



SAPIENZA
UNIVERSITÀ DI ROMA

Feature selection algorithms for corticomuscular coherence-based Brain-Computer Interfaces

Facoltà di Ingegneria dell'informazione, Informatica e Statistica
Corso di Laurea Magistrale in Intelligenza Artificiale e Robotica

Candidate

Gioele Migno
ID number 1795826

Thesis Advisor

Prof. Febo Cincotti

Co-Advisor

Prof. Emma Colamarino

Academic Year 2023/2024

Thesis defended on 22 October 2024
in front of a Board of Examiners composed by:

Prof. Massimo Mecella (chairman)
Prof. Simone Agostinelli
Prof. Bottoni Paolo Gaspare
Prof. Febo Cincotti
Prof. Francesca Cuomo
Prof. Luca Iocchi
Prof. Domenico Lembo

Feature selection algorithms for corticomuscular coherence-based Brain-Computer Interfaces

Master's thesis. Sapienza – University of Rome

© 2024 Gioele Migno. All rights reserved

This thesis has been typeset by L^AT_EX and the Sapthesis class.

Version: October 7, 2024

Author's email: gioele.migno@gmail.com

Abstract

The application field of this work is neuroengineering, in particular Brain-Computer Interface (BCI) used to assist the rehabilitation process of post-stroke patients. The focus of the research is on automated feature selection step, a key component necessary to improve this technology. The aim is to expand its use in other contexts where the expertise on BCI, and the knowledge of the neurophysiological principles that guide the feature selection, are limited. Moreover, automated feature selection process allows to improve the choice by also considering the classification performance at the end of the pipeline.

After the literature review about the feature selection algorithms used by the research community in this field, several existing methods and a novel one were compared on a group of healthy subjects. The processed data was provided by IRCCS Santa Lucia Foundation, Rome, Italy.

This text contains a chapter with the theoretical fundamentals necessary to make the content of this work accessible to researchers of artificial intelligence and data analysis fields without any background in neuroscience.

Contents

Introduction	vii
1 Theoretical fundamentals	1
1.1 Anatomical fundamentals of the nervous system	1
1.1.1 The brain	1
1.1.1.1 The neuron	2
1.1.1.2 The cerebral cortex	3
1.1.1.3 Brain lobes and Brodmann areas	4
1.1.1.4 Motor and sensory areas of the cerebral cortex	5
1.1.2 The muscles	7
1.1.2.1 Skeletal muscles	8
1.1.2.2 Upper limbs muscles	9
1.2 Brain and muscle signal acquisition	13
1.2.1 Scalp electroencephalography (EEG)	13
1.2.1.1 Electrode system	15
1.2.1.2 Spatial filtering	17
1.2.1.3 Artifacts	18
1.2.1.4 Brain rhythms	19
1.2.2 Electromyography (EMG)	20
1.3 Multivariate analysis of biomedical signals	21
1.3.1 Power Spectral Density	22
1.3.2 Coherence	23
1.3.3 Cortico-Muscular Coherence	23
1.4 Brain-Computer Interface	24
1.4.1 EEG signal features	24
1.4.2 Post-stroke rehabilitation	25
1.5 Supervised learning and feature selection	26
2 State of the art	27
2.1 Feature extraction	27
2.2 Feature selection	28
2.2.1 Filter methods	30
2.2.1.1 Mutual information	32
2.2.1.2 Principal Component Analysis	34
2.2.2 Wrapper methods	34
2.2.2.1 Genetic Algorithm	36

2.2.2.2	Stepwise Regression	40
2.2.3	Embedded methods	44
3	Materials and methods	45
3.1	Datasets	45
3.1.1	Signal acquisition and feature extraction	45
3.1.2	Feature space reduction to account for neurophysiological constraints	47
3.2	Feature selection algorithms to compare	48
3.2.1	Performance metrics	50
3.2.1.1	AUC and F1 Score	50
3.2.1.2	SoftS and HardS	50
3.2.2	Hyperparameter tuning for GAAM	51
3.2.3	Random Planet	54
3.2.3.1	Algorithm description	54
3.2.3.2	Hyperparameter tuning	56
3.3	Algorithms comparison	56
4	Results	59
4.1	Hyperparameter tuning	59
4.1.1	Genetic algorithm with aggressive mutation	59
4.1.2	Random Planet	62
4.2	Algorithms comparison	63
4.2.1	ExtL	63
4.2.1.1	NoFilter	63
4.2.1.2	Laplacian	70
4.2.2	ExtR	74
4.2.2.1	NoFilter	74
4.2.2.2	Laplacian	78
4.2.3	GraspL	82
4.2.3.1	NoFilter	82
4.2.3.2	Laplacian	87
4.2.4	GraspR	91
4.2.4.1	NoFilter	91
4.2.4.2	Laplacian	95
5	Discussion and conclusion	99
5.1	Discussion	99
5.2	Conclusion	100
Appendices		113
.1	Tables hyperparameter tuning Random Planet	113
.2	Tables methods comparison	130

Introduction

Stroke is a condition in which the brain cells suddenly die because of lack of oxygen caused by an obstruction in the blood flow. This condition has become one of the main reasons for abnormal human death. Moreover, patients who survive are usually left with disabilities, affecting especially limb movement. Thanks to the medical advances, in recent years there has been a noticeable increase in the survival rate of post-stroke patients resulting in a higher demand for advanced rehabilitation techniques. Indeed, intensive rehabilitation after stroke can really make the difference in overcoming or not disabilities. The exercises and training of rehabilitation encourage unaffected areas of the brain to learn to perform functions that were done by the damaged area [70] [9].

Among the more recent advanced rehabilitation techniques, there are the ones based on Brain-Computer Interfaces (BCI). These systems for motor rehabilitation after stroke have proven their efficacy to enhance upper limb motor recovery by reinforcing motor related brain activity [33]. During rehabilitation is fundamental to monitor if the patient is attempting to perform a desired movement by controlling muscles in a physiological way. After stroke indeed, it is common for a patient attempt to perform a movement of the limb involving maladaptive and not functional motor strategies. If this misbehavior is not corrected, the patient will not be able to restore correctly his/her motor abilities. BCI technology allows a specialist to monitor the brain activity of the patient and provide him/her feedback. However, an important step of this process is selecting the right features to monitor during the rehabilitation process to assess if the movement attempt is performed correctly or not. At the moment this feature selection step is performed by highly trained specialists. In the future, this step might be performed by an automated system allowing to make possible post-stroke patients to perform rehabilitation exercises with the supervision of clinicians not specialized.

In the present work, after a review of the literature, the most adopted feature selection methods are tested and compared. The performance are assessed by considering the metrics obtained in a classification task where the aim is to distinguish if a subject is performing a desired movement, or he/she is at rest. The current initial analysis performed in this work, is focused on healthy subjects who did not present any evidence of known history of neuromuscular disorders.

The rest of the text is organized as follows:

- **Section 1:** The theoretical fundamentals necessary to understand the application field are introduced;
- **Section 2:** A review of the state-of-the-art feature selection methods is presented;
- **Section 3:** The dataset and the compared feature selection techniques are described. Moreover, for the methods which require it, the hyperparameter tuning procedure is described;
- **Section 4:** This section contains the obtained results of the hyperparameter tuning and of the comparison among the tested methods;
- **Section 5:** Discussion and final conclusions are reported.

Chapter 1

Theoretical fundamentals

This chapter contains the theoretical fundamentals required to understand the content of the present work. It is organized as follows:

- **Section 1.1:** introduces the nervous system with focus on the brain and the upper limbs' muscles anatomies;
- **Section 1.2:** describes how to acquire signal from the brain and the muscles;
- **Section 1.3:** describes the concept of coherence which is a key component used in the feature extraction step;
- **Section 1.4:** contains an introduction to BCI technique with the focus on post-rehabilitation application;
- **Section 1.5:** describes briefly the definition of supervised learning and feature selection process.

1.1 Anatomical fundamentals of the nervous system

The *nervous system* includes the brain, spinal cord, and a complex network of nerves. This system sends messages back and forth between the brain and the body. It is made up of two parts [8]:

- **Central nervous system:** includes the brain and spinal cord;
- **Peripheral nervous system:** includes the nerves that run throughout the whole body.

In this study, the brain is the component of interest from the central nervous system. From the peripheral nervous system instead, the focus is on muscles which receiving stimuli from the nerves, allow the body to move in the surrounding environment.

1.1.1 The brain

The brain is a multiprocessing system that receives information from the surrounding environment and from the human body. It processes such information and controls

body actions accordingly. The proper action is chosen by predicting the future on the basis of the past in order to help the individual to survive and perpetuate the species [42]. The brain structure is very complex and is organized in several levels, in neuroscience its functionality is studied at different scales: a single neuron, a brain area, the entire brain and even multiple brains interacting with each other [22] [13]. Concerning the current work, this section is organized by first briefly introducing the neural cell and then the cerebral cortex with the focus on the areas associated with the control of body movements.

1.1.1.1 The neuron

The complexity of the brain relies on the combination of billions of neurons, which taken individually, have a very simple operation necessary to elaborate and process signals. In Fig. 1.1 the structure of a typical neuron is shown. On the left side there is the cell body which receives input signals from a tree structure composed by the *dendrites*. The output is then propagated to other cells by the *axon terminals* [13].

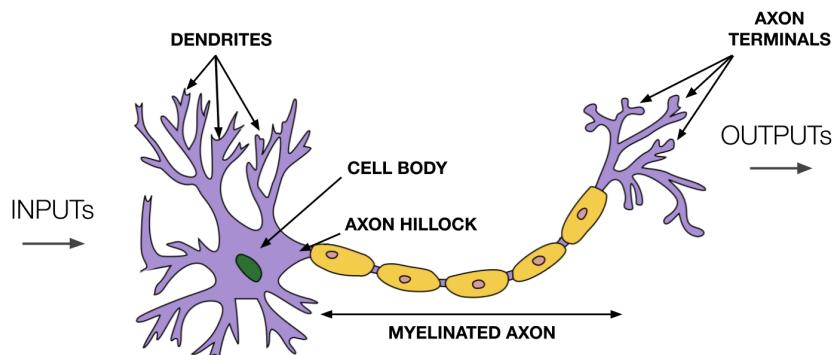


Figure 1.1. Diagram of a typical neuron in the brain. The dendrites and the axon terminals represent respectively the input and the output of the cell. The signal is generated by the axon hillock and then is propagated through the myelinated axon.¹

The function of a neuron is not only to transfer an electrical signal from a source to a destination, it has also to elaborate and process it. Indeed, to be more specific, the four main neuron functions executed in sequence are:

1. **Collection of information:** From the dendrites, a neuron acquires the inputs from other neurons. The connection is called *synapse*;
2. **Integration in time and space:** All the inputs acquired are summed both in space, since they come from different dendrites, and in time, since the timing of each input influences the final output;
3. **Generation of output signal:** The signal in output contains a single bit of information, if the sum of the input signals reach a certain threshold then a

¹Image source: adapted from <https://commons.wikimedia.org/w/index.php?curid=151245917>

spike called *action potential* is generated (high level output = 1) otherwise not (low level output = 0);

4. **Propagation of output signal:** When the action potential is generated by the *axon hillock*, the signal is propagated through the *myelinated axon* up to the *axon terminals* where other neurons are connected. Note that the same output is repeated in each axon terminal, they are not different signals.

1.1.1.2 The cerebral cortex

The first distinction that can be made in the brain is *gray matter* and *white matter*. The gray matter is composed by the cell bodies and dendrites of the neurons. The white one is instead formed by myelinated axon of the neurons. This last is the matter in charge of information propagation and the former one instead, of the information processing. The brain can be also macroscopically classified into two parts: *cerebral cortex* and *subcortical area*. The cerebral cortex is the external surface of the brain and the most of the gray matter is present there, it is in charge of advanced functions such as language, body control and vision. Subcortical area instead, it is located in the inner part of the brain and contains more primitive and vital functions like digestion, breathing and heart rate. Due to where the subcortical area is positioned, it is a challenging part to reach with non-invasive brain recording methods. For this reason, in neuroengineering field, the studies are mainly focused on just the brain cortex which is highly involved in the executive function of motor and communication behaviors. In order to increase the cortical surface in a limited volume the cerebral cortex is folded as shown in Fig. 1.2c. The outward folds are called *gyri*, the inward ones *sulci* instead. Sulci represent about 2/3 of the surface, hence most of the cortical surface is hidden making more difficult punctual recording in those areas [103] [22].

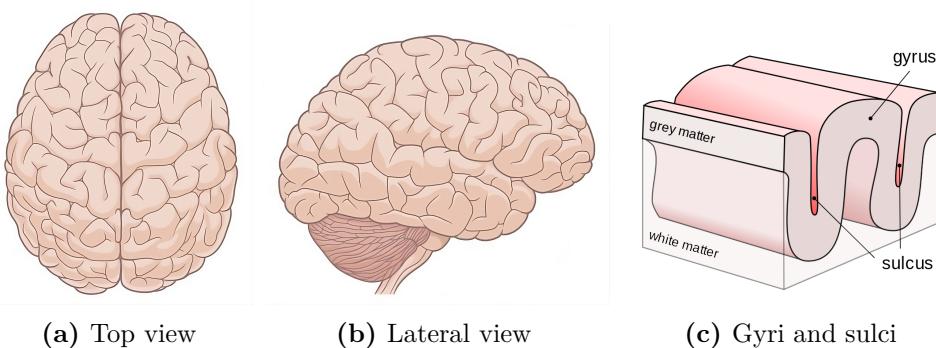


Figure 1.2. (a) Top view of the brain where left and right hemisphere are both visible. (b) Lateral view, only the right hemisphere and the *cerebellum* are visible. (c) Gyri (*gyrus singular*) and sulci (*sulcus singular*).²

²Image source: adapted from <https://www.istockphoto.com>, <https://commons.wikimedia.org/w/index.php?curid=147334212>.

1.1.1.3 Brain lobes and Brodmann areas

The cerebral cortex is divided into two hemispheres Fig. 1.2a which are mostly functional symmetrical. The hemispheres are contralateral, namely the right hemisphere controls the left part of the body and the left one the right side. Each hemisphere is partitioned into four brain lobes Fig. 1.3 [22]:

- **Frontal lobe:** It occupies the front part of the brain, and it is in charge of high level cognitive processes such as reasoning, decision-making and social skills;
- **Parietal lobe:** It is located behind the frontal lobe, and it is separated from it by the *central sulcus*. Some of the functionalities of this lobe are spelling, perception and object manipulation;
- **Occipital Lobe:** On the back side of the skull, the occipital lobe is located in. Its main functionality is the processing of visual information;
- **Temporal Lobe:** It is positioned behind the ears, and it is separated from the frontal lobe by the *lateral sulcus*. This lobe is responsible for functions related to memories and basic hearing.

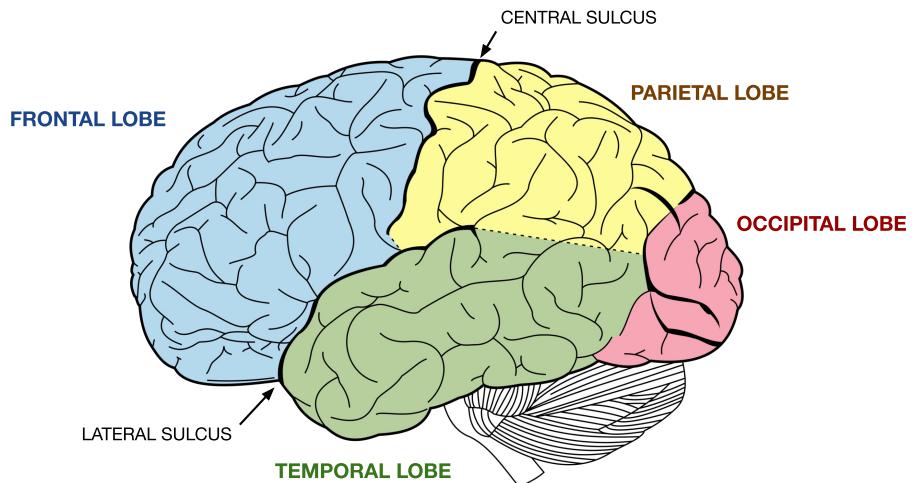


Figure 1.3. Brain lobes on lateral view of the left hemisphere. Central sulcus divides the frontal and the parietal lobe. Lateral sulcus instead, divides the temporal lobe from them. Finally, the occipital lobe is on the back.³

The cerebral cortex subdivision in lobes is mainly given by the main sulci, another approach to subdivide is according to the neural network layout. Using this approach, in the 1909, Korbinian Brodmann identified 52 areas Fig. 1.4 within the cerebral

³Image source: adapted from <https://commons.wikimedia.org/w/index.php?curid=1676555>

cortex according to their cytoarchitectures, namely the neural cells type and their organization. The areas are symmetrical on the two hemispheres [103].

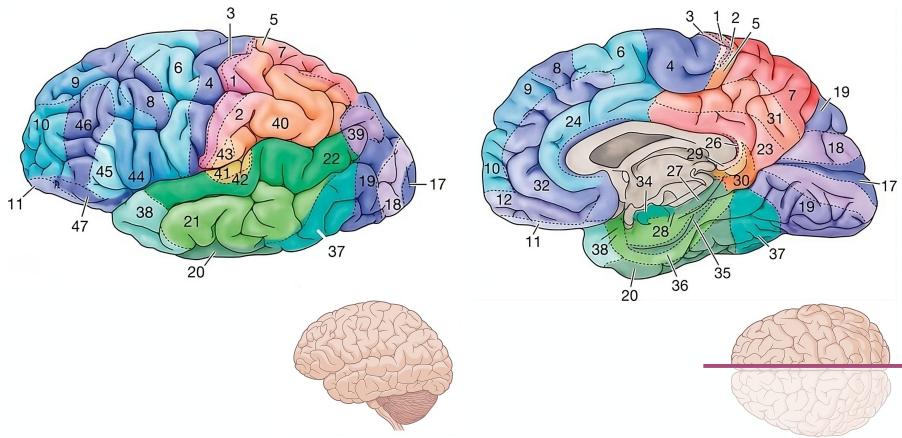


Figure 1.4. Some the Brodmann areas. Lateral view of the left hemisphere on the left and medial surface view on the right.⁴

1.1.1.4 Motor and sensory areas of the cerebral cortex

The main cortical areas involved in the planning and execution of voluntary movements reside near the central sulcus, in the frontal and parietal lobes. The focus of past and current neuroengineering researches have been on four areas [103]:

- **Primary motor cortex:** It corresponds to Brodmann area 4 and is involved in movement execution;
- **Primary somatosensory cortex:** It conveys the sensation of touch, temperature, pain and limb position. It is located in Brodmann areas 1,2 and 3;
- **Premotor cortex:** It is responsible for motion planning upon external stimulus such as light. It is positioned in Brodmann area 6;
- **Supplementary motor area:** It is similar to premotor cortex but is involved in motion planning guided by internal stimulus such as memory. This region is located in Brodmann area 6.

These areas are introduced in the follow.

Primary motor cortex (M1)

The primary motor cortex (M1) Fig. 1.5 is located in the frontal lobe and lies along the anterior wall of the central sulcus and continues into the precentral gyrus. This region corresponds to Brodmann area 4 and is involved in movement execution. The

⁴Image source: adapted from [76]

primary motor cortex is organized somatotopically, namely particular regions of it are devoted primarily to the control of particular body areas, this organization is reflected by a cortical map called Penfield's homunculus Fig. 1.6. The name comes from the distorted image of the human body used where the parts that require more finely graded control, such as hands, have disproportionately large representations [103].



Figure 1.5. Primary motor cortex. Perspective, lateral and top view.⁵

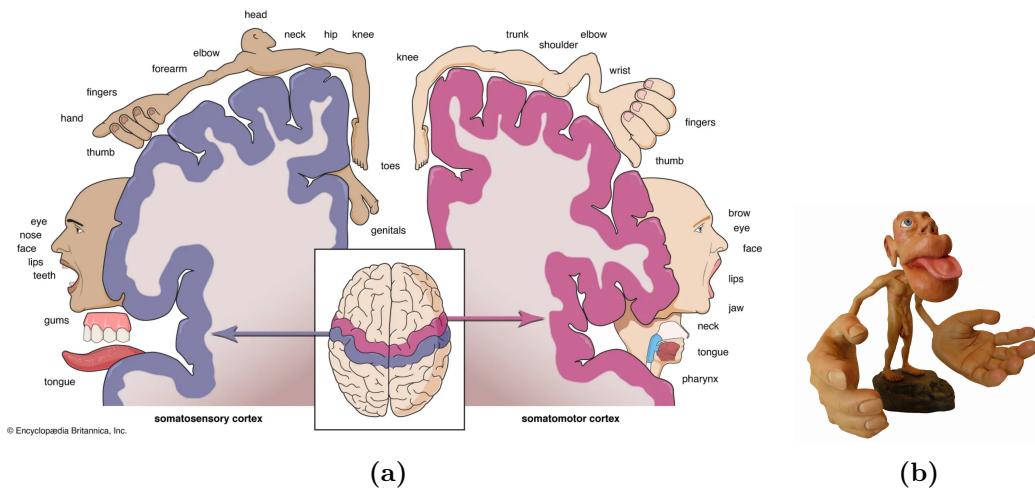


Figure 1.6. (a) Somatotopic arrangement of primary motor cortex and primary sensory cortex. (b) Homunculus. In somatosensory cortex, the more innervated an area is, the greater its representation is. In somatomotor cortex instead, bigger parts correspond to body parts where the movement has finer control such as hands.⁶

Primary somatosensory cortex (S1)

The primary somatosensory cortex (S1) Fig. 1.7 is located in the parietal lobe, Brodmann areas 1,2 and 3. This region is important in movement because it conveys the sensation of touch, temperature, pain and limb position that are crucial in guiding movement. S1, as M1, is organized somatotopically and the map is given by Penfield's homunculus Fig. 1.6.

⁵Image source: generated with <https://www.brainfacts.org/3d-brain>

⁶Image source: <https://www.britannica.com/science/homunculus-biology>

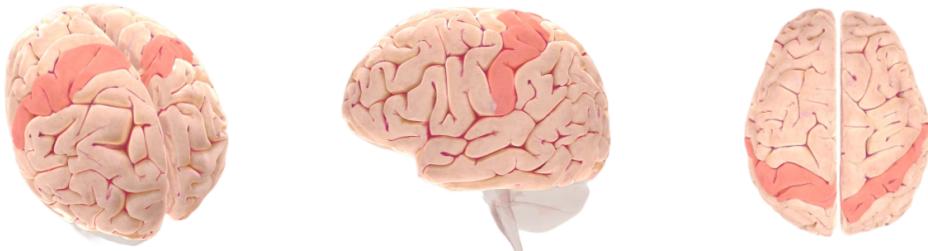


Figure 1.7. Primary somatosensory cortex. Perspective, lateral and top view.⁷

Premotor cortex and supplementary motor area

Promotor cortex (PM) Fig. 1.8 and supplementary motor area (SMA) Fig. 1.9 are located in the frontal lobe, Brodmann area 6. More precisely, SMA is placed in the interhemispheric fissure behind PM. While primary motor cortex is involved in the actual movement execution, PM and SMA are responsible for motion planning. Experimental evidence on monkeys suggests that PM is activated upon an external guided movement, for instance by a light source, and SMA instead, is related to internally triggered movements provided by memory for instance [103] [68].



Figure 1.8. Premotor cortex. Perspective, lateral and top view.⁷



Figure 1.9. Supplementary motor area. Perspective, lateral and top view.⁷

1.1.2 The muscles

This section concerns the basic concepts on how the muscles work.

⁷Image source: generated with <https://www.brainfacts.org/3d-brain>

1.1.2.1 Skeletal muscles

Three different types of muscle tissue are present in the human body: *cardiac muscle*, *smooth muscle* and *skeletal muscle*. As the name suggests, cardiac muscle typology is found only in the heart, and it is necessary to pump blood in the body. Smooth muscles instead, are located in the walls of hollow organs such as intestines and lungs, their contractions regulate the passage of material through the organs by changing their internal diameters. The last typology of muscle derives its name from the fact skeletal muscles are usually attached to the skeleton. The contraction of such type of muscles allows the movement of single parts of the body. When the contraction of several skeletal muscles happens in a coordinated way, then the entire body moves in the environment performing complex movements such as walking or object manipulation [2].

A single skeletal muscle is composed of hundreds of individual cylindrical shaped cells, called *fibers*, bound together by connective tissue. This typology of muscle is stimulated by a specific type of neurons located in our body called *motor neurons*. A motor neuron receives the impulse from the brain or spinal cord and when it delivers the signal, it causes the contraction of the controlled muscle fiber. A single motor neuron can innervate (i.e., control) several fibers but each muscle fiber is controlled by only one motor neuron. The smallest functional unit to describe the neuronal control of the muscular contraction process is called *motor unit* and is composed by the combination of a single motor neuron and all the muscle fibers it controls. Fig. 1.10 shows a diagram with two motor units [2][6].

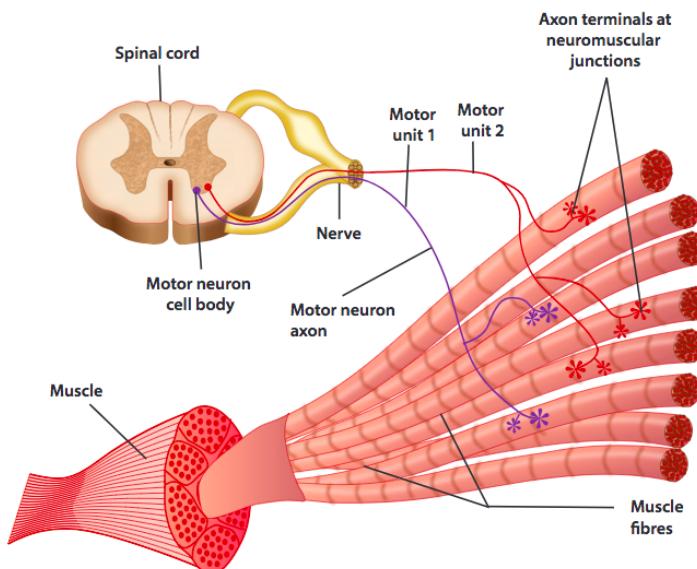


Figure 1.10. Two motor units. Each motor unit is composed by a motor neuron and the muscle fibers it controls.⁸

The activation of a single motor neuron causes the contraction of all the controlled muscle fibers causing therefore, the activation of the entire motor unity. The degree

⁸Image source: <https://github.com/iandanforth/pymuscle>

of skeletal muscle contraction is controlled by the number of motor units activated within the muscle of interest, and by the frequency of motor neuron impulses in each unit. When to perform a task is necessary to increase the strength of a muscle contraction, the brain increases the number of simultaneously active motor units within the muscle. This process is called *motor unit recruitment* [2].

1.1.2.2 Upper limbs muscles

The interest muscles in this study are the ones of the upper limbs. Fig. 1.11 shows the superficial view of the muscles on the upper part of the right arm. In Fig. 1.12 instead, the muscles of the right forearm are shown. In these figures the relevant muscles on this work are highlighted and are: *deltoid*, *pectoralis major*, *biceps brachii* and *triceps brachii* in Fig. 1.11, and *flexor digitorum superficialis* and *extensor digitorum* in Fig. 1.12. Moreover, Fig. 1.16 shows the *trapezius* [1].

In Fig. 1.14 a better view of flexor digitorum superficialis and extensor digitorum is reported. Flexor digitorum superficialis flexes the middle phalanges of the medial four digits and the wrist joint. Extensor digitorum has a similar functionality but instead of flexes it extends [1].

The remaining figures show in more details other muscles of interest during arm movement. Fig. 1.15 shows the subdivisions of biceps brachii and triceps brachii. Fig. 1.13 contains the three parts of the deltoid: *lateral deltoid*, *anterior deltoid* and *posterior deltoid*. Finally, Fig. 1.16 shows *upper*, *middle* and *lower trapezius* necessary to move the scapula. The upper trapezius also extends, laterally flexes, and contralaterally rotates the head and neck at the spinal joints [1].

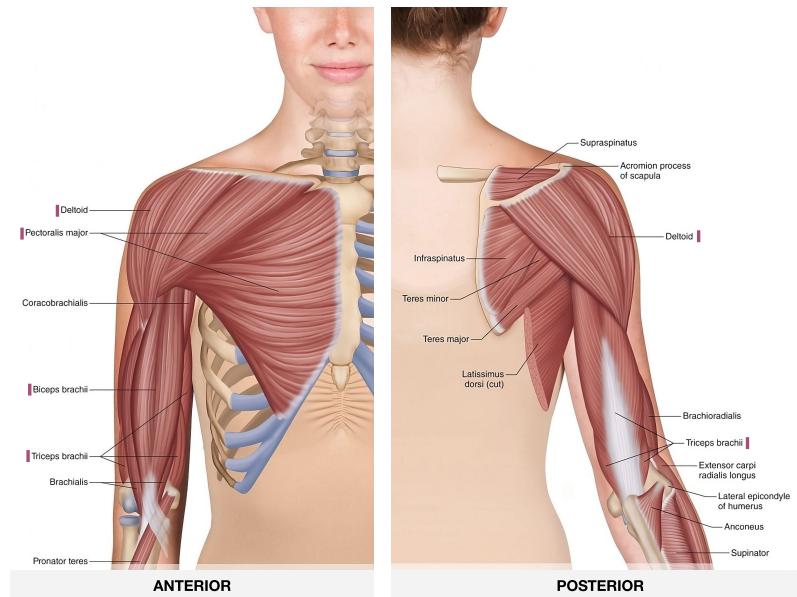


Figure 1.11. Superficial anterior and posterior view of the right arm. Relevant muscles for this study are highlighted.⁹

⁹Image source: adapted from [1]

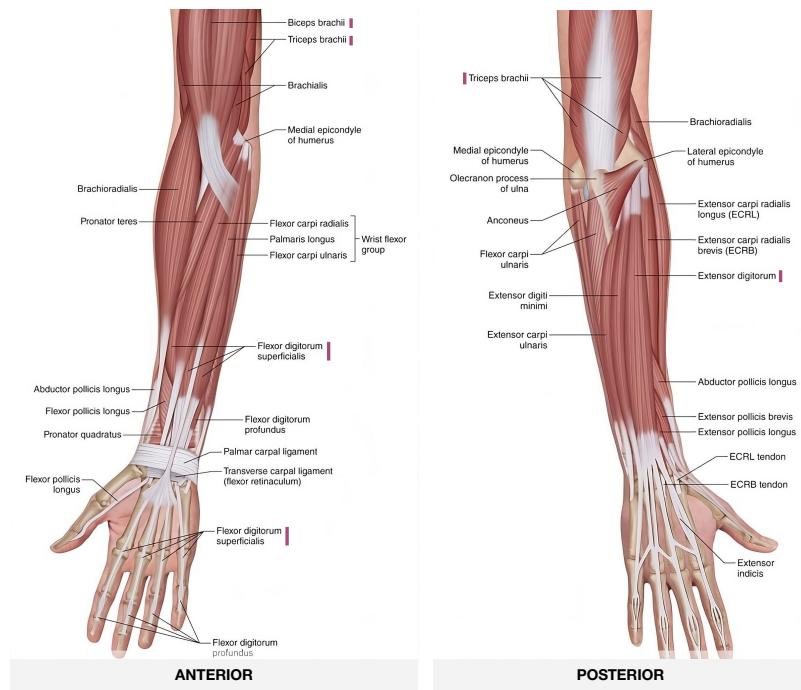


Figure 1.12. Superficial anterior and posterior view of the right forearm. In the anterior view the palm of the hand is shown while in the posterior one, the back of the hand is visible. Relevant muscles for this study are highlighted.¹⁰

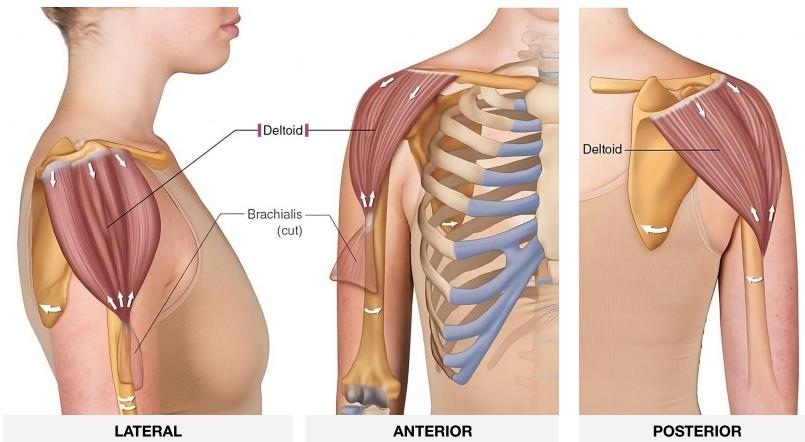


Figure 1.13. Lateral, anterior and posterior deltoid.¹⁰

¹⁰Image source: adapted from [1]

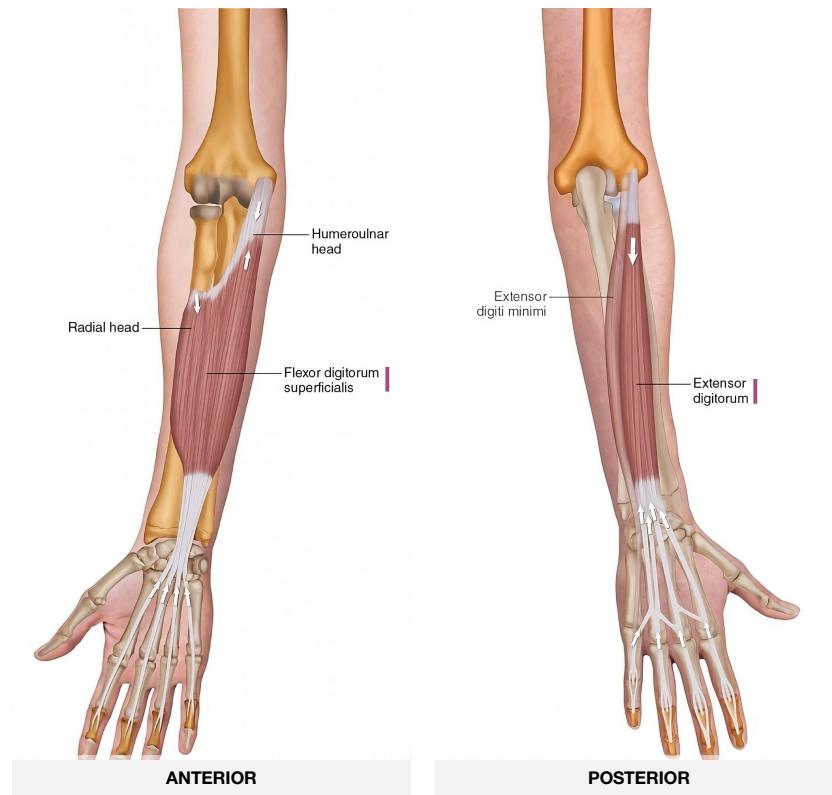


Figure 1.14. Flexor digitorum superficialis and extensor digitorum.¹¹

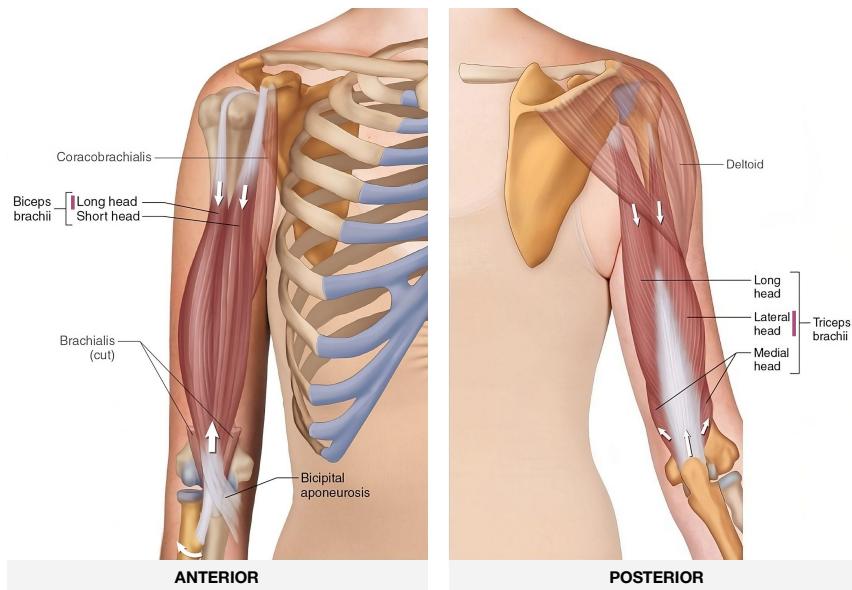


Figure 1.15. Subdivisions of biceps brachii and triceps brachii.¹¹

¹¹Image source: adapted from [1]

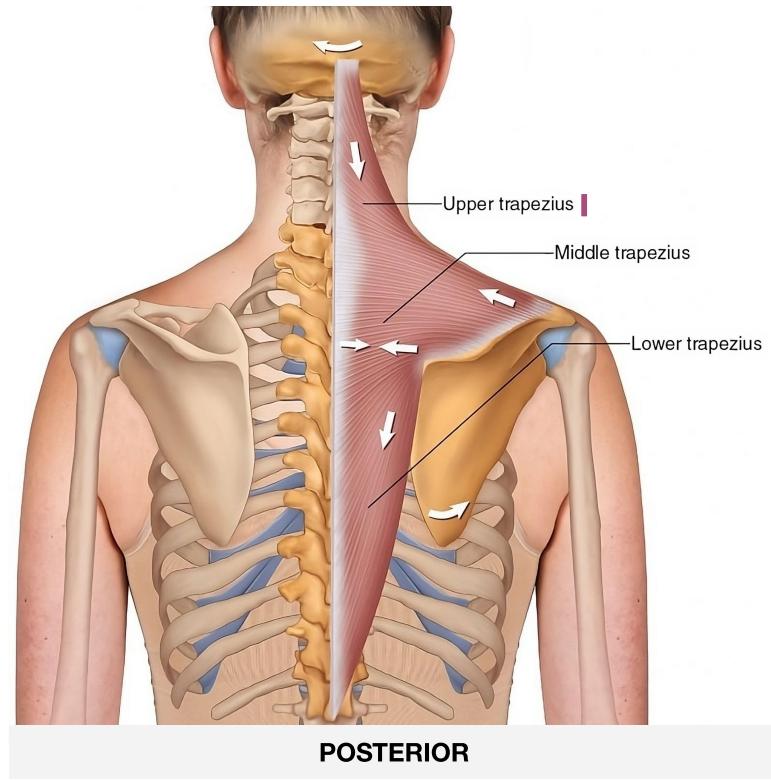


Figure 1.16. Upper, middle and lower trapezius.¹²

¹²Image source: adapted from [1]

1.2 Brain and muscle signal acquisition

In this work, in addition to brain activity, also muscle activity is acquired. This section describes how to record such activities using the most common techniques.

1.2.1 Scalp electroencephalography (EEG)

The main objective in neuroengineering is to study brain activity by acquiring the signal produced by the brain. Unfortunately, since the cerebral cortex is located inside the scalp and due to its folded structure made by gyri and sulci, signal acquisition from it is not simple [13] [103] [42].

Activities in the brain are generated when a neuron transfers a message, namely an action potential spike, to other neurons through a synapse. This signal exchange happens by ions movements that cause a change in the electrical potential inside the neuron which is measurable outside the cell due to the ion current. This current is indeed generated by the difference of ions inside and outside the neuron which causes ions flow from high concentration areas to low concentration ones [22] [13].

Common recordings methods are not able to observe a single neuron but just a group of them, more invasive the method is, higher the spatial resolution will be, and therefore, smaller the population of neurons observable. With the term electroencephalography, several acquisition techniques are included, the most common one is *scalp electroencephalography* which is usually called EEG. In addition to EEG, two more invasive methods are *stereo-electroencephalography (S-EEG)* and *electrocorticography (ECoG)*. The former is the most invasive since the electrodes, more precisely needles, are implanted directly in the brain cortex. ECoG instead, is less invasive because it is placed intracranial without penetrating the brain, see Fig. 1.17. Differently from them, EEG is non-invasive since it is performed with electrodes placed outside the scalp and for this reason, it is the most adopted method. S-EEG and ECoG are used only on patients who required to open their skull due to a disease. For instance, S-EEG is used on epileptic patients since the areas of interest are located deeply in the brain [22] [42].

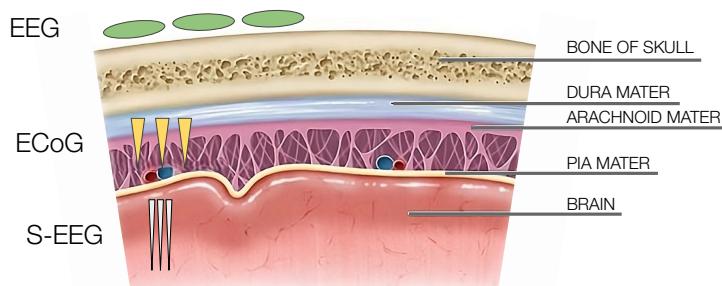


Figure 1.17. Position of electrodes of EEG, ECoG and S-EEG. EEG electrodes are placed outside the skull, ECoG electrodes between *dura mater* and *piamater*. S-EEG needles are instead positioned inside the brain tissue.¹³

¹³Image source: adapted from <https://my.clevelandclinic.org>

In order to understand how complex functionalities of the brain work, it is necessary to study the activities of more neurons at the same time, more specifically, the activities of the neurons in specific areas of the cortex. The overall activity is given by the linear summation of the activity of each neuron, this summation is not given by the combination of each action potential produced by each neuron, but it is given by the summation of the electrical potential of them. This is due to the fact that action potentials are signals with durations too short to make it possible a synchronization of them among a group of cells. Therefore, the actual signal measured is not the actual activity of the observed neural population but just a signal correlated to it [13]. The electrical potential of the observed neural population, is propagated thanks to the volume conduction provided by the tissues in the skull which contain water, so ion currents can spread instantaneously up to the scalp. Unfortunately, during the conduction, the signal is spread and attenuated with a reduction of amplitude from order of mV to order of μV [13].

In an EEG, the signals are recorded by placing metal electrodes on the scalp using a headset Fig. 1.18. The electrodes are mostly made of German silver which is an alloy of copper, nickel, and zinc. This specific metal is necessary to convert efficiently ion current to electron current which is the one used in electronic devices. The conduction is also improved using an electrolyte gel or paste applied between the electrodes and the scalp Fig. 1.19 [22].



Figure 1.18. EEG cap.¹⁴



Figure 1.19. Close up to EEG electrodes and gel application.¹⁴

¹⁴Image source: https://www.westernsydney.edu.au/marcs/labs/psychophysiology_lab

1.2.1.1 Electrode system

The placement of the electrodes on the scalp follows specific patterns called electrode systems, the two common ones are: *10-20 system* and *10-10 system*. In Fig. 1.20 10-20 system is shown, the name comes from the subdivision of the line above the scalp from the *inion* up to the *nasion*. The first and the last row of electrodes are indeed placed at 10% of the total distance, while the other ones at 20%. Fig. 1.21 instead, shows 10-10 system where electrodes have a higher density. The figure also shows the relationship between the electrodes and the brain lobes.

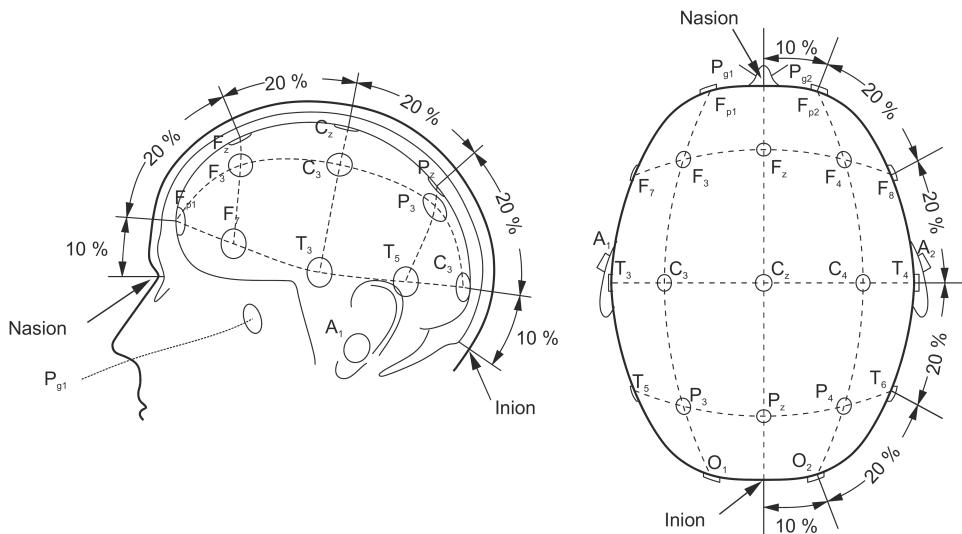


Figure 1.20. EEG 10-20 electrode system. ¹⁵

Beside the mapping between electrodes and brain lobes, a more useful and important mapping is the one that puts in relationships the electrodes with the Brodmann areas Table 1.1. This mapping was built experimentally in [92]. It is important to highlight that a single electrode is not able to acquire the activity of just one brain area due to the so-called *spatial blur effect* (Section 1.2.1.2). Therefore, this mapping cannot exactly indicate the brain area source of the signal recorded by an electrode, it is just an indicative mapping useful to get oriented during data analysis.

¹⁵Image source: [67]

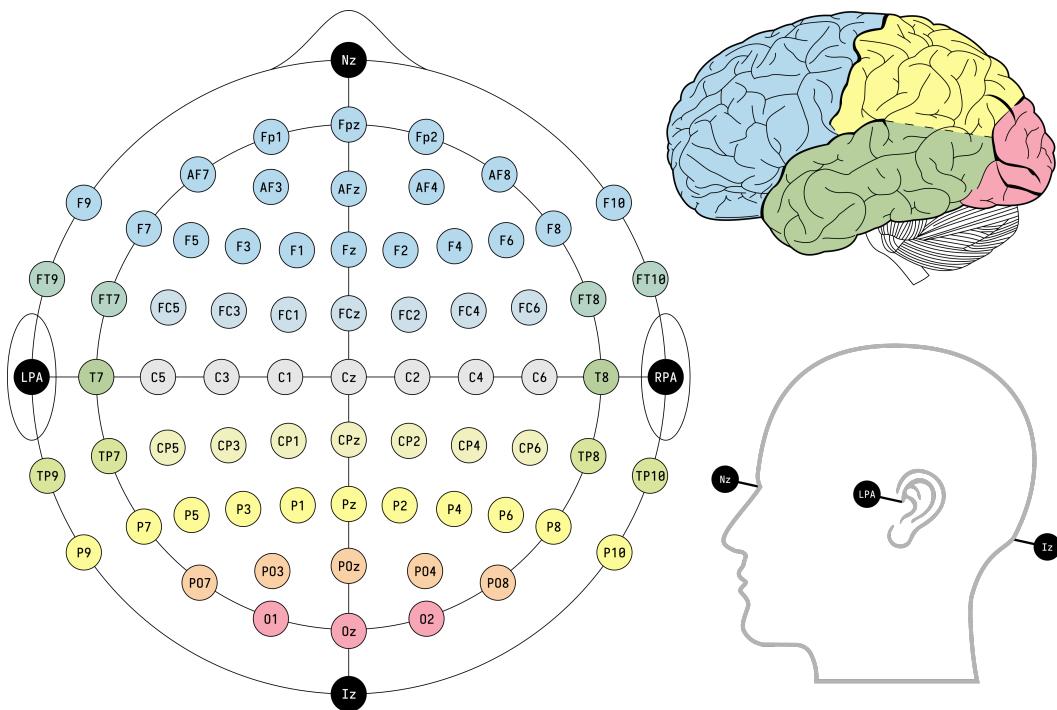


Figure 1.21. EEG 10-10 electrode system.¹⁶

Table 1.1. Brodmann area and respective EEG Electrodes of 10-10 system. The column Elec. contains electrodes names while BA the corresponding Brodmann areas of the left (Lxx) or the right (Rxx) hemisphere. Experimentally built in [92].

Elec. BA	Elec. BA	Elec. BA	Elec. BA
AF3 L09	CPZ L07	FP1 L10	P7 L19
AF4 R09	CZ R04	FP2 R10	P8 R19
AF7 L10	F1 L06/L08	FT10 R21	PO3 L19
AF8 R10	F2 R06/R08	FT7 L44	PO4 R19
AFZ L09	F3 L09	FT8 R06	PO7 L18
C1 L04	F4 R09	FT9 L38	PO8 R19
C2 R06	F5 L09	FZ L06	POZ L19
C3 L01	F6 R09	O1 L18	PZ L07
C4 R01	F7 L45	O2 R18	T7 L21
C5 L40	F8 R45	FC1 L06	T8 R22
C6 R40	FC2 R06	FC3 L06	TP7 L21
CP1 L07	FC4 R06	FC5 L06	TP8 R37
CP2 R07	FC6 R06	FCZ L06	TP9 -
CP3 L40			TP10 -
CP4 R07			
CP5 L39			
CP6 R39			

¹⁶Image source: <https://commons.wikimedia.org/w/index.php?curid=96859272>

1.2.1.2 Spatial filtering

EEG has several important advantages such as being noninvasive, portable, relatively inexpensive and easy to use. Moreover, it is scalable, namely with an arbitrary number of electrodes according to the application, and it has an excellent temporal resolution. With this last, we refer to the negligible delay from the moment of a brain activity to occur and the time when its correlated signal is measured. Unfortunately, this is not true when it comes to spatial resolution. Since the electrodes are placed on the scalp and outside the brain, the electrical signals received on a single electrode come from different sources, namely from different neuron populations and therefore from different brain areas. This effect is due to the nature of volume conduction which causes the so-called *spatial blur*. The signal coming from a brain region is spread and attenuated over a larger region of the scalp which is then collected by different electrodes at the same time. More a brain area is in depth, more the spatial blur is broad. Fig. 1.22 shows this effect. Electrical potential from superficial areas is less spread with respect to other areas such as the subcortical ones. Notice how the activity of a single brain area is measured by multiple electrodes [42].

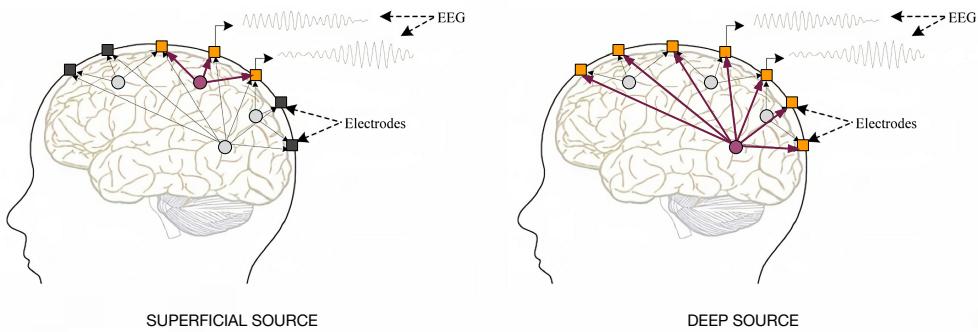


Figure 1.22. Spatial blur effect. More the source signal is in depth, broader is the diffusion.¹⁷

Besides spatial blur, other drawbacks of EEG are the low signal-to-noise ratio and the limited capability of acquiring signals from neurons located in sulci and *fissures* (i.e., deeper sulci), therefore most of the recording activity comes from gyri. This limitation is due to the orientation of the neurons in sulci which cause a mutual elimination of the electrical potential generated by the cells. There are no solutions for this last limitation, however the signal-to-noise ratio can be improved by removing noise and artifacts as explained in the following section (Section 1.2.1.3).

To limit the spatial blur and also to increase the signal-to-noise ratio, *spatial filtering* of EEG electrodes signal is widely used in literature as preprocessing step during signal analysis. During an EEG recording, the minimum number of electrodes required are three: the *active electrode* over the region of interest, the *reference electrode* and the *ground electrode*. The choice of this last is not particularly of interest, the reference electrode is very crucial instead. Usually, during data analysis, the mean value acquired from the electrodes located on the back of the ears (*mastoid*)

¹⁷Image source: adapted from [94]

is used as reference. Then during data analysis, all the signals from the electrodes of interest are translated over new reference electrodes. This translation is commonly referred as spatial filtering [42] [63] [100]. The most common techniques are *Laplacian filtering* and *Common Average Reference (CAR)*. In Laplacian filtering, the reference used is the mean of the four surrounding electrodes, in CAR instead, the average value of the entire electrode montage is used. There exist two variants of Laplacian filtering: *large Laplacian* and *small Laplacian*. The difference is shown in Fig. 1.23 [63].

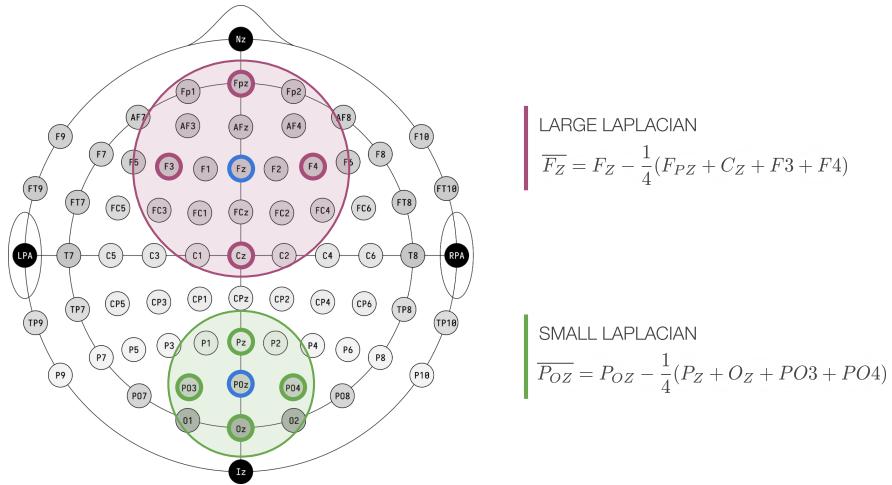


Figure 1.23. Large and small Laplacian filtering techniques.¹⁸

1.2.1.3 Artifacts

An artifact is an undesired signal that contaminates an EEG recording. The nature of such disturbance can be physiological (*biological artifact*) or instrumental (*nonbiological artifact*). Prevention of them is always preferable over removing or compensating after data acquisition, therefore well-known situations causing such artifacts must be avoided as much as possible. If possible then, remove the measurements containing artifacts and when this is not possible, remove the remaining artifacts in post-processing [42].

Physiological artifacts come from sources such as eyes, heart and muscles. These artifacts have typically higher amplitude by several orders of magnitude with respect to EEG signal. Moreover, they also may have overlapping frequency content. Ocular artifacts arise because the eye is an electrical dipole, during eye movements and eye blinks those artifacts are generated. To prevent this type of artifacts, it is useful to ask the subject to fixate on a designated point, such as a cross on the screen, and to reduce eye blinks as much as possible. A multitude of muscles are located on the face, head and neck. Their contractions are most likely to contaminate the measurements since muscle contraction is regulated by electrical impulses which can interfere with the EEG. In addition to ask the subject to avoid muscles contractions,

¹⁸Image source: adapted from <https://commons.wikimedia.org/w/index.php?curid=96859272>

it is important to make the signal acquisition setup comfortable as much as possible for the participant. Heart artifacts are caused by the strong electrical potentials generated by heart contractions. This type of artifacts, having a well-known and predictable trend, can be easily detected and removed automatically without strong supervision. Other types of physiological artifacts are due to subject movements and sweating which cause bad electrodes connections [42].

The most typical non-physiological artifacts are generated by the surrounding environment and the equipment used during measurements. A noticeable and very common artifact is power-line noise coming from electrical wall outlets. This noise is easily removable using a notch filter at 50Hz or 60Hz according to the country. Finally, different types of artifacts can arise from improperly applied or damaged EEG electrodes [42].

1.2.1.4 Brain rhythms

The first rudimentary EEG was performed in 1929 by Hans Berger, a German psychiatrist who obtained the first measure of the electrical activity on the human scalp. During data recording over occipital lobe, he noticed a large and persistent 10Hz rhythm (oscillation) that he named *alpha*. The alpha rhythm was affected by eye-opening, it was low when the subject kept his eyes open and high when his eyes were close. Fig. 1.24 shows a drawn EEG track with alpha rhythm. Years later, the same discovery was made by Lord Adrian, a British neuroscientist who proposed for the first time an idling hypothesis for brain rhythms. Idling means that a system during resting condition (close eyes) is kept warm so that it can quickly back to work when it is needed [42].

The most known and widely applied brain rhythms are alpha rhythm over the occipital lobe and the *mu rhythm* over the central sulcus. This last was described for the first time in 1952 by Henri Gestaut who noticed its reactivity to execution of motor activities in the same way alpha rhythm reacts with open or close eyes. Mu rhythm has two main frequency components one around 10Hz and another around 20Hz, this two frequencies nature can be seen by its arched shape in Fig. 1.24.

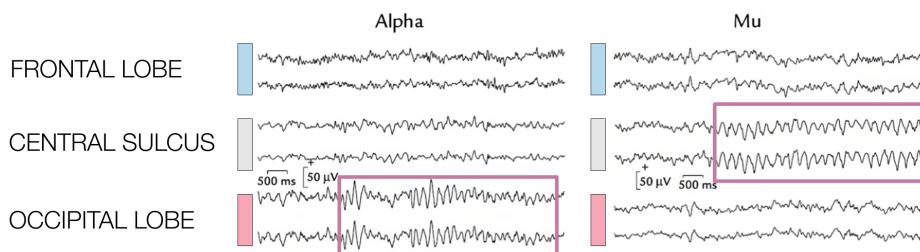


Figure 1.24. Mu and alpha rhythms are highlighted in the EEG recording. Mu rhythm is more visible over the central sulcus and occurs when the subject is at rest. Alpha rhythm instead, is visible on the occipital lobe when the subject keeps close his/her eyes.¹⁹

¹⁹Image source: adapted from [42]

When the usage of EEG rhythms in studies increased in the 1930s and 1940s, the scientific community started to characterize the rhythms in terms of their dominant frequency instead of their functions. With some changes over the years, the same frequency bands introduced at the time, are still used nowadays and are reported in Table 1.2.

Table 1.2. EEG frequency bands.

Name	Band
Delta band	< 3.5 Hz
Theta band	4 - 7.5 Hz
Alpha band	8 - 12 Hz
Beta band	13 - 30 Hz
Gamma band	> 30 Hz

It is important to highlight that the names provided in the table refer only to specific frequencies bands and not to the functionality of the rhythm. Indeed, oscillations in the Alpha band will be called alpha or mu rhythm depending on whenever they are generated by the visual or primary motor cortex [42].

1.2.2 Electromyography (EMG)

Electromyography (EMG) is a technique for recording biomedical electrical signals obtained from the neuromuscular activities. In the same way EEG measures the brain activity, EMG measures muscle activity principally from skeletal muscles. More specifically, when a motor unit is activated, the corresponding muscle fibers generate and conduct their own electrical impulses that cause the actual contraction of the fibers. The electrical impulse generated and conducted by a single fiber is very weak, however the combination of several fibers allows to produce an electric impulse strong enough to induce a voltage difference detectable on the surface of the skin [41] [2]. Fig. 1.25 shows the raw EMG corresponding to nine consecutive contractions of a muscle. Notice how an important aspect of the analysis of an EMG signal is the detection of the onset and the offset of each contraction.

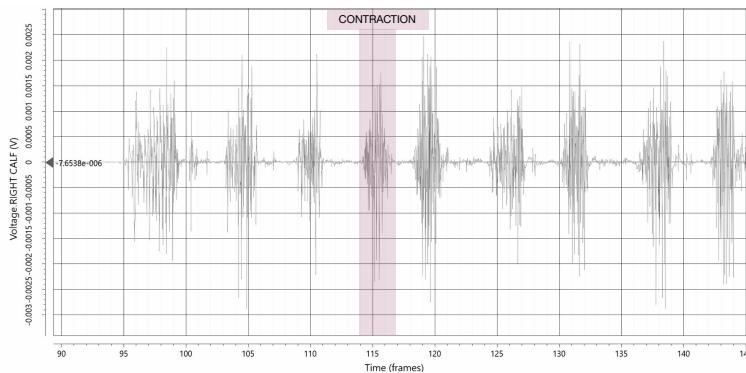


Figure 1.25. Raw EMG signal containing nine muscle contractions.²⁰

²⁰Image source: adapted from <https://logemas.com>

There are two main types of electrodes used to perform EMG measurement: *needle electrode* and *surface electrode*. The former one is invasive since it must be inserted into the muscle and therefore under the skin. The second instead, is non-invasive and is applied over the skin region corresponding to the muscle of interest [41].

Beside the types of electrodes used, the measurement of the activity of a muscle can be performed into two different ways: *unipolar* and *bipolar*. In the same way of EEG recordings, to measure the activity of a single muscle three electrodes are necessary: *active electrode*, *reference electrode* and *ground electrode*. In unipolar measurement, only the active electrode is placed over the muscle, and a point of the body with potential close to zero is used as reference. In bipolar measurement instead, both the active and the reference electrode are placed over the same muscle with an appropriate distance from each other. In both techniques, the ground electrode is placed over the subject body faraway from the muscle of interest [4]. Fig. 1.26 shows an example of the placement of three EMG bipolar electrodes over three different muscles of the right arm.

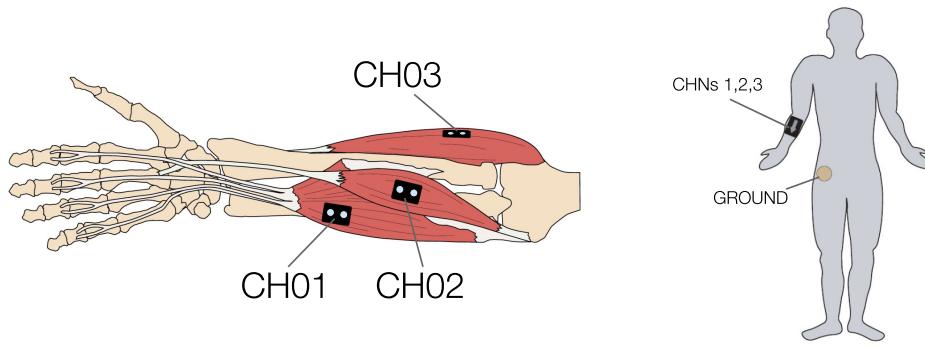


Figure 1.26. Example positions on the right arm of three bipolar electrodes to acquire the activity from three different muscles. The electrodes are superficial, but the drawing shows the muscles and the skeleton to help orientation. In the figure, the ground electrode is placed on the hip, but other locations can be chosen.²¹

1.3 Multivariate analysis of biomedical signals

During the analysis of biomedical signals, two different approaches can be followed:

- **Univariate analysis:** Each measured signal is studied independently from the others;
- **Multivariate analysis:** In addition to the study of each signal singularly, also the interdependency between them is analyzed.

In complex systems such as the nervous system, multivariate analysis is fundamental to study the overall function since analyze the interaction between each component

²¹Image source: adapted from [4]

is crucial. Considering for the moment only the brain, in order to understand its function and develop clinical and industrial applications, it is very useful to build a model of the brain circuit using brain activity information and brain connectivity. There are three different types of brain connectivity:

- **Anatomical connectivity:** Physical connection linking sets of neurons or brain areas. This kind of connectivity is relatively stable at long timescales of hours and days;
- **Functional connectivity:** It represents how much the functions of two brain regions are related between them;
- **Effective connectivity:** It refers explicitly to the influence that one neural system exerts over another.

To understand better the difference between functional and effective connectivity, let consider two brain regions A and B. Observing their activities, functional connectivity answers to the question "*Are functions of A and B similar or related?*". Effective connectivity instead answers to "*Does the activity on region A cause activity in region B?*" [12].

The focus of the present work is on functional connectivity not only related to just the brain but to the entire nervous system, in particular the interaction between brain areas and muscles.

EEG signal is produced by the synchronization between several neurons. Given the excellent temporal resolution of EEG measurements, the synchronization of different brain regions is observable in the recording and is visible in form of brain rhythms as shown earlier in this chapter. Functional connectivity is fundamental for the study of brain rhythms. Relationships between brain regions can be quantified with different measures of functional connectivity such as *correlation* and *coherence*. They both measure how much two signals are synchronized or similar, however, correlation works in the time domain while coherence in the frequency domain which is the one of interest in this work [12].

Before coherence, a fundamental quantity called *Power Spectral Density* is introduced.

1.3.1 Power Spectral Density

Given a complex values time series $x[n]$ with sampling interval T , the Fourier transform $X(f)$ of $x[n]$ is mathematically expressed as Eq. 1.1.

$$X(f) = \sum_{n=-\infty}^{\infty} x[n]e^{-j2\pi fnT} \quad (1.1)$$

Fourier transform allows moving from the initial time-discrete signal $x[n]$ to a continuous function $X(f)$ in the frequency domain. Using this last, it is possible to define an important quantity called *Power Spectral Density (PSD)*:

$$S_{xx}(f) = |X(f)|^2 \quad (1.2)$$

In general in neuroscience, Fourier transform $X(f)$ of the data collected $x[n]$ during an EEG or EMG recording has not a physical meaning, and it is not used. In the contrary, PSD is crucial because it represents the strength of a signal at a specific frequency f which is useful during brain rhythms analysis [12].

PSD can be defined also regarding two time series. Given two complex values time series $x[n]$ and $y[n]$ with sampling interval T , the *Cross Power Spectral Density (CPSD)* is defined as:

$$S_{xy}(f) = X(f)Y^*(f) \quad (1.3)$$

where $X(f)$ is the Fourier transform of $x[n]$ and $Y^*(f)$ is the transform of the conjugate series of the complex values series $y[n]$. CPSD is symmetrical, namely:

$$S_{xy}(f) = S_{yx}(f) = Y(f)X^*(f) \quad (1.4)$$

CPSD represents the quantity of power that both signals x and y share at a frequency f [12].

1.3.2 Coherence

There are several orders of coherence, the one of interest in the present work is *ordinary coherence* namely the first order coherence. Ordinary coherence is the linear correlation between two signals at a given frequency f . It is defined as the squared magnitude of the CPSD of two time series over the product of the magnitude of their PSD. Formally, given two complex values time series $x[n]$ and $y[n]$ with sampling interval T , the ordinary coherence is defined as:

$$C_{xy}(f) = \frac{|S_{xy}(f)|^2}{|S_{xx}(f)| |S_{yy}(f)|} \quad (1.5)$$

The denominator is used to normalize and take into account the ratio of exchanged power over the total power of the signals [12]. Therefore, given a frequency f_0 :

- If $C_{xy}(f_0) = 0$: The two signals x and y are independent at frequency f_0 ;
- If $C_{xy}(f_0) = 1$: The two signals are maximally correlated at frequency f_0 .

1.3.3 Cortico-Muscular Coherence

Being the nervous system composed by both the brain and the muscular nerves, beside EEG, during the study of brain activity is desirable to include some peripheral signals. This kind of signals indeed, provides information that is otherwise difficult to be extracted from merely the EEG measurement [42]. In this work, EMG recording of the upper limb muscles is the peripheral signal acquired concurrently to EEG. Therefore, instead of studying the coherence between two brain areas, the analysis is focused on the coherence between muscles and brain areas. To do that, the *Cortico-Muscular Coherence (CMC)* is used which is the non-normalized coherence computed between each EMG channel and each EEG channel. Given an EEG signal x and an EMG signal y , the CMC, a frequency-dependent function, is defined in Eq. 1.6

$$CMC_{xy}(f) = |S_{xy}(f)|^2 \quad (1.6)$$

where $S_{xy}(f)$ is the CPSD of the two signals x and y at frequency f [30].

It is important to highlight that CMC has relevant significance also in pathological subjects, such as post-stroke patients, where alterations of the brain are reflected to muscular patterns.

1.4 Brain-Computer Interface

A Brain-Computer Interface (BCI) is a system that measures Central Nervous System (CNS) activity and converts it into artificial output that replaces, restores, enhances, supplements, or improves natural CNS output and thereby changes the ongoing interactions between the CNS and its external or internal environment [103]. Such technology provides a way to develop interaction between a brain and a computer. This interaction is achieved by using control signals arising due to the brain activity. Applications of BCI technology base their functionalities on either, allowing a computer to "observe" the user mental state, or allowing the user to send an intention to complete a desired task [22] [104] [14].

The research community has initially developed BCI applications in the biomedical field, with the aim of facilitating the restoration or even replacing motor or communicative functionalities lost due to a disease or an accident [14]. Thanks to the recent advancements in neurology and engineering, BCI technology is gaining interest also in other fields outside the medical one, such entertainment and industry. In the future, BCI device can be potentially be used by also healthy people to interact with a smartphone or other devices. Moreover, for critical working environments where the workers' attention is crucial, BCI technology is applicable to monitor their mental state and detect tiredness or lack of concentration [64].

In the present work, the type of BCI used is hybrid BCI with biomedical field of interest. Hybrid BCI are characterized by the use of an additional peripheral signal in combination of the cerebral one recorded with an EEG. More specifically, the type of peripheral signal adopted is muscle activity acquired through EMG.

1.4.1 EEG signal features

An important distinction among BCI applications, is on the type of control EEG signal used to monitor the subject brain activity and more specifically, if the monitored activity has been generated by an internal or an external stimulus. When a BCI technique uses an EEG signal generated independently of external stimulation and may be fully and freely regulated by the user, then the technique is called *endogenous*. Contrary, the BCI application is called *exogenous* when the EEG signal used is produced in response to external inputs such as visual or auditory cues. An example of such type of signal is P300 which occurs approximately 300 milliseconds after the subject notices the external stimulus.

In this work, the interest is on endogenous BCI applications which monitor the *Sensory Motor Rhythms (SMR)*. This type of brain rhythm can be observed on an EEG signals acquired over the sensorimotor cortex and is modulated by actual movements, motor intentions or motor imagery performed by the subject. The modulation of SMR follows the homunculus mapping making possible to distinguish what muscle or part of the body is involved. The term *motor imagery* refers to when

the user just imagines moving one of his/her limbs without actually performing any movement in the environment. The most important aspect, is that SMR is detectable in healthy individuals as well as individuals disabled by neuromuscular diseases or injuries including spinal cord injury and stroke. Therefore, for instance the SMR can be used to develop a BCI device that allows to regain mobility to a person in a wheelchair by controlling an auxiliary mobility system such as an exoskeleton [106].

The application of interest of this article is on post-stroke rehabilitation.

1.4.2 Post-stroke rehabilitation

In the present work, the biomedical field is the one of interest, in particular assist the rehabilitation process of patients who lost the mobility of their upper limbs after stroke [64]. *Stroke* is a condition in which brain cells suddenly die because of the lack of oxygen, this can be caused by an obstruction in the blood flow. The death of brain cells can cause memory problems, loss of limb control or even the death. Rehabilitation training is critical for stroke patients, BCI technologies can improve and speed up the recovery period [70]. An example of such use is provided in [79] where the BCI approach used for post rehabilitation is based on motor imagery. The patient is asked to imagine or attempt a movement on the injured limb. If the patient attempts the movement by controlling the right brain areas and muscles, then positive feedback is given by moving a virtual limb on the screen. Otherwise, the wrong movement attempt is reported. The immediate feedback provided to the patient is crucial and speed up the recovery process helping his/her to remain motivated. Indeed, especially in the first stage of the rehabilitation process, the patient is not able to move properly or even not move at all the limb involved in the injury making frustrating the first sessions; provide a virtual visual feedback therefore is strongly helpful.

The pipeline Fig. 1.27 of the BCI for post-stroke rehabilitation of interest in this study combines both EEG and EMG signals, and it is made by four steps:

- **Signal acquisition:** EEG and EMG signals are acquired and preprocessed;
- **Feature extraction:** The signal acquired is mapped into a feature space using CMC;
- **Feature selection:** The most informative features are selected;
- **Classification:** Classify the brain activity detected into the two possible classes: **rest** or **task**. Namely, the subject is performing the asked movement, or he/she is not;
- **User feedback:** Accordingly to the classification result, feedback is provided to the user.

The focus of this work is on feature selection step which is crucial to speed up the entire pipeline and extract useful knowledge from the signal acquired.



Figure 1.27. Pipeline BCI application for post-stroke rehabilitation.

1.5 Supervised learning and feature selection

Supervised learning is a paradigm in machine learning where the learning process uses both input training examples and the corresponding desired outputs. The aim of this process is to build a function that maps new input examples to the correct output values, or in other words, built a model able to predict the outputs of new input instances.

Formally, a training dataset \mathcal{D}_{Tr} for a supervised learning task is composed by N training pair examples (x_i, y_i) , where x_i is the i -th input training example and y_i is the corresponding output also called label in classification tasks. Each input x_i belongs to the feature space X while each output y_i to the output space Y (label space in classification tasks). Therefore, given a training dataset $\mathcal{D}_{Tr} = \{(x_1, y_1) \cdots (x_N, y_N)\}$ where $x_i \in X$ and $y_i \in Y$, the aim of a learning algorithm is, using a model, build a function $f : X \rightarrow Y$ able to map correctly new inputs to their outputs. The performance of the obtained model are then measured on a different dataset called test dataset \mathcal{D}_V .

In this work, the focus is on supervised learning classification tasks where each input is a real values vector $x_i = (x_{i,1}, x_{i,2}, \cdots x_{i,n-1}, x_{i,n})^T \in \mathbb{R}^n$, in the following, the symbol $x_{*,j}$ is used to indicate the j -th feature. The output y_i instead, is an integer label representing the corresponding k -th class c_k . Assuming a binary classification task, only two classes are admissible, namely $y_i = c_k \in \{0, 1\}$. The symbols x and y without any subscript indexes, indicate the input and output variables respectively, namely generic inputs and outputs inside or outside the given datasets.

In this setting, a *feature selection task* consists in a learning algorithm seeking a selection function $S : X \rightarrow X_S$ such that $X_S \subset X$ is a subset of $s < n$ features chosen to maximize the performance of a classifier $f : X_S \rightarrow Y$ over the training dataset \mathcal{D}_{Tr} . After this process, every input $x = (x_{*,1} \cdots x_{*,n}) \in \mathbb{R}^n$ is transformed to $\bar{x} = (\bar{x}_{*,1} \cdots \bar{x}_{*,s}) \in \mathbb{R}^s$, namely a reduced vector composed just by the s selected features. Then the final performance are assessed on the test set \mathcal{D}_V .

Chapter 2

State of the art

The content of this chapter is based on the study of the reviews published in the last 12 years having as a topic the feature selection techniques applied to the motor imagery or the motor execution classification task. The search was performed on Scopus using (ALL (bci) OR ALL ("brain-computer interface") AND ALL ("motor imagery") OR ALL ("motor execution") AND ALL ("feature selection")) AND PUBYEAR > 2012 AND PUBYEAR < 2025 AND (LIMIT-TO (DOC-TYPE , "re")) AND (LIMIT-TO (LANGUAGE , "English")) as research string obtaining 144 reviews. Sorting them starting by the most cited ones, 22 reviews ([61], [53], [93], [91], [106], [14], [15], [18], [22], [54], [60], [64], [75], [82], [85], [87], [88], [98], [104], [70], [74], [96]) were then selected by ignoring the ones focused on deep learning approaches or that were too generic without any details on the actual feature extraction or feature selection techniques used. In the following, the most common feature extraction and feature selection techniques used in Motor Imagery and Execution BCI (MI and ME BCI) protocols are introduced.

2.1 Feature extraction

Feature extraction is an essential step in any machine learning application since it is the phase where the critical information of data sources is extracted. In the MI and ME BCI fields, feature extraction is more formally the mapping of large EEG data into a feature space. To obtain high performance during the final classification step, the resulting feature space must contain all the necessary discriminative information between the different classes of interest like distinguishing moments where the subject rests from the other ones where he/she imagines/executes a movement [93] [60].

In MI and ME BCI, the features extraction methods can be divided into four main categories [93]: *temporal methods*, *spectral methods*, *time-frequency methods* and *spatial methods*.

Temporal methods

During a motor imagery/execution task, sensorimotor rhythm varies making the EEG a non-stationary signal whose amplitude, phase, and frequency changes accordingly. Temporal methods for feature extraction work on each EEG channel individually

and extract temporal information related to the task with the aim of investigating how the sensorimotor rhythm modulation changes as a function of time. Examples of such methods are the ones that measure quantities like *entropy*, *mean* and *Root Mean Square (RMS)* or use statistical features like *variance* [93].

Spectral methods

Similar approaches used by temporal methods are implemented in the frequency domain by spectral methods. Spectral information is extracted by measuring the power of EEG signals in specific frequency bands such as Alpha, Beta and Gamma bands or more generally through the Power Spectral Density that describes how the power of the signal is distributed over frequencies [93].

Time-frequency methods

Temporal and spectral information can be extracted at the same time, as done by time-frequency methods such as *Short Term Fourier Transform (STFT)* that segments the signal into overlapping time frames where *Fast Fourier Transform (FFT)* is applied. Another approach is *Wavelet Transform* which decomposes the signal into wavelets, namely finite harmonic $\sin()$, $\cos()$ functions, that describe the signal in the time-frequency domain [93].

Spatial methods

Differently from the previous methods, spatial methods extract information from multiple channels at the same time combining them to generate the features. The most widely method used is *Common Spatial Pattern (CSP)* which is a supervised learning technique based on spatial filters that maximize the discriminability between classes [91] [93].

All the previous categories can be combined to generate more complex feature extraction methods. For instance, *Riemannian Manifold-based algorithms* manipulate both time and space [93], while Cortico-Muscular Coherence (CMC) [78] [34] [33] [30] combines spectral and spatial domains composed of both EEG and EMG channels.

2.2 Feature selection

After the feature extraction phase where the data has been translated into a feature space, the feature selection step can be applied to reduce the dimensionality of the resulting dataset with several benefits [61]:

- **Remove redundant information:** Some features extracted from the data might be dependent on each other and therefore using both of them is unnecessary and redundant;
- **Knowledge extraction:** Selecting only a reduced number of features allows us to observe easier what features are really useful for class discrimination. In particular in BCI, it helps to identify what features are actually related to the

mental states of interest, moreover, it can also permit to reduce the number of electrodes necessary to the recording increasing subject comfort;

- **Reduce the number of parameters for the classifier:** The number of parameters a classifier needs to optimize during the learning process is proportional to the dimensionality of the feature space of its input. Reducing the number of features allows to use a simpler classifier, namely one with lower parameters with respect to the model required without the feature selection step. This usually improves the performance of the classifier especially when the number of training examples is small with respect to the feature space dimensionality;
- **Faster prediction:** Work with a reduced amount of features usually speeds up the entire processing pipeline allowing to obtain a faster prediction. In the BCI field, this is translated to a faster feedback response to the subject.

The most important benefits in MI and ME BCI for rehabilitation are knowledge extraction and faster prediction. This last is to provide fast feedback, desirably in real time to the patient. Knowledge extraction instead, is fundamental to check whenever the patient is attempting to perform the required movement using the correct muscles which is usually not the case in the first stages of rehabilitation after a stroke.

There exist several methods for feature selection as a preprocessing step for a regression or a classification task performed by a machine learning model. In the following, we assume the final aim is a classification task. The first distinction that can be made among these methods is based on whether the importance of a feature is evaluated individually or in combination with a selected subset of features. These two main approaches are called [28]:

- **Individual evaluation:** Also called feature ranking. A weight is associated individually to each feature according to its importance;
- **Subset evaluation:** A weight is associated to a subset of features without estimation of individual feature relevance.

Besides this distinction, feature selection methods are usually divided into three main categories according to how the methodology used to extract the most informative features interacts with the learning algorithm, namely in our case, the classifier model [28] [93]:

- **Filter methods:** Their objective is to rank the features according to their informative importance and select the best ones. The estimated importance is usually based on statistical measures and does not rely on any learning algorithm [93] [60];
- **Wrapper methods:** This category of techniques relies on searching for the best subset of features using as metric the performance obtained by an external learning model [61];

- **Embedded methods:** Feature selection is performed during the process of training of a given learning model [28].

Filter methods have as their main advantages low computational cost, fast execution, and good generalization capabilities. This category of feature selection relies on individual evaluation of the features causing the incapability of filter methods of removing redundant features. Indeed, in a ranking approach, redundant features are likely to have similar ranking scores. This issue is handled by wrapper and embedded methods which use subset feature evaluation. Wrapper methods find more optimal feature subsets compared to filter approaches but with a considerably higher computational cost since for each candidate subset of features, a classifier must be trained. This makes them not suitable for datasets with high dimensionality. The other disadvantages of wrapper techniques are the high chance of overfitting and the dependency on the features selected on the classifier used. This dependency might be a problem if the classifier used during the feature selection step is different with respect to the one used in the final classification since the performance will be probably worse. This disadvantage is shared with embedded methods which however have as improvement a lower computational cost compared to wrapper approaches [28] [93].

In the following, for each of the three main categories, a resume of the most used techniques is provided.

2.2.1 Filter methods

In Table 2.1, all the filter methods mentioned in the studied reviews are shown. For each of them, the review where it is mentioned is reported together with the cited articles containing the studies that use that approach to solve a MI and ME BCI task. In some reviews, the cited paper is not related to the BCI field even though the review specifies that the mentioned method is used in MI and ME BCI or BCI literature. These cases are indicated with two asterisk symbols as described in the Table 2.1 caption.

The most cited filter methods in the studied reviews are *Mutual Information*, *Principal Component Analysis (PCA)* and *Correlation based Feature Selection (CFS)*. Mutual Information method measures the mutual dependency between the j -th feature and the class variable, higher dependency indicates high feature importance [93]. PCA is a well-known dimensionality reduction technique able to project the data onto a lower-dimensional space [40]. CFS method chooses the subset of features that are highly correlated to the class variable and not correlated with each other [93]. *Minimum Redundancy Maximum Relevance (MRMR)* [11] is a method that uses mutual information with the additional step of minimizing the redundancy of the chosen features [11]. *Coefficient of determination (R^2)* is a statistical quantity that ranges from 0 to 1 and measures the proportion of the variation in the dependent variable that is predictable from the independent one. *T-test* is the well-known statistical test used to determine if a statistically significant difference between two populations exists. It is used as a feature selection method by comparing the values assumed by a feature in the two different classes of interest. The features with

Table 2.1. Summary of the most used filter methods in MI and ME BCI applications.

Review column contains the review where the method is cited. Article column instead, indicates the papers cited by the review. Citations with two asterisks in Article refer to works not related to the BCI field but to different fields.

Feature Selection Method	Review	Article
Mutual Information	[93]	[25]**
	[61]	[17]
	[53]	[51]
Principal Component Analysis	[60]	[49]
	[54]	[24]
	[82]	[105]
	[53]	[44]
Correlation based Feature Selection	[93]	[84]
	[61]	[57]
Minimum Redundancy Maximum Relevance	[61]	[62]
	[60]	[36]
Coefficient of Determination (R^2)	[61]	[65] [102]
t-test	[93]	[46]
F-score	[93]	[52]**
Information Gain Ranking	[61]	[57]
Relief	[61]	[57]
Consistency based	[61]	[57]
Dempster-Shafer Theory based	[91]	[50]
Discriminant Filter Bank	[60]	[73]
Sparse Filter Bank	[60]	[73]

higher significant differences between the two classes are the most discriminative and therefore the more informative for a classification task [93]. *F-score* is a similar method that measures the discriminative capability of each feature [93]. *Information Gain Ranking* is a method similar to mutual information where the features are ranked according to the amount of information gained on the class label when their values are given. A feature has a high score if knowing its value allows it to easily predict the class of the given input. *Relief algorithm* evaluates each feature according to value differences between nearest neighbors data points. If two neighbor data points belonging to the same class have a big value difference concerning a feature, then the score associated with that feature is decreased. Contrary, the score is increased when two neighbors belong to the same class and the value difference of the feature is small, namely the two inputs are spatially close. *Consistency based approaches* [55] are another class of methods that evaluate how much the value of a feature is consistent in input samples belonging to the same class, namely a feature has a high score if its value is stable among data points with the same label. Finally, *Dempster-Shafer theory based*, *Discriminatin Filter Bank* and *Sparse Filter Bank* are feature selection methods specifically designed to be used in combination with CSP when this last is used as a feature extraction technique. Indeed, CSP alone has some inherent limitations in extracting optimal features from raw EEG data [91].

2.2.1.1 Mutual information

Mutual information is a statistical measure able to compute the mutual dependence and uncertainty between two random variables. This quantity and its variants are widely used to rank features' importance in the MI and ME BCI literature. In the context of a supervised classification task, mutual information can be used to perform feature selection by measuring the dependence of each feature $x_{*,j}$ and the class variable y . Higher dependence indicates a higher relevance [93].

Mutual information is strongly related to the concept of entropy which is a quantity defined in different fields. In information theory, the entropy $H(y)$ of a random variable y , is a measure of its uncertainty or in other words, how much informative that single variable is. Mutual information instead, is a measure of the amount of information one random variable contains about another. Entropy therefore provides information regarding a single variable, mutual information instead about two. It is interesting to mention that mutual information is a special case of a more general quantity called *relative entropy* that measures the distance between two probability distributions [32].

Formally, mutual information $I(y; x_{*,j})$ between two random variables y and $x_{*,j}$, is related to the entropy according to the following equation [93]:

$$I(y, x_{*,j}) = H(y) - H(y|x_{*,j}) \quad (2.1)$$

where:

- $H(y)$: Entropy of random variable y ;
- $H(y|x_{*,j})$: *Conditional entropy*. It describes the entropy of y when the variable $x_{*,j}$ is given.

From Eq. 2.1, it is more clear what mutual information measures, indeed this quantity represents the reduction in the uncertainty of y due to the knowledge of $x_{*,j}$. Mutual information cannot assume negative values and it is equal to zero if and only if the two random variables are independent Eq. 2.2 [32].

$$I(y, x_{*,j}) \geq 0 ; \quad I(y, x_{*,j}) = 0 \iff y \text{ and } x_{*,j} \text{ are independent} \quad (2.2)$$

Feature selection task aims to determine what features are more useful to predict the class y of a given input $x = (x_{*,1} \cdots x_{*,n})^T$, more a feature $x_{*,j}$ is related to the class variable y , more it gives information about y . For this reason, mutual information can be used to rank features by relevance in a filter approach, higher dependence, namely a higher value of mutual information, corresponds to higher relevance in the classification task.

An important property of mutual information, inherited by entropy, is its independence on the actual values taken by the variables $y, x_{*,j}$. Indeed, it depends only on their probability distributions [32], therefore before the computation of this quantity it is not necessary any normalization step to the dataset.

In the context of using mutual information to select the best features for a classification task in an MI and ME BCI scenario where Cortico-Muscular coherence

is used, y is a discrete random variable with discrete probability function $p(\nu)$ and each feature $x_{*,j}$ is a continuous random variable with continuous density function $\gamma(\chi_{*,j})$. By using the property in Eq. 2.3 [32]:

$$H(y|x_{*,j}) = H(x_{*,j},y) - H(x_{*,j}) \quad (2.3)$$

where $H(y, x_{*,j})$ is the *joint entropy*, the definition of mutual information in Eq. 2.1 can be rewritten removing conditional entropy obtaining Eq. 2.4:

$$I(y, x_{*,j}) = H(y) + H(x_{*,j}) - H(x_{*,j}, y) \quad (2.4)$$

Entropy and therefore mutual information can be estimated using the data collected. The dataset for supervised learning to solve a classification task is composed of a list of (x_i, y_i) examples, considering only a feature at a time we obtain a collection consisting of $(x_{i,j}, y_i)$ data points where $x_{i,j}$ is a possible value assumed by the feature $x_{*,j}$ and y_i is the relative class the sample x_i belongs to. Given the assumptions done for the two random variables $x_{*,j}$, y , and using $\mu(\chi_{*,j}, \nu)$ to indicate the underlying probability distribution of $(x_{i,j}, y_i)$ data points, the entropy of the two random variables $x_{*,j}$, y and their joint entropy are Eqs. 2.5, 2.6, 2.7 respectively.

$$H(x_{*,j}) = - \int \gamma(\chi_{*,j}) \log(\gamma(\chi_{*,j})) d\chi_{*,j} \quad (2.5)$$

$$H(y) = - \sum_{\nu} p(\nu) \log(p(\nu)) \quad (2.6)$$

$$H(x_{*,j}, y) = - \sum_{\nu} \int \mu(\chi_{*,j}, \nu) \log(\mu(\chi_{*,j}, \nu)) d\chi_{*,j} \quad (2.7)$$

Notice that the discrete probability function $p(\nu)$ of y and the continuous density $\gamma(\chi_{*,j})$ of $x_{*,j}$ can be written in term of $\mu(\chi_{*,j}, \nu)$:

$$\gamma(\chi_{*,j}) = \sum_{\nu} \mu(\chi_{*,j}, \nu) \quad (2.8)$$

$$p(\nu) = \int \mu(\chi_{*,j}, \nu) d\chi_{*,j} \quad (2.9)$$

therefore knowing $\mu(\chi_{*,j}, \nu)$, the computation of the mutual information is straightforward, however, this is usually unknown. In the case of both random variables being discrete, the probability distribution $\mu(\chi_{*,j}, \nu)$ can be estimated by counting the number of times each pair $(x_{i,j}, y_i)$ occurs in the data, this allows to easily estimate mutual information. This approach cannot be used when one of the two random variables is continuous because real-value datasets are by definition sparsely sampled therefore most real numbers will not be found, a possible naïve solution is the discretization using discrete values bins. A better approach called *nearest-neighbor method* to estimate mutual information between a discrete and a continuous random variable, was introduced in [90]. This method has a single hyperparameter k that indicates the number of neighbors to consider for each data point during the estimation, refer to the original work for more details regarding the algorithm. Authors suggest to set $k = 3$.

2.2.1.2 Principal Component Analysis

Principal Component Analysis is a dimensionality reduction technique able to project the data onto a lower-dimensional hyperplane. This hyperplane is described by the axes, namely directions, in the original feature space that contain the maximum amount of variance [40]. It is important to highlight that PCA is not able to select a subset of features from the original data, each feature in the resulting feature space is a combination of several features. To understand better this key concept, an example is illustrated in Fig. 2.1. The dataset in the original two-dimensional space is shown in the left plot, PCA reduces its dimensionality by projecting the data points to the axis A_1 , namely the principal component. The resulting dataset with a single feature $z_{*,1}$ is shown on the right plot of Fig. 2.1, this feature is a completely new feature, it is not equal to $x_{*,1}$, neither to $x_{*,2}$. In MI and ME BCI applications related to post-stroke rehabilitation, dimensionality reduction techniques are not suitable because the new features lost the physiological meanings of the original features.

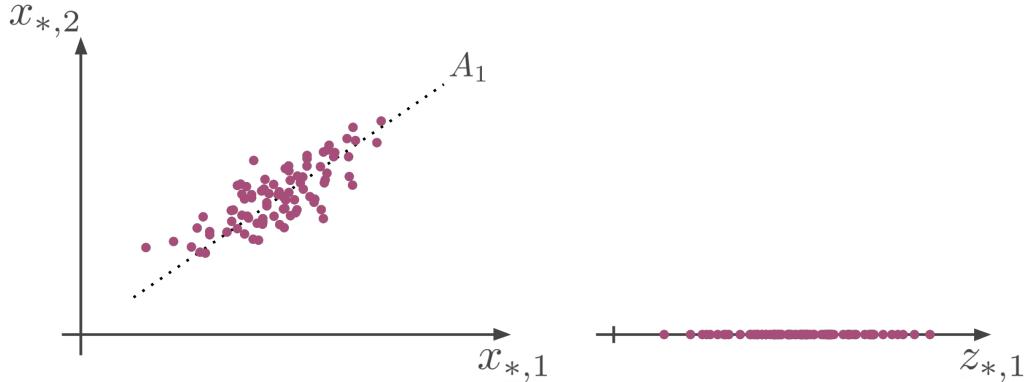


Figure 2.1. Example of dimensionality reduction performed by PCA. The original two-dimensional dataset is on the left and the reduced one-dimensional on the right. Notice how $z_{*,1}$ is a completely new feature given by the combination of both $x_{*,1}$ and $x_{*,2}$.

2.2.2 Wrapper methods

Similarly to filter methods, in Table 2.2 a summary of the most common wrapper approaches is shown and a brief description of all of them is provided below.

Mutual Information-based Rough Set Reduction is a wrapper method that uses mutual information in combination with a classifier to perform feature selection [17]. *Recursive Feature Elimination* works by progressively removing one feature at a time from the initial set until the final desired dimension is reached. At each iteration, the new subset of features is evaluated using the performance obtained by the chosen classifier, usually a Support Vector Machine model [69]. In [69], this approach is used as feature selection method with CSP as feature extraction technique.

Sequential Forward Selection (SFS) is a type of searching algorithm where the initial set of features is empty and at each iteration, a feature is added until the desired number of features is reached. The counterpart of this approach is called

Table 2.2. Summary of the most used wrapper methods in MI and ME BCI applications.

Review column contains the review where the method is cited. **Article** column instead, indicates the papers cited by the review. Three asterisks in **Article** indicate no citation was provided in the review where the method is mentioned.

Feature Selection Method	Review	Article
Mutual Information-based Rough Set Reduction (MIRSR)	[53]	[17]
Recursive Feature Elimination	[53] [61]	[69] [59]
Sequential Forward Selection	[104] [93]	[35] [58] ***
Sequential Backward Selection	[104] [93]	[35] [58] ***
Sequential Forward Floating Search	[104]	[99] [43] [66]
Sequential Backward Floating Search	[104] [93]	[99] [43] [66] ***
Evolutionary Algorithm	[96] [96] [96] [61]	[71] [56] [16] [72]
Genetic Algorithm	[54] [96] [61] [82]	[24] [86] [31] [47]
Differential Evolutionary	[93]	[23]
Particle Swarm Optimization	[93] [96] [96] [96]	[21] [19] [20] [101]
Artificial Bee Colony	[93] [82] [96] [82]	[83] [48] [83] [48]

Sequential Backward Selection (SBS) where the search proceeds in the opposite direction, the algorithm starts with the entire feature space and at each iteration a feature is removed. The main drawback of these techniques is that they are generally suboptimal and suffer from the so-called nesting effect, namely once a decision has been made, it cannot be changed after during the search. In SFS this means that an added feature cannot be removed and in SBS a removed feature will never be considered again. *Sequential Forward Floating Selection* (SFFS) and *Sequential Backward Floating Selection* (SBFS) coped with the nesting effect by adding the possibility of adding or removing the same feature during the search, therefore floating between including and excluding features. Although both SFFS and SBFS switch direction during the search, they are based on two different algorithms according to the dominant direction [80] [104]. An approach that relies on SFFS/SBFS is

Stepwise Regression where statistical quantities are used to guide the search. In the Section 2.2.2.2 this method is explained in more detail being the technique currently used at BCI lab at IRCCS Santa Lucia Foundation, Rome, Italy for the selection of the EEG control features suitable for MI-based BCI supported protocols [30].

In classic search algorithms, the key component is the heuristic function which is necessary to guide the search towards the optimal solution. This function is problem specific therefore must be designed and chosen properly. For instance, in the case of SFS/SBS or SFFS/SBFS, the heuristic must be designed to solve the problem of finding the best subset of features. A more generic and problem-independent approach is metaheuristic search where the heuristic used is suitable for a broad range of types of search and does not require the design of a heuristic function. The most common metaheuristic search strategies for feature selection in MI and ME BCI have been inspired by nature. *Evolutionary Algorithms* is a family of search algorithms that use mechanisms from biological evolution such as reproduction, mutation, gene recombination, and natural selection. This kind of algorithm is called population-based since the search progresses from one set of solutions to another, each solution is considered like an individual in a population. *Genetic Algorithm* and *Differential Evolutionary* are two types of Evolutionary Algorithms, the main difference between them is that Differential Evolutionary uses an explicit updating equation during population evolution while Genetic Algorithm does not. Moreover, Differential Evolutionary is mainly used in continuous search space even though is applicable also to discrete search spaces like in a feature selection setting [23]. *Particle Swarm Optimization* is still a metaheuristic population-based but instead of imitating biological evolution, it imitates the motion of particles in space considering quantities such as velocity, acceleration, and inertia [19]. *Artificial Bee Colony* algorithm imitates the behavior of bees in a colony during food research. It is a population-based method as the previous ones, but it divides the population into three categories, each one with a task. This class division helps the colony to reach the food and select the best food source among the ones available [23].

2.2.2.1 Genetic Algorithm

Genetic Algorithm (GA) [45] is the most known Evolutionary Algorithm used to perform metaheuristic search. The algorithm proceeds from one set of candidate solutions to another. Each solution is represented as a string, namely an array of characters or numbers, and is considered as an individual who belongs to a population. From the biological point of view, the characters are the genes that compose each chromosome (i.e., individual). At each iteration, the population produces new individuals by performing gene combinations between current individuals and random genes mutations on the offsprings. In order to maintain constant the number of individuals, at the end of each phase a natural selection step is performed with the aim of selecting the best individuals for the next generation. Each chromosome is evaluated using the so-called fitness function that returns a high score on high-quality individuals and low scores on bad solutions. High-quality solutions have a higher chance of being part of the next stage.

During the implementation of GA, the key components to specify are the method for coding problem solutions as strings and the *fitness function*. When used in

in the feature selection process, the most common fitness function is the performance obtained by a classifier that uses just the selected features. There exist instead two main approaches to encoding a feature selection solution as a string in GA:

- **Binary mask:** Each individual is represented as a binary string 0/1 with a length equal to the dimensionality of the complete feature space. If the bit in position j is 1, the corresponding feature $x_{*,j}$ is selected, otherwise it is excluded in the final features set [31] [47];
- **Explicit selection:** Each gene is an integer number that explicitly indicates the index of the feature to include in the final set. The string is as long as the desired number of features to select [86].

Assuming an initial feature space of dimensionality equal to 100 and having 2 as the number of features to select, Fig. 2.2 shows an example of both binary mask and explicit selection representation.

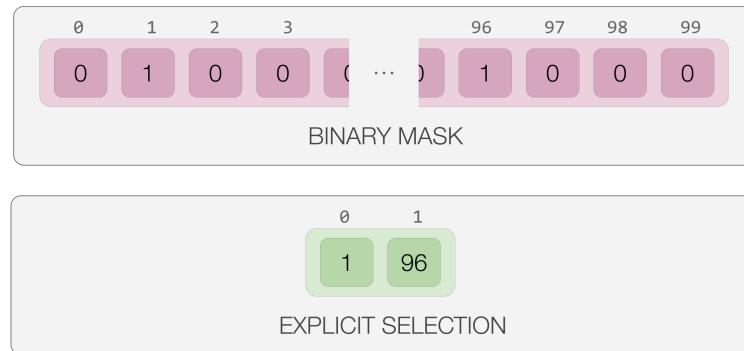


Figure 2.2. Binary and explicit encodings of an individual for the Genetic Algorithm used to select 2 features out of 100.

Binary mask approach is not suitable when applied to select a specified low number of features from a large initial feature space. The first issue arises in creating the initial population. Consider the example in Fig. 2.2 where we want to select the best 2 features from the initial 100 ones. If we create the initial population using a uniform probability distribution, then each solution selects about half of the features, namely about 50, which is not what we want. This issue is easily solvable by setting all the bits to 0 and choosing 2 indexes with uniform distribution to set to 1. The other issue in using binary mask is during offsprings generation, most of the new individuals will select too many features, and excluding them using a fitness function is not immediate since the classification performance is probably higher when using a higher number of features with respect to the desired one. This issue could be solved by adding some kind of penalties to individuals which select too many features or alternatively eliminate them directly without allowing them to survive up to the next generation. The process of manually eliminating individuals and generating new chromosomes until they satisfy the constraints is computationally expensive and goes against the idea of natural selection of GA. It

should be clear at this point that explicit selection encoding is preferred in BCI applications where the required reduction of the feature set is so large, usually higher than 95% of the initial dimensionality [86].

Genetic Algorithm with Aggressive Mutation

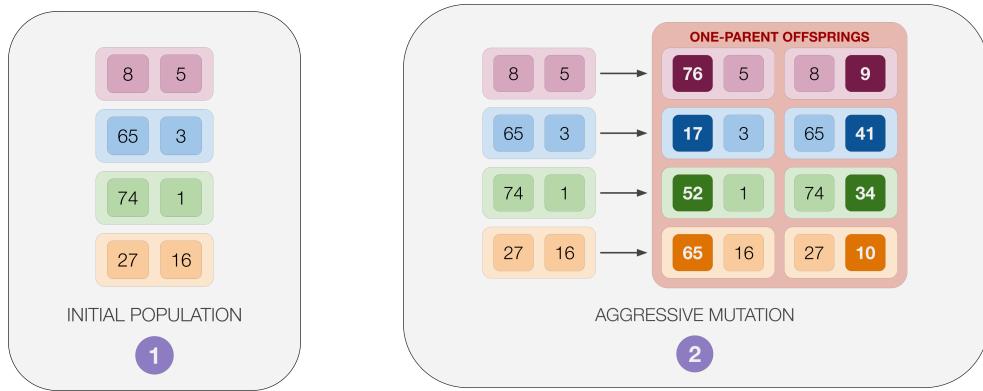
Explicit selection encoding is used in the work presented in [86] where the authors proposed a variant of GA called *Genetic Algorithm with Aggressive Mutation (GAAM)*. The algorithm is composed of the following steps:

1. Set algorithm hyperparameters:

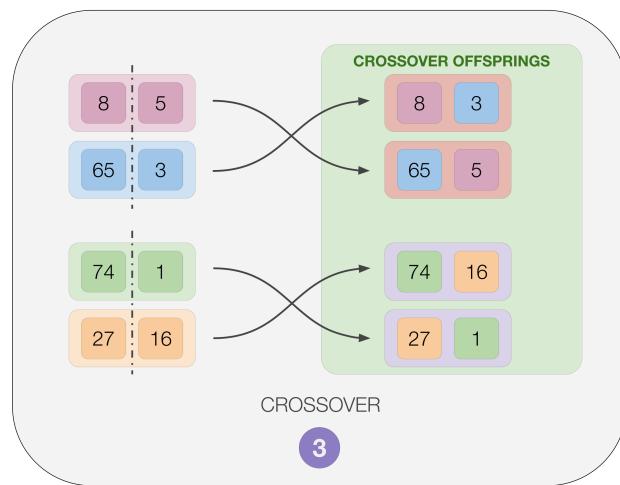
- NG = Number of Generations;
- PS = Population Size;
- LC = Length of each Chromosome (i.e., number of genes in an individual);
- FS = Initial Feature Space size.

In the example shown in Fig. 2.3: $\text{PS}=4$, $\text{LC}=2$, $\text{FS}=100$;

2. **Create initial population:** Using uniform random distribution, generates PS arrays of integers with values in $\{0, \text{FS} - 1\}$. Fig. 2.3a;
3. **Aggressive mutation:** Each individual of the previous population generate LC offsprings by performing mutations on a single gene at a time. This step generates $\text{LC} * \text{PS}$ individuals which are called *one-parent offspring*. Fig. 2.3b;
4. **Crossover:** Perform classic Holland crossover operation [45] on individuals from the old generation with a probability equal to one. The offsprings generated at this step are called *crossover offsprings*. Fig. 2.3c;
5. **Build mother population:** Adding the one-parent offsprings and the crossover offsprings to the old population, we obtain the so-called *mother population* composed by $\text{LC} * \text{PS} + \text{PS} + \text{PS} = (\text{LC} + 2) * \text{PS}$ individuals. Fig. 2.3d;
6. **Natural selection:** Apply fitness function to each individual of the mother population and discard $\text{LC} * \text{PS} + \text{PS}$ individuals with the lower score. The best PS individuals will survive to the next generation. Fig. 2.3e;
7. **Repeat** from point 3 until reached the maximum number of generations NG ;
8. **Natural selection on last generation:** The set of feature selected is obtained by selecting the best individual in the last generation.

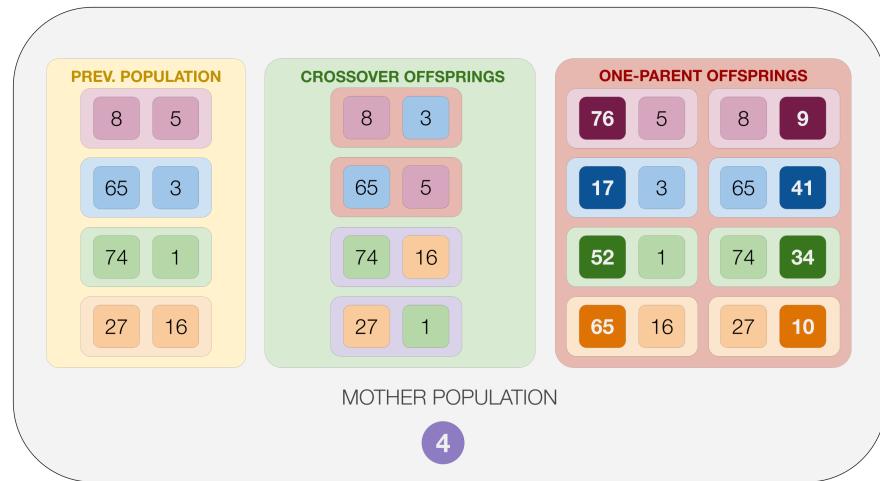


- (a) The initial population is composed by $PS=4$ individuals. Each individual has a different color to keep track of it during the algorithm execution in the figures below
- (b) Each individual generates $LC=2$ offsprings by mutating a gene at a time. The mutated gene has a darker color with white text

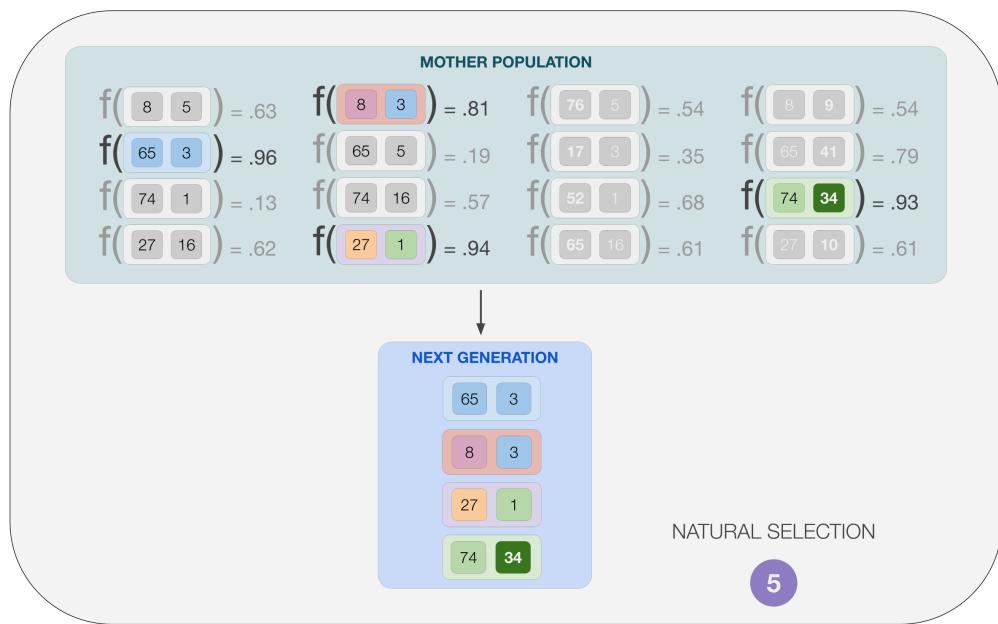


- (c) Individuals from the previous generation are combined in pairs by performing crossover operations. Two parents generate two offsprings.

Figure 2.3. Example of the first iteration of Genetic Algorithm with Aggressive genes Mutation.



- (d) Mother population is composed by the PS=4 individuals from the previous generation, PS=4 offsprings generated during crossover operation and LC*PS=2*4 offsprings from aggressive genes mutation



- (e) The best PS=4 individuals from the mother population are selected using the fitness function $f(\cdot)$. The best individuals which will be part of the next generation are colored while the others are in gray.

Figure 2.3. Example of the first iteration of Genetic Algorithm with Aggressive genes Mutation.

2.2.2.2 Stepwise Regression

In statistics, a regression model aims to estimate a linear function that describes the relationship between an independent variable x and a dependent variable y . When

the number of independent variables is more than one, this approach takes the name of multilinear regression. Assuming n independent variables $x = (x_{*,1}, \dots, x_{*,n})^T$, the model is described by the following equation [27]:

$$\hat{y} = q + a_1 x_{*,1} + \dots + a_n x_{*,n} \quad (2.10)$$

where:

- \hat{y} : Estimated value of the dependent variable y ;
- q : Intercept of the estimated linear function;
- a_j : j -th coefficient associated with the j -th independent variable $x_{*,j}$.

In the field of machine learning, the independent variables $x_{*,j}$ are the features that describe the dataset data examples and the dependent variable y is the class or the real value associated with it [40].

Stepwise Regression [37] is a feature selection method based on Sequential Floating Forward/Backward Search (SFFS/SFBS) that uses as heuristic the coefficient associated with each feature during the optimization procedure of a multilinear regression model [5]. In the following, we assume the forward direction of the search, namely the initial set of selected features is empty. At each step, the algorithm uses a statistical test to decide what features (i.e., terms) add by comparing the linear model with and without a potential term. In particular, it performs an F-statistic with the null hypothesis: *The new term $x_{*,j}$ would have a zero coefficient ($a_j = 0$) if included in the model, or in other words it would be useless.* If there is sufficient evidence to reject the null hypothesis, the term is added. Stepwise Regression method uses *least squares method* to estimate the model coefficients when a term is added or removed. The overall algorithm is composed of the following steps [5]:

1. Initial empty model and set hyperparameters:

- `P_enter` = Maximum p -value to reject null hypothesis to add a new term;
- `P_remove` = Minimum p -value to not reject the null hypothesis and then remove a current term from the model. This hyperparameter must be higher than `P_enter`;
- `max_iter` = Maximum number of terms insertions and removals. The final model will select at maximum `max_iter` features.

2. F-statistic on out-model terms: For each term not currently in the model, perform the comparison between the current model with and without that term;

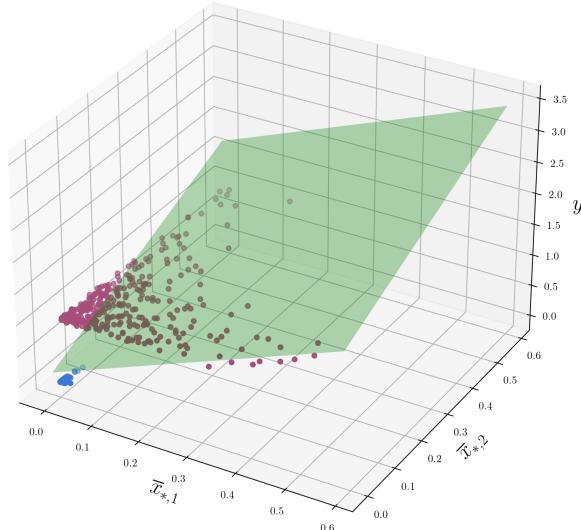
3. Add new term: Add to the current model the term with the lowest p -value lower than `P_enter`. If no candidate new term has a sufficient low p -value then go to step 5;

4. Repeat or exit: Go to step 2 or exit if reached the maximum number of iterations `max_iter`;

5. **F-statistic on in-model terms:** For each term currently in the model, perform the comparison between the current model with and without that term;
6. **Remove a term:** If any terms in the model have p -value greater than P_remove , remove the one with the largest one and go to step 2. If reached the number of maximum iterations or there is no term with sufficient high p -value then exit.

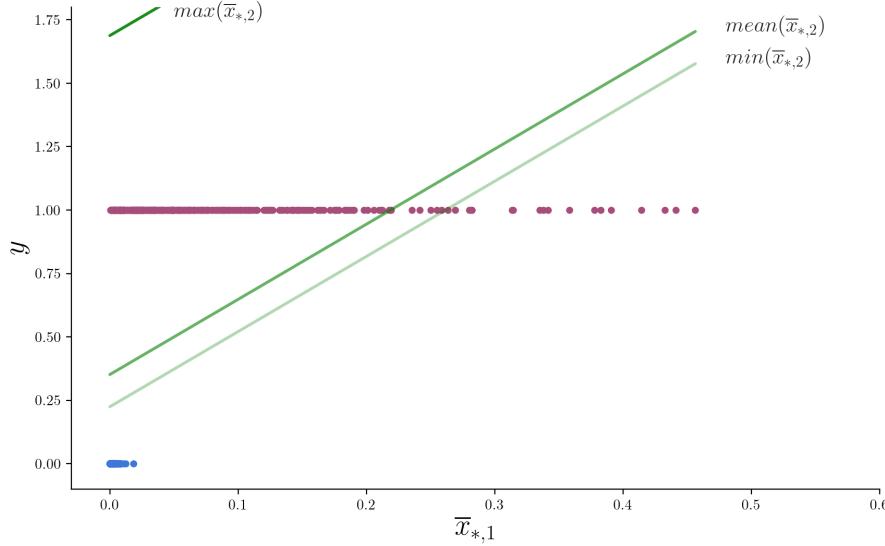
Stepwise regression can be used as feature selection technique both for classification tasks where y is an integer variable and for regression tasks where y is a real-value variable. However, the final linear model can be used in the final classification step without any changes only for regression tasks, indeed the estimated value \hat{y} from Eq. 2.10 is always a real value quantity. For binary classification tasks, the model is usable by just applying the logistic function [40]. Fig. 2.4 shows an example of the resulting multilinear model to select the two best features for a binary classification task.

Notice that stepwise regression could also be considered as an embedded method since the feature selection process relies on the training of a learning model. In this work though, we classified it as a wrapper one due to its search-oriented approach and because the final model obtained cannot be directly applicable in our final classification step.

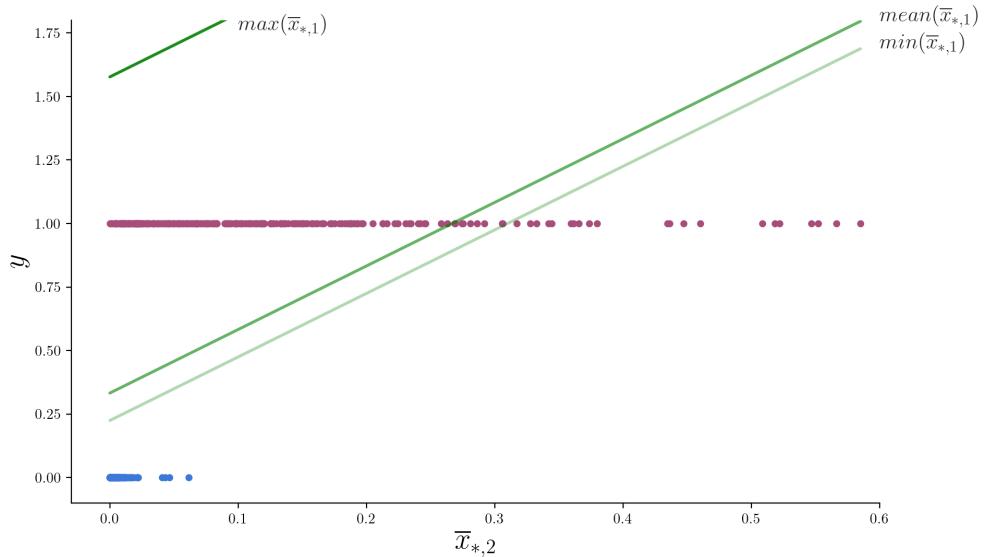


(a) Representation of the result in a three-dimensional space where the multilinear model is a plane in green

Figure 2.4. Example of the resulting multilinear model built during the selection of the best 2 features in a binary classification task. As also indicated by y axis, the data points in blue belong to the class 0 while the others in magenta to the class 1. Intercept $q = 0.22$, the weight associated to the first chosen feature $\bar{x}_{*,1}$ is $a_1 = 2.96$ while for $\bar{x}_{*,2}$ is $a_2 = 2.50$.



(b) Result considering just the first feature $\bar{x}_{*,1}$ with the other one fixed at three constant values: its minimum, medium, and maximum. The computed green hyperplane is seen as a line with this viewpoint for each chosen constant value of $\bar{x}_{*,2}$.



(c) Result considering just the second feature $\bar{x}_{*,2}$ with the other one fixed at three constant values: its minimum, medium, and maximum. The computed green hyperplane is seen as a line with this viewpoint for each chosen constant value of $\bar{x}_{*,1}$.

Figure 2.4. Example of the resulting multilinear model built during the selection of the best 2 features in a binary classification task. As also indicated by y axis, the data points in blue belong to the class 0 while the others in magenta to the class 1. Intercept $q = 0.22$, the weight associated to the first chosen feature $\bar{x}_{*,1}$ is $a_1 = 2.96$ while for $\bar{x}_{*,2}$ is $a_2 = 2.50$.

2.2.3 Embedded methods

Table 2.3 summarizes the most common embedded methods used in literature. *Linear Discriminant Analysis (LDA)* is a linear classification algorithm able to perform dimensionality reduction during the training phase. LDA is very similar to PCA but it uses labels of the data points to learn the most discriminative axes between the classes. These axes are then used to define the hyperplane with a lower dimension where to project the dataset. The benefit of this strategy is that the projection will keep classes as far apart as possible [40]. In [44], authors use a modified version called Kernel LDA that uses a kernel function to enable non-linear classification capability.

Decision Tree [29] [81] [40] is a well-known machine learning method that can perform both classification and regression tasks. The goal of the method is to automatically create a decision tree, namely a tree structure composed of simple decision rules, that predicts the value of a target variable y when given as input a features vector x . A Decision Tree model is suitable also for feature selection tasks since during the learning process it computes the importance of each feature during y prediction [40]. *Random Forest* [97] as Decision Tree can be used both to solve a regression or a classification task and to perform feature selection. The model is an ensemble of Decision Trees [40].

Table 2.3. Summary of the most used embedded methods in MI and ME BCI applications.

Review column contains the review where the method is cited. Article column instead, indicates the papers cited by the review. Citations with the asterisk in Article refer to papers containing just the description of the method without practical application on MI and ME BCI or other fields. Three asterisks instead, indicate no citation was provided in the review where the method is mentioned.

Feature Selection Method	Review	Article
Linear Discriminant Analysis	[60]	[38]*
	[54]	[24]
	[96]	[95]
Kernel Linear Discriminant Analysis (KLDA)	[53]	[44]
Decision Tree	[61]	[29]* [81]*
	[60]	[73]
Random Forest	[60]	***

Chapter 3

Materials and methods

This chapter is divided into three sections:

- **Section 3.1:** It contains the description of the signal acquisition and feature extraction process performed by IRCCS Santa Lucia Foundation, Rome, Italy. Then the datasets built in this work are introduced;
- **Section 3.2:** The feature selection methods chosen to be compared are listed, and the hyperparameter tuning steps performed for the algorithms that required it, are described;
- **Section 3.3:** The procedure used to compare the feature selection algorithms is introduced.

3.1 Datasets

The processed data used in this work was provided by IRCCS Santa Lucia Foundation, Rome, Italy. In Section 3.1.1, the signal acquisition procedure and the computation of Cortico-Muscular Coherence (CMC) features are briefly described. Finally, Section 3.1.2 describes the datasets built in this work. More precisely, the types of EEG spatial filtering and feature preselections tested.

3.1.1 Signal acquisition and feature extraction

Data were collected from 14 healthy participants. The signal acquisition was performed by recording simultaneously EEG and EMG signal with sampling rates of 1000 and 2000 Hz respectively. EEG signal acquisition was obtained using 61 active electrodes (Fp1, Fpz, Fp2, AF7, AF3, AFz, AF4, AF8, F7, F5, F3, F1, Fz, F2, F4, F6, F8, FT7, FC5, FC3, FC1, FCz, FC2, FC4, FC6, FT8, T7, C5, C3, C1, Cz, C2, C4, C6, T8, TP7, CP5, CP3, CP1, CPz, CP2, CP4, CP6, TP8, P7, P5, P3, P1, Pz, P2, P4, P6, P8, P07, P03, P0z, P04, P08, O1, O2, Oz) according to an extension of the 10-20 International System (Section 1.2.1.1). EMG was instead recorded using surface bipolar electrodes over 16 muscles, in particular 8 muscles from each upper limb side:

- **Extensor Digitorum (EXTDIG):** Fig. 1.14 in Section 1.1.2.2;
- **Flexor Digitorum superficialis (FLEXDIG):** Fig. 1.14 in Section 1.1.2.2;

- **Lateral head of the Triceps muscle (TRICEPS):** Fig. 1.15 in Section 1.1.2.2;
- **Long head of the Biceps brachii muscle (BICEPS):** Fig. 1.15 in Section 1.1.2.2;
- **Pectoralis major (PECTORALIS):** Fig. 1.11 in Section 1.1.2.2;
- **Lateral Deltoid (DEL_LAT):** Fig. 1.13 in Section 1.1.2.2;
- **Anterior Deltoid (DEL_ANT):** Fig. 1.13 in Section 1.1.2.2;
- **Upper Trapezius (TRAPEZIUS):** Fig. 1.16 in Section 1.1.2.2.

The data was collected from healthy subjects performing four different movements:

- **ExtL:** Finger extension with the left hand;
- **ExtR:** Finger extension with the right hand;
- **GraspL:** Finger grasping with the left hand;
- **GraspR:** Finger grasping with the right hand.

For finger extension movements (hand opening), **ExtL** and **ExtR**, the *target muscle*, namely the one more involved in the movement, is **EXTDIG** from the left and the right side of the body respectively. For grasping tasks (hand closing), **GraspL** and **GraspR**, instead, the target muscle is **FLEXDIG**. These two muscles are antagonistic to each other.

For each subject and for each movement activity, the recording was repeated for 40 times (i.e., 40 trials). In 20 trials the subject was at rest while in the remaining 20, the participant performed the asked movement. The trials where the subject performed the motion task are called **task** trials while the others **rest** trials. The **task** trials have a duration of 8 s and are composed by an initial preparation rest phase of 4 s, which is then followed by a visual clue that signal the subject to execute the required movement for the remaining 4 s. The **rest** trials instead, last 4 s and the participant has to stay relaxed without performing any task Fig. 3.1. During dataset preparation, the initial rest phases in **task** trials were removed by using the EMG signal acquired from the target muscle. Indeed, the EMG onset provides the exact instant when the subject starts to actually perform the movement. The signal onset is shown in Fig. 3.1 as a red dotted line. This procedure was necessary in order to remove the preparation phase, which is not of interest, from **task** trials. Therefore, for each subject and each movement, the data obtained is composed by 20 **task** trials, with a duration of about 4 s, and 20 **rest** trials with an exact duration of 4 s. The duration of **task** trials, once removed the preparation phase, slightly varies across the subjects due to their different reaction times.

For each trial, the Cortico-Muscular Coherence (CMC) (Section 1.3.3) was computed for each EEG-EMG channels pairs using a moving window of 1 s with 125 ms of overlapping. The CMC was calculated at the characteristic frequencies in Alpha, Beta and Gamma bands (Table 1.2 in Section 1.2.1.4). The characteristic

frequency is the frequency at which the CMC peak occurred in a band. More information about acquisition procedure, signal processing, artifacts removal and CMC features extraction, please refer to [33].

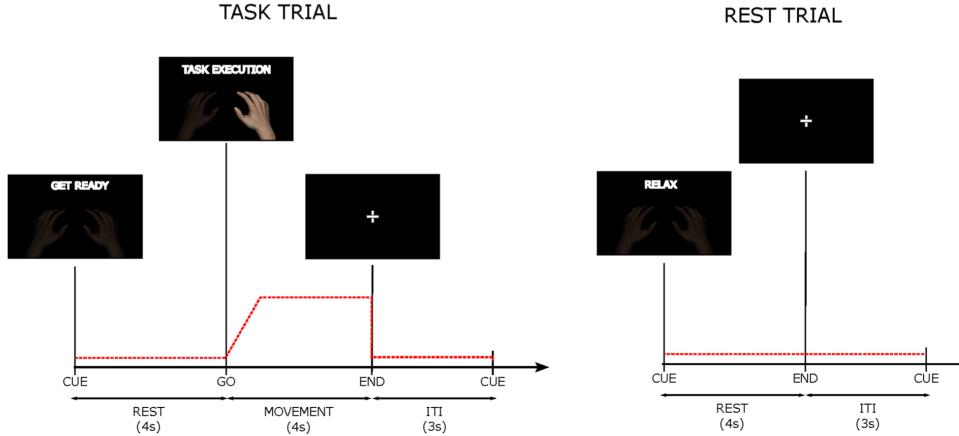


Figure 3.1. Experiment timeline for **task** and **rest** trials. The red dotted line represents the activation profile required for the target muscle (EXTDIG for Ext and FLEXDIG for Grasp). ITI stands for **I**nter-**Trial **I**nterval defined as pause period between the end of a trial and the beginning of the next one.¹**

3.1.2 Feature space reduction to account for neurophysiological constraints

In this study, the analysis was focused on single subject analysis, therefore for each subject and each movement (ExtL, ExtR, GraspL, GraspR) a dataset was built. The features space is composed by 2928 features (61 EEG chns * 16 EMG chns * 3 bands) and the number of data examples is about $1000 = (2 \text{ classes} * (25 \text{ windows} * 20 \text{ trials}))$ with a sufficient balance of **rest** and **task** classes. The number of data examples is not exact and slightly varies across the subjects due to the different reaction times of the participants during the **task** trials.

In order to study how each feature selection algorithm behaves according to the initial set of features, several preselection approaches of EEG and EMG channels were considered:

- **ALL_(EEG)-ALL_(EMG)-ALL_(BAND)**: No preselection;
- **SN_MT-<LEFT/RIGHT>-ALL_(BAND)**: EEG electrodes restricted over and nearby sensorimotor region (SN_MT: FC5, FC3, FC1, FCz, FC2, FC4, FC6, C5, C3, C1, Cz, C2, C4, C6, CP5, CP3, CP1, CPz, CP2, CP4, CP6, P5, P3, P1, Pz, P2, P4, P6). Moreover, EMG channels are restricted to the side of the body involved in the movement;
- **SN_MT-<LEFT/RIGHT>-<ALPHA/BETA/GAMMA>**: Same preselection of the previous case on EEG and EMG channels. Signal band restricted to just a single band, i.e., Alpha, Beta and Gamma bands, considered individually.

¹Image source: [78]

In addition to channels preselection, also the influence of spatial filtering on EEG electrodes was taken into account. Indeed, the feature selection methods were tested considering EEG data (before the CMC computation) not filtered (**NoFilter**) or filtered by means of the small Laplacian filter (**Laplacian**). P5, P1, P2 and P6 electrodes were excluded when spatial filtering was applied because they do not have nearby electrodes for Laplacian computation.

To summarize, for each subject and for each movement, the following eight datasets were built:

- **NoFilter-<ExtL/ExtR/GraspL/GraspR>-ALL_(EEG)-ALL_(EMG)-ALL_(BAND):**
Feature space dimensionality = $61*16*3 = 2928$;
- **NoFilter-<ExtL/ExtR/GraspL/GraspR>-SN_MT-<LEFT/RIGHT>-ALL_(BAND):**
Feature space dimensionality = $28*8*3 = 672$;
- **NoFilter-<ExtL/ExtR/GraspL/GraspR>-SN_MT-<LEFT/RIGHT>-ALPHA:**
Feature space dimensionality = $28*8*1 = 224$;
- **NoFilter-<ExtL/ExtR/GraspL/GraspR>-SN_MT-<LEFT/RIGHT>-BETA:**
Feature space dimensionality = $28*8*1 = 224$;
- **NoFilter-<ExtL/ExtR/GraspL/GraspR>-SN_MT-<LEFT/RIGHT>-GAMMA:**
Feature space dimensionality = $28*8*1 = 224$;
- **Laplacian-<ExtL/ExtR/GraspL/GraspR>-ALL_(EEG)-ALL_(EMG)-ALL_(BAND):**
Feature space dimensionality = $27*16*3 = 1296$;
- **Laplacian-<ExtL/ExtR/GraspL/GraspR>-SN_MT-<LEFT/RIGHT>-ALL_(BAND):**
Feature space dimensionality = $24*8*3 = 576$;
- **Laplacian-<ExtL/ExtR/GraspL/GraspR>-SN_MT-<LEFT/RIGHT>-ALPHA:**
Feature space dimensionality = $24*8*1 = 192$;
- **Laplacian-<ExtL/ExtR/GraspL/GraspR>-SN_MT-<LEFT/RIGHT>-BETA:**
Feature space dimensionality = $24*8*1 = 192$;
- **Laplacian-<ExtL/ExtR/GraspL/GraspR>-SN_MT-<LEFT/RIGHT>-GAMMA:**
Feature space dimensionality = $24*8*1 = 192$.

Each dataset was then randomly divided into a training set (80%) and a test set (20%).

3.2 Feature selection algorithms to compare

In Section 2.2, a summary of the most used feature selection techniques was provided. According to the literature review performed, for each type of method (*filter*, *wrapper* and *embedded*) one or more approaches were chosen. Principal Component Analysis and Linear Discriminant Analysis were excluded being dimensionality reduction techniques, namely they project the dataset into a new lower dimensional space which does not contain the original features and therefore the features lose their

physiological meanings. From filter category, Mutual Information was selected due to its simplicity and immediate interpretability of the results. Among wrapper methods instead, Evolutionary Algorithm, in particular Genetic Algorithm with Aggressive Mutation (GAAM) was chosen. GAAM was adopted in place of a standard genetic algorithm due to its explicit representation of the selected features which is preferable over binary mask selection as explained in Section 2.2.2.1. In addition to GAAM, also Stepwise Regression (a type of Sequential Forward Floating Search) was selected among the other methods in this category. Stepwise Regression is indeed the standard approach used at IRCCS Santa Lucia Foundation, Rome, Italy to perform feature selection. Finally, among embedded category, Decision Tree and a novel method called Random Planet proposed in this work, were tested. *Random Planet* is a model that combines multiple Random Forest models in order to stabilize the feature selection decision, more details are provided in Section 3.2.3.

To summarize, in this work the feature selection techniques implemented in Python language and compared are the following:

- **Decision Tree:** Algorithm implemented by the function `DecisionTreeClassifier()` provided by Scikit-learn library [77]. The default parameters were used;
- **Genetic Algorithm with Aggressive Mutation:** The algorithm designed in [86] and described in Section 2.2.2.1 was implemented in this work using some functions provided by PyGAD library [39];
- **Mutual Information:** The algorithm is based on the work [90] cited in Section 2.2.1.1. It was implemented using the function `mutual_info_classif()` provided by Scikit-learn library [77]. The default arguments were used except for `discrete_features=False`;
- **Random Planet:** Algorithm designed in this work and introduced below in Section 3.2.3. The underlying Random Forest algorithm was implemented using Scikit-learn library [77];
- **Stepwise Regression:** The method was described in Section 2.2.2.2. The algorithm was executed in Python using the MATLAB [7] function `stepwisefit()`. Indeed, from MATLAB R2023b, it is possible to execute MATLAB code in Python. The default parameters of the function were used with the search starting by an initial empty multilinear model.

Section 3.2.2 and Section 3.2.3.2 contain the hyperparameter tuning process used for GAAM and Random Planet respectively. The training set from `Laplacian-ExtL-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)` dataset was used during hyperparameter tuning, this choice was made to reduce the overall amount of time required for the procedure. This choice did not show noticeable lower performance on the other movements. The other methods do not need hyperparameter tuning. Indeed, Mutual Information does not require hyperparameter optimization by definition, for Stepwise Regression instead, the default hyperparameters are the ones used in IRCCS Santa Lucia Foundation, Rome, Italy. Finally, for Decision Tree the hyperparameter tuning was not applied because the default hyperparameters do not limit the growth

of the tree and all the features are considered during branch generation. With these conditions, Decision Tree is able to evaluate the quality of all the features. It is important to recall that this model is used only for feature selection step and not for classification where hyperparameter tuning is instead crucial.

3.2.1 Performance metrics

The metrics adopted in this work are: *Area Under Curve (AUC) of the Receiver Operating Characteristic curve*, *F1 Score*, *Soft Stability (SoftS)* and *Hard Stability (HardS)*.

3.2.1.1 AUC and F1 Score

Area Under Curve (AUC) of the Receiver Operating Characteristic curve and F1 Score are common used metrics adopted to assess classification performance. These metrics are defined by considering a binary classification task with a positive class (+1) and a negative class (-1).

The Receiver Operating Characteristic (ROC) curve plots the *Truth Positive Rate (TPR)* on y-axis against the *False Positive Rate (FPR)* on x-axis. TPR is also called *recall* or *sensitivity* and is defined as the number of positive instances that are correctly detected by the classifier (TP) over the total number of positive instances in the dataset. FPR instead, is the number of negative instances that are incorrectly classified as positive (FP) over the total number of negative instances (Eq. 3.1).

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad \text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}} \quad (3.1)$$

The area under ROC curve is indeed called AUC and provides a performance measure of the classifier. A perfect classifier has a value of AUC equal to 1, while a purely random classifier has a value equal to 0.5 [40].

F1 Score (Eq. 3.2) combines two other metrics namely recall (TPR) and *precision*. This last is defined as the ratio between TP and TP+FP. Instead of computing the mean between these two metrics, F1 Score uses the harmonic mean of precision and recall which allows to obtain a high score only when both precision and recall are high [40].

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \quad (3.2)$$

3.2.1.2 SoftS and HardS

Soft Stability (SoftS) and Hard Stability (HardS) are two metrics introduced in this work to measure how stable the feature selection decision is across several runs of the same method, using different random seeds when present, over the same exact dataset. These two quantities are computed as the ratio between the number of times a subset of feature is selected over the total number of executions. The difference between these two metrics is that, contrary to HardS, SoftS does not consider the ranking of the selected features, namely the order of importance assigned to them is ignored.

To understand better SoftS and HardS, consider the example shown in Fig. 3.2 where the results of ten executions of the same feature selection method using different random seeds, are reported. Features are indicated with uppercase letters and the ranking assigned is given by their vertical order. Fig. 3.2b contains the computation of HardS and Fig. 3.2c of SoftS instead. Notice that HardS is always lower or equal to SoftS by definition.

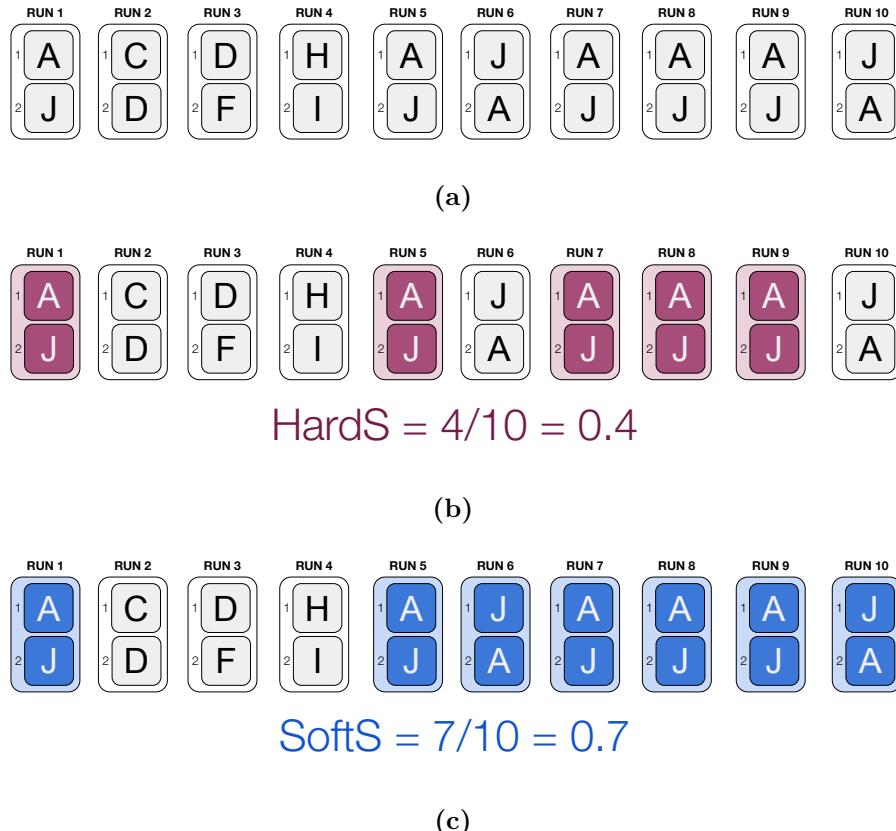


Figure 3.2. (a) Example of HardS and SoftS over ten runs of a feature selection method in charge of selecting two features using different random seeds. (b) $[A, J]$ is the most selected ordered pair of features. It is selected four times over ten. (c) The unordered pair $\{A, J\}$ is the most common and is chosen seven times.

3.2.2 Hyperparameter tuning for GAAM

The type of Evolutionary Algorithm chosen to be tested in this work is Genetic Algorithm with Aggressive Mutation (GAAM) proposed in [86]. The algorithm was explained in Section 2.2.2.1. In the following, the hyperparameter tuning procedure applied on GAAM is described.

The main components and hyperparameters of GAAM that need to be defined are [86] [45]:

- **Number of generations (NG):** Corresponds to the number of iterations of the algorithm, namely how many times the evolutionary cycle is repeated;
- **Population Size (PS):** Number of individuals that compose the population at each iteration;
- **Initial feature space size:** Defines the admissible values of individuals' digits;
- **Length of each chromosome:** Number of characters or digits used to represent an individual;
- **Fitness function:** Necessary to evaluate each individual in the current population during natural selection procedure.

In the paper [86] where GAAM was proposed, the authors tested the algorithm to perform feature selection for a BCI application with the aim of selecting 6 features from an initial set of 324 frequency band power features. Table 3.1 shows the initial feature set size, the components and the hyperparameters selected as the best for their work.

Table 3.1. Number of initial features, components and hyperparameters adopted in [86] by GAAM's authors.

Initial features set size	324
Number of generations (NG)	100
Population size (PS)	10
Length of chromosomes	6
Fitness function	accuracy linear SVM

The hyperparameter tuning procedure performed in this work used Table 3.1 as reference for choosing the values to test. The hyperparameters NG and PS influence the performance of GAAM since the convergence of the algorithm to the optimal solution depends on them. Higher the dimensionality of the search space is, higher the number of iterations (NG) and population size (PS) required will be. For hyperparameters testing, the dataset used was ExtL-Laplacian-LEFT-SN_MT_-ALL(BAND). In this case, the initial feature set is composed of 576 features, which is slightly less than double of the initial features of Table 3.1. Therefore, considering a factor of two, the possible values to test for NG and PS are 200 and 20 respectively, namely the double of the one shown in Table 3.1. In addition to them, also two other values for each hyperparameter were tested, the grid search was indeed performed for values in Table 3.2.

Notice that the dimensionality of the search space in the current settings is less than the one in Table 3.1 by several orders of magnitude, therefore considering a duplicative factor of two in the choice of NG and PS, is more than enough to guarantee similar performance to the one obtained in [86]. Indeed, the dimensionality of the search space in a feature selection task is given by the number of feature subsets of

Table 3.2. Number of generations or iterations **NG** and population size **PS**. Values tested for the hyperparameters to be optimized.

NG	[100, 200, 300]
PS	[10, 20, 40]

size k which can be obtained by an initial set of n features, this quantity corresponds to the binomial coefficient $\binom{n}{k}$. For the dataset used during hyperparameter tuning in this study, the total number of choices is 165,600 and for the work in [86] is about 1.534e12.

The number of genes for the current application was set to two and corresponds to the number of features to select. As done in [86], a linear SVM classifier was used in the present work to compute the fitness function, however, instead of accuracy metric, AUC obtained in the training set was adopted.

The hyperparameter tuning was performed by using a grid search approach, namely testing every possible combination of the two hyperparameters in Table 3.2. In the following, the term *setting* is used as a synonym of hyperparameters combination. The testing procedure was performed on all the healthy subjects by restricting the dataset as described before with a reduced amount of starting features. This choice was made to limit the overall testing time. Every setting was evaluated according to the performance on the training set obtained by a linear SVM model trained with just the two features selected by GAAM. The performance was measured using different metrics but focusing on AUC and Soft Stability (SoftS) in this order of importance. Hard Stability (HardsS) metric was excluded in the case of GAAM because this feature selection technique does not return importance weights to the selected features, therefore the ordering between them is irrelevant and random. SoftS was computed across ten runs.

The effect of hyperparameter values on performance was assessed using the Analysis of Variance (ANOVA) with repeated measures (rmANOVA) and Wilcoxon signed-rank test was used as post-hoc test. This last was applied in place of paired t-test as consequence of the violation of normality hypothesis of samples assessed by means of the Shapiro-Wilk test [89]. The total number of different settings is 9 and the number of subjects is 14. Since the number of observations (14) is not enough higher than the number of different groups (9), in order to avoid a false positive result (type I error), rmANOVA test was performed in two steps where in each of them only a hyperparameter varies and the other remains constant. In the first set of analyses, the number of generations (**NG**) was fixed, and just the population size (**PS**) changed. Thus, the population size was the independent variable (3 levels, i.e. 10, 20, 40) and the AUC the dependent variable. rmANOVA was performed for each level of the variable **NG**, i.e. 100, 200, 300. In the second set of analyses, the population size (**PS**) was fixed, and just the number of generations (**NG**) changed. Thus, the number of generations was the independent variable (3 levels, i.e. 100, 200, 300) and the AUC the dependent variable. The test was performed for each level of the variable **PS**, i.e. 10, 20, 40. The significance level was set to 0.05 and multiple Wilcoxon signed-rank tests were applied as post hoc tests with false discovery rate correction [26] for multiple comparisons [89]. Results are reported in Section 4.1.1.

3.2.3 Random Planet

During the initial research phases of the present work, one of the discovered limitation of Decision Tree and Random Forest was their instability in selecting the best set of features when the algorithms were executed multiple times with different random seeds, over the same exact dataset. In the case of Decision Tree, even if the chosen default hyperparameters do not constraint the tree growth significantly, a different random seed has a relevant impact on the selected features. This instability is caused by the nature of the datasets used in this work which are easily linear separable, this means that among all the features, there exist more than a set of two features with similar discriminative characteristics during the construction of the tree [3]. Random Forest instead, in addition to the easily linear separable datasets, owes its instability to the algorithm itself which indeed relies heavily on randomness.

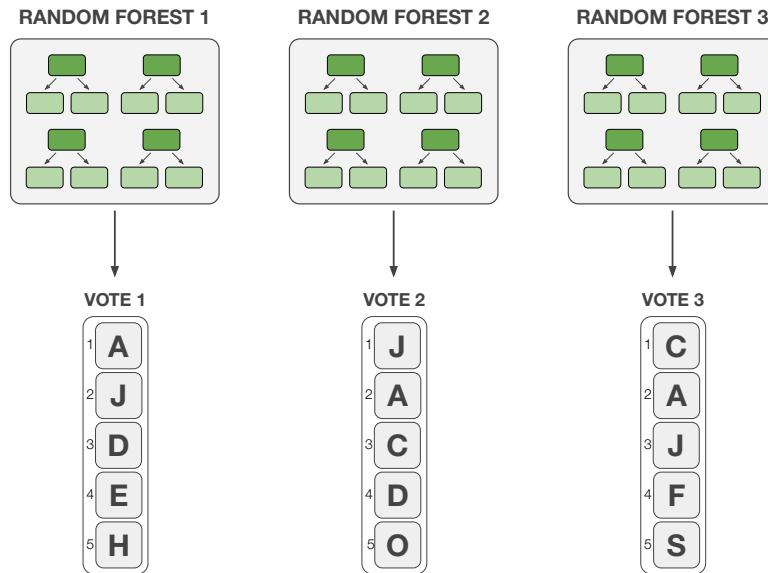
In biomedical field where explainability is a fundamental aspect to take into account, this instability and the strong influence the random seed has, is something not desirable. In order to reduce the instability of Decision Tree and more importantly of Random Forest, in this study, a novel ensemble method called Random Planet that combines multiple Random Forest models, was developed and tested. In the same way an ensemble of Decision Tree models gives a Random Forest, a combination of Random Forest models gives a Random Planet. Random Planet indeed, combines several Random Forest models with the aim of stabilizing the choice of the selected features making it invariant as much as possible upon random seed variation. The ranking of the selected features is then optimized using the classification performance obtained from a linear Support Vector Machine. A review of the literature on Scopus using the search string: (ALL ("random forest") AND ALL ("feature selection") AND ALL (combined) OR ALL (parallel) OR ALL (multiple) OR ALL (ensemble)) AND PUBYEAR > 2018 AND PUBYEAR < 2024 AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (DOCTYPE , "ar")) was performed to find works with a feature selection approach similar to Random Planet. The research returned 412 articles with more than 100 citations, and 1350 results with less than 100 and more than 50 citations. After the review of the articles with more than 100 citations, no similar methods to Random Planet were found applied in the literature so far.

3.2.3.1 Algorithm description

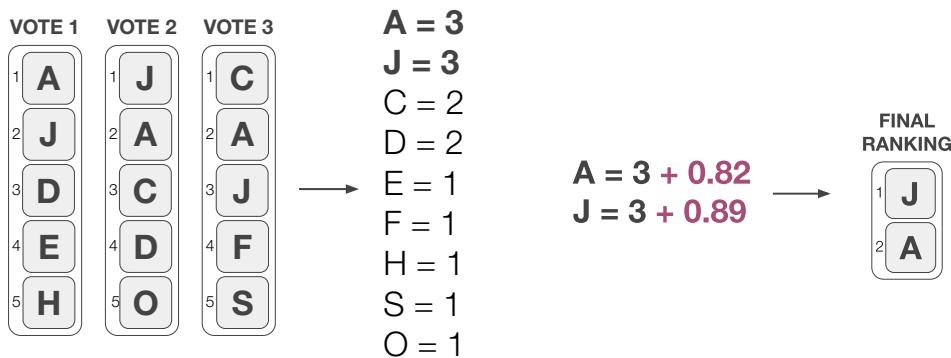
Random Planet combines several Random Forest models with a hard voting ensemble fashion where each model expresses its preference on the best NT features. The votes are then collected and the desired number of features is obtained by choosing the most voted ones. Since it might happen that two or more features obtain the same number of votes, to make the tiebreaker, the weight of each feature is given by the number of votes obtained summed to the AUC performance obtained by a SVM model trained on the training set restricted to just that single feature.

Fig. 3.3 shows graphically an example of execution. In this example, the Random Planet model is composed by NF=3 Random Forest models and each model votes for NT=5 features. The features are indicated with upper case letters Fig. 3.3a. Considering all the votes, a temporary ranking of features is computed Fig. 3.3b.

In this example, the desired number of features is two, since A and J are the most voted features, they are selected. In the final step Fig. 3.3c, to impose an order of importance to the selected features, for each feature a SVM model is trained on a training set composed by just a feature (only A or only J). Then the AUC performance obtained by the SVM in the training set is added to the number of votes the feature got in the previous step.



(a) Each Random Forest model expresses its preference on the best NT features.



(b) The votes are aggregated and the number of desired features (2) is selected.

(c) The AUC obtained in the training set by a SVM model trained on the reduced training set, composed by just a feature, is added to the number of votes that specific feature got.

Figure 3.3. Example feature selection performed by a Random Planet model.

3.2.3.2 Hyperparameter tuning

Each Random Forest model in Random Planet is trained on the training set and using a different random seed. The Decision Tree models in each Random Forest are trained with bootstrap technique and without any limitation on the max depth, max number of leaves or on the maximum number of features to take in consideration during trees construction. Refer to the algorithm of Random Forest in [10] and [40] for more information about the meaning of the choices made.

The hyperparameters of interest in a Random Planet model that must be tuned are:

- **Number of Forest (NF):** Number of Random Forest models used;
- **Number of Estimator in each forest (NE):** Number of Decision Tree models used in each Random Forest model;
- **Number of Top features each Random Forest votes (NT):** Number of features voted by each model.

Similarly to the procedure followed in Section 3.2.2 to perform hyperparameter tuning for GAAM, the hyperparameters for Random Planet were assigned using a grid search approach on the values shown in Table 3.3. Moreover, the same dataset with the reduced set of initial features adopted for GAAM is used also for Random Planet hyperparameter tuning.

Table 3.3. Values tested for the hyperparameters to be optimized. Number of Forest (NF), Number of Estimator in each forest (NE) and Number of Top features each Random Forest votes (NT).

NF	[1, 20, 50, 100, 150]
NE	[20, 50, 70]
NT	[2, 5, 10]

The total number of settings to test (i.e., hyperparameters combination) is 45 and the number of observations is equal to 14. Due to insufficient number of observations, rmANOVA test was not applicable and therefore, the effect of hyperparameters values on performance was assessed using Wilcoxon signed-rank test. Results are shown in Section 4.1.2.

3.3 Algorithms comparison

In the present work, as reported in Section 3.2, the feature selection algorithms tested are:

- **Decision Tree (DTree);**
- **Genetic Algorithm with Aggressive Mutation (GAAM);**
- **Mutual Information (MInfo);**
- **Random Planet (RPlanet);**

- Stepwise Regression (**StepwiseR**).

For each different dataset introduced in Section 3.1.2, the performance of the tested methods were compared using ANOVA with repeated measures (rmANOVA) and Wilcoxon signed-rank test as post hoc. The metric used during the statistical analysis was AUC obtained in the test set by a linear SVM model trained on the reduced dataset. AUC was chosen because is the most adopted metric in biomedical classification tasks. Moreover, a complete report of the AUC, F1 Score and SoftS performance obtained by each method in each dataset, was built. Finally, in order to study what features each algorithm tends to select, a visual representation of the choice made by the methods over all the subjects, is provided through heatmaps. This last type of comparison was performed on the datasets with feature preselection on Beta band. This choice was motivated by previous studies which identified this band as the typical one for CMC [33]. The results organized by the movements and the types of EEG filtering used, are reported in Section 4.2.

Chapter 4

Results

This chapter contains the results obtained in this study. It is divided as follows:

- **Section 4.1:** reports the results of the hyperparameter tuning procedures performed for GAAM and Random Planet;
- **Section 4.2:** contains the performance comparison across the different feature selection algorithms tested.

4.1 Hyperparameter tuning

In this section, the results of the statistical analyses performed during hyperparameter tuning are reported.

4.1.1 Genetic algorithm with aggressive mutation

The tuning procedure involved two different hyperparameters:

- **Number of Generations (NG):** Corresponds to the number of iterations of the algorithm, namely how many times the evolutionary cycle is repeated. Values tested in [100, 200, 300];
- **Population Size (PS):** Number of individuals that compose the population at each iteration. Values tested in [10, 20, 40].

The statistical analysis is divided into two parts:

- **NG_PSx:** Number of Generation (NG) is fixed at 100, 200 or 300 and the Population Size (PS) varies in [10, 20, 40];
- **NGx_PS:** NG varies in [100, 200, 300] and PS is fixed at 10, 20 or 40.

NG_PSx

Table 4.1 shows repeated measures ANOVA (rmANOVA) test results and the average values across participants (14) of the performance obtained in the training set. In the first two tests, namely NG100_PSx and NG200_PSx, statistical differences were

found across the levels of the PS variable for each performance metric computed in the training set (AUC, F1 score, Soft Stability). In the third test NG300_PSx instead, just for AUC and F1 Score.

Table 4.1. Results of the rmANOVA tests ($F(\text{Num_DF}, \text{Den_DF})$ and P-value) and average performance (across 14 healthy participants) measured on NG100_PSx = [NG100_PS10, NG100_PS20, NG100_PS40], NG200_PSx = [NG200_PS10, NG200_PS20, NG200_PS40] and NG300_PSx = [NG300_PS10, NG300_PS20, NG300_PS40]. Color indicates significant statistic differences P-value ≤ 0.05 .

		AUC	F1 Score	Soft Stability
NG100_PSx	F(2.0, 26.0)	16.2	13.8	26.6
	P-value	<.0001	.0001	<.0001
	Mean	.9778	.9770	.5976
NG200_PSx	F(2.0, 26.0)	9.8	7.2	14.8
	P-value	.0007	.0033	.0001
	Mean	.9782	.9775	.6905
NG300_PSx	F(2.0, 26.0)	10.4	6.2	1.7
	P-value	.0005	.0061	.2031
	Mean	.9783	.9776	.7071

Multiple Wilcoxon signed-rank tests were applied as post hoc tests to determine the best value for PS that maximizes the performance. Table 4.2 shows the results of the six tests performed between the settings NG100_PS10, NG100_PS20 and NG100_PS40. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis: *AUC mean obtained with the i -th setting is less than the one of the j -th setting*. In order to deal with the type I error accumulation from each of the six paired tests, false discovery rate correction was applied. In this way the overall type I error is limited to 0.05 as in the previous rmANOVA test.

Table 4.2. Wilcoxon signed-rank test comparing AUC between NG100_PS10, NG100_PS20 and NG100_PS40. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *AUC mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.05 .

	NG100_PS10	NG100_PS20	NG100_PS40
NG100_PS10	-	.0009	.0009
NG100_PS20	1	-	.0188
NG100_PS40	1	1	-

Using the less-than symbol ($\theta < \varphi$) to indicate that the performance mean obtained with setting θ is statistically less than the one obtained with φ , the interpretation of the results in Table 4.2 are shown in Eq. 4.1.

$$\left. \begin{array}{l} \text{NG100_PS10} < \text{NG100_PS20} \\ \text{NG100_PS10} < \text{NG100_PS40} \\ \text{NG100_PS20} < \text{NG100_PS40} \end{array} \right\} \implies \text{NG100_PS40 highest AUC} \quad (4.1)$$

Namely, with $\text{NG}=100$, the best value to assign to PS is 40. The same procedure was applied with NG fixed to 200 and 300. Table 4.3, 4.4 show the results of the Wilcoxon signed-rank tests and the interpretations are in Eq. 4.2, 4.3, respectively.

Table 4.3. Wilcoxon signed-rank test comparing AUC between NG200_PS10 , NG200_PS20 and NG200_PS40 . The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *AUC mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.05 .

	NG200_PS10	NG200_PS20	NG200_PS40
NG200_PS10	-	.0354	.0108
NG200_PS20	1	-	.0679
NG200_PS40	1	1	-

$$\left. \begin{array}{l} \text{NG200_PS10} < \text{NG200_PS20} \\ \text{NG200_PS10} < \text{NG200_PS40} \end{array} \right\} \implies \begin{array}{ll} \text{NG200_PS20} & \text{highest AUC} \\ \text{NG200_PS40} & \end{array} \quad (4.2)$$

Table 4.4. Wilcoxon signed-rank test comparing AUC between NG300_PS10 , NG300_PS20 and NG300_PS40 . The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *AUC mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.05 .

	NG300_PS10	NG300_PS20	NG300_PS40
NG300_PS10	-	.0123	.0123
NG300_PS20	1	-	.1663
NG300_PS40	1	1	-

$$\left. \begin{array}{l} \text{NG300_PS10} < \text{NG300_PS20} \\ \text{NG300_PS10} < \text{NG300_PS40} \end{array} \right\} \implies \begin{array}{ll} \text{NG300_PS20} & \text{highest AUC} \\ \text{NG300_PS40} & \end{array} \quad (4.3)$$

From this first set of analysis, $\text{PS}=40$ is the best value for any tested value of NG . Indeed, it is the best when $\text{NG}=100$, and it is one of the best when $\text{NG}=[200, 300]$ where both $\text{PS}=20$ and $\text{PS}=40$ allow to reach the same performance.

NGx_PS

In the second set of analyses the population size (PS) is fixed, and the number of generations (NG) changes. Table 4.5 shows rmANOVA test results and the mean values of the performance obtained in the training set considering all the possible values of PS .

The only row of interest in Table 4.5 is the third one NGx_PS40 useful to determine the best value of NG when $\text{PS}=40$. In this case, there are no statistically significant differences for all metrics (AUC, F1 score, SoftS). With the aim of minimize computation time $\text{NG}=100$ was selected. The final hyperparameter combination used is then NG100_PS40 . Table 4.6 summarizes the initial feature set size, the components, and the hyperparameters selected as the best for the present work.

Table 4.5. Results of the rmANOVA tests (F(Num_DF, Den_DF) and P-value) and average performance (across 14 healthy participants) measured on NGx_PS10 = [NG100_PS10, NG200_PS10, NG300_PS10], NGx_PS20 = [NG100_PS20, NG200_PS20, NG300_PS20] and NGx_PS30 = [NG100_PS40, NG200_PS40, NG300_PS40]. Color indicates significant statistic differences P-value ≤ 0.05 .

		AUC	F1 Score	Soft Stability
NGx_PS10	F(2.0, 26.0)	20.7	20.0	24.0
	P-value	<.0001	<.0001	<.0001
	Mean	.9776	.9769	.5333
NGx_PS20	F(2.0, 26.0)	6.7	5.1	4.2
	P-value	.0046	.0137	.0259
	Mean	.9782	.9775	.7167
NGx_PS40	F(2.0, 26.0)	2.5	1.8	0.7
	P-value	.1030	.1769	.5006
	Mean	.9784	.9778	.7452

Table 4.6. Best hyperparameters selected for GAAM in the present work.

Initial features set size	576
Number of generations (NG)	100
Population size (PS)	40
Length of chromosomes	2
Fitness function	AUC linear SVM

4.1.2 Random Planet

The tuning procedure involved three different hyperparameters:

- **Number of Forest (NF):** Values tested in [1, 20, 50, 100, 150]. Notice that when NF=1, just one forest is used, therefore Random Planet is equivalent to a Random Forest model;
- **Number of Estimator in each forest (NE):** The values for the number of trees composing each model were tested in [20, 50, 70];
- **Number of Top features (NT):** The tested numbers of preferences each model can provide were in [2,5,10].

The performance were compared adopting the same training set used for GAAM tuning and using AUC, F1 score, HardS and SoftS as metrics.

Each pair of settings was compared using Wilcoxon signed-rank test using false discovery rate correction and with the same alternative hypothesis used in the post hoc test for GAAM. No statistical significant differences with confidence interval higher than 95% were found on any metrics. In order to perform hyperparameters tuning for Random Planet, the confidence interval was lowered to 85% and statistical differences were found for SoftS metric. The table showing all the Wilcoxon signed-rank tests with correction performed for SoftS are reported in Appendix .1.

From the initial 45 settings, all the hyperparameters combinations with performance worse than at least another one were removed. In order to minimize the computation time, the selected setting was NF20_NE20_NT2.

4.2 Algorithms comparison

In this section, the results of the comparison between the different tested feature selection algorithms are provided. The following text is divided into the four movement tasks (ExtL, ExtR, GraspL, GraspR) introduced in Section 3.1.2. Each subsection is in turn divided according to the type of EEG spatial filtering applied (NoFilter, Laplacian).

4.2.1 ExtL

In ExtL task, each subject performed finger extension movement using the left hand. During this task, the target muscle is *Extensor Digitorum* of the left forearm (Fig. 1.14 in Section 1.1.2.2).

4.2.1.1 NoFilter

For each subject, the datasets used for ExtL task when applied no EEG spatial filtering, are the following ones. The difference among them was explained in Section 3.1.2:

- ExtL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- ExtL-NoFilter-SN_MT-LEFT-ALL_(BAND)¹
- ExtL-NoFilter-SN_MT-LEFT-ALPHA
- ExtL-NoFilter-SN_MT-LEFT-BETA
- ExtL-NoFilter-SN_MT-LEFT-GAMMA

In order to perform a comparison of the classification performance obtained using the features selected by each method, a common classification model was used. The chosen model is a linear Support Vector Machine trained with just the two features each algorithm selected. The classification metric considered is AUC obtained in the test set by the linear SVM model.

The statistical analysis was performed using Analysis of Variance (ANOVA) with repeated measures (rmANOVA) and Wilcoxon signed-rank test with false discovery rate correction as post hoc test. This last was used in place of paired T-test due to lack of the normality assumption on paired comparison assessed using Shapiro-Wilk test. For each dataset, namely different type of features preselection, a rmANOVA test was performed. The number of observations (i.e., number of subjects) was 14. The independent variable was the feature selection algorithm adopted among the chosen ones in this work:

¹SN_MT = EEG electrodes over and nearby sensorimotor area

- Stepwise Regression (**StepwiseR**);
- Random Planet (**RPlanet**);
- Mutual Information (**MInfo**);
- Decision Tree (**DTree**);
- Genetic Algorithm with Aggressive Mutation (**GAAM**).

The dependent variable was instead AUC classification performance obtained in the test set. The results are shown in Table 4.7. The significant level was set to 0.05.

Table 4.7. Results of the rmANOVA tests (F(Num_DF, Den_DF) and P-value) across 14 healthy participants. The independent variable is the feature selection method used while the dependent variable is AUC in the test set. Color indicates significant statistic differences P-value ≤ 0.05 .

Task	Filter	EEG	EMG	Band		AUC Test
ExtL	NoFil	ALL	ALL	ALL	F(4.0, 52.0)	3.616
					P-value	.0113
ExtL	NoFil	SN_MT	LEFT	ALL	F(4.0, 52.0)	5.013
					P-value	.0017
ExtL	NoFil	SN_MT	LEFT	ALPHA	F(4.0, 52.0)	2.529
					P-value	.0515
ExtL	NoFil	SN_MT	LEFT	BETA	F(4.0, 52.0)	3.906
					P-value	.0076
ExtL	NoFil	SN_MT	LEFT	GAMMA	F(4.0, 52.0)	4.484
					P-value	.0035

For each dataset where rmANOVA shown statistical significant differences, Wilcoxon signed-rank test was used as post hoc test. The difference were confirmed in the following datasets:

- ExtL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- ExtL-NoFilter-SN_MT-LEFT-BETA

The tables with all the post hoc tests performed for this dataset and the following ones, are attached in Appendix .2.

Table 4.8 and 4.9, report the results of post hoc analysis with false discovery rate correction applied on ExtL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(band) and ExtL-NoFilter-SN_MT-LEFT-BETA respectively. The element in the i -th row and j -th column indicates the P-value of the test having as alternative hypothesis: *AUC mean obtained by the linear SVM model in the test set with the i -th feature selection algorithm, is less than the one obtained using the j -th feature selection algorithm.*

From Table 4.8, GAAM results better than RPlanet, Minfo and Dtree in ExtL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND) dataset. This same difference was also detected over ExtL-NoFilter-SN_MT-LEFT-BETA in Table 4.9. In this last, RPlanet and MInfo show worse performance with respect to StepwiseR too.

Table 4.8. Wilcoxon signed-rank test comparing AUC performance on ExtL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND). Color indicates significant statistic differences P-value ≤ 0.05 .

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.4913
RPlanet	.8395	-	1	1	.0266
MInfo	.8676	1	-	.8676	.0266
DTree	1	1	1	-	.0266
GAAM	1	1	1	1	-

Table 4.9. Wilcoxon signed-rank test comparing AUC performance on ExtL-NoFilter-SN_-MT-LEFT-BETA. Color indicates significant statistic differences P-value ≤ 0.05 .

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.5315
RPlanet	.0066	-	1	.3553	.0220
MInfo	.0066	1	-	.3553	.0220
DTree	.3795	1	1	-	.0220
GAAM	1	1	1	1	-

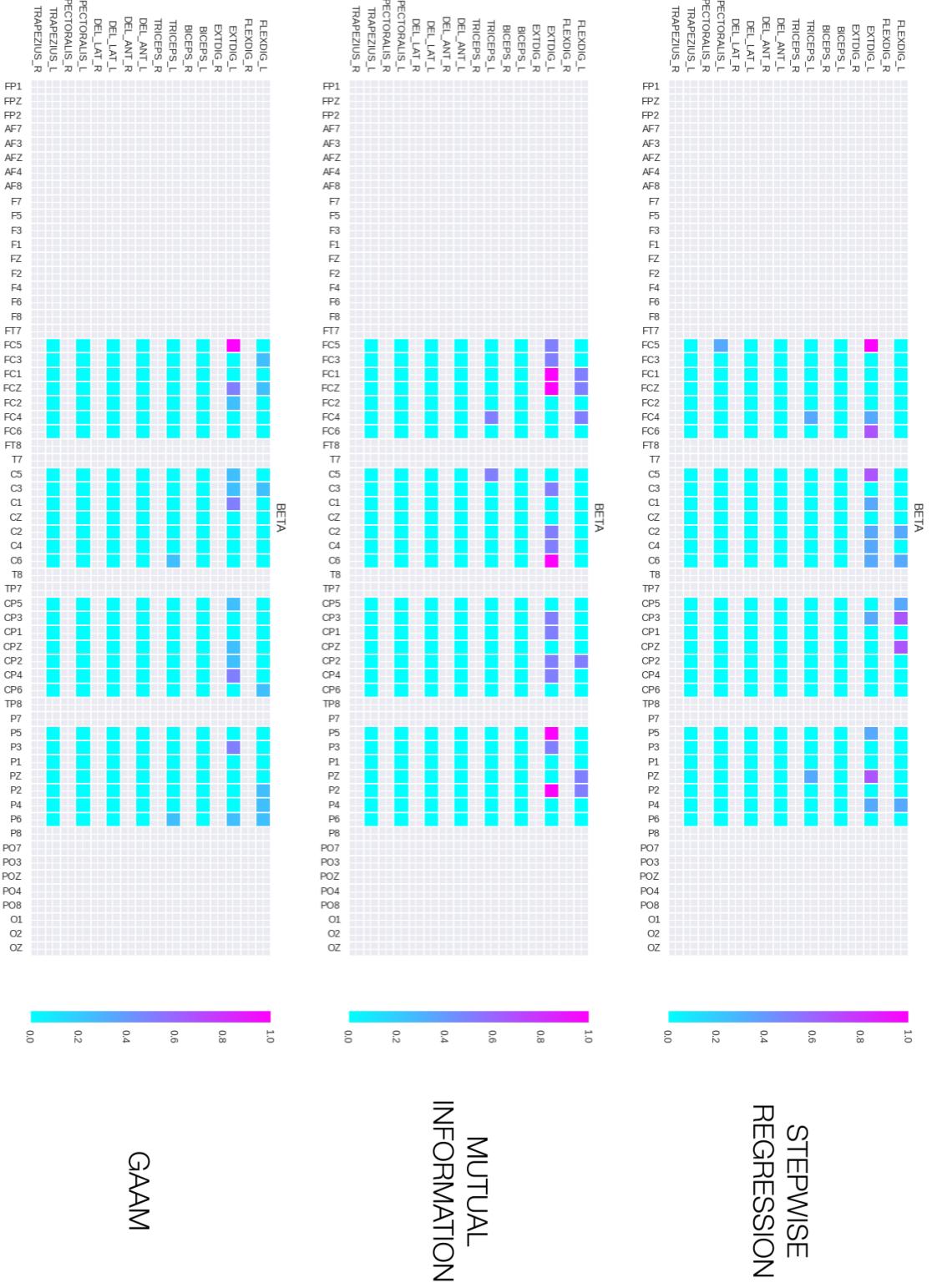
Table 4.10 reports the performance for the datasets of ExtL task and without any EEG spatial filtering. In addition to AUC, also F1 Score classification metric and Soft Stability (SoftS) are shown. F1 Score is shown because is one of the most common metric adopted by machine learning community for classification tasks. SoftS instead, is reported to give a measure of how consistent is the choice made by each feature selection algorithm across several runs on the same exact dataset. The values of AUC and F1 Score were computed as the mean across the 14 healthy subjects. As explained before, these metrics refer to the classification performance obtained by a linear SVM model trained on the reduced feature space composed by just the two features each method selected. SoftS instead, was computed for each subject by repeating the selection process using the same exact dataset for five times adopting always a different random seed for the methods which use it. Then the mean value of SoftS was computed across the subjects. The most stable method is StepwiseR with maximum stability equal to one. Also the stability of MInfo is high but not always equal to one. The worst stability was obtained by GAAM in ExtL-NoFilter-ALL_(EEG)-_ALL_(EMG)-ALL_(BAND), this is due to the higher number of initial features which causes the algorithm to remain stuck in a different local maximum at each run. Finally, it is interesting to notice that, as expected, RPlanet is always more stable than DTree. These behaviors on SoftS metric were noticed also in all the other tested datasets.

Considering only the dataset ExtL-NoFilter-SN_MT-LEFT-BETA, it is interesting to see the different features chosen by each method. The choice of Beta band preselection is motivated by previous studies which identified this band as the typical one for CMC [33]. Figs. 4.1, 4.2 show the features selected by each method on all the subjects. More precisely, the figures show a heatmap for each method with values ranging from 0 to 1. A value near to 0 indicates that the feature was never or

rarely selected in one of the healthy subjects. A value close to 1 instead, means that the same feature was selected as the best for several subjects. Grey blocks indicate the features removed in the preselection. The methods with more selection peaks on target muscle **EXTDIG_L** are Mutual Information and Random Planet. However, these algorithms are prone to select features on electrodes **Px** line located at the end of the parietal lobe near the occipital lobe.

Table 4.10. Soft Stability of each algorithm and classification performance obtained using the features selected by each method. Mean \pm std.

Task	Filter	EEG	EMG	Band	Method	AUC Train	AUC Test	F1 Train	F1 Test	SoftS
ExtL	NoFill	ALL	ALL	ALL	DTree	0.953 \pm 0.03	0.950 \pm 0.03	0.951 \pm 0.03	0.947 \pm 0.04	0.629 \pm 0.33
					GAAM	0.978 \pm 0.02	0.967 \pm 0.03	0.977 \pm 0.03	0.965 \pm 0.03	0.371 \pm 0.17
					MInfo	0.938 \pm 0.06	0.933 \pm 0.06	0.931 \pm 0.07	0.924 \pm 0.08	0.986 \pm 0.05
					RPlanet	0.949 \pm 0.03	0.940 \pm 0.04	0.946 \pm 0.03	0.935 \pm 0.04	0.800 \pm 0.19
ExtL	NoFill	SN_MT	LEFT	ALL	Stepwiser	0.962 \pm 0.03	0.954 \pm 0.03	0.961 \pm 0.03	0.952 \pm 0.04	1.0 \pm 0.0
					DTree	0.945 \pm 0.05	0.944 \pm 0.06	0.939 \pm 0.07	0.938 \pm 0.07	0.586 \pm 0.32
					GAAM	0.974 \pm 0.03	0.964 \pm 0.03	0.973 \pm 0.03	0.963 \pm 0.03	0.829 \pm 0.21
					MInfo	0.935 \pm 0.06	0.929 \pm 0.07	0.926 \pm 0.08	0.919 \pm 0.09	0.957 \pm 0.15
ExtL	NoFill	SN_MT	LEFT	ALPHA	RPlanet	0.938 \pm 0.06	0.933 \pm 0.06	0.930 \pm 0.07	0.923 \pm 0.08	0.814 \pm 0.21
					Stepwiser	0.951 \pm 0.04	0.943 \pm 0.05	0.948 \pm 0.05	0.940 \pm 0.06	1.0 \pm 0.0
					DTree	0.921 \pm 0.05	0.915 \pm 0.06	0.912 \pm 0.07	0.903 \pm 0.08	0.771 \pm 0.31
					GAAM	0.961 \pm 0.03	0.944 \pm 0.03	0.959 \pm 0.03	0.940 \pm 0.03	0.900 \pm 0.16
ExtL	NoFill	SN_MT	LEFT	ALPHA	MInfo	0.926 \pm 0.06	0.920 \pm 0.07	0.917 \pm 0.08	0.908 \pm 0.08	0.971 \pm 0.1
					RPlanet	0.932 \pm 0.06	0.927 \pm 0.07	0.924 \pm 0.07	0.916 \pm 0.09	0.943 \pm 0.14
					Stepwiser	0.935 \pm 0.05	0.934 \pm 0.05	0.929 \pm 0.06	0.927 \pm 0.06	1.0 \pm 0.0
					DTree	0.937 \pm 0.04	0.931 \pm 0.04	0.931 \pm 0.05	0.924 \pm 0.05	0.643 \pm 0.28
ExtL	NoFill	SN_MT	LEFT	BETA	GAAM	0.967 \pm 0.03	0.954 \pm 0.03	0.965 \pm 0.03	0.951 \pm 0.04	0.914 \pm 0.12
					MInfo	0.915 \pm 0.09	0.901 \pm 0.09	0.897 \pm 0.13	0.878 \pm 0.15	0.986 \pm 0.05
					RPlanet	0.932 \pm 0.05	0.919 \pm 0.05	0.925 \pm 0.06	0.911 \pm 0.06	0.900 \pm 0.18
					Stepwiser	0.949 \pm 0.04	0.942 \pm 0.05	0.946 \pm 0.05	0.939 \pm 0.05	1.0 \pm 0.0
ExtL	NoFill	SN_MT	LEFT	GAMMA	DTree	0.943 \pm 0.04	0.944 \pm 0.04	0.939 \pm 0.04	0.939 \pm 0.04	0.729 \pm 0.31
					GAAM	0.966 \pm 0.04	0.958 \pm 0.03	0.964 \pm 0.04	0.956 \pm 0.03	0.857 \pm 0.21
					MInfo	0.934 \pm 0.04	0.934 \pm 0.05	0.929 \pm 0.05	0.928 \pm 0.06	1.0 \pm 0.0
					RPlanet	0.940 \pm 0.03	0.942 \pm 0.04	0.937 \pm 0.04	0.938 \pm 0.04	0.871 \pm 0.19
ExtL	NoFill	SN_MT	LEFT	Stepwiser	Stepwiser	0.938 \pm 0.04	0.943 \pm 0.03	0.934 \pm 0.04	0.939 \pm 0.04	1.0 \pm 0.0



ExtL_NoFilter_Sensorimotor_LEFT_BETA

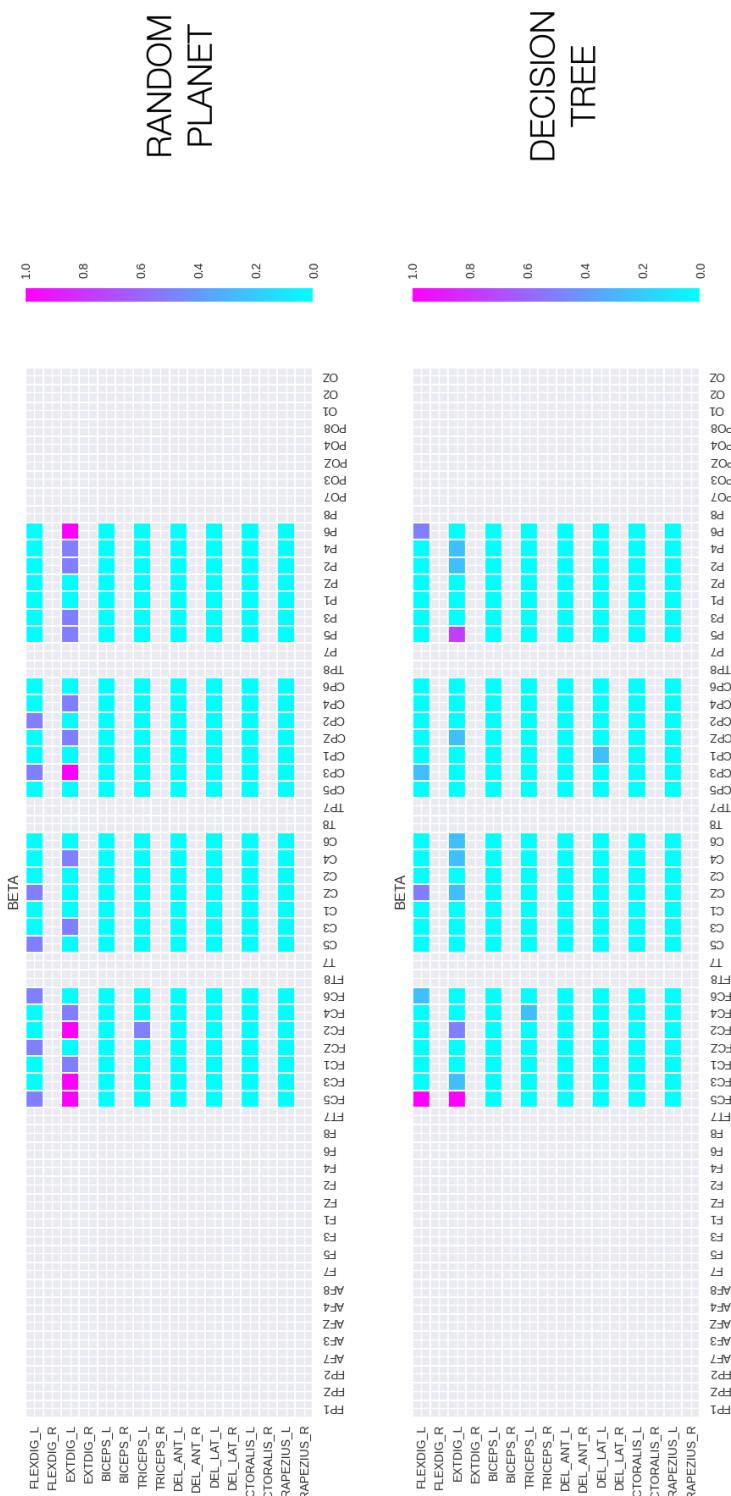


Figure 4.2. Heatmap features selected by each method in ExtL-NoFilter-SN_MT-LEFT-BETA dataset for all 14 subjects.

4.2.1.2 Laplacian

Similarly to the previous case, for each subject, the datasets used for ExtL task with Laplacian spatial filtering are:

- ExtL-Laplacian-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- ExtL-Laplacian-SN_MT-LEFT-ALL_(BAND)
- ExtL-Laplacian-SN_MT-LEFT-ALPHA
- ExtL-Laplacian-SN_MT-LEFT-BETA
- ExtL-Laplacian-SN_MT-LEFT-GAMMA

The statistical comparison method used before was applied also for Laplacian filtering. Table 4.11 show the rmANOVA test performed. In this case, no statistical significant differences were found across the classification performance of the linear SVM using the various feature selection methods.

Table 4.11. Results of the rmANOVA tests (F(Num_DF, Den_DF) and P-value) across 14 healthy participants. The independent variable is the feature selection method used while the dependent variable is AUC in the test set.

Task	Filter	EEG	EMG	Band		AUC Test
ExtL	Lapl	ALL	ALL	ALL	F(4.0, 52.0) P-value	1.928 .1196
ExtL	Lapl	SN_MT	LEFT	ALL	F(4.0, 52.0) P-value	2.016 .1057
ExtL	Lapl	SN_MT	LEFT	ALPHA	F(4.0, 52.0) P-value	2.536 .0510
ExtL	Lapl	SN_MT	LEFT	BETA	F(4.0, 52.0) P-value	2.156 .0870
ExtL	Lapl	SN_MT	LEFT	GAMMA	F(4.0, 52.0) P-value	1.93 .1192

Table 4.12 contains the complete report of the performance of each method. The heatmap for the selected features among the 14 healthy subjects is shown in Figs. 4.3, 4.4. Stepwise Regression and GAAM are the two methods that select more features on the target muscle EXTDIG_L. Stepwise Regression selects several times features on Px electrodes and GAAM contains a selection peak at PZ.

Table 4.12. Soft Stability of each algorithm and classification performance obtained using the features selected by each method. Mean \pm std.

Task	Filter	EEG	EMG	Band	Method	AUC Train	AUC Test	F1 Train	F1 Test	SoftS
ExtL	Lapl	ALL	ALL	DTree	0.944 \pm 0.04	0.935 \pm 0.04	0.940 \pm 0.04	0.930 \pm 0.05	0.800 \pm 0.29	
				GAAM	0.979 \pm 0.02	0.960 \pm 0.03	0.978 \pm 0.02	0.958 \pm 0.03	0.671 \pm 0.21	
				MInfo	0.922 \pm 0.11	0.917 \pm 0.11	0.889 \pm 0.22	0.885 \pm 0.22	1.0 \pm 0.0	
				RPlanet	0.956 \pm 0.03	0.942 \pm 0.03	0.953 \pm 0.03	0.938 \pm 0.04	0.914 \pm 0.16	
				Stepwiser	0.957 \pm 0.04	0.953 \pm 0.03	0.955 \pm 0.04	0.950 \pm 0.03	1.0 \pm 0.0	
ExtL	Lapl	SN_MT	LEFT	DTree	0.947 \pm 0.03	0.935 \pm 0.04	0.945 \pm 0.04	0.929 \pm 0.05	0.800 \pm 0.32	
				GAAM	0.978 \pm 0.02	0.964 \pm 0.03	0.978 \pm 0.02	0.962 \pm 0.03	0.800 \pm 0.17	
				MInfo	0.928 \pm 0.11	0.923 \pm 0.11	0.897 \pm 0.22	0.891 \pm 0.21	1.0 \pm 0.0	
				RPlanet	0.961 \pm 0.02	0.954 \pm 0.03	0.959 \pm 0.02	0.951 \pm 0.03	0.786 \pm 0.26	
				Stepwiser	0.958 \pm 0.04	0.955 \pm 0.03	0.956 \pm 0.04	0.953 \pm 0.03	1.0 \pm 0.0	
ExtL	Lapl	SN_MT	ALPHA	DTree	0.944 \pm 0.04	0.935 \pm 0.04	0.939 \pm 0.04	0.929 \pm 0.05	0.671 \pm 0.32	
				GAAM	0.969 \pm 0.02	0.950 \pm 0.04	0.967 \pm 0.03	0.947 \pm 0.04	0.914 \pm 0.18	
				MInfo	0.914 \pm 0.12	0.902 \pm 0.12	0.879 \pm 0.22	0.865 \pm 0.22	1.0 \pm 0.0	
				RPlanet	0.952 \pm 0.02	0.941 \pm 0.03	0.950 \pm 0.03	0.937 \pm 0.04	0.914 \pm 0.16	
				Stepwiser	0.944 \pm 0.04	0.946 \pm 0.04	0.940 \pm 0.05	0.941 \pm 0.05	1.0 \pm 0.0	
ExtL	Lapl	SN_MT	LEFT	DTree	0.949 \pm 0.03	0.948 \pm 0.03	0.946 \pm 0.03	0.944 \pm 0.03	0.771 \pm 0.28	
				GAAM	0.974 \pm 0.02	0.964 \pm 0.02	0.973 \pm 0.02	0.962 \pm 0.02	0.8571 \pm 0.21	
				MInfo	0.910 \pm 0.13	0.907 \pm 0.12	0.868 \pm 0.25	0.865 \pm 0.24	1.0 \pm 0.0	
				RPlanet	0.952 \pm 0.03	0.947 \pm 0.03	0.949 \pm 0.03	0.943 \pm 0.04	0.9143 \pm 0.12	
				Stepwiser	0.951 \pm 0.05	0.945 \pm 0.04	0.948 \pm 0.05	0.942 \pm 0.05	1.0 \pm 0.0	
ExtL	Lapl	SN_MT	LEFT	DTree	0.956 \pm 0.03	0.956 \pm 0.03	0.953 \pm 0.03	0.953 \pm 0.03	0.729 \pm 0.33	
				GAAM	0.975 \pm 0.02	0.967 \pm 0.02	0.974 \pm 0.03	0.966 \pm 0.02	0.8571 \pm 0.23	
				MInfo	0.954 \pm 0.05	0.954 \pm 0.04	0.949 \pm 0.06	0.949 \pm 0.05	0.9714 \pm 0.1	
				RPlanet	0.942 \pm 0.08	0.938 \pm 0.08	0.929 \pm 0.12	0.925 \pm 0.12	0.9 \pm 0.16	
				Stepwiser	0.950 \pm 0.04	0.946 \pm 0.05	0.946 \pm 0.05	0.941 \pm 0.06	1.0 \pm 0.0	

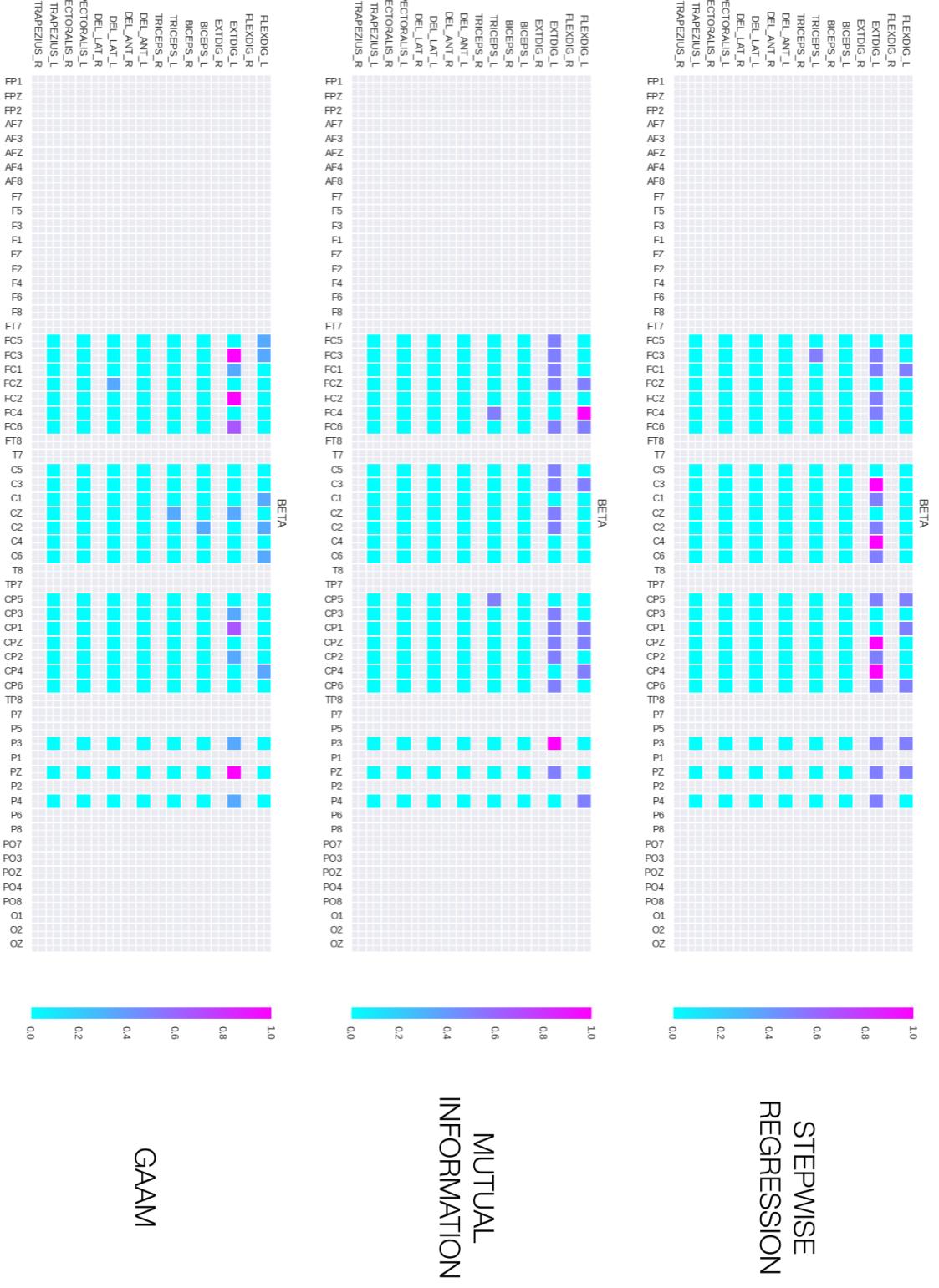


Figure 4.3. Heatmap features selected by each method in ExtL-Laplacian-SN_MT-LEFT-BETA dataset for all 14 subjects.

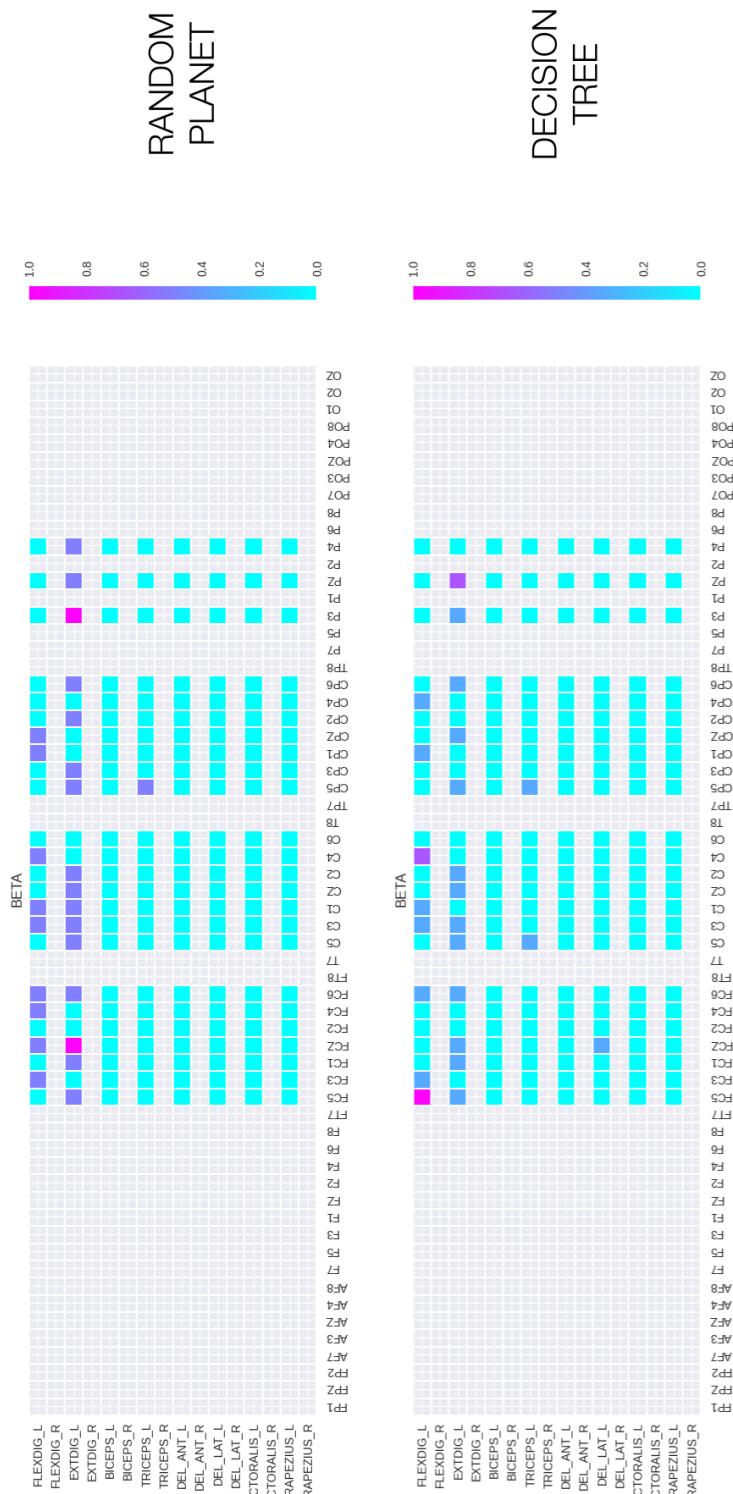


Figure 4.4. Heatmap features selected by each method in ExtL-Laplacian-SN_MT-LEFT-BETA dataset for all 14 subjects.

4.2.2 ExtR

In ExtR task, each subject performed finger extension movement using the right hand. During this task, the target muscle is EXTDIG_R of the right forearm (Fig. 1.14 in Section 1.1.2.2). This subsection is organized as the previous one.

4.2.2.1 NoFilter

For each subject, the datasets used for ExtR task without EEG spatial filtering are the following ones. The difference among them was explained in Section 3.1.2:

- ExtR-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- ExtR-NoFilter-SN_MT-RIGHT-ALL_(BAND)
- ExtR-NoFilter-SN_MT-RIGHT-ALPHA
- ExtR-NoFilter-SN_MT-RIGHT-BETA
- ExtR-NoFilter-SN_MT-RIGHT-GAMMA

The same statistical comparison method used before, was applied also for ExtR without EEG spatial filtering. Table 4.13 shown the rmANOVA test performed. No statistical significant differences were found across the classification performance of the linear SVM using the various feature selection methods.

Table 4.13. Results of the rmANOVA tests (F(Num_DF, Den_DF) and P-value) across 14 healthy participants. The independent variable is the feature selection method used while the dependent variable is AUC in the test set.

Task	Filter	EEG	EMG	Band		AUC Test
ExtR	NoFil	ALL	ALL	ALL	F(4.0, 52.0)	0.916
					P-value	.4617
ExtR	NoFil	SN_MT	RIGHT	ALL	F(4.0, 52.0)	1.155
					P-value	.3412
ExtR	NoFil	SN_MT	RIGHT	ALPHA	F(4.0, 52.0)	1.42
					P-value	.2404
ExtR	NoFil	SN_MT	RIGHT	BETA	F(4.0, 52.0)	1.876
					P-value	.1286
ExtR	NoFil	SN_MT	RIGHT	GAMMA	F(4.0, 52.0)	1.557
					P-value	.1997

Table 4.14 contains the complete report of the performance of each method. The heatmap for the selected features among the 14 healthy subjects is shown in Figs. 4.7, 4.8. The method with more selection peaks is Stepwise Regression, however it is prone to selects features from both EXTDIG_R and FLEXDIG_R muscle. Moreover, this method and GAAM have a peak on Px electrodes.

Table 4.14. Soft Stability of each algorithm and classification performance obtained using the features selected by each method. Mean \pm std.

Task	Filter	EEG	EMG	Band	Method	AUC Train	AUC Test	F1 Train	F1 Test	SoftS
ExtR	NoFill	ALL	ALL	ALL	DTree	0.964 \pm 0.02	0.962 \pm 0.03	0.962 \pm 0.02	0.959 \pm 0.03	0.443 \pm 0.31
					GAAM	0.986 \pm 0.01	0.972 \pm 0.02	0.986 \pm 0.01	0.971 \pm 0.02	0.443 \pm 0.15
					MInfo	0.964 \pm 0.02	0.961 \pm 0.03	0.962 \pm 0.02	0.959 \pm 0.03	1.0 \pm 0.0
					RPlanet	0.974 \pm 0.01	0.970 \pm 0.02	0.973 \pm 0.01	0.969 \pm 0.02	0.714 \pm 0.31
ExtR	NoFill	SN_MT	LEFT	ALL	Stepwiser	0.969 \pm 0.02	0.964 \pm 0.03	0.968 \pm 0.03	0.962 \pm 0.03	1.0 \pm 0.0
					DTree	0.958 \pm 0.03	0.958 \pm 0.03	0.955 \pm 0.03	0.955 \pm 0.03	0.529 \pm 0.28
					GAAM	0.984 \pm 0.01	0.969 \pm 0.03	0.984 \pm 0.01	0.967 \pm 0.03	0.643 \pm 0.22
					MInfo	0.959 \pm 0.02	0.960 \pm 0.03	0.957 \pm 0.02	0.957 \pm 0.03	1.0 \pm 0.0
ExtR	NoFill	SN_MT	LEFT	ALL	RPlanet	0.964 \pm 0.02	0.964 \pm 0.03	0.963 \pm 0.02	0.962 \pm 0.03	0.814 \pm 0.28
					Stepwiser	0.966 \pm 0.03	0.966 \pm 0.03	0.964 \pm 0.03	0.964 \pm 0.03	1.0 \pm 0.0
					DTree	0.946 \pm 0.03	0.953 \pm 0.02	0.942 \pm 0.03	0.950 \pm 0.03	0.814 \pm 0.27
					GAAM	0.964 \pm 0.03	0.953 \pm 0.03	0.962 \pm 0.03	0.950 \pm 0.03	0.843 \pm 0.19
ExtR	NoFill	SN_MT	LEFT	ALPHA	MInfo	0.942 \pm 0.03	0.945 \pm 0.03	0.938 \pm 0.04	0.941 \pm 0.04	1.0 \pm 0.0
					RPlanet	0.943 \pm 0.03	0.944 \pm 0.03	0.939 \pm 0.04	0.940 \pm 0.03	0.843 \pm 0.19
					Stepwiser	0.941 \pm 0.04	0.943 \pm 0.03	0.937 \pm 0.04	0.939 \pm 0.03	1.0 \pm 0.0
					DTree	0.963 \pm 0.02	0.961 \pm 0.03	0.961 \pm 0.02	0.959 \pm 0.03	0.600 \pm 0.28
ExtR	NoFill	SN_MT	LEFT	BETA	GAAM	0.978 \pm 0.02	0.967 \pm 0.02	0.977 \pm 0.02	0.965 \pm 0.02	0.957 \pm 0.11
					MInfo	0.954 \pm 0.03	0.959 \pm 0.03	0.951 \pm 0.03	0.956 \pm 0.03	1.0 \pm 0.0
					RPlanet	0.958 \pm 0.02	0.954 \pm 0.03	0.955 \pm 0.03	0.951 \pm 0.04	0.8 \pm 0.25
					Stepwiser	0.960 \pm 0.02	0.962 \pm 0.03	0.957 \pm 0.03	0.960 \pm 0.03	1.0 \pm 0.0
ExtR	NoFill	SN_MT	LEFT	GAMMA	DTree	0.957 \pm 0.03	0.959 \pm 0.04	0.954 \pm 0.04	0.956 \pm 0.05	0.586 \pm 0.31
					GAAM	0.975 \pm 0.03	0.966 \pm 0.04	0.974 \pm 0.03	0.964 \pm 0.04	0.914 \pm 0.18
					MInfo	0.953 \pm 0.04	0.955 \pm 0.05	0.950 \pm 0.04	0.951 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.959 \pm 0.03	0.959 \pm 0.04	0.956 \pm 0.03	0.957 \pm 0.04	0.800 \pm 0.28
ExtR	NoFill	SN_MT	LEFT	Stepwiser	Stepwiser	0.956 \pm 0.04	0.949 \pm 0.05	0.953 \pm 0.04	0.945 \pm 0.06	1.0 \pm 0.0

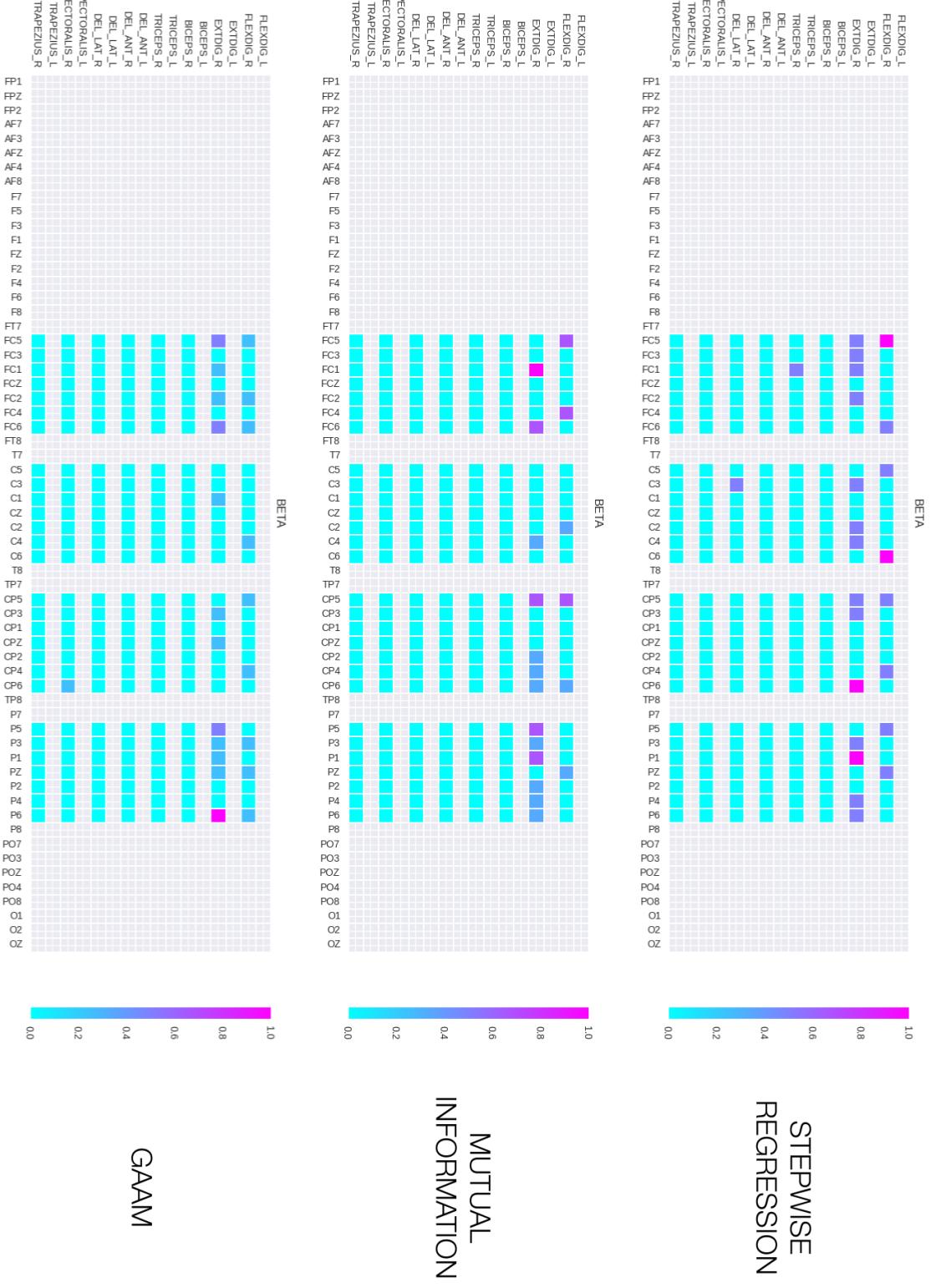


Figure 4.5. Heatmap features selected by each method in NoFilter-ExtR-SN_MT-RIGHT-BETA dataset for all 14 subjects.

ExtR_NoFilter_Sensorimotor_RIGHT_BETA

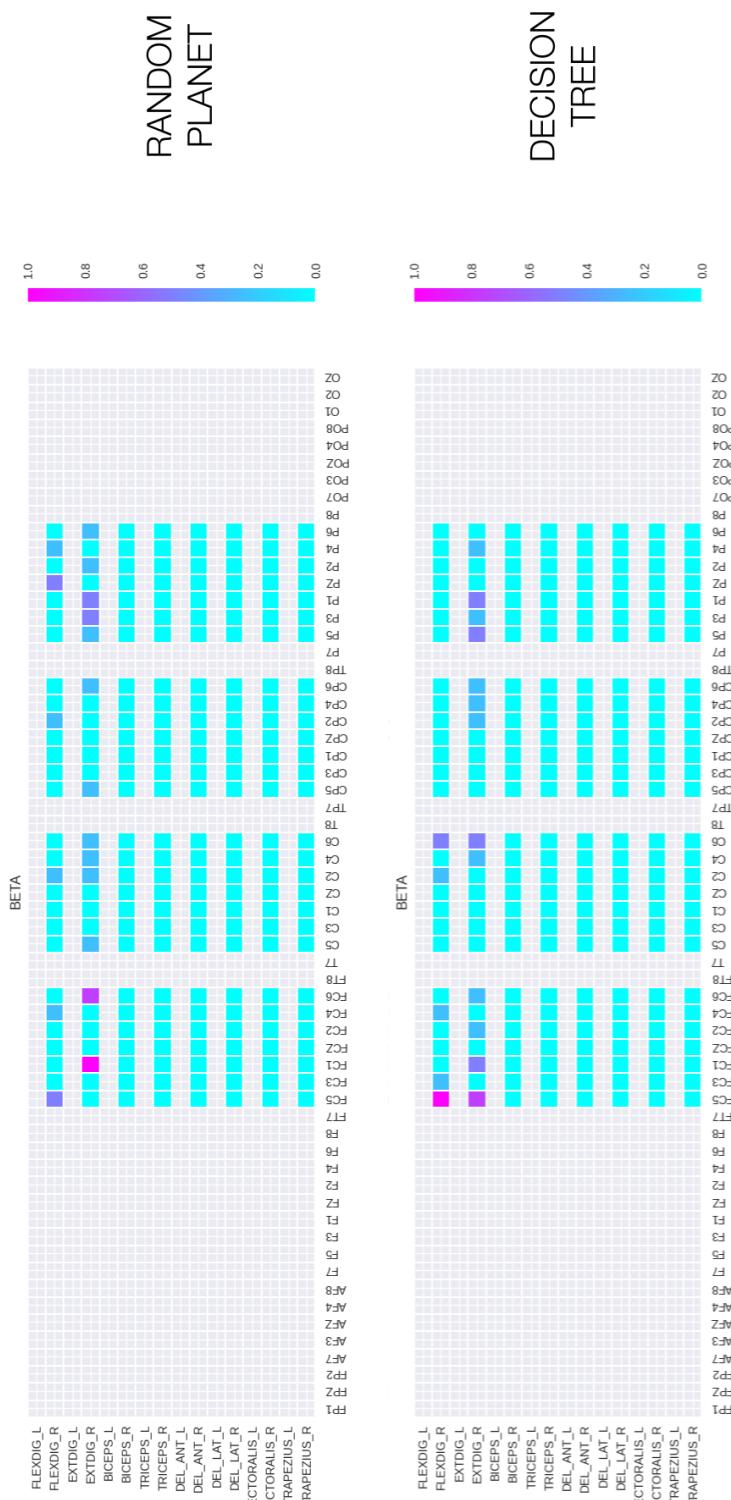


Figure 4.6. Heatmap features selected by each method in NoFilter-ExtR-SN_MT-RIGHT-BETA dataset for all 14 subjects.

4.2.2.2 Laplacian

For each subject, the datasets used for ExtR task with EEG Laplacian spatial filtering are the following ones. The difference among them was explained in Section 3.1.2:

- ExtR-Laplacian-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- ExtR-Laplacian-SN_MT-RIGHT-ALL_(BAND)
- ExtR-Laplacian-SN_MT-RIGHT-ALPHA
- ExtR-Laplacian-SN_MT-RIGHT-BETA
- ExtR-Laplacian-SN_MT-RIGHT-GAMMA

Table 4.15 shown the rmANOVA test performed. Wilcoxon signed-rank test with false discovery rate correction was then applied as post hoc test. No statistical significant differences were found after post hoc across the classification performance of the linear SVM using the various feature selection methods.

Table 4.15. Results of the rmANOVA tests (F(Num_DF, Den_DF) and P-value) across 14 healthy participants. The independent variable is the feature selection method used while the dependent variable is AUC in the test set. Color indicates significant statistic differences P-value ≤ 0.05 .

Task	Filter	EEG	EMG	Band		AUC Test
ExtR	Lapl	ALL	ALL	ALL	F(4.0, 52.0)	3.002
					P-value	.0265
ExtR	Lapl	SN_MT	RIGHT	ALL	F(4.0, 52.0)	1.997
					P-value	.1087
ExtR	Lapl	SN_MT	RIGHT	ALPHA	F(4.0, 52.0)	4.855
					P-value	.0021
ExtR	Lapl	SN_MT	RIGHT	BETA	F(4.0, 52.0)	1.587
					P-value	.1917
ExtR	Lapl	SN_MT	RIGHT	GAMMA	F(4.0, 52.0)	2.84
					P-value	.0333

Table 4.16 contains the complete report of the performance of each method. The heatmap for the selected features among the 14 healthy subjects is shown in Figs. 4.7, 4.8. In this case, all the methods have peaks on both EXTDIG_R and FLEXDIG_R muscles. Mutual Information and Decision Tree have the same peak at EXTDIG_R:P4.

Table 4.16. Soft Stability of each algorithm and classification performance obtained using the features selected by each method. Mean \pm std.

Task	Filter	EEG	EMG	Band	Method	AUC Train	AUC Test	F1 Train	F1 Test	SoftS
ExtR	Lapl	ALL	ALL	ALL	DTree	0.963 \pm 0.02	0.955 \pm 0.03	0.961 \pm 0.02	0.952 \pm 0.03	0.529 \pm 0.36
					GAAM	0.988 \pm 0.01	0.977 \pm 0.02	0.988 \pm 0.01	0.976 \pm 0.02	0.686 \pm 0.24
					MInfo	0.970 \pm 0.02	0.966 \pm 0.02	0.969 \pm 0.02	0.965 \pm 0.02	1.0 \pm 0.0
					RPlanet	0.972 \pm 0.02	0.971 \pm 0.02	0.970 \pm 0.02	0.969 \pm 0.03	0.786 \pm 0.28
					Stepwiser	0.972 \pm 0.02	0.966 \pm 0.02	0.971 \pm 0.02	0.965 \pm 0.02	1.0 \pm 0.0
ExtR	Lapl	SN_MT	LEFT	ALL	DTree	0.965 \pm 0.02	0.958 \pm 0.03	0.963 \pm 0.02	0.955 \pm 0.03	0.643 \pm 0.36
					GAAM	0.988 \pm 0.01	0.975 \pm 0.01	0.988 \pm 0.01	0.974 \pm 0.02	0.671 \pm 0.25
					MInfo	0.969 \pm 0.02	0.967 \pm 0.02	0.968 \pm 0.02	0.965 \pm 0.03	1.0 \pm 0.0
					RPlanet	0.970 \pm 0.02	0.967 \pm 0.02	0.969 \pm 0.02	0.965 \pm 0.02	0.686 \pm 0.31
					Stepwiser	0.973 \pm 0.01	0.971 \pm 0.02	0.973 \pm 0.01	0.970 \pm 0.02	1.0 \pm 0.0
ExtR	Lapl	SN_MT	LEFT	ALPHA	DTree	0.960 \pm 0.02	0.956 \pm 0.02	0.957 \pm 0.02	0.953 \pm 0.02	0.743 \pm 0.33
					GAAM	0.975 \pm 0.02	0.971 \pm 0.02	0.974 \pm 0.02	0.970 \pm 0.02	0.800 \pm 0.24
					MInfo	0.961 \pm 0.02	0.957 \pm 0.02	0.959 \pm 0.02	0.954 \pm 0.02	1.0 \pm 0.0
					RPlanet	0.961 \pm 0.02	0.955 \pm 0.02	0.959 \pm 0.02	0.952 \pm 0.02	0.757 \pm 0.24
					Stepwiser	0.951 \pm 0.04	0.939 \pm 0.05	0.947 \pm 0.05	0.933 \pm 0.06	1.0 \pm 0.0
ExtR	Lapl	SN_MT	LEFT	BETA	DTree	0.968 \pm 0.01	0.960 \pm 0.02	0.966 \pm 0.02	0.957 \pm 0.02	0.771 \pm 0.32
					GAAM	0.984 \pm 0.01	0.970 \pm 0.02	0.983 \pm 0.01	0.969 \pm 0.02	0.943 \pm 0.16
					MInfo	0.971 \pm 0.01	0.962 \pm 0.03	0.970 \pm 0.01	0.959 \pm 0.03	1.0 \pm 0.0
					RPlanet	0.970 \pm 0.01	0.956 \pm 0.03	0.968 \pm 0.01	0.953 \pm 0.03	0.757 \pm 0.24
					Stepwiser	0.967 \pm 0.02	0.958 \pm 0.02	0.965 \pm 0.02	0.956 \pm 0.02	1.0 \pm 0.0
ExtR	Lapl	SN_MT	LEFT	GAMMA	DTree	0.967 \pm 0.02	0.961 \pm 0.03	0.966 \pm 0.02	0.959 \pm 0.03	0.557 \pm 0.33
					GAAM	0.985 \pm 0.01	0.978 \pm 0.01	0.985 \pm 0.01	0.978 \pm 0.01	0.914 \pm 0.15
					MInfo	0.964 \pm 0.02	0.965 \pm 0.02	0.963 \pm 0.02	0.963 \pm 0.02	1.0 \pm 0.0
					RPlanet	0.965 \pm 0.02	0.961 \pm 0.02	0.963 \pm 0.03	0.959 \pm 0.02	0.800 \pm 0.26
					Stepwiser	0.968 \pm 0.02	0.964 \pm 0.02	0.966 \pm 0.02	0.962 \pm 0.02	1.0 \pm 0.0

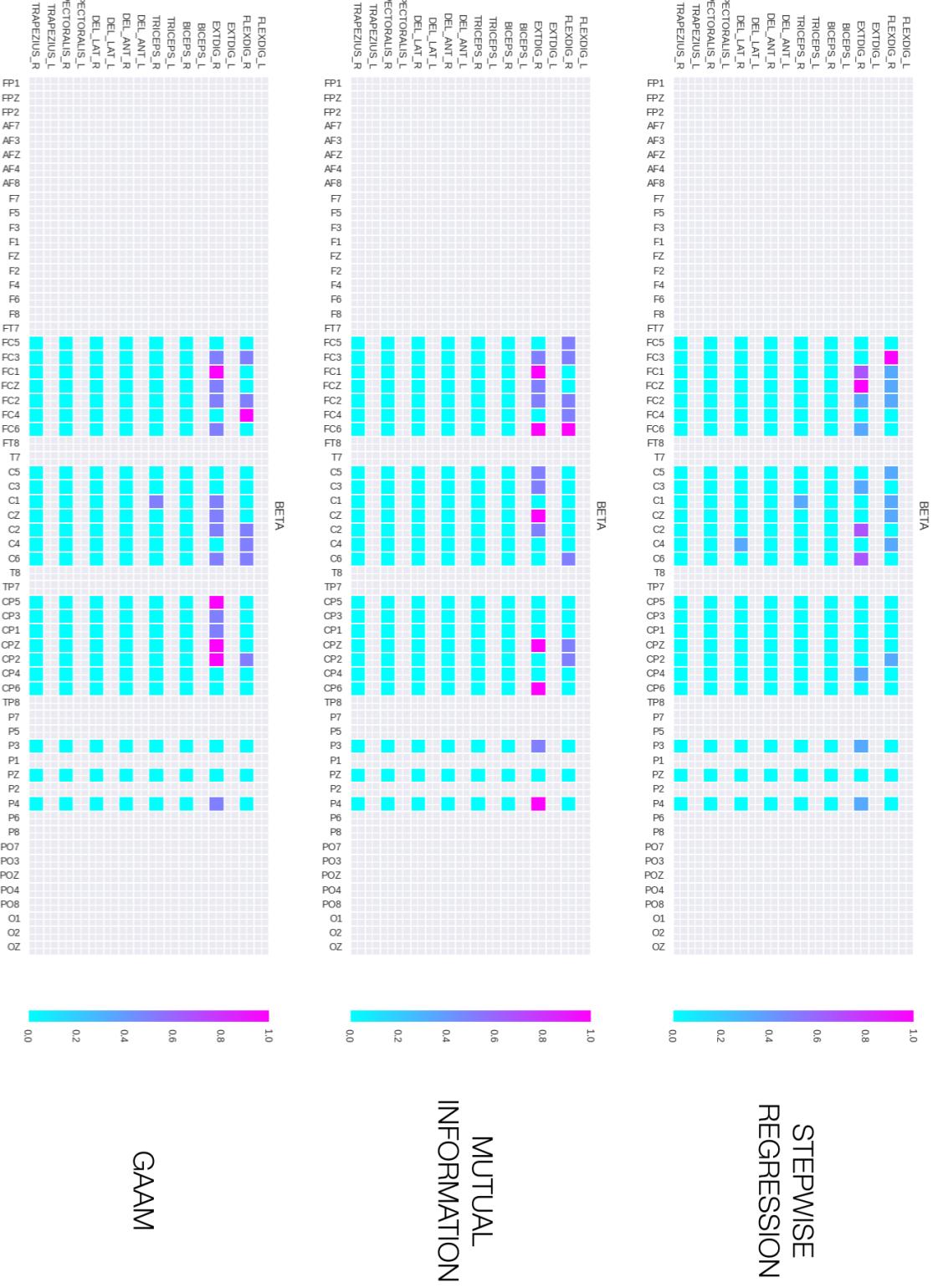


Figure 4.7. Heatmap features selected by each method in Laplacian-Extr-SN-MT-LEFT-BETA dataset, for all 14 subjects.

ExtR_Laplacian_Sensorimotor_RIGHT_BETA

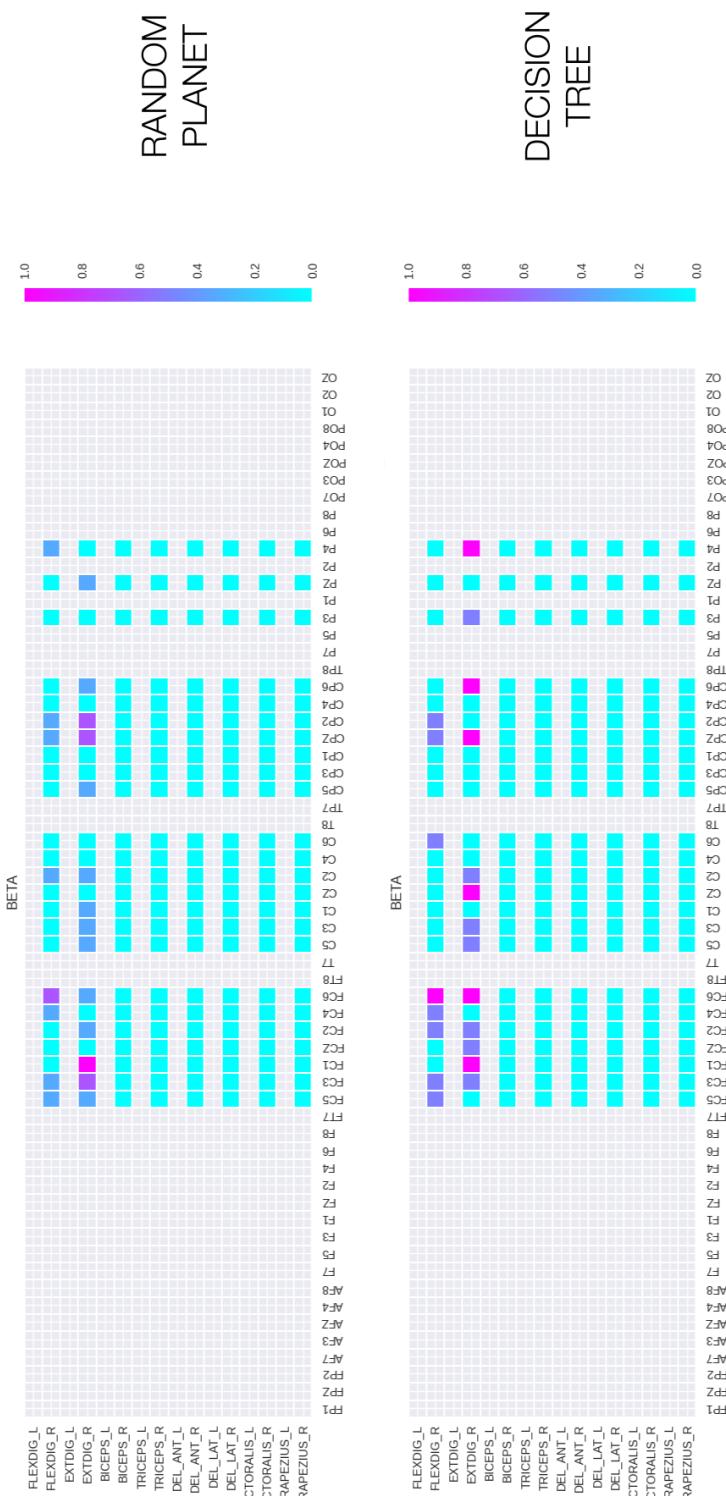


Figure 4.8. Heatmap features selected by each method in Laplacian-ExtR-SN_MT-LEFT-BETA dataset for all 14 subjects.

4.2.3 GraspL

In GraspL task, each subject performed finger grasping movement using the left hand. During this task, the target muscle is FLEXDIG_L of the left forearm (Fig. 1.14 in Section 1.1.2.2). This subsection is organized as the previous one.

4.2.3.1 NoFilter

For each subject, the datasets used for GraspL task without EEG spatial filtering are the following ones. The difference among them was explained in Section 3.1.2:

- GraspL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- GraspL-NoFilter-SN_MT-LEFT-ALL_(BAND)
- GraspL-NoFilter-SN_MT-LEFT-ALPHA
- GraspL-NoFilter-SN_MT-LEFT-BETA
- GraspL-NoFilter-SN_MT-LEFT-GAMMA

Table 4.17 shown the rmANOVA test performed.

Table 4.17. Results of the rmANOVA tests (F(Num_DF, Den_DF) and P-value) across 14 healthy participants. The independent variable is the feature selection method used while the dependent variable is AUC in the test set. Color indicates significant statistic differences P-value ≤ 0.05 .

Task	Filter	EEG	EMG	Band		AUC Test
GraspL	NoFil	ALL	ALL	ALL	F(4.0, 52.0)	2.87
					P-value	.0319
GraspL	NoFil	SN_MT	LEFT	ALL	F(4.0, 52.0)	2.648
					P-value	.0436
GraspL	NoFil	SN_MT	LEFT	ALPHA	F(4.0, 52.0)	2.216
					P-value	.0800
GraspL	NoFil	SN_MT	LEFT	BETA	F(4.0, 52.0)	4.633
					P-value	.0028
GraspL	NoFil	SN_MT	LEFT	GAMMA	F(4.0, 52.0)	3.483
					P-value	.0136

For each dataset where rmANOVA shown statistical significant differences, Wilcoxon signed-rank test was used as post hoc test with false detection rate correction. The difference was confirmed only in:

- GraspL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)

Table 4.18 reports the results of post hoc analysis for this dataset. The element in the i -th row and j -th column indicates the P-value of the test having as alternative hypothesis: *AUC mean obtained by the linear SVM model in the test set with the i -th feature selection algorithms, is less than the one obtained using the j -th feature selection algorithms.* In this dataset, MInfo performed worse than GAAM.

Table 4.18. Wilcoxon signed-rank test comparing AUC performance on GraspL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND). Color indicates significant statistic differences P-value ≤ 0.05 .

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.2492
RPlanet	1	-	1	1	.0724
MInfo	1	1	-	1	.0220
DTree	1	1	1	-	.0517
GAAM	1	1	1	1	-

Table 4.19 contains the complete report of the performance of each method. The heatmaps for the selected features among the 14 healthy subjects are shown in Figs. 4.9, 4.10. GAAM and Decision Tree are the methods with more peaks on the target muscle FLEXDIG_L. Stepwise Regression is the only methods with a peak on EXTDIG_L.

Table 4.19. Soft Stability of each algorithm and classification performance obtained using the features selected by each method. Mean \pm std.

Task	Filter	EEG	EMG	Band	Method	AUC Train	AUC Test	F1 Train	F1 Test	SoftS
Graspl	NoFil	ALL	ALL	ALL	DTree	0.959 \pm 0.02	0.957 \pm 0.02	0.957 \pm 0.02	0.955 \pm 0.03	0.343 \pm 0.28
					GAM	0.984 \pm 0.02	0.975 \pm 0.02	0.984 \pm 0.02	0.975 \pm 0.02	0.443 \pm 0.17
					Minfo	0.957 \pm 0.03	0.952 \pm 0.04	0.954 \pm 0.04	0.948 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.960 \pm 0.02	0.954 \pm 0.03	0.958 \pm 0.02	0.951 \pm 0.03	0.671 \pm 0.27
Graspl	NoFil	SN_MT	LEFT	ALL	StepwiseR	0.966 \pm 0.02	0.964 \pm 0.02	0.964 \pm 0.03	0.962 \pm 0.03	1.0 \pm 0.0
					DTree	0.957 \pm 0.03	0.954 \pm 0.04	0.954 \pm 0.03	0.951 \pm 0.04	0.514 \pm 0.34
					GAM	0.981 \pm 0.02	0.975 \pm 0.02	0.980 \pm 0.02	0.974 \pm 0.02	0.800 \pm 0.25
					Minfo	0.956 \pm 0.04	0.955 \pm 0.04	0.953 \pm 0.04	0.951 \pm 0.05	1.0 \pm 0.0
Graspl	NoFil	SN_MT	LEFT	ALL	RPlanet	0.964 \pm 0.02	0.966 \pm 0.03	0.963 \pm 0.02	0.964 \pm 0.03	0.714 \pm 0.26
					StepwiseR	0.964 \pm 0.03	0.964 \pm 0.02	0.962 \pm 0.03	0.962 \pm 0.03	1.0 \pm 0.0
					DTree	0.955 \pm 0.03	0.952 \pm 0.03	0.953 \pm 0.03	0.948 \pm 0.04	0.671 \pm 0.36
					GAM	0.972 \pm 0.02	0.963 \pm 0.02	0.971 \pm 0.02	0.961 \pm 0.02	0.800 \pm 0.25
Graspl	NoFil	SN_MT	LEFT	ALPHA	Minfo	0.954 \pm 0.02	0.952 \pm 0.03	0.951 \pm 0.03	0.949 \pm 0.03	1.0 \pm 0.0
					RPlanet	0.953 \pm 0.02	0.952 \pm 0.03	0.950 \pm 0.02	0.948 \pm 0.03	0.914 \pm 0.15
					StepwiseR	0.954 \pm 0.03	0.945 \pm 0.04	0.951 \pm 0.04	0.941 \pm 0.05	1.0 \pm 0.0
					DTree	0.955 \pm 0.02	0.962 \pm 0.03	0.952 \pm 0.03	0.959 \pm 0.04	0.629 \pm 0.32
Graspl	NoFil	SN_MT	LEFT	BETA	GAM	0.974 \pm 0.02	0.973 \pm 0.02	0.973 \pm 0.02	0.971 \pm 0.02	0.929 \pm 0.14
					Minfo	0.949 \pm 0.04	0.947 \pm 0.04	0.945 \pm 0.04	0.942 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.956 \pm 0.02	0.950 \pm 0.03	0.953 \pm 0.03	0.947 \pm 0.03	0.843 \pm 0.22
					StepwiseR	0.956 \pm 0.03	0.958 \pm 0.02	0.953 \pm 0.03	0.955 \pm 0.03	1.0 \pm 0.0
Graspl	NoFil	SN_MT	LEFT	GAMMA	DTree	0.942 \pm 0.03	0.949 \pm 0.03	0.938 \pm 0.04	0.945 \pm 0.04	0.643 \pm 0.35
					GAM	0.972 \pm 0.02	0.968 \pm 0.02	0.970 \pm 0.02	0.967 \pm 0.02	0.957 \pm 0.11
					Minfo	0.955 \pm 0.02	0.958 \pm 0.03	0.953 \pm 0.03	0.955 \pm 0.03	1.0 \pm 0.0
					RPlanet	0.949 \pm 0.03	0.953 \pm 0.03	0.945 \pm 0.03	0.950 \pm 0.04	0.871 \pm 0.22
Graspl	NoFil	SN_MT	LEFT	GAMMA	StepwiseR	0.950 \pm 0.03	0.953 \pm 0.03	0.947 \pm 0.03	0.950 \pm 0.04	1.0 \pm 0.0

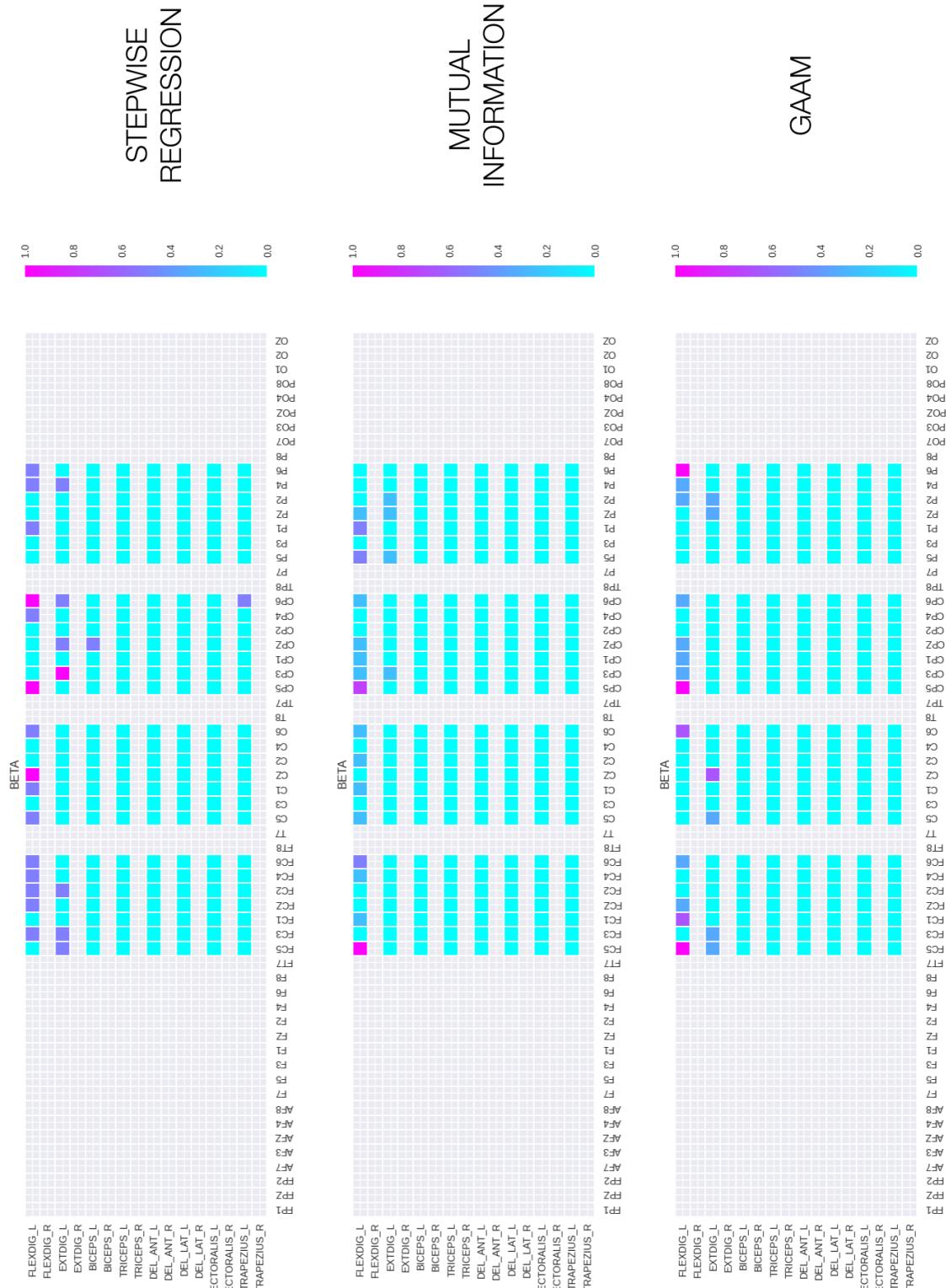


Figure 4.9. Heatmap features selected by each method in GraspL-NoFilter-SN_MT-LEFT-BETA dataset for all 14 subjects.

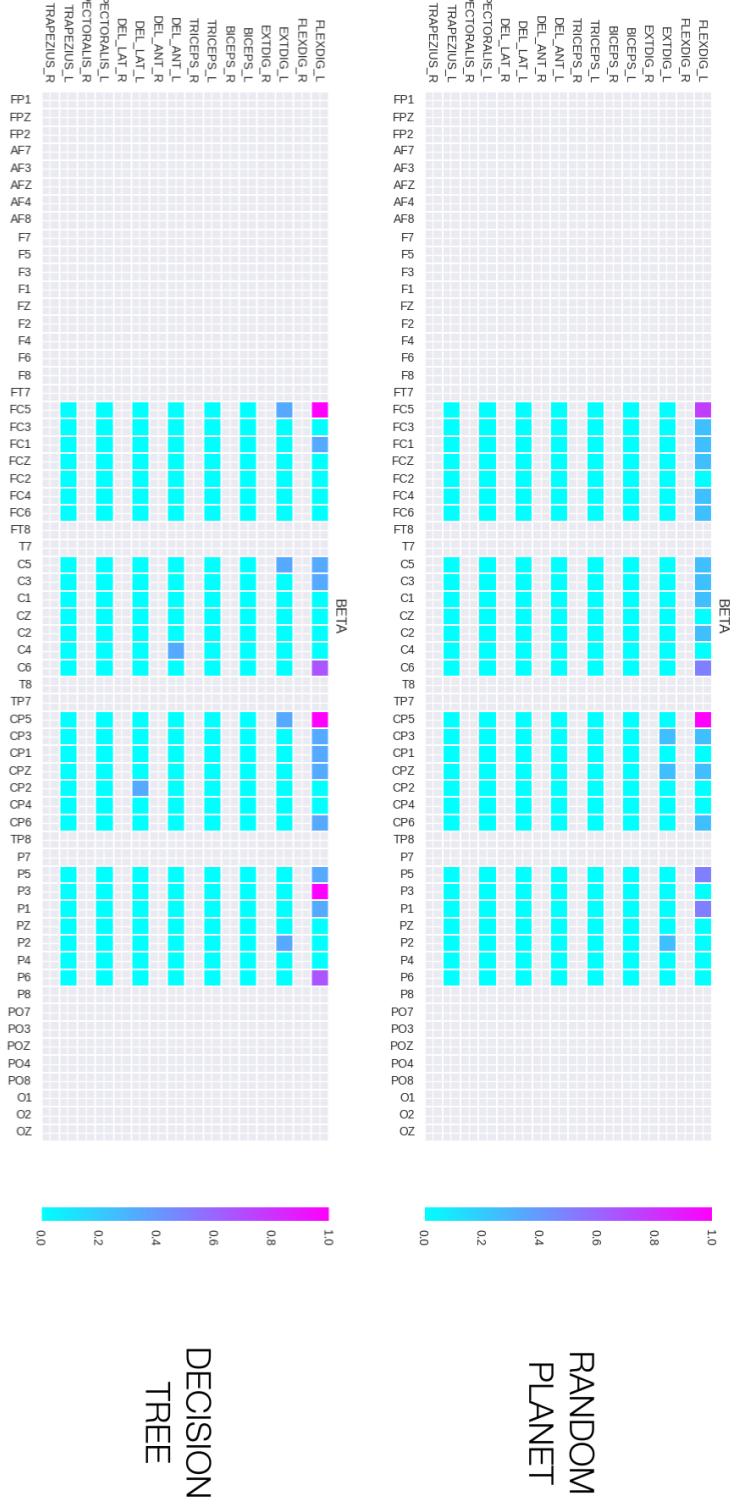


Figure 4.10. Heatmap features selected by each method in Graspl-NoFilter-SN_MT-LEFT-BETA dataset for all 14 subjects.

4.2.3.2 Laplacian

For each subject, the datasets used for GraspL task with EEG Laplacian spatial filtering are the following ones. The difference among them was explained in Section 3.1.2:

- GraspL-Laplacian-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- GraspL-Laplacian-SN_MT-LEFT-ALL_(BAND)
- GraspL-Laplacian-SN_MT-LEFT-ALPHA
- GraspL-Laplacian-SN_MT-LEFT-BETA
- GraspL-Laplacian-SN_MT-LEFT-GAMMA

Table 4.20 shown the rmANOVA test performed. Wilcoxon signed-rank test with false discovery rate correction was then applied as post hoc test. No statistical significant differences were found after post hoc across the classification performance of the linear SVM using the various feature selection methods.

Table 4.20. Results of the rmANOVA tests ($F(\text{Num_DF}, \text{Den_DF})$ and P-value) across 14 healthy participants. The independent variable is the feature selection method used while the dependent variable is AUC in the test set. Color indicates significant statistic differences P-value ≤ 0.05 .

Task	Filter	EEG	EMG	Band		AUC Test
GraspL	Lapl	ALL	ALL	ALL	$F(4.0, 52.0)$	2.625
					P-value	.0450
GraspL	Lapl	SN_MT	LEFT	ALL	$F(4.0, 52.0)$	2.65
					P-value	.0434
GraspL	Lapl	SN_MT	LEFT	ALPHA	$F(4.0, 52.0)$	1.657
					P-value	.1740
GraspL	Lapl	SN_MT	LEFT	BETA	$F(4.0, 52.0)$	2.534
					P-value	.0511
GraspL	Lapl	SN_MT	LEFT	GAMMA	$F(4.0, 52.0)$	1.988
					P-value	.1100

Table 4.21 contains the complete report of the performance of each method. The heatmap for the selected features among the 14 healthy subjects is shown in Figs. 4.11, 4.12. The method with the highest number of peaks on the target muscle FLEXDIG_L is Decision Tree. Stepwise Regression is the only method with a peak on EXTDIG_L.

Table 4.21. Soft Stability of each algorithm and classification performance obtained using the features selected by each method. Mean \pm std.

Task	Filter	EEG	EMG	Band	Method	AUC Train	AUC Test	F1 Train	F1 Test	SoftS
Graspl	Lapl	ALL	ALL	ALL	DTree	0.940 \pm 0.07	0.942 \pm 0.08	0.929 \pm 0.1	0.928 \pm 0.12	0.486 \pm 0.3
					GAM	0.982 \pm 0.03	0.976 \pm 0.02	0.981 \pm 0.03	0.975 \pm 0.02	0.586 \pm 0.19
					Minfo	0.934 \pm 0.1	0.929 \pm 0.11	0.911 \pm 0.18	0.902 \pm 0.2	1.0 \pm 0.0
					RPlanet	0.952 \pm 0.05	0.950 \pm 0.04	0.947 \pm 0.06	0.946 \pm 0.05	0.771 \pm 0.25
					StepwiseR	0.960 \pm 0.04	0.957 \pm 0.06	0.957 \pm 0.05	0.952 \pm 0.07	1.0 \pm 0.0
Graspl	Lapl	SN_MT	LEFT	ALL	DTree	0.943 \pm 0.07	0.941 \pm 0.08	0.933 \pm 0.1	0.929 \pm 0.12	0.457 \pm 0.22
					GAM	0.983 \pm 0.02	0.977 \pm 0.02	0.982 \pm 0.02	0.976 \pm 0.02	0.814 \pm 0.19
					Minfo	0.935 \pm 0.11	0.932 \pm 0.12	0.908 \pm 0.2	0.902 \pm 0.22	1.0 \pm 0.0
					RPlanet	0.935 \pm 0.09	0.931 \pm 0.1	0.918 \pm 0.14	0.906 \pm 0.19	0.786 \pm 0.26
					StepwiseR	0.958 \pm 0.05	0.956 \pm 0.05	0.954 \pm 0.06	0.951 \pm 0.07	1.0 \pm 0.0
Graspl	Lapl	SN_MT	LEFT	ALPHA	DTree	0.932 \pm 0.09	0.929 \pm 0.1	0.915 \pm 0.14	0.909 \pm 0.15	0.629 \pm 0.29
					GAM	0.975 \pm 0.03	0.961 \pm 0.04	0.974 \pm 0.03	0.958 \pm 0.04	0.914 \pm 0.16
					Minfo	0.930 \pm 0.1	0.930 \pm 0.11	0.909 \pm 0.16	0.905 \pm 0.18	1.0 \pm 0.0
					RPlanet	0.922 \pm 0.11	0.925 \pm 0.11	0.895 \pm 0.19	0.895 \pm 0.21	0.843 \pm 0.17
					StepwiseR	0.946 \pm 0.05	0.943 \pm 0.05	0.941 \pm 0.06	0.938 \pm 0.06	1.0 \pm 0.0
Graspl	Lapl	SN_MT	LEFT	BETA	DTree	0.946 \pm 0.05	0.941 \pm 0.05	0.940 \pm 0.07	0.935 \pm 0.06	0.643 \pm 0.28
					GAM	0.975 \pm 0.03	0.970 \pm 0.03	0.974 \pm 0.03	0.968 \pm 0.03	0.943 \pm 0.16
					Minfo	0.931 \pm 0.12	0.924 \pm 0.12	0.898 \pm 0.23	0.889 \pm 0.23	1.0 \pm 0.0
					RPlanet	0.941 \pm 0.08	0.937 \pm 0.08	0.929 \pm 0.12	0.923 \pm 0.12	0.929 \pm 0.18
					StepwiseR	0.953 \pm 0.05	0.951 \pm 0.05	0.949 \pm 0.06	0.946 \pm 0.07	1.0 \pm 0.0
Graspl	Lapl	SN_MT	LEFT	GAMMA	DTree	0.954 \pm 0.05	0.952 \pm 0.06	0.949 \pm 0.06	0.945 \pm 0.09	0.543 \pm 0.24
					GAM	0.980 \pm 0.02	0.975 \pm 0.02	0.979 \pm 0.02	0.974 \pm 0.02	0.929 \pm 0.12
					Minfo	0.945 \pm 0.08	0.942 \pm 0.1	0.933 \pm 0.12	0.924 \pm 0.16	1.0 \pm 0.0
					RPlanet	0.947 \pm 0.06	0.949 \pm 0.08	0.940 \pm 0.08	0.939 \pm 0.11	0.857 \pm 0.21
					StepwiseR	0.953 \pm 0.06	0.951 \pm 0.06	0.948 \pm 0.07	0.944 \pm 0.08	1.0 \pm 0.0

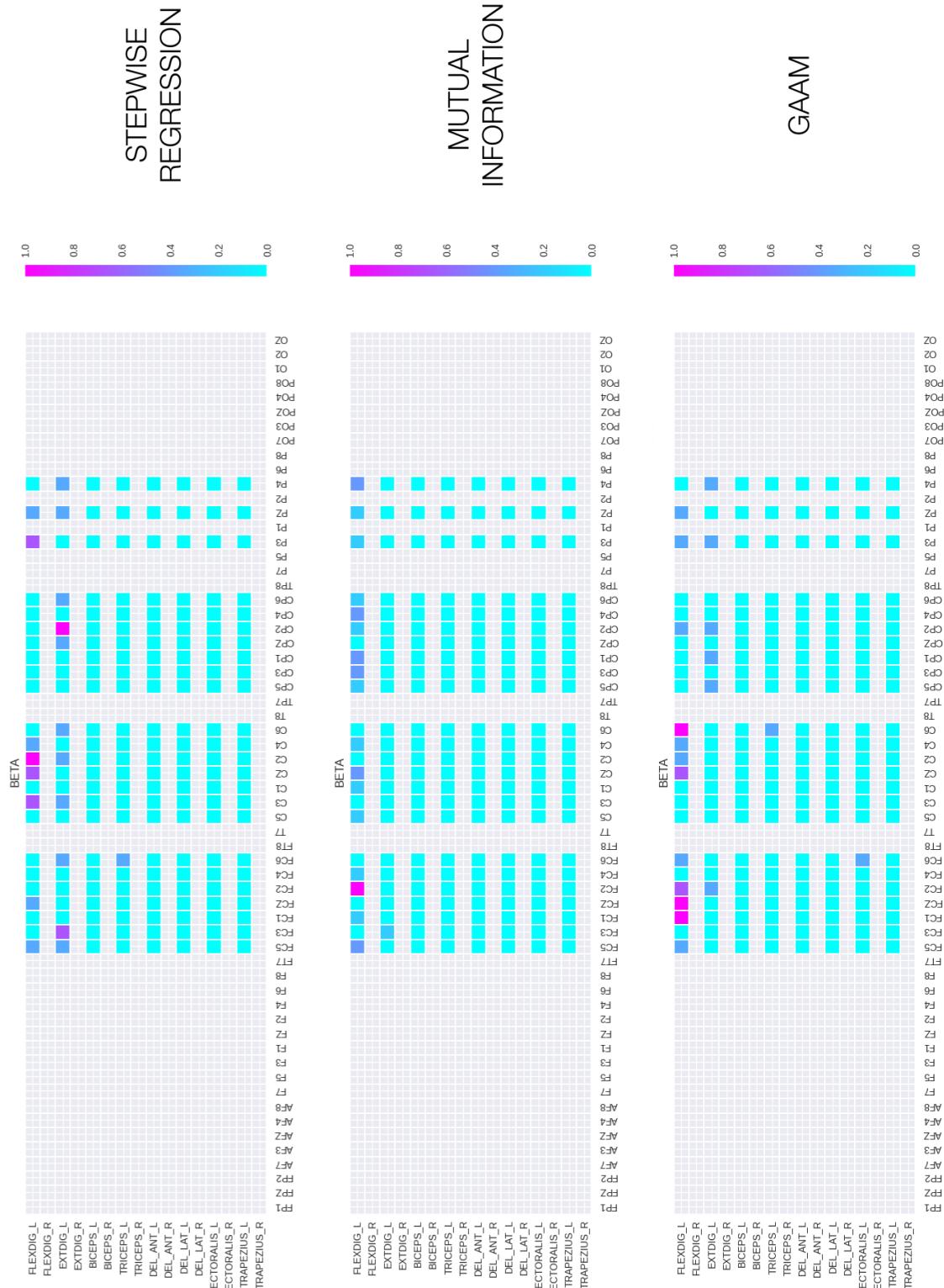


Figure 4.11. Heatmap features selected by each method in GraspL-Laplacian-SN_MT-LEFT-BETA dataset for all 14 subjects.

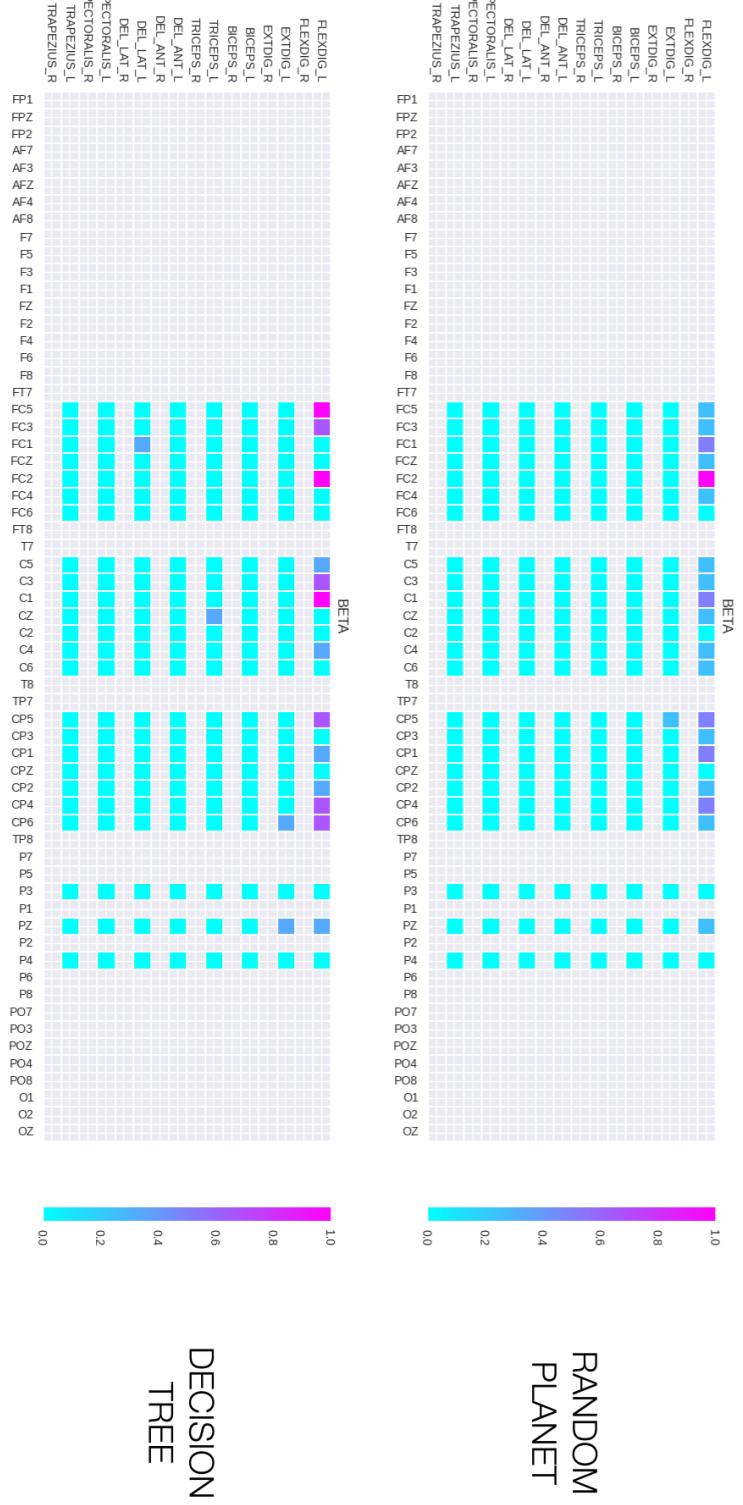


Figure 4.12. Heatmap features selected by each method in GraspL-Laplacian-SN_MT-LEFT-BETA dataset for all 14 subjects.

4.2.4 GraspR

In GraspR task, each subject performed finger grasping movement using the right hand. During this task, the target muscle is FLEXDIG_R of the right forearm (Fig. 1.14 in Section 1.1.2.2). This subsection is organized as the previous one.

4.2.4.1 NoFilter

For each subject, the datasets used for GraspR task without any EEG spatial filtering are the following ones. The difference among them was explained in Section 3.1.2:

- GraspR-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- GraspR-NoFilter-SN_MT-RIGHT-ALL_(BAND)
- GraspR-NoFilter-SN_MT-RIGHT-ALPHA
- GraspR-NoFilter-SN_MT-RIGHT-BETA
- GraspR-NoFilter-SN_MT-RIGHT-GAMMA

Table 4.22 shown the rmANOVA test performed. No statistical significant differences were found during post hoc test.

Table 4.22. Results of the rmANOVA tests (F(Num_DF, Den_DF) and P-value) across 14 healthy participants. The independent variable is the feature selection method used while the dependent variable is AUC in the test set. Color indicates significant statistic differences P-value ≤ 0.05 .

Task	Filter	EEG	EMG	Band		AUC Test
GraspR	NoFil	ALL	ALL	ALL	F(4.0, 52.0)	4.19
					P-value	.0051
GraspR	NoFil	SN_MT	RIGHT	ALL	F(4.0, 52.0)	2.129
					P-value	.0903
GraspR	NoFil	SN_MT	RIGHT	ALPHA	F(4.0, 52.0)	1.54
					P-value	.2042
GraspR	NoFil	SN_MT	RIGHT	BETA	F(4.0, 52.0)	1.968
					P-value	.1131
GraspR	NoFil	SN_MT	RIGHT	GAMMA	F(4.0, 52.0)	3.89
					P-value	.0077

Table 4.23 contains the complete report of the performance of each method. The heatmap for the selected features among the 14 healthy subjects is shown in Figs. 4.13, 4.14. GAAM is the algorithms which selects more features on the target muscle FLEXDIG_R. Stepwise Regression is the only method with more peaks in EXTDIG_R than in FLEXDIG_R.

Table 4.23. Soft Stability of each algorithm and classification performance obtained using the features selected by each method. Mean \pm std.

Task	Filter	EEG	EMG	Band	Method	AUC Train	AUC Test	F1 Train	F1 Test	SoftS
GraspR	NoFil	ALL	ALL	ALL	DTree	0.928 \pm 0.06	0.921 \pm 0.06	0.917 \pm 0.08	0.910 \pm 0.07	0.400 \pm 0.29
					GAM	0.981 \pm 0.02	0.968 \pm 0.02	0.980 \pm 0.02	0.966 \pm 0.02	0.443 \pm 0.17
					Minfo	0.951 \pm 0.05	0.947 \pm 0.06	0.946 \pm 0.06	0.939 \pm 0.07	1.0 \pm 0.0
					RPlanet	0.950 \pm 0.05	0.935 \pm 0.06	0.945 \pm 0.06	0.926 \pm 0.08	0.786 \pm 0.26
GraspR	NoFil	SN_MT	LEFT	ALL	StepwiseR	0.961 \pm 0.03	0.949 \pm 0.05	0.958 \pm 0.03	0.944 \pm 0.05	1.0 \pm 0.0
					DTree	0.935 \pm 0.04	0.935 \pm 0.05	0.928 \pm 0.05	0.927 \pm 0.07	0.486 \pm 0.34
					GAM	0.978 \pm 0.02	0.964 \pm 0.03	0.977 \pm 0.02	0.962 \pm 0.03	0.829 \pm 0.2
					Minfo	0.933 \pm 0.05	0.940 \pm 0.05	0.924 \pm 0.06	0.933 \pm 0.06	1.0 \pm 0.0
GraspR	NoFil	SN_MT	LEFT	ALPHA	RPlanet	0.938 \pm 0.04	0.935 \pm 0.04	0.931 \pm 0.05	0.929 \pm 0.05	0.814 \pm 0.22
					StepwiseR	0.953 \pm 0.04	0.948 \pm 0.04	0.949 \pm 0.05	0.944 \pm 0.05	1.0 \pm 0.0
					DTree	0.930 \pm 0.04	0.933 \pm 0.04	0.924 \pm 0.04	0.926 \pm 0.05	0.643 \pm 0.3
					GAM	0.960 \pm 0.03	0.943 \pm 0.03	0.958 \pm 0.03	0.940 \pm 0.03	0.900 \pm 0.16
GraspR	NoFil	SN_MT	LEFT	ALPHA	Minfo	0.934 \pm 0.04	0.930 \pm 0.04	0.929 \pm 0.04	0.923 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.930 \pm 0.03	0.934 \pm 0.04	0.925 \pm 0.04	0.928 \pm 0.04	0.843 \pm 0.2
					StepwiseR	0.925 \pm 0.04	0.920 \pm 0.04	0.918 \pm 0.05	0.912 \pm 0.05	1.0 \pm 0.0
					DTree	0.941 \pm 0.05	0.940 \pm 0.05	0.934 \pm 0.06	0.934 \pm 0.06	0.557 \pm 0.31
GraspR	NoFil	SN_MT	LEFT	BETA	GAM	0.970 \pm 0.03	0.961 \pm 0.03	0.968 \pm 0.03	0.959 \pm 0.03	0.900 \pm 0.16
					Minfo	0.931 \pm 0.06	0.929 \pm 0.07	0.921 \pm 0.08	0.917 \pm 0.09	1.0 \pm 0.0
					RPlanet	0.944 \pm 0.05	0.946 \pm 0.04	0.938 \pm 0.06	0.941 \pm 0.05	0.857 \pm 0.16
					StepwiseR	0.938 \pm 0.06	0.937 \pm 0.06	0.930 \pm 0.08	0.928 \pm 0.08	1.0 \pm 0.0
GraspR	NoFil	SN_MT	LEFT	GAMMA	DTree	0.938 \pm 0.04	0.940 \pm 0.04	0.931 \pm 0.05	0.934 \pm 0.05	0.471 \pm 0.29
					GAM	0.963 \pm 0.03	0.958 \pm 0.03	0.960 \pm 0.04	0.955 \pm 0.03	0.900 \pm 0.16
					Minfo	0.927 \pm 0.05	0.933 \pm 0.04	0.918 \pm 0.06	0.927 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.927 \pm 0.05	0.936 \pm 0.04	0.919 \pm 0.06	0.930 \pm 0.04	0.857 \pm 0.14
GraspR	NoFil	SN_MT	LEFT	GAMMA	StepwiseR	0.937 \pm 0.04	0.935 \pm 0.04	0.931 \pm 0.05	0.928 \pm 0.05	1.0 \pm 0.0

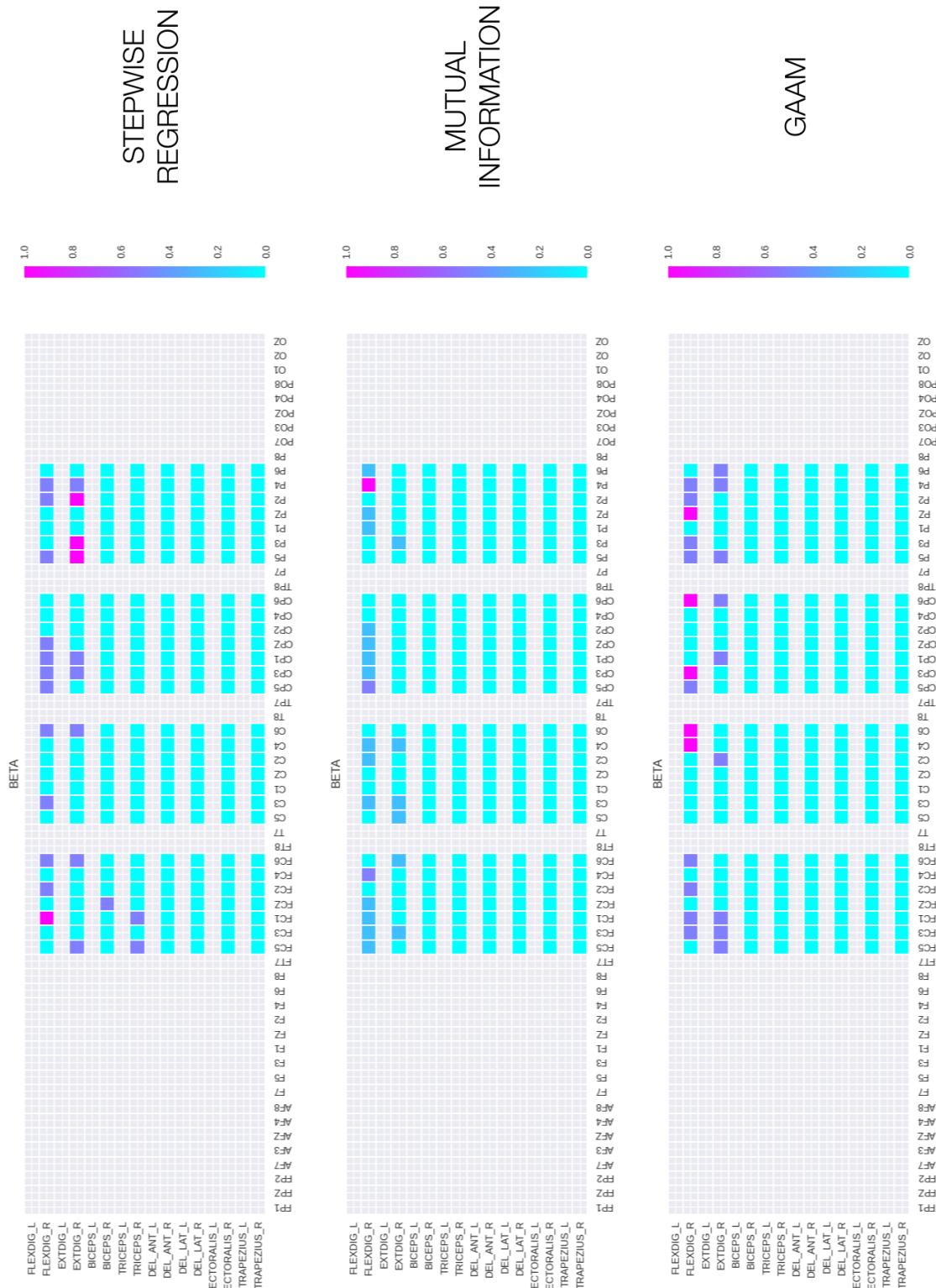


Figure 4.13. Heatmap features selected by each method in GraspR-NoFilter-SN_MT-RIGHT-BETA dataset for all 14 subjects.

GraspR_NoFilter_Sensorimotor_RIGHT_BETA

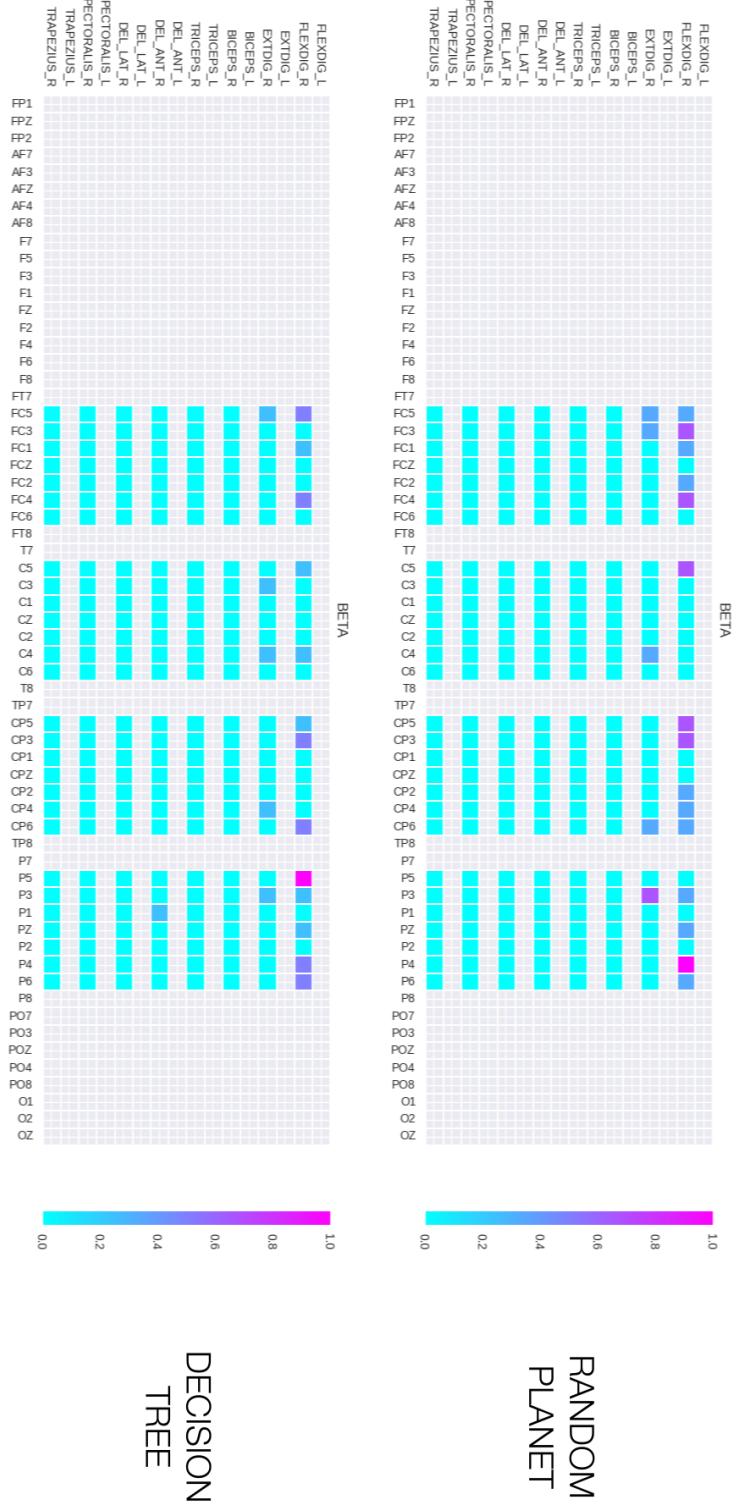


Figure 4.14. Heatmap features selected by each method in GraspR-NoFilter-SN_MT-RIGHT-BETA dataset for all 14 subjects.

4.2.4.2 Laplacian

For each subject, the datasets used for GraspR task with EEG Laplacian spatial filtering are the following ones. The difference among them was explained in Section 3.1.2:

- GraspR-Laplacian-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)
- GraspR-Laplacian-SN_MT-RIGHT-ALL_(BAND)
- GraspR-Laplacian-SN_MT-RIGHT-ALPHA
- GraspR-Laplacian-SN_MT-RIGHT-BETA
- GraspR-Laplacian-SN_MT-RIGHT-GAMMA

Table 4.24 shown the rmANOVA test performed. No statistical significant differences were found across the classification performance of the linear SVM using the various feature selection methods.

Table 4.24. Results of the rmANOVA tests ($F(\text{Num_DF}, \text{Den_DF})$ and P-value) across 14 healthy participants. The independent variable is the feature selection method used while the dependent variable is AUC in the test set.

Task	Filter	EEG	EMG	Band		AUC Test
GraspR	Lapl	ALL	ALL	ALL	$F(4.0, 52.0)$	2.23
					P-value	.0784
GraspR	Lapl	SN_MT	RIGHT	ALL	$F(4.0, 52.0)$	1.605
					P-value	.1870
GraspR	Lapl	SN_MT	RIGHT	ALPHA	$F(4.0, 52.0)$	2.029
					P-value	.1039
GraspR	Lapl	SN_MT	RIGHT	BETA	$F(4.0, 52.0)$	1.51
					P-value	.2128
GraspR	Lapl	SN_MT	RIGHT	GAMMA	$F(4.0, 52.0)$	1.32
					P-value	.2747

Table 4.25 contains the complete report of the performance of each method. The heatmaps for the selected features among the 14 healthy subjects are shown in Figs. 4.3, 4.4. The algorithm with more selection peaks on the target muscle FLEXDIG_R is Stepwise Regression. This method has also a peak on EXTDIG_R:P4.

Table 4.25. Soft Stability of each algorithm and classification performance obtained using the features selected by each method. Mean \pm std.

Task	Filter	EEG	EMG	Band	Method	AUC Train	AUC Test	F1 Train	F1 Test	SoftS
GraspR	Lapl	ALL	ALL	ALL	DTree	0.944 \pm 0.04	0.940 \pm 0.05	0.938 \pm 0.05	0.933 \pm 0.06	0.557 \pm 0.36
					GAM	0.985 \pm 0.02	0.974 \pm 0.02	0.984 \pm 0.02	0.973 \pm 0.03	0.786 \pm 0.21
					Minfo	0.952 \pm 0.04	0.951 \pm 0.04	0.947 \pm 0.05	0.946 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.954 \pm 0.04	0.949 \pm 0.05	0.949 \pm 0.05	0.943 \pm 0.06	0.757 \pm 0.28
					StepwiseR	0.946 \pm 0.07	0.941 \pm 0.08	0.939 \pm 0.09	0.933 \pm 0.1	1.0 \pm 0.0
GraspR	Lapl	SN_MT	LEFT	ALL	DTree	0.949 \pm 0.04	0.948 \pm 0.04	0.945 \pm 0.04	0.942 \pm 0.05	0.486 \pm 0.34
					GAM	0.983 \pm 0.02	0.969 \pm 0.02	0.983 \pm 0.02	0.967 \pm 0.03	0.814 \pm 0.24
					Minfo	0.952 \pm 0.04	0.953 \pm 0.04	0.948 \pm 0.05	0.948 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.955 \pm 0.04	0.955 \pm 0.05	0.951 \pm 0.05	0.950 \pm 0.06	0.729 \pm 0.26
					StepwiseR	0.946 \pm 0.07	0.942 \pm 0.07	0.939 \pm 0.09	0.934 \pm 0.09	1.0 \pm 0.0
GraspR	Lapl	SN_MT	LEFT	ALPHA	DTree	0.945 \pm 0.03	0.934 \pm 0.03	0.941 \pm 0.03	0.928 \pm 0.03	0.729 \pm 0.31
					GAM	0.973 \pm 0.03	0.957 \pm 0.03	0.971 \pm 0.03	0.954 \pm 0.03	0.943 \pm 0.14
					Minfo	0.956 \pm 0.02	0.950 \pm 0.03	0.953 \pm 0.03	0.947 \pm 0.03	1.0 \pm 0.0
					RPlanet	0.955 \pm 0.02	0.948 \pm 0.03	0.952 \pm 0.03	0.944 \pm 0.03	0.943 \pm 0.12
					StepwiseR	0.937 \pm 0.06	0.938 \pm 0.06	0.930 \pm 0.08	0.930 \pm 0.09	1.0 \pm 0.0
GraspR	Lapl	SN_MT	LEFT	BETA	DTree	0.950 \pm 0.03	0.947 \pm 0.04	0.947 \pm 0.03	0.942 \pm 0.04	0.543 \pm 0.35
					GAM	0.979 \pm 0.02	0.965 \pm 0.03	0.977 \pm 0.03	0.963 \pm 0.03	0.886 \pm 0.2
					Minfo	0.950 \pm 0.04	0.949 \pm 0.05	0.946 \pm 0.04	0.943 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.953 \pm 0.04	0.949 \pm 0.05	0.949 \pm 0.05	0.943 \pm 0.06	0.843 \pm 0.2
					StepwiseR	0.938 \pm 0.07	0.935 \pm 0.08	0.929 \pm 0.09	0.926 \pm 0.11	1.0 \pm 0.0
GraspR	Lapl	SN_MT	LEFT	GAMMA	DTree	0.955 \pm 0.04	0.952 \pm 0.05	0.951 \pm 0.04	0.947 \pm 0.06	0.443 \pm 0.31
					GAM	0.978 \pm 0.02	0.967 \pm 0.03	0.977 \pm 0.03	0.965 \pm 0.03	0.929 \pm 0.18
					Minfo	0.952 \pm 0.04	0.957 \pm 0.04	0.949 \pm 0.04	0.953 \pm 0.05	1.0 \pm 0.0
					RPlanet	0.955 \pm 0.03	0.956 \pm 0.03	0.952 \pm 0.04	0.953 \pm 0.04	0.829 \pm 0.22
					StepwiseR	0.961 \pm 0.03	0.949 \pm 0.03	0.958 \pm 0.04	0.945 \pm 0.04	1.0 \pm 0.0

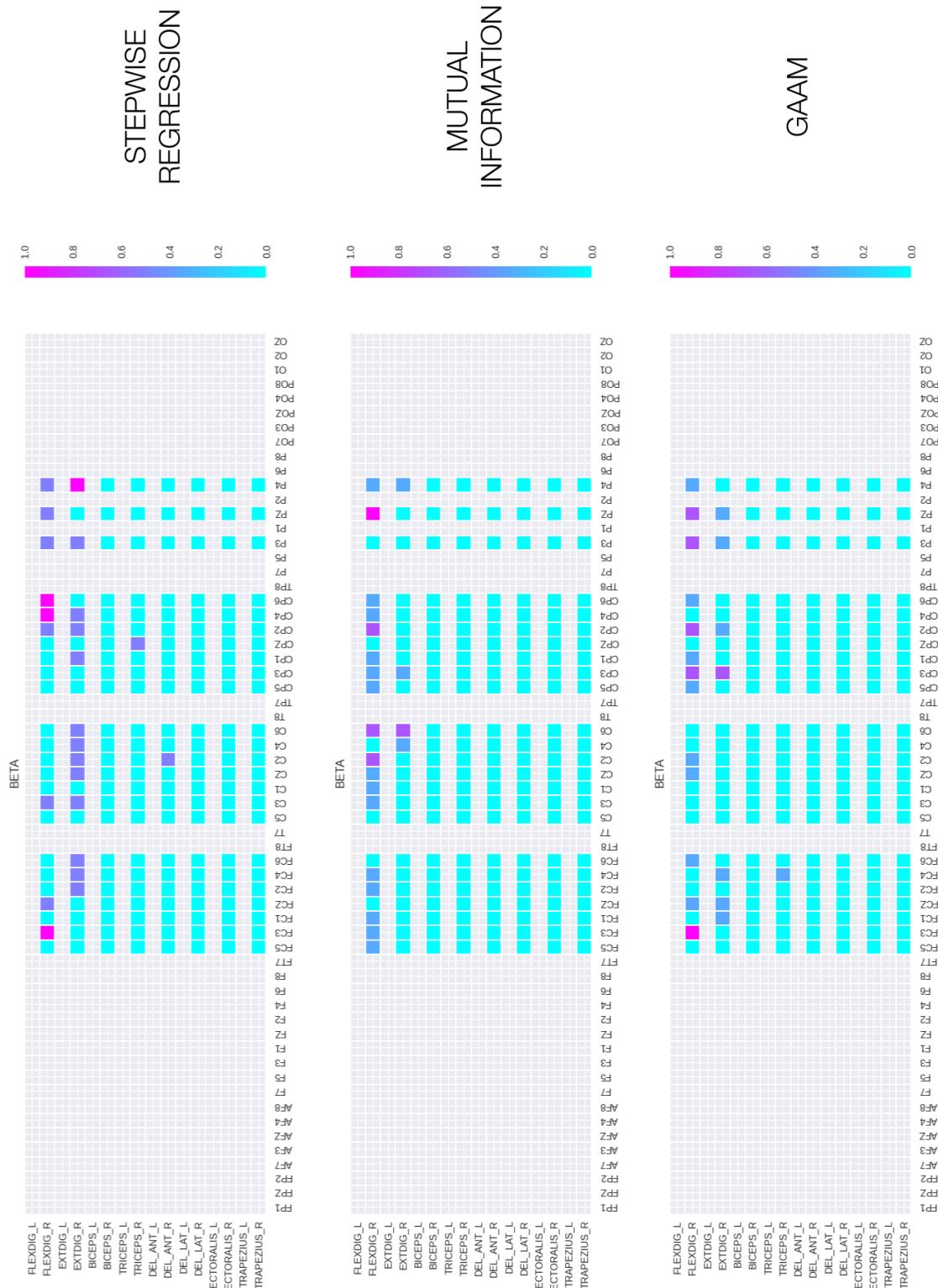
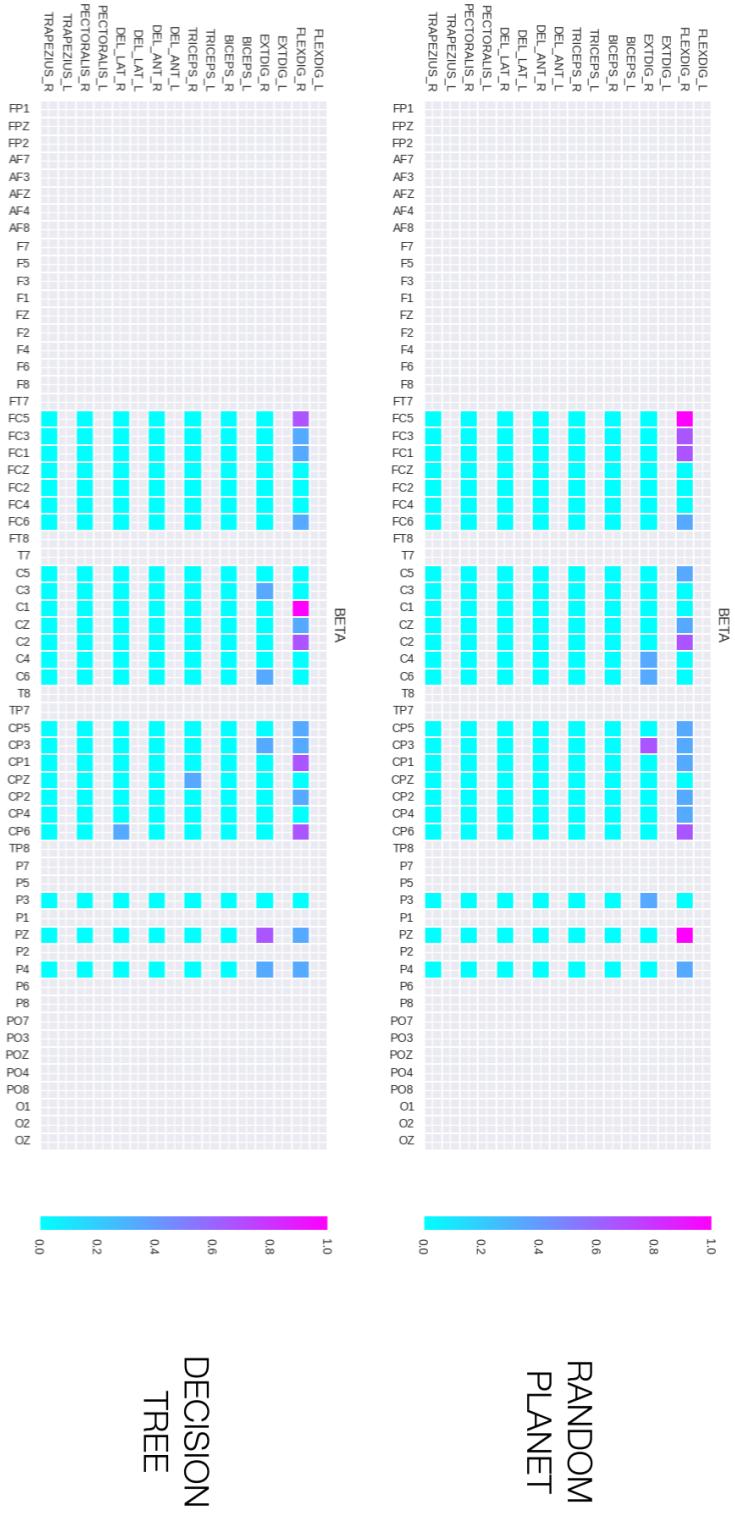


Figure 4.15. Heatmap features selected by each method in GraspR-Laplacian-SN_MT-RIGHT-BETA dataset for all 14 subjects.

GraspR_Laplacian_Sensorimotor__RIGHT_BETA

**Figure 4.16.** Heatmap features selected by each method in GraspR-Laplacian-SN_MT-RIGHT-BETA dataset for all 14 subjects.

Chapter 5

Discussion and conclusion

5.1 Discussion

The results reported in Section 4 shown that, after hyperparameter tuning, all the feature selection algorithms tested in this work (Decision Tree **DTree**, Genetic Algorithm with Aggressive Mutation **GAAM**, Mutual Information **MInfo**, Random Planet **RPlanet** and Stepwise Regression **StepwiseR**) allow to obtain high classification performance. Indeed, using a linear Support Vector Machine model trained on the reduced feature space, the performance shown in the test set are always higher than 90% on both AUC and F1 Score metrics. In most of the considered datasets, no statistical significant differences were found across the algorithms according to AUC metric calculated in the test set. The only significant differences were in a couple of datasets in **ExtL** and **GraspL** tasks. In **ExtL** task without any spatial filtering and considering the entire initial feature space (**ExtL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)**), **GAAM** shown higher performance than **RPlanet**, **MInfo** and **DTree**. The same difference was detected when restricting the initial feature space to the EEG electrodes located on and nearby the sensorimotor area (**ExtL-NoFilter-SN_MT-LEFT-BETA**). In this case though, also **StepwiseR** shown higher performance to other methods, namely **RPlanet** and **MInfo**. Finally, **GAAM** shown better AUC performance than **MInfo** also for **GraspL** task without spatial filtering and considering the entire initial feature space (**GraspL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND)**).

Concerning Soft Stability metric (SoftS), a metric defined in this study to measure how consistent is the decision of a feature selection method across several runs on the same exact dataset using different random seeds, **RPlanet** always shown higher SoftS with respect to **DTree** as expected. Indeed, **RPlanet**, a novel ensemble method for feature selection, was specifically proposed to overcome the limitations of Decision Tree and Random Forest models, namely their strongly influence from the random seed adopted. However, in the results obtained, the stability of **RPlanet** was never high as the one obtained by **StepwiseR** and **MInfo**. Also **GAAM** has a high stability (>80%) but only on datasets with preselection. Indeed, it has shown poor stability performance when the entire initial feature space was considered.

From an overall perspective, **GAAM** has the same high AUC performance of **StepwiseR**, the feature selection method used at IRCCS Santa Lucia Foundation, Rome, Italy. Indeed, there no exist statistical differences under this metric. However, **GAAM**

shown a noticeable lower stability which is an undesired property in a biomedical application. Concerning **DTree** and **RPlanet**, these two methods have similar performance under AUC metric but **RPlanet** has a higher stability. The only advantage of **DTree** over **RPlanet** is its high explainability that allows a clinician to easily understand the choices made during feature selection.

Restricting the analysis to only the datasets with preselection on the EEG electrodes over and near the sensorimotor area, with EMG channels from the single upper limb of interest, and on just the Beta band, graphic comparisons of the selected features across the algorithms were provided. In most of the cases, all the methods shown selection peaks on the target muscle, however different peaks were found also on the antagonist muscle. Neurophysiology expert consulting is necessary to further considerations on what algorithms select the most relevant features from a physiological point of view. This last is an important aspect to take into account in addition to AUC and SoftS metrics.

Future works should increase the number of subjects in order to have more data to highlight more differences across the classification performance each feature selection algorithm allows one to obtain. Moreover, the comparison should be extended to post-stroke patients considering also the features a neurophysiologist would select as reference.

5.2 Conclusion

In this study, the theoretical fundamentals of neuroengineering and signal acquisition of brain and muscle activities were provided. This chapter was necessary to make this text accessible to researches of artificial intelligence and data analysis fields without any background in neuroscience. After the review of the state-of-the-art feature selection algorithms, four methods were used and an additional novel technique was proposed and applied. The classification performance obtained using the five chosen methods were compared using statistical analysis methods which did not highlight any differences in most of the tested datasets. Finally, visual representations of what features each method is prone to select on healthy subjects, were reported in form of heatmaps.

Bibliography

- [1] Anatomy Encyclopedia. <https://learnmuscles.com/glossary/>.
- [2] Data Acquisition, Loggers, Amplifiers, Transducers, Electrodes | BIOPAC. <https://www.biopac.com/wp-content/uploads/EMG-Guide.pdf>.
- [3] DecisionTreeClassifier. Available from: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>.
- [4] Delsys – Wearable Sensors for Movement Sciences. <https://delsys.com/downloads/TECHNICALNOTE/101-emg-sensor-placement.pdf>.
- [5] Fit linear regression model using stepwise regression - MATLAB stepwisefit - MathWorks Italia. <https://it.mathworks.com/help/stats/stepwisefit.html>.
- [6] Herman & Wallace Pelvic Rehabilitation Continuing Education - Home. https://hermanwallace.com/download/The_ABC_of_EMG_by_Peter_Konrad.pdf.
- [7] MATLAB version: 23.2.0 (R2023b). The MathWorks Inc.
- [8] Nervous System (for Parents). <https://kidshealth.org/en/parents/brain-nervous-system.html>.
- [9] Overview of Stroke - Brain, Spinal Cord, and Nerve Disorders. <https://www.msdmanuals.com/home/brain-spinal-cord-and-nerve-disorders/stroke/overview-of-stroke>.
- [10] RandomForestClassifier. Available from: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.
- [11] Rank features for classification using minimum redundancy maximum relevance (MRMR) algorithm - MATLAB fscmrmr - MathWorks Italia. <https://it.mathworks.com/help/stats/fscmrmr.html>.
- [12] *Introduction to the Various Connectivity Analyses*, pp. 339–352. The MIT Press (2014). ISBN 978-0-262-31955-3. doi:10.7551/mitpress/9609.003.0032.
- [13] *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, Mass (2023). ISBN 978-0-262-31142-7.

- [14] ABDULKADER, S., ATIA, A., AND MOSTAFA, M.-S. Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, **16** (2015), 213. doi:10.1016/j.eij.2015.06.002.
- [15] ABOALAYON, K., FAEZIPOUR, M., ALMUHAMMADI, W., AND MOSLEHPOUR, S. Sleep stage classification using EEG signal analysis: A comprehensive survey and new investigation. *Entropy*, **18** (2016). doi:10.3390/e18090272.
- [16] AMARASINGHE, K., SIVILS, P., AND MANIC, M. EEG feature selection for thought driven robots using evolutionary Algorithms. In *2016 9th International Conference on Human System Interactions (HSI)*, pp. 355–361. IEEE, Portsmouth, United Kingdom (2016). ISBN 978-1-5090-1729-4. doi:10.1109/HSI.2016.7529657.
- [17] ANG, K. K., CHIN, Z. Y., WANG, C., GUAN, C., AND ZHANG, H. Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b. *Frontiers in Neuroscience*, **6** (2012). doi:10.3389/fnins.2012.00039.
- [18] ANITHA, T., SHANTHI, N., SATHIYASHEELAN, R., EMAYAVARAMBAN, G., AND RAJENDRAN, T. Brain-computer interface for persons with motor disabilities - A review. *Open Biomedical Engineering Journal*, **13** (2019), 127. doi:10.2174/1874120701913010127.
- [19] ATYABI, A., LUERSSEN, M. H., FITZGIBBON, S. P., AND POWERS, D. M. W. The Impact of PSO Based Dimension Reduction on EEG Classification. In *Brain Informatics* (edited by D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, F. M. Zanzotto, S. Tsumoto, N. Taatgen, and Y. Yao), vol. 7670, pp. 220–231. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). ISBN 978-3-642-35138-9 978-3-642-35139-6. doi:10.1007/978-3-642-35139-6_21.
- [20] ATYABI, A., LUERSSEN, M. H., AND POWERS, D. M. PSO-based dimension reduction of EEG recordings: Implications for subject transfer in BCI. *Neurocomputing*, **119** (2013), 319. doi:10.1016/j.neucom.2013.03.027.
- [21] ATYABI, A., SHIC, F., AND NAPLES, A. Mixture of autoregressive modeling orders and its implication on single trial EEG classification. *Expert Systems with Applications*, **65** (2016), 164. doi:10.1016/j.eswa.2016.08.044.
- [22] BABLANI, A., EDLA, D., TRIPATHI, D., AND CHERUKU, R. Survey on brain-computer interface: An emerging computational intelligence paradigm. *ACM Computing Surveys*, **52** (2019). doi:10.1145/3297713.
- [23] BAIG, M. Z., ASLAM, N., SHUM, H. P., AND ZHANG, L. Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery EEG. *Expert Systems with Applications*, **90** (2017), 184. doi:10.1016/j.eswa.2017.07.033.

- [24] BASHASHATI, A., FATOURECHI, M., WARD, R. K., AND BIRCH, G. E. A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. *Journal of Neural Engineering*, **4** (2007), R32. doi:10.1088/1741-2560/4/2/R03.
- [25] BATTITI, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, **5** (1994), 537. doi:10.1109/72.298224.
- [26] BENJAMINI, Y. AND YEKUTIELI, D. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, **29** (2001). doi:10.1214/aos/1013699998.
- [27] BEVANS, R. Multiple Linear Regression | A Quick Guide (Examples). <https://www.scribbr.com/statistics/multiple-linear-regression/> (2020).
- [28] BOLÓN-CANEDO, V., SÁNCHEZ-MAROÑO, N., AND ALONSO-BETANZOS, A. A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, **34** (2013), 483. doi:10.1007/s10115-012-0487-8.
- [29] BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A., AND STONE, C. J. *Classification And Regression Trees*. Routledge, 1 edn. (2017). ISBN 978-1-315-13947-0. doi:10.1201/9781315139470.
- [30] COLAMARINO, E., DE SETA, V., MASCIULLO, M., CINCOTTI, F., MATTIA, D., PICHIORRI, F., AND TOPPI, J. Corticomuscular and Intermuscular Coupling in Simple Hand Movements to Enable a Hybrid Brain–Computer Interface. *International Journal of Neural Systems*, **31** (2021), 2150052. doi:10.1142/S0129065721500520.
- [31] CORRALEJO, R., HORNERO, R., AND ALVAREZ, D. Feature selection using a genetic algorithm in a motor imagery-based Brain Computer Interface. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 7703–7706. IEEE, Boston, MA (2011). ISBN 978-1-4244-4121-1 978-1-4577-1589-1. doi:10.1109/IEMBS.2011.6091898.
- [32] COVER, T. M. AND THOMAS, J. A. *Elements of Information Theory*. Wiley Series in Telecommunications. J. Wiley & sons, New York [etc.] (1991). ISBN 978-0-471-06259-2.
- [33] DE SETA, V., TOPPI, J., COLAMARINO, E., MOLLE, R., CASTELLANI, F., CINCOTTI, F., MATTIA, D., AND PICHIORRI, F. Cortico-muscular coupling to control a hybrid brain-computer interface for upper limb motor rehabilitation: A pseudo-online study on stroke patients. *Frontiers in Human Neuroscience*, **16** (2022), 1016862. doi:10.3389/fnhum.2022.1016862.
- [34] DE SETA, V., TOPPI, J., PICHIORRI, F., MASCIULLO, M., COLAMARINO, E., MATTIA, D., AND CINCOTTI, F. Towards a hybrid EEG-EMG feature for the classification of upper limb movements: Comparison of different processing pipelines. In *2021 10th International IEEE/EMBS Conference on Neural*

- Engineering (NER)*, pp. 355–358. IEEE, Italy (2021). ISBN 978-1-72814-337-8.
doi:10.1109/NER49283.2021.9441390.
- [35] DIAS, N. S., KAMRUNNAHAR, M., MENDES, P. M., SCHIFF, S. J., AND CORREIA, J. H. Feature selection on movement imagery discrimination and attention detection. *Medical & Biological Engineering & Computing*, **48** (2010), 331. doi:10.1007/s11517-010-0578-1.
- [36] DING, C. AND PENG, H. MINIMUM REDUNDANCY FEATURE SELECTION FROM MICROARRAY GENE EXPRESSION DATA. *Journal of Bioinformatics and Computational Biology*, **03** (2005), 185. doi:10.1142/S0219720005001004.
- [37] DRAPER, N. R. AND SMITH, H. *Applied Regression Analysis*. Wiley Series in Probability and Statistics. Wiley, New York, 3rd ed edn. (1998). ISBN 978-0-471-17082-2.
- [38] FISHER, R. A. THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. *Annals of Eugenics*, **7** (1936), 179. doi:10.1111/j.1469-1809.1936.tb02137.x.
- [39] GAD, A. F. PyGAD: An Intuitive Genetic Algorithm Python Library (2021). doi:10.48550/ARXIV.2106.06158.
- [40] GÉRON, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Covid-19 Collection. O'Reilly, Beijing Boston Farnham Sebastopol Tokyo, second edition edn. (2019). ISBN 978-1-4920-3264-9 978-1-09-812597-4.
- [41] GOHEL, V. AND MEHENDALE, N. Review on electromyography signal acquisition and processing. *Biophysical Reviews*, **12** (2020), 1361. doi:10.1007/s12551-020-00770-w.
- [42] HARI, R. AND PUCE, A. *MEG-EEG Primer*. Oxford University Press, New York, NY (2017). ISBN 978-0-19-049777-4.
- [43] HASAN, B. A. S. AND GAN, J. Q. Unsupervised adaptive GMM for BCI. In *2009 4th International IEEE/EMBS Conference on Neural Engineering*, pp. 295–298. IEEE, Antalya, Turkey (2009). ISBN 978-1-4244-2072-8. doi:10.1109/NER.2009.5109291.
- [44] HEKMATMANESH, A., WU, H., JAMALOO, F., LI, M., AND HANDROOS, H. A combination of CSP-based method with soft margin SVM classifier and generalized RBF kernel for imagery-based brain computer interface applications. *Multimedia Tools and Applications*, **79** (2020), 17521. doi:10.1007/s11042-020-08675-2.
- [45] HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI (1975).

- [46] HOMRI, I. AND YACOUB, S. A hybrid cascade method for EEG classification. *Pattern Analysis and Applications*, **22** (2019), 1505. doi:10.1007/s10044-018-0737-9.
- [47] HSU, W.-Y. Improving classification accuracy of motor imagery EEG using genetic feature selection. *Clinical EEG and neuroscience*, **45** (2014), 163. doi:10.1177/1550059413491559.
- [48] HSU, W.-Y. AND HU, Y.-P. Artificial Bee Colony Algorithm for Single-Trial Electroencephalogram Analysis. *Clinical EEG and Neuroscience*, **46** (2015), 119. doi:10.1177/1550059414538808.
- [49] HU, L. AND ZHANG, Z. (eds.). *EEG Signal Processing and Feature Extraction*. Springer Singapore, Singapore (2019). ISBN 9789811391125 9789811391132. doi:10.1007/978-981-13-9113-2.
- [50] JIN, J., XIAO, R., DALY, I., MIAO, Y., WANG, X., AND CICHOCKI, A. Internal Feature Selection Method of CSP Based on L1-Norm and Dempster–Shafer Theory. *IEEE Transactions on Neural Networks and Learning Systems*, **32** (2021), 4814. doi:10.1109/TNNLS.2020.3015505.
- [51] KAI KENG ANG, ZHANG YANG CHIN, HAIHONG ZHANG, AND CUNTAI GUAN. Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 2390–2397. IEEE, Hong Kong, China (2008). ISBN 978-1-4244-1820-6. doi:10.1109/IJCNN.2008.4634130.
- [52] KASHEF, S., NEZAMABADI-POUR, H., AND NIKPOUR, B. Multilabel feature selection: A comprehensive review and guiding experiments. *WIREs Data Mining and Knowledge Discovery*, **8** (2018), e1240. doi:10.1002/widm.1240.
- [53] KHADEMI, Z., EBRAHIMI, F., AND KORDY, H. A review of critical challenges in MI-BCI: From conventional to deep learning methods. *Journal of Neuroscience Methods*, **383** (2023). doi:10.1016/j.jneumeth.2022.109736.
- [54] KHAN, S., KHAN, A., AND FAROOQ, O. Selection of features and classifiers for EMG-EEG-Based upper limb assistive devices - A review. *IEEE Reviews in Biomedical Engineering*, **13** (2020), 248. doi:10.1109/RBME.2019.2950897.
- [55] KILHO, S., DANNY, F., AND SEIYA, M. Consistency Measures for Feature Selection: A Formal Definition, Relative Sensitivity Comparison and a Fast Algorithm. doi:10.5591/978-1-57735-516-8/IJCAI11-251.
- [56] KIMOVSKI, D., ORTEGA, J., ORTIZ, A., AND BANOS, R. Feature selection in high-dimensional EEG data by parallel multi-objective optimization. In *2014 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 314–322. IEEE, Madrid, Spain (2014). ISBN 978-1-4799-5548-0. doi:10.1109/CLUSTER.2014.6968782.

- [57] KOPRINSKA, I. Feature Selection for Brain-Computer Interfaces. In *New Frontiers in Applied Data Mining* (edited by D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, T. Theeramunkong, C. Nattee, P. J. L. Adeodato, N. Chawla, P. Christen, P. Lenca, J. Poon, and G. Williams), vol. 5669, pp. 106–117. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). ISBN 978-3-642-14639-8 978-3-642-14640-4. doi:10.1007/978-3-642-14640-4_8.
- [58] LAKANY, H. AND CONWAY, B. Understanding intention of movement from electroencephalograms. *Expert Systems*, **24** (2007), 295. doi:10.1111/j.1468-0394.2007.00435.x.
- [59] LAL, T., SCHRODER, M., HINTERBERGER, T., WESTON, J., BOGDAN, M., BIRBAUMER, N., AND SCHOLKOPF, B. Support Vector Channel Selection in BCI. *IEEE Transactions on Biomedical Engineering*, **51** (2004), 1003. doi:10.1109/TBME.2004.827827.
- [60] LAZCANO-HERRERA, A., FUENTES-AGUILAR, R., CHAIREZ, I., ALONSO-VALERDI, L., GONZALEZ-MENDOZA, M., AND ALFARO-PONCE, M. Review on BCI Virtual Rehabilitation and Remote Technology Based on EEG for Assistive Devices. *Applied Sciences (Switzerland)*, **12** (2022). doi:10.3390/app122312253.
- [61] LOTTE, F., BOUGRAIN, L., CICHOCKI, A., CLERC, M., CONGEDO, M., RAKOTOMAMONJY, A., AND YGER, F. A review of classification algorithms for EEG-based brain-computer interfaces: A 10 year update. *Journal of Neural Engineering*, **15** (2018). doi:10.1088/1741-2552/aab2f2.
- [62] MÄHL, C., JEUNET, C., AND LOTTE, F. EEG-based workload estimation across affective contexts. *Frontiers in Neuroscience*, **8** (2014). doi:10.3389/fnins.2014.00114.
- [63] MCFARLAND, D. J., MCCANE, L. M., DAVID, S. V., AND WOLPAW, J. R. Spatial filter selection for EEG-based communication. *Electroencephalography and Clinical Neurophysiology*, **103** (1997), 386. doi:10.1016/S0013-4694(97)00022-2.
- [64] MRIDHA, M., DAS, S., KABIR, M., LIMA, A., ISLAM, MD.R., AND WATANOBE, Y. Brain-computer interface: Advancement and challenges. *Sensors*, **21** (2021). doi:10.3390/s21175746.
- [65] MÜLLER, K. R., KRAULEDAT, M., DORNHEGE, G., CURIO, G., AND BLANKERTZ, B. Machine Learning and Applications for Brain-Computer Interfacing. In *Human Interface and the Management of Information. Methods, Techniques and Tools in Information Design* (edited by D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, M. J. Smith, and G. Salvendy), vol. 4557,

- pp. 705–714. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). ISBN 978-3-540-73344-7 978-3-540-73345-4. doi:10.1007/978-3-540-73345-4_80.
- [66] NEUPER, C., SCHERER, R., WRIESSNEGGER, S., AND PFURTSCHELLER, G. Motor imagery and action observation: Modulation of sensorimotor brain rhythms during mental control of a brain–computer interface. *Clinical Neurophysiology*, **120** (2009), 239. doi:10.1016/j.clinph.2008.11.015.
- [67] NICOLAS-ALONSO, L. F. AND GOMEZ-GIL, J. Brain Computer Interfaces, a Review. *Sensors*, **12** (2012), 1211. doi:10.3390/s120201211.
- [68] NINOMIYA, T., INOUE, K.-I., HOSHI, E., AND TAKADA, M. Layer specificity of inputs from supplementary motor area and dorsal premotor cortex to primary motor cortex in macaque monkeys. *Scientific Reports*, **9** (2019), 18230. doi:10.1038/s41598-019-54220-z.
- [69] NOVI, Q., GUAN, C., DAT, T. H., AND XUE, P. Sub-band Common Spatial Pattern (SBCSP) for Brain-Computer Interface. In *2007 3rd International IEEE/EMBS Conference on Neural Engineering*, pp. 204–207. IEEE, Kohala Coast, HI (2007). ISBN 978-1-4244-0791-0. doi:10.1109/CNE.2007.369647.
- [70] ORBAN, M., ELSAMANTY, M., GUO, K., ZHANG, S., AND YANG, H. A Review of Brain Activity and EEG-Based Brain–Computer Interfaces for Rehabilitation Application. *Bioengineering*, **9** (2022). doi:10.3390/bioengineering9120768.
- [71] ORTEGA, J., ASENSIO-CUBERO, J., GAN, J. Q., AND ORTIZ, A. Evolutionary Multiobjective Feature Selection in Multiresolution Analysis for BCI. In *Bioinformatics and Biomedical Engineering* (edited by F. Ortúñoz and I. Rojas), vol. 9043, pp. 347–359. Springer International Publishing, Cham (2015). ISBN 978-3-319-16482-3 978-3-319-16483-0. doi:10.1007/978-3-319-16483-0_35.
- [72] ORTEGA, J., ASENSIO-CUBERO, J., GAN, J. Q., AND ORTIZ, A. Classification of motor imagery tasks for BCI with multiresolution analysis and multiobjective feature selection. *BioMedical Engineering OnLine*, **15** (2016), 73. doi:10.1186/s12938-016-0178-x.
- [73] PADFIELD, N., ZABALZA, J., ZHAO, H., MASERO, V., AND REN, J. EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges. *Sensors*, **19** (2019), 1423. doi:10.3390/s19061423.
- [74] PAHUJA, S. AND VEER, K. Recent Approaches on Classification and Feature Extraction of EEG Signal: A Review. *Robotica*, **40** (2022), 77. doi:10.1017/S0263574721000382.
- [75] PALUMBO, A., GRAMIGNA, V., CALABRESE, B., AND IELPO, N. Motor-imagery EEG-based BCIs in wheelchair movement and control: A systematic literature review. *Sensors*, **21** (2021). doi:10.3390/s21186285.
- [76] PANG, L. Hallucinations Experienced by Visually Impaired: Charles Bonnet Syndrome. *Optometry and Vision Science*, **93** (2016), 1466. doi:10.1097/OPX.0000000000000959.

- [77] PEDREGOSA, F., ET AL. Scikit-learn: Machine Learning in Python. (2012). doi:[10.48550/ARXIV.1201.0490](https://doi.org/10.48550/ARXIV.1201.0490).
- [78] PICHIORRI, F., TOPPI, J., DE SETA, V., COLAMARINO, E., MASCIULLO, M., TAMBURELLA, F., LORUSSO, M., CINCOTTI, F., AND MATTIA, D. Exploring high-density corticomuscular networks after stroke to enable a hybrid Brain-Computer Interface for hand motor rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, **20** (2023), 5. doi:[10.1186/s12984-023-01127-6](https://doi.org/10.1186/s12984-023-01127-6).
- [79] PICHIORRI, F., ET AL. Brain-computer interface boosts motor imagery practice during stroke recovery. *Annals of Neurology*, **77** (2015), 851. doi:[10.1002/ana.24390](https://doi.org/10.1002/ana.24390).
- [80] PUDIL, P., NOVOVIČOVÁ, J., AND KITTNER, J. Floating search methods in feature selection. *Pattern Recognition Letters*, **15** (1994), 1119. doi:[10.1016/0167-8655\(94\)90127-9](https://doi.org/10.1016/0167-8655(94)90127-9).
- [81] QUINLAN, J. R. Induction of decision trees. *Machine Learning*, **1** (1986), 81. doi:[10.1007/BF00116251](https://doi.org/10.1007/BF00116251).
- [82] QUITADAMO, L., CAVRINI, F., SBERNINI, L., RIILLO, F., BIANCHI, L., SERI, S., AND SAGGIO, G. Support vector machines to detect physiological patterns for EEG and EMG-based human-computer interaction: A review. *Journal of Neural Engineering*, **14** (2017). doi:[10.1088/1741-2552/14/1/011001](https://doi.org/10.1088/1741-2552/14/1/011001).
- [83] RAKSHIT, P., BHATTACHARYYA, S., KONAR, A., KHASNOBISH, A., TIBAREWALA, D. N., AND JANARTHANAN, R. Artificial Bee Colony Based Feature Selection for Motor Imagery EEG Data. In *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)* (edited by J. C. Bansal, P. Singh, K. Deep, M. Pant, and A. Nagar), vol. 202, pp. 127–138. Springer India, India (2013). ISBN 978-81-322-1040-5 978-81-322-1041-2. doi:[10.1007/978-81-322-1041-2_11](https://doi.org/10.1007/978-81-322-1041-2_11).
- [84] RAMOS, A. C., HERNANDEZ, R. G., AND VELLASCO, M. Feature Selection methods applied to Motor Imagery task classification. In *2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1–6. IEEE, Cartagena, Colombia (2016). ISBN 978-1-5090-5105-2. doi:[10.1109/LA-CCI.2016.7885731](https://doi.org/10.1109/LA-CCI.2016.7885731).
- [85] RASHID, M., SULAIMAN, N., P. P. ABDUL MAJEED, A., MUSA, R., AB. NASIR, A., BARI, B., AND KHATUN, S. Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface: A Comprehensive Review. *Frontiers in Neurorobotics*, **14** (2020). doi:[10.3389/fnbot.2020.00025](https://doi.org/10.3389/fnbot.2020.00025).
- [86] REJER, I. Genetic algorithm with aggressive mutation for feature selection in BCI feature space. *Pattern Analysis and Applications*, **18** (2015), 485. doi:[10.1007/s10044-014-0425-3](https://doi.org/10.1007/s10044-014-0425-3).

- [87] REMESEIRO, B. AND BOLON-CANEDO, V. A review of feature selection methods in medical applications. *Computers in Biology and Medicine*, **112** (2019). doi:10.1016/j.combiomed.2019.103375.
- [88] ROSA, M. AND SEYMOUR, B. Decoding the matrix: Benefits and limitations of applying machine learning algorithms to pain neuroimaging. *Pain*, **155** (2014), 864. doi:10.1016/j.pain.2014.02.013.
- [89] ROSNER, B. Fundamentals of Biostatistics. (2010).
- [90] ROSS, B. C. Mutual Information between Discrete and Continuous Data Sets. *PLoS ONE*, **9** (2014), e87357. doi:10.1371/journal.pone.0087357.
- [91] SAEIDI, M., KARWOWSKI, W., FARAHANI, F., FIOK, K., TAIAR, R., HANCOCK, P., AND AL-JUAID, A. Neural decoding of eeg signals with machine learning: A systematic review. *Brain Sciences*, **11** (2021). doi:10.3390/brainsci11111525.
- [92] SCRIVENER, C. L. AND READER, A. T. Variability of EEG electrode positions and their underlying brain regions: Visualizing gel artifacts from a simultaneous EEG-fMRI dataset. *Brain and Behavior*, **12** (2022), e2476. doi:10.1002/brb3.2476.
- [93] SINGH, A., HUSSAIN, A. A., LAL, S., AND GUESGEN, H. W. A Comprehensive Review on Critical Issues and Possible Solutions of Motor Imagery Based Electroencephalography Brain-Computer Interface. *Sensors (Basel, Switzerland)*, **21** (2021). doi:10.3390/s21062173.
- [94] SU, Y., LI, W., BI, N., AND LV, Z. Adolescents Environmental Emotion Perception by Integrating EEG and Eye Movements. *Frontiers in Neurorobotics*, **13** (2019), 46. doi:10.3389/fnbot.2019.00046.
- [95] SUBASI, A. AND ISMAIL GURSOY, M. EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Systems with Applications*, **37** (2010), 8659. doi:10.1016/j.eswa.2010.06.065.
- [96] TAHERNEZHAD-JAVAZM, F., AZIMIRAD, V., AND SHOARAN, M. A review and experimental study on the application of classifiers and evolutionary algorithms in EEG-based brain-machine interface systems. *Journal of Neural Engineering*, **15** (2018). doi:10.1088/1741-2552/aa8063.
- [97] TIN KAM HO. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282. IEEE Comput. Soc. Press, Montreal, Que., Canada (1995). ISBN 978-0-8186-7128-9. doi:10.1109/ICDAR.1995.598994.
- [98] TIWARI, N., EDLA, D., DODIA, S., AND BABLANI, A. Brain computer interface: A comprehensive survey. *Biologically Inspired Cognitive Architectures*, **26** (2018), 118. doi:10.1016/j.bica.2018.10.005.

- [99] TOWNSEND, G., GRAIMANN, B., AND PFURTSCHELLER, G. A Comparison of Common Spatial Patterns With Complex Band Power Features in a Four-Class BCI Experiment. *IEEE Transactions on Biomedical Engineering*, **53** (2006), 642. doi:10.1109/TBME.2006.870237.
- [100] TSUCHIMOTO, S., SHIBUSAWA, S., IWAMA, S., HAYASHI, M., OKUYAMA, K., MIZUGUCHI, N., KATO, K., AND USHIBA, J. Use of common average reference and large-Laplacian spatial-filters enhances EEG signal-to-noise ratios in intrinsic sensorimotor activity. *Journal of Neuroscience Methods*, **353** (2021), 109089. doi:10.1016/j.jneumeth.2021.109089.
- [101] UDHAYA KUMAR, S. AND HANNAH INBARANI, H. PSO-based feature selection and neighborhood rough set-based classification for BCI multiclass motor imagery task. *Neural Computing and Applications*, **28** (2017), 3239. doi:10.1007/s00521-016-2236-5.
- [102] VAUGHAN, T., MCFARLAND, D., SCHALK, G., SARNACKI, W., KRUSIENSKI, D., SELLERS, E., AND WOLPAW, J. The wadsworth BCI research and development program: At home with BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **14** (2006), 229. doi:10.1109/TNSRE.2006.875577.
- [103] WOLPAW, J. AND WOLPAW, E. W. *Brain-Computer InterfacesPrinciples and Practice*. Oxford University Press (2012). ISBN 978-0-19-538885-5. doi:10.1093/acprof:oso/9780195388855.001.0001.
- [104] YADAV, D., YADAV, S., AND VEER, K. A comprehensive assessment of Brain Computer Interfaces: Recent trends and challenges. *Journal of Neuroscience Methods*, **346** (2020). doi:10.1016/j.jneumeth.2020.108918.
- [105] YU, X., PARK, S.-M., KO, K.-E., AND SIM, K.-B. Discriminative Power Feature Selection Method for Motor Imagery EEG Classification in Brain Computer Interface Systems. *International Journal of Fuzzy Logic and Intelligent Systems*, **13** (2013), 12. doi:10.5391/IJFIS.2013.13.1.12.
- [106] YUAN, H. AND HE, B. Brain-computer interfaces using sensorimotor rhythms: Current state and future perspectives. *IEEE Transactions on Biomedical Engineering*, **61** (2014), 1425. doi:10.1109/TBME.2014.2312397.

Appendices

.1 Tables hyperparameter tuning Random Planet

Table .1. 1/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

	NF100_NE20_NT2	NF100_NE50_NT2	NF100_NE70_NT2	NF150_NE20_NT2	NF150_NE50_NT2	NF150_NE70_NT2	NF1_NE20_NT2	NF1_NE50_NT2	NF1_NE70_NT2	NF20_NE20_NT2	NF20_NE50_NT2	NF20_NE70_NT2
NF100_NE20_NT2	-	1	1	1	1	1	1	1	1	1	1	1
NF100_NE50_NT2	1	-	.7621	1	1	.9494	1	1	1	1	1	1
NF100_NE70_NT2	1	1	-	1	1	1	1	1	1	1	1	1
NF150_NE20_NT2	1	1	.8973	-	1	.7733	1	1	1	1	1	1
NF150_NE50_NT2	1	1	1	1	-	1	1	1	1	1	1	1
NF150_NE70_NT2	1	1	1	1	1	-	1	1	1	1	1	1
NF1_NE20_NT2	.1223	.1223	.1223	.1223	.1223	-	.2114	.1484	.1223	.1223	.1223	.1223
NF1_NE50_NT2	.1223	.1223	.1223	.1223	.1223	1	-	1	.1385	.1223	.1223	.1223
NF1_NE70_NT2	.1223	.1223	.1223	.1385	.1223	1	1	-	.1626	.1385	.1385	.1385
NF20_NE20_NT2	.2861	.3943	.2861	.392	.3943	.2861	1	1	1	-	.7113	.5349
NF20_NE50_NT2	.5349	.5435	.5349	.5349	.5349	1	1	1	1	-	1	1
NF20_NE70_NT2	.5349	.5349	.392	1	.5349	.5349	1	1	1	1	-	-

Table 2. 2/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

Table 3. 3/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

Table 4. 4/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

Table .5. 5/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

	NF100_NE20_NT2	NF100_NE50_NT2	NF100_NE70_NT2	NF150_NE20_NT2	NF150_NE50_NT2	NF150_NE70_NT2	NF1_NE20_NT2	NF1_NE50_NT2	NF1_NE70_NT2	NF20_NE20_NT2	NF20_NE50_NT2	NF20_NE70_NT2
NF50_NE20_NT2	.5331	.8286	.3844	1	.8243	.5349	1	1	1	1	1	1
NF50_NE50_NT2	1	1	.3631	1	1	.3631	1	1	1	1	1	1
NF50_NE70_NT2	1	1	.7733	1	1	.5331	1	1	1	1	1	1
NF100_NE20_NT2	.3844	.392	.2859	.3878	.392	.2861	1	1	1	1	1	1
NF100_NE50_NT5	.2859	.392	.2114	.5921	.392	.2114	1	1	1	1	1	1
NF100_NE70_NT5	1	1	.3878	1	1	.3878	1	1	1	1	1	1
NF150_NE20_NT5	.2777	.3844	.2839	.3844	.3918	.2859	1	1	1	1	1	1
NF150_NE50_NT5	.8286	.7053	.2859	1	.7053	.2859	1	1	1	1	1	1
NF150_NE70_NT5	1	.8752	.2859	1	.8752	.2859	1	1	1	1	1	1
NF1_NE20_NT5	.1223	.1223	.1223	.1223	-	.2114	.1484	.1223	.1223	.1223	.1223	.1223
NF1_NE50_NT5	.1223	.1223	.1223	.1223	1	-	1	.1385	.1223	.1223	.1223	.1223

Table 6. 6/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

Table .7. $7/16$. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

	$NF_1_NE70_NT5$	$NE20_NE20_NT5$	$NE20_NE50_NT5$	$NE20_NE70_NT5$	$NE50_NE20_NT5$	$NE50_NE50_NT5$	$NE50_NE70_NT5$	$NE100_NE20_NT10$	$NE100_NE50_NT10$	$NE100_NE70_NT10$	$NE150_NE20_NT10$
$NF50_NE20_NT2$	1	1	1	1	1	1	1	1	1	1	1
$NF50_NE50_NT2$	1	1	1	1	1	1	1	1	1	1	1
$NF50_NE70_NT2$	1	1	1	1	1	1	1	1	1	1	1
$NF100_NE20_NT5$	1	1	1	1	1	1	1	1	1	1	1
$NF100_NE50_NT5$	1	1	1	1	1	1	1	1	1	1	1
$NF100_NE70_NT5$	1	1	1	1	1	1	1	1	1	1	1
$NF150_NE20_NT5$	1	1	1	1	1	1	1	1	1	1	1
$NF150_NE50_NT5$	1	1	1	1	1	1	1	1	1	1	1
$NF150_NE70_NT5$	1	1	1	1	1	1	1	1	1	1	1
$NF1_NE20_NT5$.1484	.1507	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223
$NF1_NE50_NT5$	1	.3758	.1385	.1793	.1223	.1385	.1223	.1223	.2777	.1223	.1223

Table .8. 8/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis $SoftS$ mean obtained with the i -th setting is less than the one of the j -th setting. Color indicates significant statistic differences P-value ≤ 0.15 .

	NF150_NE50_NT10	NF1_NE20_NT10	NF1_NE70_NT10	NF1_NE50_NT10	NF20_NE20_NT10	NF20_NE50_NT10	NF20_NE70_NT10	NF20_NE50_NT10	NF50_NE20_NT10	NF50_NE50_NT10	NF50_NE70_NT10	NF50_NE50_NT10	NF50_NE70_NT10	NF50_NE20_NT10	NF50_NE50_NT10	NF50_NE70_NT10
NF50_NE20_NT2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF50_NE50_NT2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF50_NE70_NT2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF100_NE20_NT5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF100_NE50_NT5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF100_NE70_NT5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF150_NE20_NT5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF150_NE50_NT5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF150_NE70_NT5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
NF1_NE20_NT5	.1223	.1223	-	.2114	.1484	.5349	.1633	.1484	.1223	.1386	.1385					
NF1_NE50_NT5	.1385	.1223	1	-	1	1	1	1	1	.4765	.5349	.5035				

Table .9. 9/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

	NE100_NE20_NT2	NE100_NE50_NT2	NE100_NE70_NT2	NE150_NE20_NT2	NE150_NE50_NT2	NE150_NE70_NT2	NE1_NE20_NT2	NE1_NE50_NT2	NE1_NE70_NT2	NE20_NE20_NT2	NE20_NE50_NT2	NE20_NE70_NT2
NF1_NE70_NT5	.1223	.1223	.1223	.1385	.1223	1	1	-	.1626	.1385	.1385	
NF20_NE20_NT5	.1223	.1223	.1223	.1223	.1223	1	1	1	.1626	.1385		
NF20_NE50_NT5	.1385	.1626	.1385	.1626	.1633	.1385	1	1	1	.3903	.2114	
NF20_NE70_NT5	.1626	.1626	.1385	.6015	.1626	.1385	1	1	1	1	.7576	
NF50_NE20_NT5	.5349	.7733	.5349	1	.7979	.5349	1	1	1	1	1	
NF50_NE50_NT5	.1626	.1626	.2623	.1626	.1626	1	1	1	1	1	.5945	
NF50_NE70_NT5	.2973	.3182	.1626	1	.3182	.1626	1	1	1	1	1	
NF100_NE20_NT10	.2114	.2859	.2114	.2859	.2859	.2114	1	1	1	1	.9536	
NF100_NE50_NT10	.1626	.1626	.1626	.1626	.1626	1	1	1	.4343	.2114	.2114	
NF100_NE70_NT10	.1626	.2114	.2114	.1626	.1626	1	1	1	1	1	.3943	
NF150_NE20_NT10	.5435	.7733	.7733	.7979	.5349	1	1	1	1	1	1	

Table 10. 10/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

Table 11. 11/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

Table 12. 12/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

Table .13. 13/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

	NF100_NE20_NT2	NF100_NE50_NT2	NF100_NE70_NT2	NF150_NE20_NT2	NF150_NE50_NT2	NF150_NE70_NT2	NE1_NE20_NT2	NE1_NE50_NT2	NE1_NE70_NT2	NE20_NE20_NT2	NE20_NE50_NT2	NE20_NE70_NT2
NF150_NE50_NT10	.1626	.2114	.1626	.1626	.2114	.1626	1	1	1	.4343	.2859	
NF150_NE70_NT10	.1626	.2114	.1626	.2114	.2114	.1626	1	1	1	1	.3943	
NF1_NE20_NT10	.1223	.1223	.1223	.1223	.1223	.1223	-.2114	.1484	.1223	.1223	.1223	
NF1_NE50_NT10	.1223	.1223	.1223	.1223	.1223	.1223	1	-.1	.1385	.1223	.1223	
NF1_NE70_NT10	.1223	.1223	.1385	.1223	.1223	.1223	1	1	-.1626	.1385	.1385	
NF20_NE20_NT10	.1223	.1223	.1515	.1223	.1223	.1223	1	1	.5349	.1484	.1385	
NF20_NE50_NT10	.1223	.1223	.1223	.1223	.1223	.1223	1	1	.1385	.1223	.1223	
NF20_NE70_NT10	.1223	.1223	.1385	.1223	.1223	.1223	1	1	.1626	.1385	.1385	
NF50_NE20_NT10	.1385	.1633	.1385	.1223	.1626	.1223	1	1	1	.5439	.368	
NF50_NE50_NT10	.1385	.1385	.1385	.1385	.1385	.1223	1	1	1	.3044	.1626	.1626
NF50_NE70_NT10	.1223	.1385	.1385	.1385	.1223	.1223	1	1	.7028	.2102	.1626	

Table .14. 14/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

	NF150_NE50_NT10	NF150_NE70_NT10	NF1_NE20_NT10	NF1_NE50_NT10	NF1_NE70_NT10	NF20_NE20_NT10	NF20_NE50_NT10	NF20_NE70_NT10	NF50_NE20_NT10	NF50_NE50_NT10	NF50_NE70_NT10
NF150_NE50_NT10	.2848	.2249	.1626	.3943	.5435	.7733	.4343	.392	.7733	.1	.1
NF150_NE70_NT10	.3943	.3943	.2107	.9536	1	1	1	1	1	1	.1
NF1_NE20_NT10	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.2114
NF1_NE50_NT10	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223	-
NF1_NE70_NT10	.1223	.1626	.1385	.1626	.3279	.5349	.2102	.2656	.5349	.1	.1
NF20_NE20_NT10	.1223	.2039	.152	.1633	.3758	.357	.2599	.2114	.4436	.1	.1
NF20_NE50_NT10	.1223	.1223	.1223	.1223	.1626	.2102	.1223	.1385	.2102	.1	.1
NF20_NE70_NT10	.1223	.1626	.1385	.1626	.3279	.5349	.2102	.2656	.5349	.1	.1
NF50_NE20_NT10	.2114	.3903	.1385	.2973	1	1	.4343	1	1	.1	.1
NF50_NE50_NT10	.1385	.1626	.1385	.1484	.2114	.2114	.1451	.2114	.2114	.1	.1
NF50_NE70_NT10	.1385	.2102	.1385	.2114	.3943	.6186	.2649	.3238	.6186	.1	.1

Table .15. 15/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

	NF1_NE70_NT5	NF20_NE20_NT5	NF20_NE50_NT5	NF20_NE70_NT5	NF50_NE20_NT5	NF50_NE50_NT5	NF50_NE70_NT5	NF100_NE20_NT10	NF100_NE50_NT10	NF100_NE70_NT10	NF150_NE20_NT10
NF150_NE50_NT10	1	1	1	1	.3521	1	.5811	.9536	1	1	.4343
NF150_NE70_NT10	1	1	1	1	.6608	1	1	1	1	1	1
NF1_NE20_NT10	.1484	.1507	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223	.1223
NF1_NE50_NT10	1	.3758	.1385	.1793	.1223	.1385	.1223	.1223	.2777	.1223	.1223
NF1_NE70_NT10	-	1	.5349	.2777	.1385	.7733	.2861	.2114	1	.392	.2861
NF20_NE20_NT10	1	1	1	1	.1626	.5349	.284	.2037	1	1	.2861
NF20_NE50_NT10	1	.8632	.2114	.3464	.1223	.2859	.1385	.1223	.8243	.1626	.1484
NF20_NE70_NT10	1	1	.5349	.2777	.1385	.7733	.2861	.2114	1	.392	.2861
NF50_NE20_NT10	1	1	1	.2346	1	1	.7071	1	1	1	.8241
NF50_NE50_NT10	1	1	.3943	1	.1385	.2859	.2114	.1626	1	.4343	.1626
NF50_NE70_NT10	1	1	.8755	.5331	.1626	1	.3707	.2861	1	.5349	.3744

Table .16. 16/16. Random Planet hyperparameter tuning. Wilcoxon signed-rank test comparing Soft Stability. The element in the i -th row and j -th column indicates the P-value of the statistical test with alternative hypothesis *SoftS mean obtained with the i -th setting is less than the one of the j -th setting*. Color indicates significant statistic differences P-value ≤ 0.15 .

.2 Tables methods comparison

Table .17. ExtL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND). Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.4913
RPlanet	.8395	-	1	1	.0266
MInfo	.8676	1	-	.8676	.0266
DTree	1	1	1	-	.0266
GAAM	1	1	1	1	-

Table .18. ExtL-NoFilter-SN_MT-LEFT-ALL_(BAND). Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.1052
RPlanet	1	-	1	.3334	.1176
MInfo	.8117	1	-	.0549	.0549
DTree	1	1	1	-	.3477
GAAM	1	1	1	1	-

Table .19. ExtL-NoFilter-SN_MT-LEFT-BETA. Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.5315
RPlanet	.0066	-	1	.3553	.0220
MInfo	.0066	1	-	.3553	.0220
DTree	.3795	1	1	-	.0220
GAAM	1	1	1	1	-

Table .20. ExtL-NoFilter-SN_MT-LEFT-GAMMA. Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.1283
RPlanet	1	-	1	1	.2080
MInfo	.8336	1	-	1	.1283
DTree	1	1	1	-	.1990
GAAM	1	1	1	1	-

Table .21. ExtR-Laplacian-ALL_(EEG)-ALL_(EMG)-ALL_(BAND). Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.4098
RPlanet	1	-	1	1	1
MInfo	1	1	-	1	.4098
DTree	1	.6722	1	-	.4098
GAAM	1	1	1	1	-

Table .22. ExtR-Laplacian-SN_MT-RIGHT-ALPHA. Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	.8927	.0515
RPlanet	1	-	1	1	.0515
MInfo	1	1	-	1	.1208
DTree	1	1	1	-	.0530
GAAM	1	1	1	1	-

Table .23. ExtR-Laplacian-SN_MT-RIGHT-GAMMA. Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.3173
RPlanet	1	-	1	1	.1673
MInfo	1	1	-	1	.2774
DTree	1	1	1	-	.2415
GAAM	1	1	1	1	-

Table .24. GraspL-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND). Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.2492
RPlanet	1	-	1	1	.0724
MInfo	1	1	-	1	.0220
DTree	1	1	1	-	.0517
GAAM	1	1	1	1	-

Table .25. GraspL-NoFilter-SN_MT-LEFT-ALL_(BAND). Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.9976
RPlanet	1	-	1	1	.8930
MInfo	1	1	-	1	.2071
DTree	1	.9976	1	-	.2944
GAAM	1	1	1	1	-

Table .26. GraspL-NoFilter-SN_MT-LEFT-BETA. Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.1994
RPlanet	1	-	1	.1994	.1537
MInfo	1	1	-	.1994	.1537
DTree	1	1	1	-	.7168
GAAM	1	1	1	1	-

Table .27. GraspL-NoFilter-SN_MT-LEFT-GAMMA. Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.2955
RPlanet	1	-	1	1	.1932
MInfo	1	1	-	1	1
DTree	1	1	.2955	-	.1932
GAAM	1	1	1	1	-

Table .28. GraspL-Laplacian-ALL_(EEG)-ALL_(EMG)-ALL_(BAND). Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.4172
RPlanet	1	-	1	1	.1578
MInfo	.4172	1	-	1	.1578
DTree	.7972	1	1	-	.1991
GAAM	1	1	1	1	-

Table .29. GraspL-Laplacian-SN_MT-LEFT-ALL_(BAND). Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.1179
RPlanet	1	-	1	1	.1179
MInfo	1	1	-	1	.1179
DTree	1	1	1	-	.1179
GAAM	1	1	1	1	-

Table .30. GraspR-NoFilter-ALL_(EEG)-ALL_(EMG)-ALL_(BAND). Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.6270
RPlanet	.6270	-	.6270	1	.3543
MInfo	1	1	-	1	.8848
DTree	.6270	.5759	.1086	-	.1086
GAAM	1	1	1	1	-

Table .31. GraspR-NoFilter-SN_MT-RIGHT-GAMMA. Wilcoxon signed-rank test comparing AUC performance. The alternative hypothesis is: *AUC mean obtained by the linear SVM model in the test set with the i-th feature selection algorithm, is less than the one obtained using the j-th feature selection algorithm.* Red color indicates significant statistic differences P-value ≤ 0.05 . Orange color instead, values near significant difference (P-value $\in [0.05, 0.10]$).

	StepwiseR	RPlanet	MInfo	DTree	GAAM
StepwiseR	-	1	1	1	.1449
RPlanet	1	-	1	1	.2955
MInfo	1	1	-	1	.1733
DTree	1	1	1	-	.2525
GAAM	1	1	1	1	-