deep Learning based Prediction of EEG Motor Imagery of Stroke Patients’ for Neuro-Rehabilitation Application”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. ISSN: 2161-4407. July 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9206884. url: https://ieeexplore.ieee.org/document/9206884 (visited on 04/29/2024).

[33] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. en. arXiv:1802.03426 [cs, stat]. Sept. 2020. url: http://arxiv.org/abs/1802.03426 (visited on 06/25/2024).

[34] Peter Wassenaar, Pierre Guetschel, and Michael Tangermann. *Approximate UMAP allows for high-rate online visualization of high-dimensional data streams*. arXiv:2404.04001 [cs, eess]. Apr. 2024. doi: 10.48550/arXiv.2404.04001. url: http://arxiv.org/abs/2404.04001 (visited on 06/20/2024).

[35] Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. “Parametric UMAP Embeddings for Representation and Semisupervised Learning”. en. In: *Neural Computation* (Aug. 2021), pp. 1–27. issn: 0899-7667, 1530-888X. doi: 10.1162/neco_a_01434. url: https://direct.mit.edu/neco/article/doi/10.1162/neco_a_01434/107068/Parametric-UMAP-Embeddings-for-Representation-and (visited on 06/24/2024).

[36] Luca Tonin et al. “Learning to control a BMI-driven wheelchair for people with severe tetraplegia”. English. In: *iScience* 25.12 (Dec. 2022). Publisher: Elsevier. issn: 2589-0042. doi: 10.1016/j.isci.2022.105418. url: https://www.cell.com/iscience/abstract/S2589-0042(22)01690-X (visited on 07/14/2024).

[37] Fabien Lotte and Camille Jeunet. “Defining and quantifying users’ mental imagery-based BCI skills: a first step”. en. In: *Journal of Neural Engineering* 15.4 (Aug. 2018), p. 046030. issn: 1741-2560, 1741-2552. doi: 10.1088/1741-2552/aac577. url: https://iopscience.iop.org/article/10.1088/1741-2552/aac577 (visited on 06/25/2024).

[38] F Mattioli, C Porcaro, and G Baldassarre. “A 1D CNN for high accuracy classification and transfer learning in motor imagery EEG-based brain-computer interface”. en. In: *Journal of Neural Engineering* 18.6 (Dec. 2021), p. 066053. issn: 1741-2560, 1741-2552. doi: 10.1088/1741-2552/ac4430. url: https://iopscience.iop.org/article/10.1088/1741-2552/ac4430 (visited on 04/18/2024).

[39] Johannes Höhne et al. “Motor Imagery for Severely Motor-Impaired Patients: Evidence for Brain-Computer Interfacing as Superior Control Solution”. In: *PLoS ONE* 9.8 (Aug. 2014), e104854. issn: 1932-6203. doi: 10.1371/journal.pone.0104854. url: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4146550/ (visited on 08/23/2024).

[40] Minmin Miao et al. “Spatial-Frequency Feature Learning and Classification of Motor Imagery EEG Based on Deep Convolution Neural Network”. en. In: *Computational and Mathematical Methods in Medicine* 2020 (July 2020), pp. 1–13. issn: 1748-670X, 1748-6718. doi: 10.1155/2020/1981728. url: https://www.hindawi.com/journals/cmmm/2020/1981728/ (visited on 04/18/2024).

[41] A. Rozza et al. “Novel high intrinsic dimensionality estimators”. en. In: *Machine Learning* 89.1 (Oct. 2012), pp. 37–65. issn: 1573-0565. doi: 10.1007/s10994-012-5294-7. url: https://doi.org/10.1007/s10994-012-5294-7 (visited on 09/12/2024).

[42] Francesco Camastra and Antonino Staiano. “Intrinsic dimension estimation: Advances and open problems”. In: *Information Sciences* 328 (Jan. 2016), pp. 26–41. issn: 0020-

0255. doi: 10.1016/j.ins.2015.08.029. url: https://www.sciencedirect.com/science/article/pii/S0020025515006179 (visited on 09/16/2024).

[43] Jonathan Bac et al. "Scikit-Dimension: A Python Package for Intrinsic Dimension Estimation". en. In: *Entropy* 23.10 (Oct. 2021). Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, p. 1368. issn: 1099-4300. doi: 10.3390/e23101368. url: https://www.mdpi.com/1099-4300/23/10/1368 (visited on 06/27/2024).

[44] Feng-Lei Fan et al. "On Interpretability of Artificial Neural Networks: A Survey". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 5.6 (Nov. 2021). Conference Name: IEEE Transactions on Radiation and Plasma Medical Sciences, pp. 741–760. issn: 2469-7303. doi: 10.1109/TRPMS.2021.3066428. url: https://ieeexplore.ieee.org/document/9380482/?arnumber=9380482 (visited on 09/13/2024).

[45] Yu Zhang et al. "A Survey on Neural Network Interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (Oct. 2021). Conference Name: IEEE Transactions on Emerging Topics in Computational Intelligence, pp. 726–742. issn: 2471-285X. doi: 10.1109/TETCI.2021.3100641. url: https://ieeexplore.ieee.org/document/9521221/?arnumber=9521221 (visited on 09/13/2024).

[46] Quan-shi Zhang and Song-chun Zhu. "Visual interpretability for deep learning: a survey". en. In: *Frontiers of Information Technology & Electronic Engineering* 19.1 (Jan. 2018), pp. 27–39. issn: 2095-9230. doi: 10.1631/FITEE.1700808. url: https://doi.org/10.1631/FITEE.1700808 (visited on 09/13/2024).

[47] Ji-Seon Bang and Seong-Whan Lee. "Interpretable Convolutional Neural Networks for Subject-Independent Motor Imagery Classification". In: *2022 10th International Winter Conference on Brain-Computer Interface (BCI)*. ISSN: 2572-7672. Feb. 2022, pp. 1–5. doi: 10.1109/BCI53720.2022.9734822. url: https://ieeexplore.ieee.org/abstract/document/9734822 (visited on 09/13/2024).

[48] Pierre Guetschel, Sara Ahmadi, and Michael Tangermann. *Review of Deep Representation Learning Techniques for Brain-Computer Interfaces and Recommendations*. en. arXiv:2405.19345 [cs, eess]. May 2024. url: http://arxiv.org/abs/2405.19345 (visited on 06/24/2024).

[49] Till Nierhaus et al. "Immediate brain plasticity after one hour of brain–computer interface (BCI)". en. In: *The Journal of Physiology* 599.9 (2021). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1113/JP278118, pp. 2435–2451. issn: 1469-7793. doi: 10.1113/JP278118. url: https://onlinelibrary.wiley.com/doi/abs/10.1113/JP278118 (visited on 09/16/2024).

[50] Yu Liu, Wei Chen, and Michael Lew. "A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision". en. In: *International Journal of Multimedia Information Retrieval* 9.3 (Sept. 2020), pp. 135–170. issn: 2192-662X. doi: 10.1007/s13735-019-00183-w. url: https://doi.org/10.1007/s13735-019-00183-w (visited on 09/13/2024).

[51] Michael E. Houle. "Dimensionality, Discriminability, Density and Distance Distributions". In: *2013 IEEE 13th International Conference on Data Mining Workshops*. ISSN: 2375-9259. Dec. 2013, pp. 468–473. doi: 10.1109/ICDMW.2013.139. url: https://ieeexplore.ieee.org/document/6753958/?arnumber=6753958 (visited on 09/13/2024).

[52] Phillip Pope et al. *The Intrinsic Dimension of Images and Its Impact on Learning*. en. Apr. 2021. url: https://arxiv.org/abs/2104.08894v1 (visited on 09/13/2024).

[53] Brett D. Roads and Bradley C. Love. "The Dimensions of dimensionality". en. In: *Trends in Cognitive Sciences* (Aug. 2024), S136466132400189X. issn: 13646613. doi: 10.1016/j.tics.2024.07.005. url: https://linkinghub.elsevier.com/retrieve/pii/S136466132400189X (visited on 09/12/2024).

[54] Hyeon Jeon et al. "Classes are Not Clusters: Improving Label-Based Evaluation of Dimensionality Reduction". In: *IEEE Transactions on Visualization and Computer Graphics* 30.1 (Jan. 2024). Conference Name: IEEE Transactions on Visualization and Computer Graphics, pp. 781–791. issn: 1941-0506. doi: 10.1109/TVCG.2023.3327187. url: https://ieeexplore.ieee.org/abstract/document/10308618 (visited on 09/12/2024).

[55] Bárbara C. Benato, Alexandre X. Falcão, and Alexandru C. Telea. "Measuring the quality of projections of high-dimensional labeled data". In: *Computers & Graphics* 116 (Nov. 2023), pp. 287–297. issn: 0097-8493. doi: 10.1016/j.cag.2023.08.023. url: https://www.sciencedirect.com/science/article/pii/S0097849323001929 (visited on 09/12/2024).