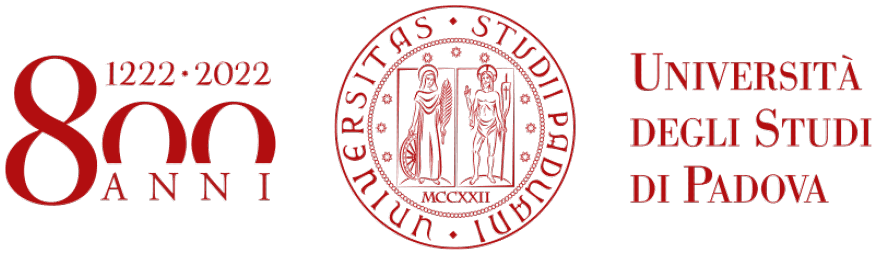
UNIVERSITÁ DEGLI STUDI DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE
IN INGEGNERIA INFORMATICA

Deployment of a real-time quality control algorithm and visualization of EEG recordings for embedded platforms.

Relatore
Prof. Luca Tonin
Prof. Lars Kai Hansen
Prof. Per Baekgaard

Laureando
Giovanni Grego

ANNO ACCADEMICO 2021/22

Abstract

Epilepsy is one of the most common neurological diseases with around 50 million cases globally, and nearly 80% of them appear in low- and middle-income countries (LMICs). It is estimated that around 75% of epilepsy cases in LMICs are not treated correctly when for up to 70% of those, it would be possible to live seizure-free thanks to the medicines. One of the most consistent predictors of seizure recurrence, as well as the most common test used to diagnose epilepsy condition, is electroencephalography (EEG), a noninvasive method for recording brain electrical activity [1].

In this scenario, some companies are developing solutions to bring affordable devices able to respond to LMICs setups, e.g. expert-independent and in a mobile environment. Therefore, there is a necessity for an intuitive, automatic and real-time EEG recording quality control feature running on a mobile phone.

The problem inspected in the thesis is the feasibility of such a feature. In the **Introduction** and **State of the art** chapters, the problem is stated and the standard methods used are exposed through the PRISMA standard. The **Design Thinking Process** chapter highlights the framework used to structure the project. The **users' needs and system requirements** chapter addresses the necessities of the users and defines their needs. The data is recorded and organized in the **Data collection** chapter. The artefact detection algorithm is designed and tuned for the specific data in the **Riemannian Potato Field (RPF) and practical design** chapter. Eventually, the implementation and the development of the quality feature in the mobile app are explained in **RPF implementation for embedded systems** and **Mobile app quality feature prototype** chapters. The specific methods, results, conclusions, discussions and future works are examined for each chapter. Finally, the outcomes of each chapter and the influenced discussions are linked together in the final **Discussions and conclusions** chapter.

Abstract

L'epilessia è una delle malattie neurologiche più comuni, con circa 50 milioni di casi a livello globale, di cui quasi l'80% di essi si manifesta nei Paesi a basso e medio reddito. Si stima che circa il 75% dei casi di epilessia nei Paesi a basso reddito non sia trattato correttamente, mentre per il 70% di questi sarebbe possibile vivere senza episodi epilettici grazie ai farmaci. Uno dei predittori più stabili di ricorrenza delle crisi, nonché il test più comunemente utilizzato per diagnosticare la condizione di epilessia, è l'elettroencefalogramma (EEG), un metodo non invasivo per registrare l'attività elettrica del cervello [1].

In questo scenario, alcune aziende stanno sviluppando soluzioni per fornire dispositivi a prezzi accessibili in grado di rispondere alle particolari necessità dei Paesi meno sviluppati, ad esempio soluzioni che non richiedano esperti del settore, e in un setup mobile. Pertanto, è necessario un metodo di controllo della qualità della registrazione EEG che sia intuitivo, automatico e in tempo reale, e che funzioni su una applicazione installata su uno smartphone.

Il problema esaminato nella tesi è la fattibilità di tale metodo. Nei capitoli **Introduction** e **State of the art**, viene esposto il problema e i metodi normalmente utilizzati, attraverso lo standard PRISMA. Il capitolo **Design Thinking Process** evidenzia il framework di riferimento utilizzato per strutturare il progetto. Il capitolo **users' needs and system requirements** affronta le necessità degli utenti e definisce i loro bisogni. I dati vengono registrati e organizzati nel capitolo **Data collection**. L'algoritmo di classificazione degli artefatti viene progettato e ne vengono calibrati i parametri, per i dati raccolti, nel capitolo **Riemannian Potato Field (RPF) and practical design**. Infine, l'implementazione e lo sviluppo della funzione di controllo della qualità nell'applicazione per smartphone sono spiegati nei capitoli RPF implementation for embedded systems e Mobile app quality feature prototype. Per ogni capitolo vengono esaminati i metodi specifici, i risultati, le conclusioni, le discussioni e i lavori futuri. Infine, i risultati di ciascun capitolo e le discussioni che ne sono derivate sono collegate tra loro nei capitoli finali **Discussions and conclusions**.

Contents

1 Introduction

1.1 Epilepsy

Epilepsy is one of the most common neurological diseases globally [1].

Epilepsy was defined conceptually in 2005 as a disorder of the brain characterized by an enduring predisposition to generate epileptic seizures. This definition is usually practically applied as having two unprovoked seizures greater than 24 hours apart [2].

A seizure is a brief episode of involuntary movement that involves the entire body or parts of it. It is often related to loss of consciousness or bladder function.

Seizure episodes are a result of excessive electrical discharges in a group of brain cells. They can appear in several different ways, hitting parts of the brain or being generalized. They can vary from causing prolonged convulsions to just brief lapses of attention. Finally, they can vary in frequency, from less than one per year to several per day. [3]

Around 50 million people have epilepsy in the world, with an estimation of 5 million new people diagnosed with epilepsy worldwide every year. Nearly 80% of epilepsy cases appear in low- and middle-income countries (then LMICs) [1].

People with epilepsy have three times higher risk of premature death than the general population. It is estimated that around 75% of epilepsy cases in LMICs are not treated correctly. This is also known as the ”treatment gap”. On the other side, seizures can be controlled with the appropriate use of antiseizure medicines. For up to 70% of people living with epilepsy, it would be possible to live seizure-free thanks to the medicines [1].

Electroencephalography (EEG) is one of the most consistent predictors of seizure recurrence [1], as well as the most common test used to diagnose epilepsy condition.

1.2 Electroencephalography

Electroencephalography (EEG) is a non-invasive method for recording brain electrical activity by measuring the voltage between pairs of electrodes located uniformly on the scalp. Nowadays, EEG is one of the most important techniques for recording cortical brain activities, given its balance between accuracy and practicality, cost and easiness of use.

The electrical activity of the brain is mainly caused by the post-synaptic potentials (PSPs), in particular, for a change in membrane conductance and transmembrane potential provoked by the neurotransmitter released by the travel of the action potential to the nerve terminal [4].

The signal can lead to an excitatory post-synaptic potential (EPSP) on the dendrites or to inhibitory post-synaptic potentials (IPSPs) on the cell body of the neuron. The combination of EPSPs and IPSPs induces currents that flow within and around the neuron. The EEG is essentially measuring these voltage changes in the extracellular matrix. The measurement is taken on the scalp by a system of electrodes and amplifiers [4].

The interaction between the cortex and thalamus as well as the functional properties of large neuronal networks in the cortex that have an intrinsic capacity for rhythmicity create recognizable EEG patterns, varying in different areas of the neocortex that allow us to make sense of the complex world of brain waves [5].

One of the drawbacks of this method is the sensitivity to external sources of information other than brain activity, that contaminate the signal. These external sources are called artifacts, and they can interfere with the brain signal in the time-frequency-space domain, making the tasks of detection and the removal of the noise cumbersome [6].

The sources of artifacts can be classified as physiological activities of the user (e.g. movements, muscle tensions, ECG..) or external sources such as environmental interferences, electrode pop-ups or cable movement. They can be periodic events (e.g. ECG) or they can appear irregularly. In addition, they can influence a group of channels (global) or only a single channel (local) [6].

1.3 Example of epilepsy diagnosis in LMICs

Numerous companies are active in the EEG device market, with most of them targeting high-income countries (HICs), and with different purposes than epilepsy diagnosis [7]. For these reasons, EEG machines are expensive to buy ($15k min), difficult to operate, and rare [8].

BrainCapture is ”a Medical Technology Company bringing affordable EEG scans and interpretations to everyone” [9]. Enabling cheap diagnosis, a larger portion of epilepsy cases could get the needed treatment, solving the cost and the distribution problems.

The biggest challenge to be faced remains the lack of experienced professionals able to acquire and validate EEG recordings, risking that all recorded data must be discarded and re-measured. On this side, BrainCapture is aiming at developing methods and strategies to provide non-professional EEG operators with the necessary support to perform a successful scan.

The principal solution developed by the company consists of a mobile application to support and guide the EEG operator through all the steps necessary to perform an epilepsy EEG test [9].

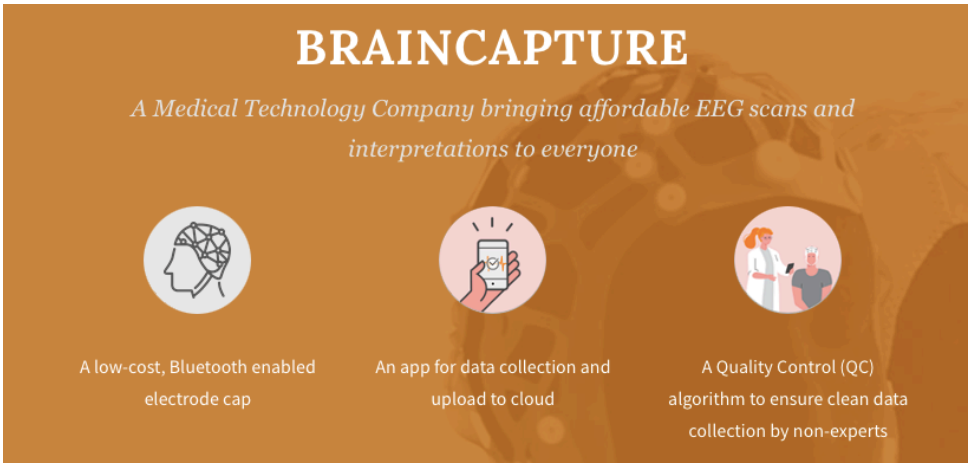


Figure 1.1: BrainCapture’s landing page. Picture from their website [9]. Three deliverables are shown, and the last is the quality control feature, demonstrating the centrality of this project in their vision.

1.4 Problem statement

When a recording is performed by non-expert technicians it is easily affected by EEG artifacts, which are external sources of noise that contaminate the brain signal. These recordings result in poor-quality tests, which are unreadable to neurologists and therefore not usable for diagnosis. This aspect is of such significance that it is considered a separate deliverable on BrainCapture's landing page, as reported in Figure 1.1.

In this work, an operative solution to this problem is researched. The idea is to inspect if it is possible to make non-expert EEG operators acquire recordings of good quality, that can be used for an epilepsy diagnosis. In particular, the main goal is to analyse the feasibility of an online artifact detection algorithm, running on an embedded device, to support them in the data acquisition.

The problem is composed of three main parts:

- The end users of these recordings are neurologists, that need to read the scans to provide a diagnosis. For this reason, it is important to understand what a good-quality recording is for them. Thus, understanding what their necessities are and which aspects are problematic. In addition, the direct users of the system are the non-expert technicians. It is fundamental to understand the difficulties and doubts they may face during the acquisition process, and which are the best strategies to support them.
- An online detection algorithm for EEG artifacts has to be researched, collecting training data, and analysing its accuracy and reliability.
- Finally, the main goal of this project, and overall research question, is to verify that the implementation of the online detection algorithm is feasible to be run on an embedded system, integrable on an existing application.

2 State of the art

2.1 EEG market for LMICs

The review for the competitors' side has been elaborated using different sources. The most interesting insights, as one might expect, came from conversations and private documents with colleagues within the company and in the company network [10][11][12]. From that point, the various devices mentioned were analysed in more detail through their websites and private conversations. A summary of this work is given below.

The four key assets necessary to unlock the LMICs market are affordability, LMIC usability, expert independence and telemedicine accessibility to neurologists [7]. Numerous companies are active in the EEG device market [7]. They can be categorized as stationary (highly reliable) traditional medical-grade EEG and innovative wireless solutions companies.

The biggest players in the first category are, amongst others, companies like Natus [13], Cadwell [14], and Compumedics [15]. These and all the traditional companies are USA-based and focus on highly profitable MedTech devices for high-income countries (HIC). The data recorded with these devices is of high fidelity and high quality. On the other side, the medical grade is expensive. In addition, the traditional EEG requires specialists on site and the necessity of having a stationary setup with stable grid power.

For the second category, it is worth citing companies like Zeto [16] and Brain products [17]. The products of these players try to overcome some of the traditional problems. For example, they are wireless and do not rely on a stable source of power. The weakest edge is the quality, as the products are intended for consumer or research purposes, and the resolution is insufficient for key medical applications.

BrainCapture's aim is to keep the advantages of both sides, overcoming the disadvantages. The strongest competitive edge is the hardware price, which is about 20 times cheaper than any other available product in the market. In addition, the solution provided by BrainCapture merges together all the other three assets listed above: it is based on a mobile infrastructure that is more widely available than grid power; it is independent of expert technicians, as the mobile application aims to guide the nurse through all the necessary steps; and the cloud storage of the recordings allow neurologists to access the data in a telemedicine manner.

Below is the competitive business radar chart, highlighting the key assets and BrainCapture positioning with respect to the other competition classes (i.e. traditional and innovative companies).

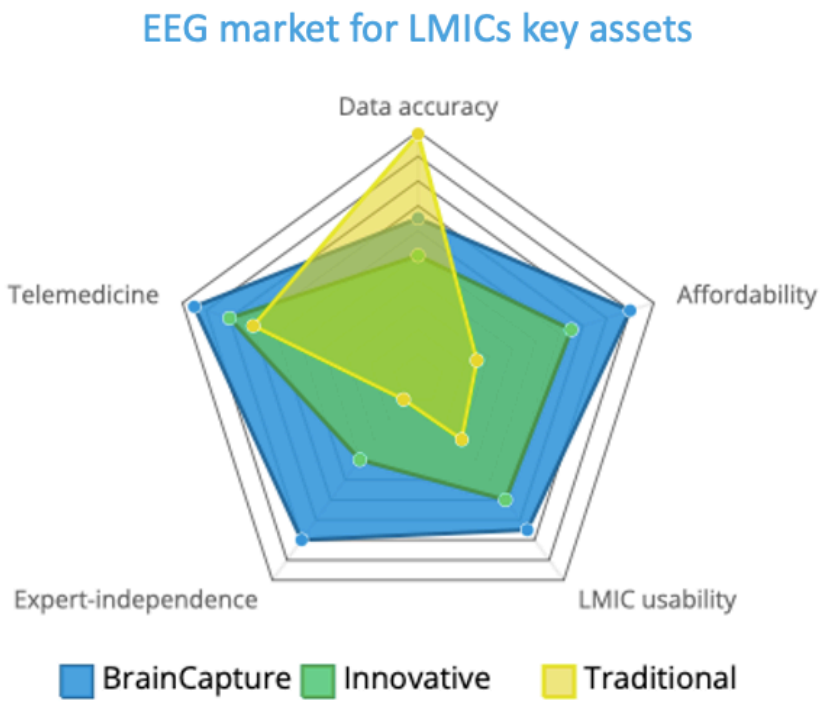


Figure 2.1: Qualitative radar graph of EEG market for LMICs. Traditional solutions overtake BrainCapture in data accuracy but perform poorly in all the other key assets for LMICs. Other innovative solutions can compete on the remaining assets, but are not as focused as BrainCapture on epilepsy diagnosis, therefore less specialized. The framework used is quantitative, but the analysis is only qualitative.

As shown in the figure above, traditional companies overtake BrainCapture in the accuracy of the data recordings but are not competitive in the ease of use and LMIC conditions. The innovative solutions, on the other hand, are more competitive on the four LMIC key assets, but not as focused on them as BrainCapture, and therefore less specialized for the peculiar challenges.

2.2 Artifact detection and removal: Literature review

To analyse the different methods for artifact detection, the PRISMA methodology has been followed, and in particular the ”study selection” point: ”Describe the results of the search and selection process, from the number of records identified in the search to the number of studies included in the review, ideally using a flow diagram” [18][19].

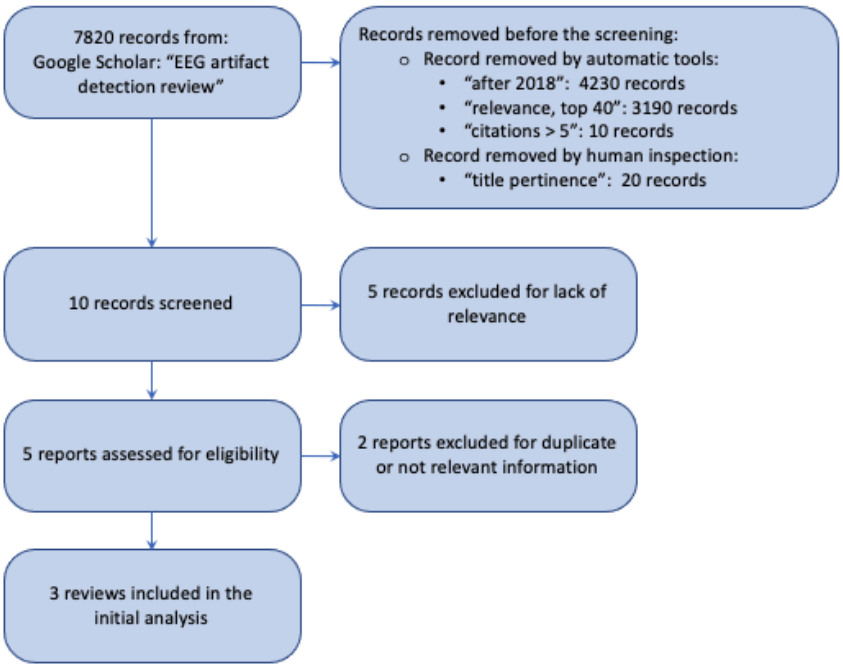


Figure 2.2: Flow diagram for the systematic review of reviews, as described in the PRISMA method[19]. The results of the first research have been filtered out through automatic tools and human inspection, to reach three final reviews used as references for further analysis.

In Figure 2.2 the retrieval and filtering process for collecting the reviews on artifact detection and removal is shown. From general research, the results have been filtered out to finally obtain 3 review papers [20][21][22]. Using the methods reported there as a starting point, a more detailed and targeted analysis has been performed for each particular method.

While the problem of artifacts is well known in the literature[6][20][21][22][23][24], a single universal and complete method for artifact detection and removal does still not exist.

Different methods have been elaborated on over the last years. The most popular techniques are based on blind source separation (e.g ICA and CCA) [25], time-frequency representation (e.g. wavelet transform) [26], adaptive filtering and machine learning applications [23] [24] [27] [28].

None of them alone is the solution to all problems of EEG practical applications. In fact, hybrid methods (e.g. EMD-BSS, Wavelet-BSS, BSS-SVM) are recently overtaking the classic standalone methods [21].

In practical solutions, there are several factors that can affect the choice of the technique to implement, and each case has to be treated separately. In general, nowadays, more applications involve BCI and neurofeedback, requiring computational light and online methods [29].

In the following, a comparative analysis of the most common methods is shown. The category used reflects the principle of adaptability to general situations.

	online	embedded	automatic
Regression	No	No	Yes
Wavelet	No	No	Yes
ICA	Yes	No	No
CCA	Yes	Yes	No
Adaptive filter	Yes	Yes	Yes
BSS	No	No	No

Table 2.1: Comparative analysis of different artifacts removal/detection methods

For most applications, artifact detection and removal algorithms have to run in real-time and without the intervention of human actions to classify clean or artifact signals. Nowadays, mobile applications are always more popular, especially in portable and home healthcare environments, so attention to embedded-friendly algorithms is also important.

Finally, numerous toolboxes have been developed over the years to work with EEG signals and to perform artifacts detection. Among the most important, EEGlab is a MATLAB toolkit that provides a GUI for processing EEG signals using independent component analysis (ICA) and time/frequency analysis, with artifact rejection built-in functionalities [30].

Two extensions of EEGlab are of particular interest in this context: BCIlab and ERICA [31]. BCIlab [32] is a MATLAB-based toolbox built to provide methods for BCI development and testing. This extends EEGlab with real-time functions. BCIlab is the latest of a sequence of toolkits published for BCI purposes. This series includes BioSig, developed at Graz University and focused on the offline analysis of a broad range of biosignals [33], and OpenViBE, a framework that allows the design of functional BCI using a GUI [34]. ERICA is an experimental environment for real-time interactive analysis and control of multimodal experiments.

3 Design Thinking Process - methods overview

The entire thesis conception and actuation followed a framework known as the ”Design Thinking Process”, proposed by the Hasso Plattner Institute of Design at Stanford [35].

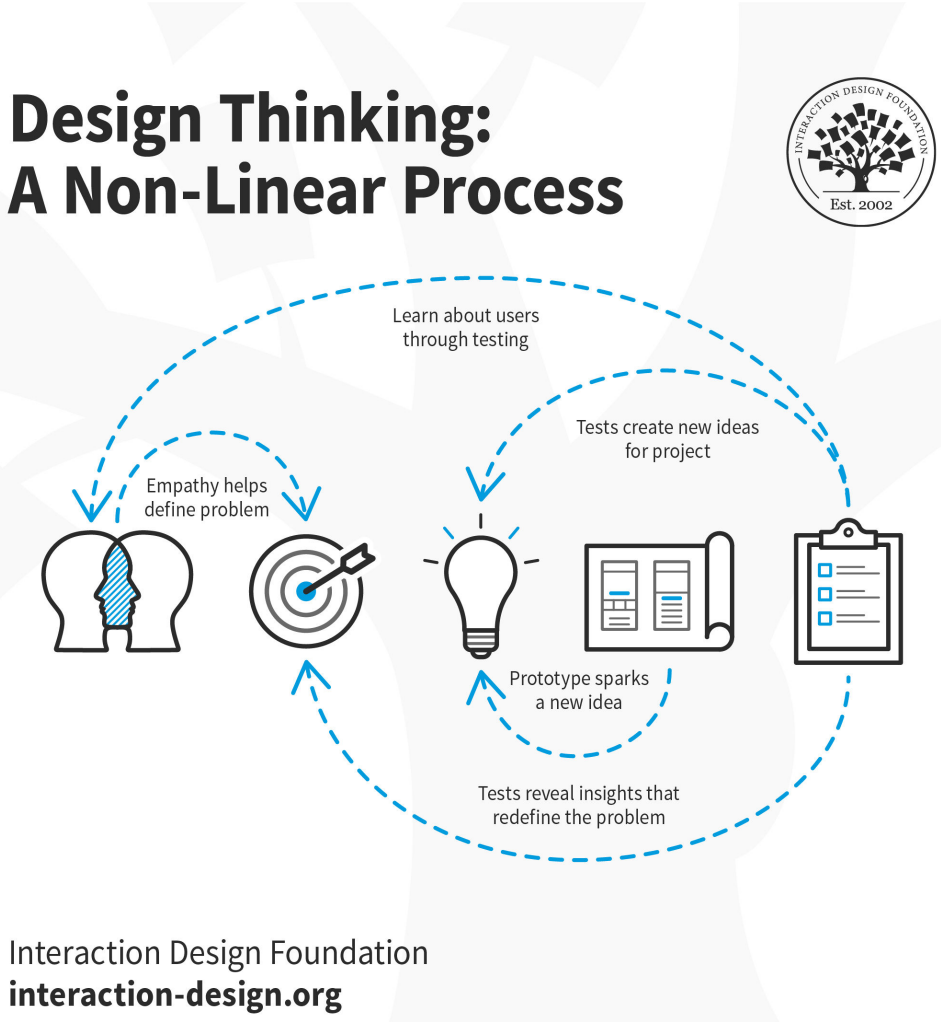


Figure 3.1: Visualization of the Design thinking framework. Picture from Interaction design foundation [36]. 5 stages form the framework: Empathize, Define, Ideate, Prototype and Test. The interactions are not linear, with each phase influencing all the others.

This framework scratches some guidelines to develop solutions for a problem. This framework is composed of 5 stages: Empathize (research your users' needs), Define (state your users' needs and problems), Ideate (challenge assumptions and create ideas), Prototype (start to create solutions), Test (try your solutions out) [36].

In the empathisation phase, all possible knowledge about the problem is collected by consulting experts and conducting meetings to engage and empathize with all the stakeholders. In this phase, it is common to gain experience and insights while also visiting the users' physical environment.

In the definition phase, the problem has to be defined and scoped, always caring about the human-centred side, in a ”user-centric problem statement”.

In the third phase, the knowledge acquired in the first two stages is used to ideate solutions to the problem statement. Different ideation techniques have been developed in the past years, like ”Brainstorm”, ”Brainwrite”, ”Worst Possible Idea” and ”SCAMPER” [36].

In the prototyping phase, inexpensive and scaled-down versions of the product, or parts of it, are developed to analyse the solutions of the Ideate stage. This stage is of fundamental importance to start facing the real problems and understand better the different limitations.

In the last stage, scientific methods and metrics are developed to measure the performances of the best solutions prototyped before, as well as the user interaction.

Finally, it is important to remember that these stages are only logically split and in a sequence, whereas, in a real case, borders are more shadowed. Moreover, even if the stages have been explained in a linear order, they are strongly interconnected. In fact, as shown in Figure 3.1, each stage contributes to refining, ideating, testing and improving all the other sections, in an iterative manner.

3.1 Thesis organization

Following this framework, the thesis has been split into different chapters, each including the methods used, the results obtained and the relative comments, and the final discussions and future work. In particular, the structure is as followed:

Users' needs and system requirements

This chapter is of extreme importance. In fact, it includes parts of the Empathy stage, the Define stage and also of the Ideate stage.

For the Empathy side, the points of view of doctors and expert EEG technicians have been started to be analysed through semi-structured interviews already available in the company. Then, to refine and scope the knowledge, and to better understand the entire system, further semi-structured questionnaires have been performed to expert technicians and non-expert technicians. A brief epilepsy lesson and a trial recording have been proposed to the non-experts, before letting them answer the questionnaire. In addition, different meetings have been conducted with company colleagues to have a clear idea of the entire system.

In the different iterations of the interviews, also several ideas have been proposed thinking about possible solutions. This is part of the Ideate stage.

All the knowledge acquired in this chapter in all the different iterations has been summarized in a lean canvas (LC) and in a user story map (USM). This process corresponds to the Define phase.

Data collection
This chapter is also part of the Empathy and Define stages. In fact, in this phase, data have been collected at the Filadelfia neurological hospital. Several sessions have been set for the recordings, for a total of 14 recordings. These provided the possibility to acquire hands-on experience by facing technical problems, observing all the critical points of an EEG scan, and discussing them with doctors and EEG technicians. In addition, numerous scans have been analysed and annotated under the supervision of expert EEG technicians, of fundamental importance for a deep understanding of the different artifacts involved.

Riemannian Potato Field (RPF) and practical design
After understanding and scoping the problem, and ideating some solutions in the previous chapters, in this one the artifacts detection algorithm is prototyped and tested. The field is designed from a knowledge-based perspective and the algorithm is tuned with a model-based optimization. In particular, hyperparameters are tuned with a bayesian optimizer and use the F2 score, partial dependence plots, and visualizations of the predictions to explain the optimization process. In addition, running time analysis has been performed to test the feasibility on embedded platforms.

RPF implementation for embedded systems
The Prototype and Test stages continue in this chapter. To assess the feasibility of the online quality feature on an embedded system, the RPF algorithm had to be converted from Python language to Java. For this reason, all classes and methods necessary to implement the RPF have been refactored to Java language. This requested to re-design the libraries to fit the new language, taking into account also the future inevitable cooperation with Kotlin language, for the final implementation in the already existing mobile application from BrainCapture.

All the code has been tested to assure the exact same behaviour with respect to the Python program.

Mobile app quality feature prototype
Also, this chapter involves the Prototype and Test phases. The quality control feature is implemented in Kotlin using the libraries developed before and UI tools proper for android development. Different tests have been idealized to analyse the final performances of the system.

4 Users' needs and system requirements

The purpose of this chapter is to analyze the needs of the users, interacting with the feature and to get the requirements that the MVP should fulfill. This chapter is of extreme importance in the entire process, as all the further work will be shaped and designed focusing on the most relevant requirements for this project.

4.1 Methods

The methods used for this analysis consist of surveys and continuous and iterative conversations with all the different stakeholders and users of the system. The continuous relationship with all the stakeholders assured systematic communication and allowed for keeping the system updated and controlled at each stage. As a starting point, initial discussions have been carried out with the colleagues in BrainCapture [37],[38],[39] to understand the context and the prior needs.

Documents previously available

Secondly, all the documents and material collected by BrainCapture have been read and analyzed to have a starting point for further inspections. In particular, the documents available regard users and market descriptions, as well as the scope of the amplifier [appendix A] [40]. In addition, an interview with a neurological doctor [appendix B] and an interview with an EEG technician [appendix C] on the important aspects of EEG recordings were available. Finally, a private document on "Top Technical Tips" [41] from TeleEEG [42] was provided. All these documents together shaped an initial picture of the opportunities and challenges that this project has to face. The interviews were not intended for the particular purpose of this elaborate, so information had to be extrapolated from the documents, and subsequently integrated and enriched with further conversations with the colleagues, in an iterative setup.

Further analysis

From the knowledge acquired with the initial overview, deeper interviews have been proposed with expert technicians and non-expert users. In particular, three semi-structured interviews [appendixes D] have been carried out with expert EEG technicians and doctors from Filadelfia neurological hospital [43].

During the same period of this project, usability tests have been performed in the company. They consisted of teaching lessons to non-expert technicians and in particular nurses. For each nurse, a brief introduction to EEG, to artifacts, to BC's device, and to the quality of EEG recording is given. Then, an initial demonstration of the entire recording process is performed, followed by three practice runs for each nurse. During the entire session, notes of the behavior have been taken. In addition, semi-structured interviews have been taken with the nurses, at the end of the day. The resulting affinity map (the only result ready at the time of this thesis) has been used to enrich the scenario [44].

Finally, a semi-structured interview with the goal of inspecting doubts and techniques for improving the recordings' quality has been proposed to the nurses, in the role of non-expert EEG technicians, on a voluntary base. Unfortunately, none of them replied to the survey. For this reason, as representatives of the non-expert operator, an experiment has been proposed for students without prior knowledge of EEG and EEG recordings. The scope of the experiment was to get an in-deep understanding of the doubts

that a non-expert user can think of when recording an EEG signal, and which requirements the system should implement to make the users aware of the quality of the data recorded. The experiment reproduced the tests described before: a brief introduction about epilepsy, EEG scan, artifacts, and acquisition quality (all based on knowledge from the expert technicians), followed by a hands-on trial, consisting of a recording executed by the subject. Subsequently, a semi-structured interview has been performed with each participant [appendix E]. The participants were all male between 23 and 25 years old, with no prior knowledge of EEG. One participant was a student of a UX course, therefore understanding the principles of UX and mobile development and visualization.

Lastly, with all the knowledge acquired, a lean canvas [45] and a user story map (USM) [46] have been designed, to summarize, organize and focus on the requirements for the feature for the mobile application.

4.2 Results and conclusions

Patient journey and interactions of users

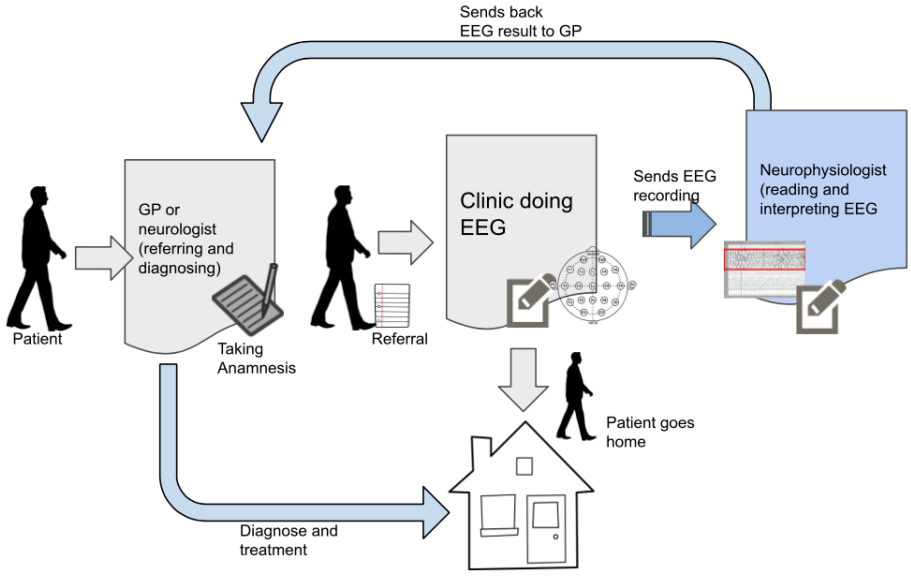


Figure 4.1: Patient journey and how the users are connected in the EEG diagnosing flow in Denmark. Picture from document [40]. The patient, typically after the manifestation of seizures, consults the General Practitioner (GP), who can refer the patient to the EEG scan procedure. Once the recordings are acquired, they are sent to the neurologist that interprets them and sends back the results to the GP.

As shown in the figure above, the patient who experiences seizures will go to his General Practitioner (GP)/neurologist. From the GP/neurologist, the patient can be referred to an EEG clinic to have an EEG Scan done.

There are different types of EEG recordings: Standard EEG, 1-hour sleep-deprived EEG, 4-hour EEG, and 1-5 day EEG recording where the patient is admitted to the hospital. Since the Standard EEG is the most commonly used type, the device is designed to support this process [40]. The standard procedure consists of different stages for calibration and seizure stimulation that is out of the interest for this project, and at least 20 minutes of resting state good quality data [47]. This time is intended to have probabilities of recording at least a seizure.

At the clinic, an EEG operator will perform the EEG recording on the patient and send the result to the neurophysiologist who then looks at the recording and interprets it. The interpretation and conclusion report is then sent to the GP/neurologist who can then diagnose and start treatment on the patient.

The scope of the amplifier, for what extends this project, is to support the EEG recording process done by the EEG operator. As the system is designed for LMIC, and the first market is expected to be Kenya, the device should apply to both facilities that already provide epilepsy services and smaller facilities that don't have any epilepsy services yet but want to take them on [40]. Therefore the users stretch from experts in epilepsy diagnosing, EEG recording, and EEG interpreting to non-experts that have never tried doing EEG-related services before. For this reason, the aim is to develop a system/device that can also be used by non-specialists. Therefore the EEG operator user is divided into 2 personas: EEG technician and nurse/healthcare assistant.

Relevant user personas

The patient is an indirect user (interacts indirectly with the system). The patient does not handle any parts of the EEG equipment directly but is exposed to the equipment upon having the EEG recording done on him/her [40]. This user is interested in a comfortable and quick EEG session and in getting the right diagnosis and treatment (indirectly meaning a quality EEG recording). This user group contains a severe variance and intra-differences [appendix A, pag. 12-13.]. For the purpose of this project, the important categories to focus on are cooperative and non-cooperative patients.

General Practitioner (GP)/Neurologist is an indirect user. The GP/Neurologist is the one diagnosing the patient, prescribing medicine, and following up on treatment with the patient. GP/Neurologist can refer the patient to an EEG Scanning to support the diagnosis [40]. This user is interested in diagnosing his patients correctly and therefore needs someone to do a quality EEG recording and get a quality EEG report back [40].

The EEG operator is a direct user. The EEG operator is the one actually handling the EEG equipment and performing the EEG recording on the patient. [40] This user is interested in doing a good, quality EEG Recording and handling the patient with care [appendix A, pag. 13-18]

EEG artifacts dealing - experts point of view

The complete questionnaire developed from the knowledge acquired by the already available documents is reported in [appendixes C]. In the following, all already available information as well as the following inspections have been summarized and organized in a discursive manner.

From an expert point of view, the most important aspects to keep under control, in order to record quality data, are the impedance of the electrodes and the EMG artifacts.

The impedance of the electrodes influences heavily the quality of the data and it is the most essential aspect to keep under control. This is because a high impedance load acquires noise from external sources, especially 50/60Hz from the power grid in the environment [48]. Once the impedance for each electrode is lower than a certain threshold (usually 10 KOhm [47]), the next steps can be performed.

Muscles and movement artifacts are the dominant artifacts in EEG recordings. These are particularly seen in babies and children patients, as well as non-cooperative patients. The solution in most cases is to try to keep the patient calm and to not move, and to remember to release the stress around the jaw and the shoulders. In non-cooperative or extreme cases, the only solution is to prolong the recording or to take the best out of the data acquired [appendixes B, C, D].

EEG artifacts dealing - non-experts point of view

As expected, a user without prior knowledge of the EEG procedure has different doubts with respect to an expert technician. In fact, the focus on the time series signal disappears, and the quality of the data is not assessed from the waves themselves. Therefore, new statistics and visualizations have to be developed in order to show the data quality.

The experiment conducted with the students resulted that a non-expert technician slavishly follows the instructions given. They focused a lot on the list of good practices to obtain a good quality scan, explained in the training and shown below:

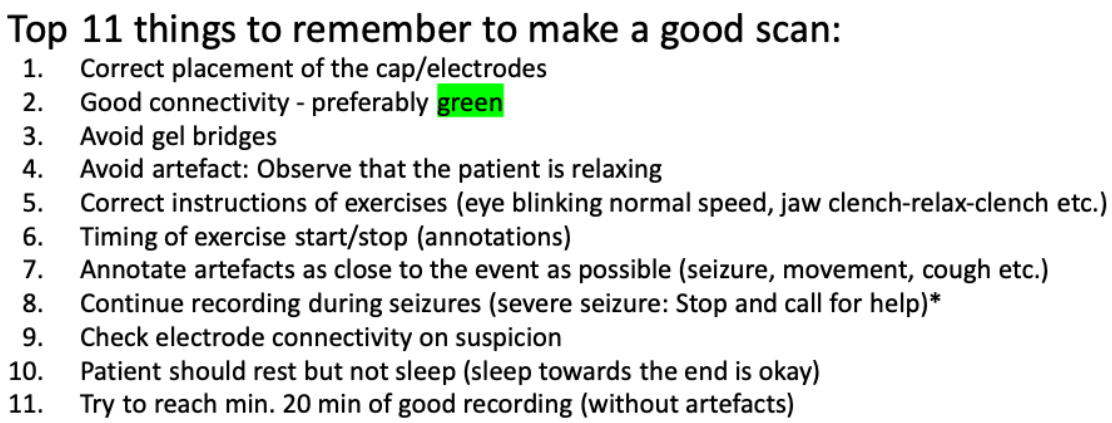
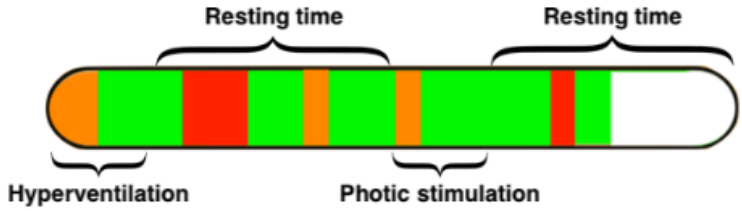


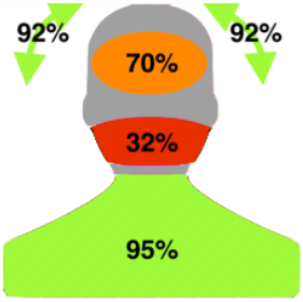
Figure 4.2: List of good practices for a proper quality scan

For the same reason, they assess the quality of the data from this list. The problem is that specific artifacts (e.g. jaw clenching, the stress in the shoulders, eye movements with eyes closed...) are not visible just by looking at the patient. For example, the second participant reported that the patient was not moving, when the quality of the scan was corrupted by stress on the jaw area. Moreover, even though all of them judged the scan taken of good quality, none of them had an explanation, but they were only referring to the mentioned list. This means that they didn't have control over this aspect. This is because they didn't have any measurement or feedback from the system, apart from the impedance feedback. Thus the idea of giving the user a tool for checking the quality of the data, without having prior EEG knowledge and without looking at the signals.

Therefore, it has been asked both expert and non-expert operators whether the two tools shown below could help them in assessing the quality of the data recorded.



(a) Progression bar for data quality over time. A color scale (green: good, orange: medium, red: poor) is used to visualize the quality of the segments of data.



(b) Bust for data quality over space. The main sources of artifacts are visualized as body regions. Eyes region for eye movements, the jaw region for stress in the jaw area (clenching, chewing,...), the shoulders for artifacts from the bottom part of the body, and top arrows for head movements. The same color scale is used to visualize the quality of the region.

Figure 4.3: Proposed tools for assessing data quality over time (a) and space (b). A color scale (green: good, orange: medium, red: poor) is used to visualize the quality of data.

The visualization on top, Figure 4.3(a), represents the quality of the data over time. A green segment indicates that the quality in that portion of the recording was good. An orange segment shows that the quality in that portion of time was still appropriate, while a red segment indicates that the quality is poor. The white segment at the end represents the time that still misses for the completion of the recording. On the progression bar, indications of the stages of the standard protocol are given. An additional clock will show the operator the time remaining for 20 minutes of a good recording. The image on the bottom, Figure 4.3(b), shows the punctual quality. At every moment, the quality of each part of the body is shown. A green area means that the patient is creating none or few artifacts from that part of the body. Hence, no further actions have to be taken. An orange area shows that, from that specific part of the body, some artifacts are being created. A red region means a lot of artifacts are due to movements or stress in that part. For example, in the figure above, the region of the jaw is affecting the scan, so the operator will focus on taking actions to release the stress from there, such as opening a bit the mouth.

It's interesting how the results between expert and non-expert operators agree. Both groups tend to give positive feedback to the progression bar, appreciating its usefulness and intuitiveness. In other words, a progression bar is an object that all users are familiar with, and in a glance gives the idea of how much time the scan has to be prolonged. On the other side, the bust hasn't been appreciated. The main reasons are that is not easy to read and that the direct user prefers to fulfil the list above in Figure 4.2, and feels confident of the quality checking those. It is the opinion of the author that this aspect has to be investigated more. There are cases in which the patient seems relaxed and not moving, whereas tensions can affect the quality of the data. For this reason, an indication of the source of the noise could improve the quality and the simplicity of finding the solution. In any case, it has been decided to include the progression bar in the MVP and to inspect the visualization of the artifacts in the following iteration.

Tools for system requirements outline

Finally, all the information acquired in this section has been collected and organized with two UX-specific tools: the Lean Canvas (LC) and the User Story Map (USM).

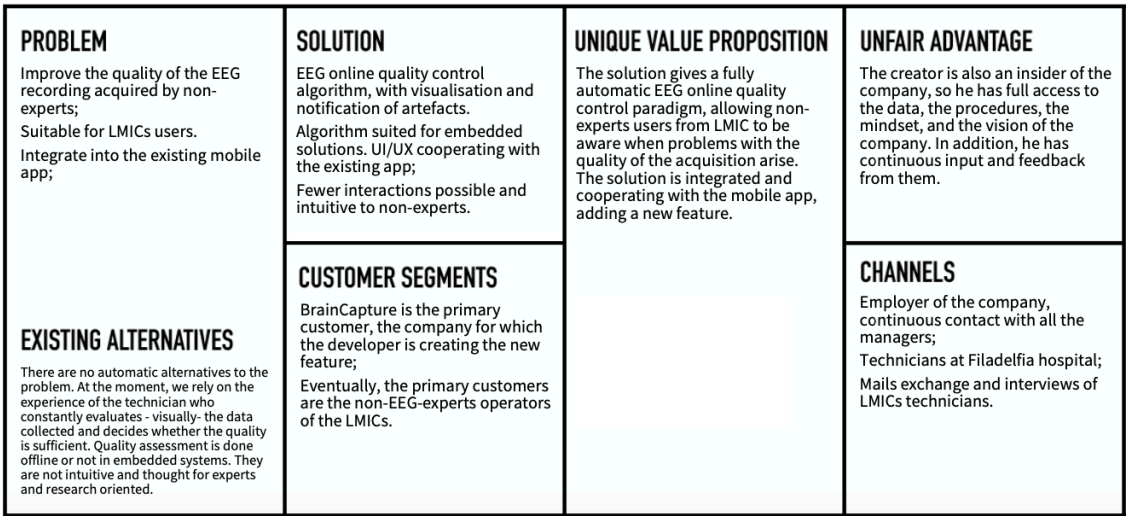


Figure 4.4: Lean Canvas from the point of view of a student implementing the Quality Control feature. It summarizes problems, solutions, and methods, scoping other important aspects.

Figure 4.4 reports the Lean Canvas from the point of view of an employer of the company, developing the quality control feature integrated into the existing mobile application. To summarize all knowledge above as much as possible, the problem is to improve the quality and support the recording of EEG data for epilepsy diagnosis by non-expert nurses in the LMICs. The solution is a feature in the existing mobile application that implements an online quality control algorithm for artifact detection and an intuitive visualization of the quality, in order to extend the recording to reach a 20-minute good-quality scan.

In Figure 4.5 below, the User Story Map for the quality feature is reported. In particular, the minimum valuable product is on the left side. Aiming at acquiring at least twenty minutes of useful EEG data, the MVP focuses on estimating how much of the data recorded at any time is clean, and suggests an estimate prolonging time for reaching the goal. In addition, a progress bar visualizes the segments of the recording, coloring them accordingly to their quality.

On following iterations, the operator will be notified when the data is corrupted, so that the user can take immediate actions, resulting in a smaller loosing of time. In addition, initial checks on the quality of the data will be performed with heuristics even before starting the recording. For example, an extremely high level of 50Hz noise can mean a not appropriate environment or connection, even having a proper impedance on each channel. Also, leakages, channel bridges, broken electrodes, and systematic artifacts can be checked in this phase. Finally, user customization will be implemented, for example for deactivating the feature for non-cooperative users, or for removing specific artifacts from the detection. For time constraints, these features are going to be implemented in the following stages of the application.

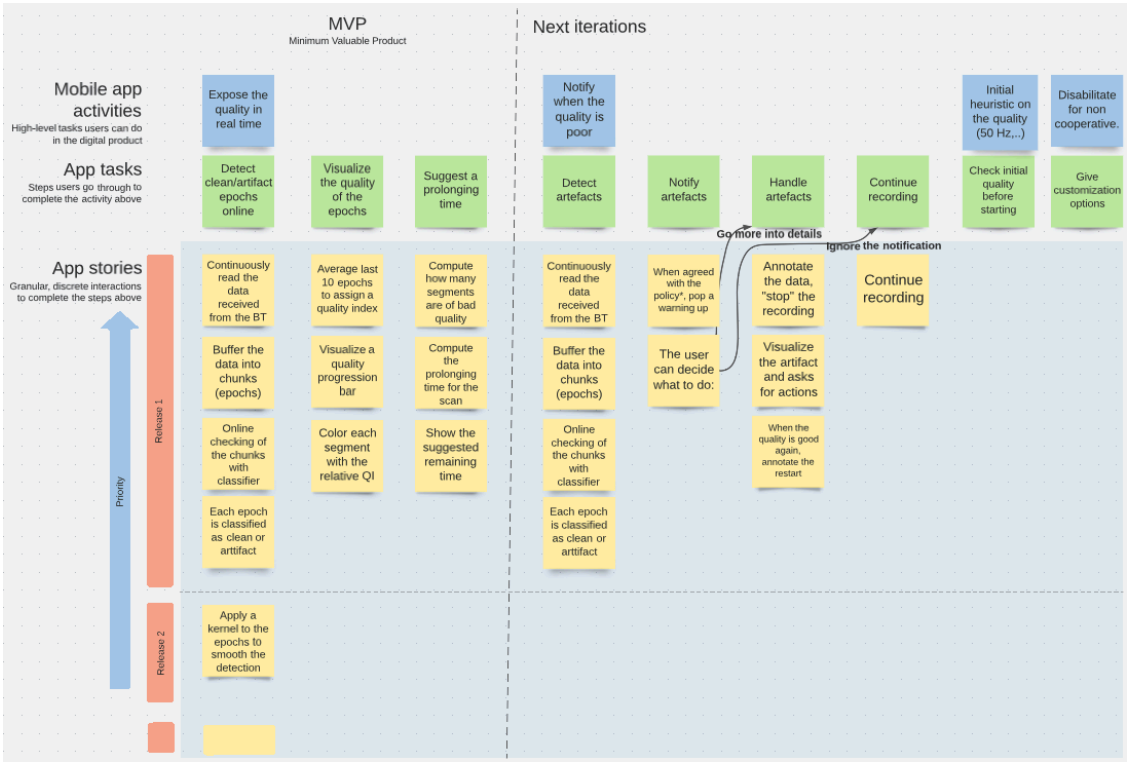


Figure 4.5: User story map (USM) for the quality control feature. The Minimum Valuable Product (MVP) is shown on the left side. For the MVP, the quality feature is implemented and the artifact epochs are visualized with a progression bar showing segments of time of data colored with respect to their quality. In addition, a suggested prolonging time will be shown.

In the most schematic way, the outcomes are that the problem to be addressed is to acquire EEG recordings for which the quality is sufficient to diagnose epilepsy. The standard sets 20 minutes of clean data the minimum for having enough information to diagnose epilepsy [47]. Therefore, the goal is to acquire at least 20 minutes of clean data. For this reason, the idea is to develop a feature for the already existing mobile application to support the acquisition and to show the operator the remaining time for reaching the minimum requirements.

One aspect that has not been yet examined in detail is the detection algorithm. From all the information in this chapter, it is known that the algorithm is going to run in an embedded system and that it needs to operate in an online matter, to immediately give feedback to the operator. In addition, it is requested to be automatic, to assess the needs of non-expert users. All these constraints are shown in the comparison Table 2.1 in the literature review chapter. As shown in the table, the adaptive filter methods are the most promising. Many different algorithms belong to this category. For ease of understanding, availability of examples and code, and interest in analyzing Riemann geometry applied to the detection of EEG artifacts, the Riemannian Potato Field algorithm has been chosen [49].

4.3 Discussion and future work

This chapter plays the most important role in guiding all the further decisions. As frequently highlighted, this procedure is at the beginning from a logical point of view, but it has to be intended as a background process always evolving and improving for each iteration.

The methods used for the users' needs surveys are standard and no other techniques have been considered useful for significantly improving the understanding of the problem in this first iteration of the project. The empathisation in this stage has been perceived as successful. All the exchanges of views, ideas, and information performed with colleagues, doctors, experts, and non-expert EEG technicians gave the right background to continue with the definition and ideation of the problems and their solutions.

On the other hand, the set of people contacted could have been enlarged and enriched. For example, although for external reasons, no nurses (i.e. the main target as non-expert operators) could be interviewed, and instead general figures have played their role.

In addition, further aspects could have been inspected with subjects from the target countries, especially for the usability and artifact visualization parts. It is believed that further investigations could have brought more detailed and focused insights. Again, this has not been considered necessary for the minimum valuable product, but an aspect to consider in the following iterations is to improve and customize the final feature.

5 Data collection

This chapter focuses on the acquisition of the data needed for the detection algorithm. Several aspects have influenced the process and many subjects played a role in the design of the procedure. More precisely, the data collection had to deal with four projects: this thesis, a Ph.D. work on artifacts detection, a medical study on epilepsy diagnosis through wireless EEG devices, and BC's test data comparison between their wireless amplifier and a medical amplifier [50][51]. For this reason, the protocol has been adjusted for fulfilling all different needs, balancing the time and effort necessary for each recording. The data is covered by IRB given to the project by DTU. For accessing the data, contact Tobias Andersen, toban@dtu.dk.

5.1 Methods

The recordings have been performed in Filadelfia hospital (then FIL). For BC's data comparison (i.e. another project in the company), the signal has been recorded simultaneously by BC's amplifier and the medical amplifier through the use of splitters. BC's cap and ECG pads were connected as an input to two splitters, which outputs were then directed one to BC's device and one to FIL's device, as shown in Figure 5.1 below.

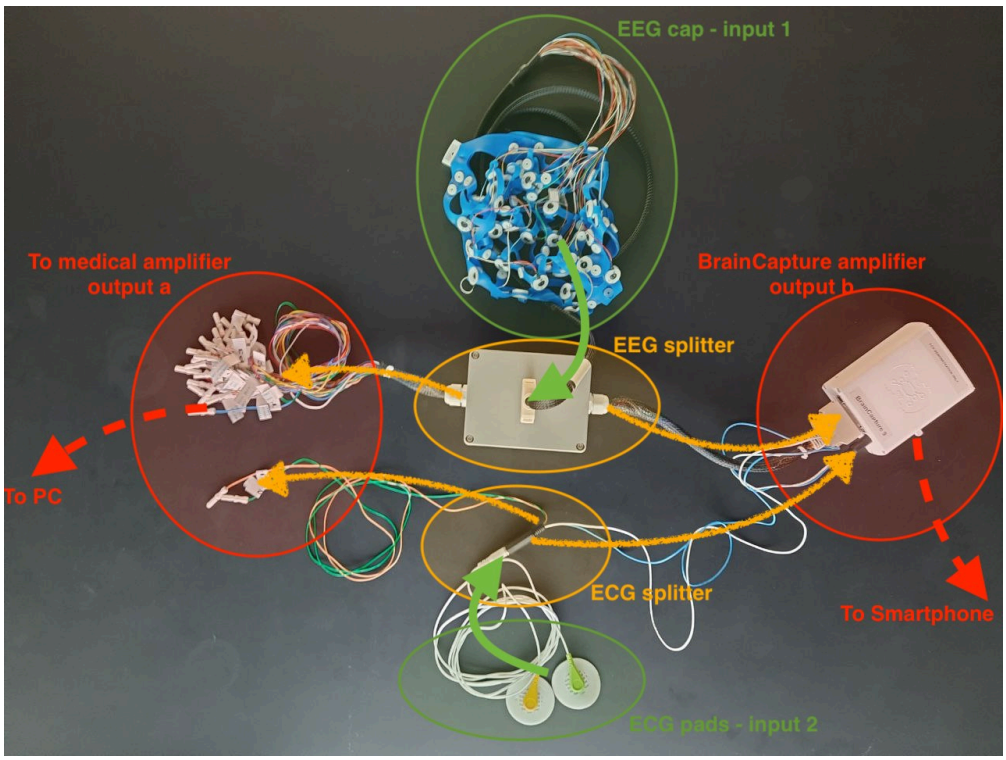


Figure 5.1: System setup of the cap-splitters-amplifiers. EEG and ECG signals are captured by the EEG cap and ECG pads (encapsulated by the green ovals in the picture). The splitters (in yellow) take these signals and split them redirecting the data to the BrainCapture's amplifier (red on the right) and to the Natus' amplifier (red on the left). The signals are then recorded by the relative softwares. The data is therefore recorded simultaneously and from the same electrode positions.

As it can be seen from Figure 5.1, the EEG cap (green on top) and the ECG pads (green on bottom) are connected respectively to the EEG and the ECG splitters (yellow at the center). From there, both inputs are split and redirected one to the medical device, and from there to the PC via cable (red circle on left), and the other to BC's amplifier and then to a smartphone via Bluetooth. In this way, the same signal can be recorded almost simultaneously and from the same location.

The device used by FIL hospital is the Natus Nicolet Cephalon V32, and the signal was recorded with Natus Nicolet ICU Monitor [52] for which the same montage had been set. The montage used by BC is a channel-to-reference montage, following the 10-20 standard system [53]. The channels used are shown in Figure 5.2. For BC's recording, BC's amplifier has been used and the data was recorded with BC's mobile application.

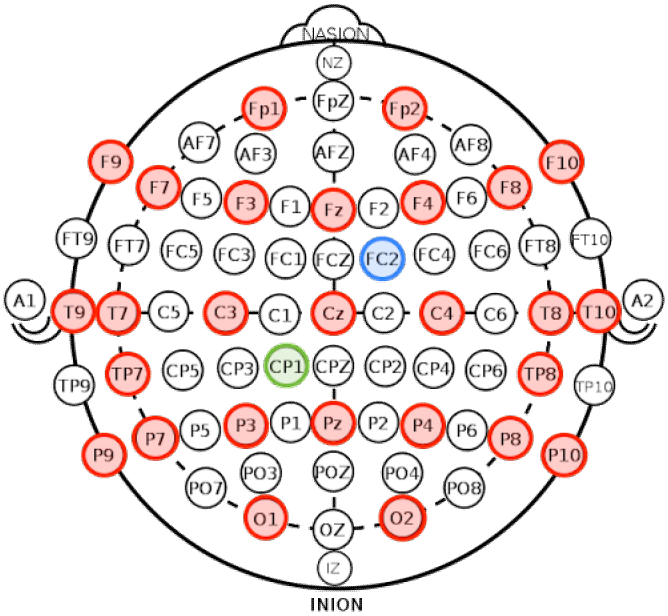


Figure 5.2: Montage of the BC cap, following the 10-20 standard system. The red circles are the electrodes used in the 10-20 system, the blue circle is the "GND" channel and the green is the "REF" channel.

Fourteen participants joined the experiment. Most of them were BC's employers or belonged to its network. 8 of them were between 23 and 25 years old, the remainder were from 35 to 45 years old. 64% of users were male. All participants were healthy and with no known previous epilepsy seizures. None of them had impairments on the scalp that requested particular setups.

For each participant, the EEG technician applied the gel to obtain a value of impedance lower than 10 KOhm [47] for each electrode. Then, after the person lay on a bed, a one-minute scan of the resting state with eyes closed was recorded to assure that the system was working properly and for having a calibration scan. During this minute, the technician checked that the participant was relaxed, that the eyes were closed, and that no artifacts were produced or, in case, annotated.

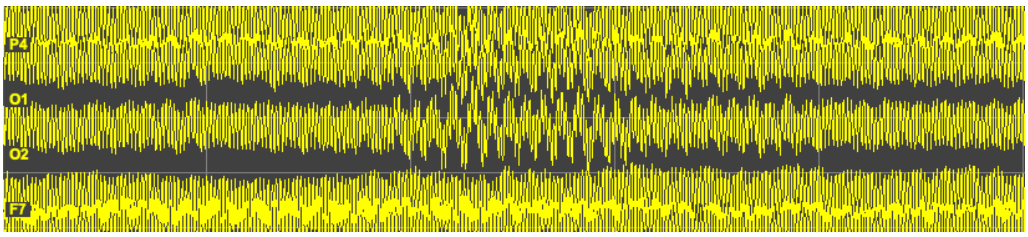
Subsequently, to each participant, a series of tasks had been assigned, following the protocol shown in appendix F. The protocol fulfills the different needs of the projects, and it is intended to have a ground truth on what the user is doing at that moment. It can be seen as divided into three phases. The first one consists of 10 minutes of resting state. The participant is asked to relax as much as possible, keeping the eyes closed, and not falling asleep. In the case of systematic artifacts created, different strategies have been applied to keep them under control (e.g. a light mask on the closed eyes to reduce unintentional later eye movement). Sporadic artifacts have been annotated. The second phase introduced artifacts in the recording. To obtain them, the person was asked to perform a series of movements to generate the artifacts. In particular, in sequence and for 30 seconds for each task, the participant was asked to blink the eyes (without forcing the movement [REF!]), to move the eyes from left to right and back (reproducing later eyes movement), to move the head from left to right and back, to chew, to repeatedly clench the jaw, and finally to shiver (i.e. moving the body convulsively). All the sequence was repeated two times. Lastly, the third phase inspects different scenarios of resting states. The participant, always keeping the eyes closed and relaxing, was asked to rest for 2 minutes. Then, for another 2 minutes, an arithmetical exercise consisting of counting back by step of 7, from 200, was performed to study a focus state. Lastly, to simulate electrode pops, some electrodes, randomly chosen, were moved and popped (always including the ”REF” channel as last).

BC’s application automatically annotates the beginning and end times for each task. On the FIL’s scan, instead, the technician was asked to annotate every not expected artifact. In the second stage, the two signals (i.e. BC’s and FIL’s scans) have been synchronized, and the annotations are shared between the two files. In this way, each file has annotations for the ranges of the exercises, and, within the exercise, all the artifacts produced. This will make the following processes easier and more automatable. Finally, each scan has been re-analyzed by the technician, checking that all annotations were correct.

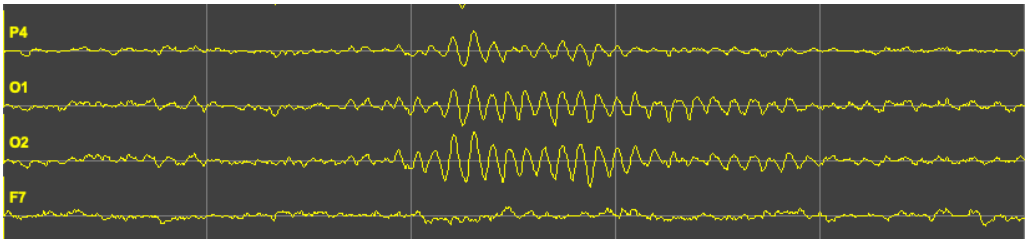
5.2 Results and conclusions

All the recordings acquired by the Natus device are correct and standard. On the other side, some problems appeared on the BC's scans. In particular, three scans had been corrupted, therefore not useful. The main reason for corruption is due to the splitter [38]. More details will be given in the next sections.

In all the visualizations shown in this chapter, BC's data has been band passed with a fourth-order Butterworth filter between 0.5 and 70 Hz, and then a resonator model notch filter of 20 Q-factor at 50 Hz has been applied to the data. This is because, while FIL's device has internal analog filters that clear the data, BC's device does not contain any, and this results in noisy data that would not be possible to visualize, as shown below in Figure 5.3.



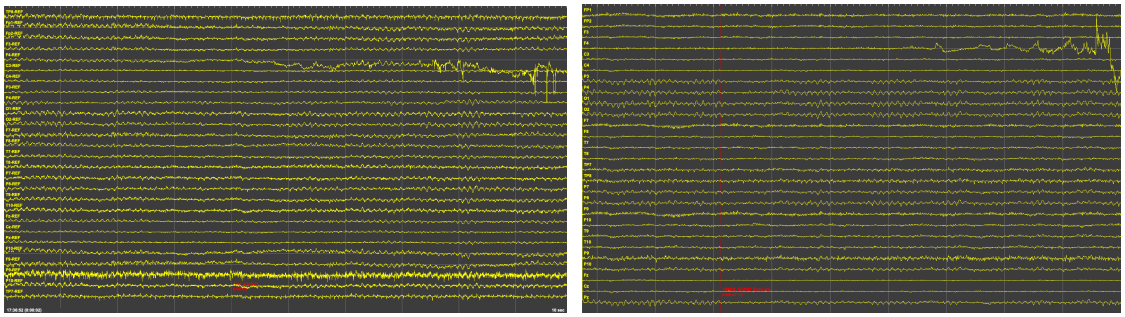
(a) BC's recording before filtering. The x-division is $1s$, the y-division is $100\mu V$.



(b) BC's recording after filtering. The x-division is $1s$, the y-division is $100\mu V$.

Figure 5.3: BC's data visualization before (a) and after (b) filtering. The x-division is $1s$, the y-division is $100\mu V$. The signal after filtering is clearer than before. This is especially due to the power line $50Hz$ noise captured by the unfiltered BC's amplifier, that overlaps the brain signal.

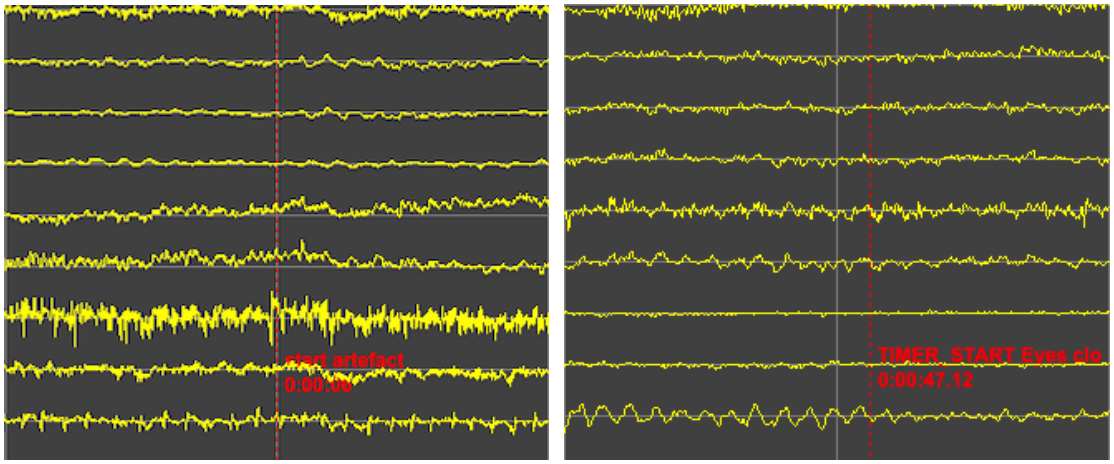
Calibration



(a) FIL's recording (b) BC's recording

Figure 5.4: Resting state data used for calibration. The x-division is $1s$, the y-division is $100\mu V$. The pictures are shown only for giving an overview of the data used. They are not intuitive, the details are shown in Figures 5.5 and 5.6.

Figure 5.4 shows two examples of chunks of recording from BC (b) and FIL's (a) amplifiers. The pictures are not intuitive at a glance, so the details will be pointed out in the following. Apart from the order of the channels shown by the viewer program, it can be seen that the channels recorded are the same, with the same montage (i.e. each channel referenced to reference). Therefore, the same procedures and algorithms can be used alternatively on the two different files. This can be useful to compare the performances of the two different sources of data. In addition, as described in the methods, it can be seen in Figure 5.5 that in the FIL recording (a), the artifact (electrode pop) on the channel "F4" is annotated, and the beginning of the exercise (i.e. "TIMER START eyes closed") is annotated on the BC scan (b).



(a) FIL's annotations. In the files from filadelfia's system, the artifacts within "clean" exercises (i.e. resting state, eyes closed, eyes open and still, and eyes closed exercises) are annotated.

(b) BC's annotations. In the files from BrainCapture's system, the beginning and end of each exercise are annotated.

Figure 5.5: Annotations in FIL's (i.e. artifacts) and BC (i.e. exercises extremes) files. The x-division is $1s$, the y-division is $100\mu V$. Putting the annotations together, is going to be possible to create the ground truth for training an algorithm in an automatic way.

Finally, it can be seen in Figure 5.6 that artifacts in the calibration phase have been annotated. As it will be shown in the following chapters, this will be of extreme importance in the correctness of the calibration, to remove those parts from it.

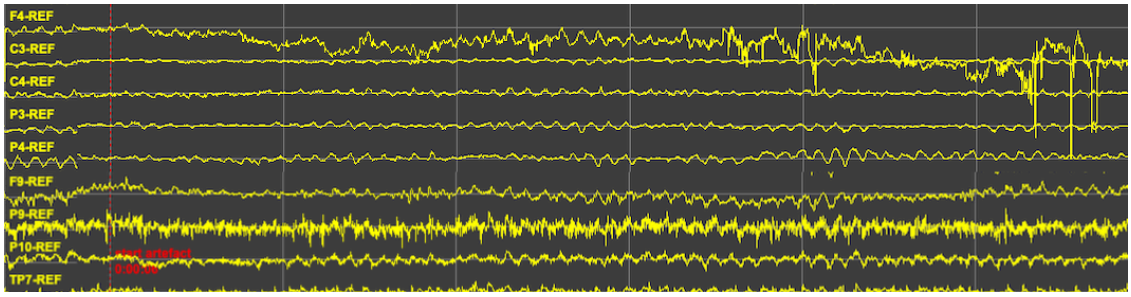
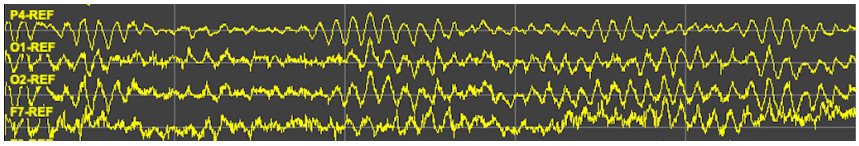


Figure 5.6: Annotation of the artifacts in the calibration phase (FIL file). The x-division is $1s$, the y-division is $100\mu V$. In this case, only FIL's annotations are relevant, because no exercises are being performed in the calibration, but only resting state.

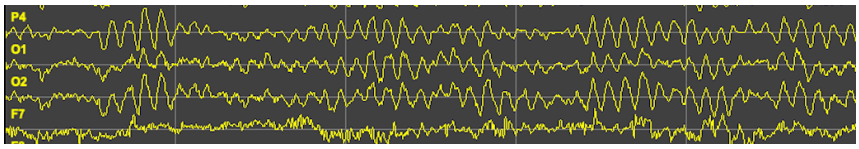
The calibration stage will be useful in the following stages of the project, in particular, is needed for the artifact detection algorithm. The idea of a calibration stage popped up from hands-on experiences collected from a pilot study for debugging the entire system.

Resting state and mental arithmetic

After verifying with the calibration state that the entire system was working properly, the protocol started with ten minutes of resting state, thus the participant lying on a bed, with eyes closed and relaxed. As for the case before, the picture of the entire scan would not be informative, therefore only some portion of them will be shown.



(a) Alpha waves in FIL's recording



(b) Alpha waves in BC's recording

Figure 5.7: Visualization of alpha waves during resting state. The x-division is $1s$, the y-division is $100\mu V$. Alpha waves are characterized by having a frequency between 8-13Hz. Easily verified in the pictures above as 10 peaks are present for each time division. Alpha waves appear when the patient is calm and focused. This is the state in which the doctors prefer the patient to be, for the best EEG readability.

In Figure 5.7 a visualization of the resting state is shown. It can be seen that a clear pattern is created especially on the occipital lobes. This pattern, called alpha waves, has a particular frequency between 8-13 Hz [54]. It is the normal behavior of the EEG for the resting state, and it is the most common for the standard epilepsy protocol [55]. In this mode, it is easy, for the doctor, to detect seizures, as the pattern of the data is very simple and clear, and any change can be easily visually detected. Also in this case, all the artifacts have been annotated, building the ground truth for any possible learning algorithm. In Figure 5.7 (a), the FIL's data is shown, while (b) visualizes BC's data. It can be seen that, as expected, there are no behavioral differences in the data. Some channels are though noisier (e.g. "F7"). This is a well-known problem from the company that, up to the end of this project, didn't find a reason for it. The researches are focused on the hardware and on the splitter setup [56].

As there are no relevant differences between the visualizations of the recordings acquired by the two different devices, for the next cases, only FIL's data will be shown.

For the mental arithmetic exercise, no relevant differences appear with respect to the resting state. The most relevant difference is in the presence of more eye movement, due to the drowsiness of the participant, as the exercise was the semi-last of the protocol. An example can be seen in Figure 5.8 below, where shifts of the signals can be seen in channels "F7" and "F8", indicating eyes movements:

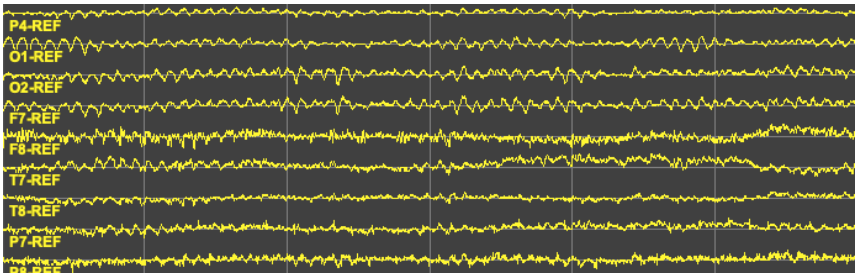


Figure 5.8: Visualization of EEG data during the mental exercise. The x-division is $1s$, the y-division is $100\mu V$. Also in this case, alpha waves predominate the EEG. The EEG is clean and readable by doctors.

In rare cases [57], midline theta rhythm, also known as "Ciganek" [58], can appear on the EEG scan, as shown in the figure below. In this case, it can be seen that also the ECG is visible on the EEG.

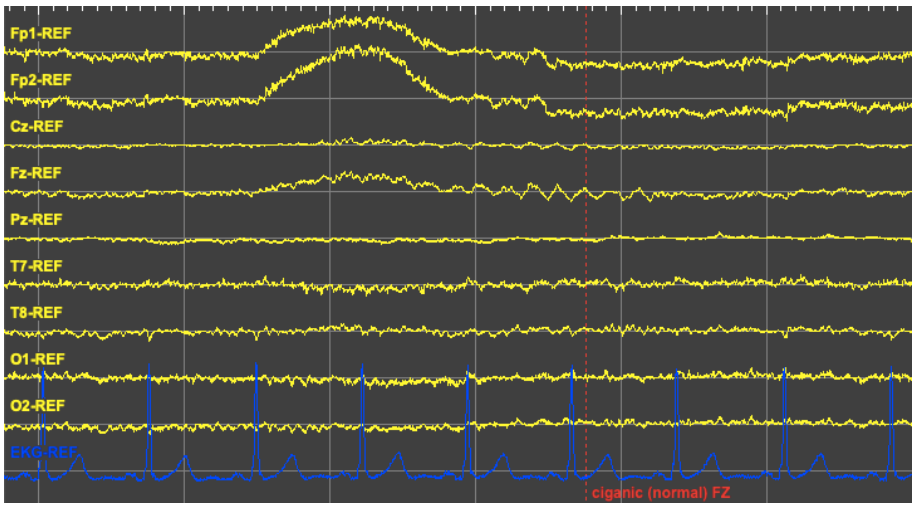


Figure 5.9: Visualization of EEG data of Ciganek rhythm on ”FZ”. The x-division is $1s$, the y-division is $100\mu V$. ECG signal (”EKG-REF” channels) has been resized to $500\mu V$. Ciganek rhythm is visible on the ”FZ-REF” channel and shown with an annotation. In this particular scan, it can be seen that the ECG signal is visible also in the EEG channels. For example, in ”T8-REF” channel a peak synchronized with the ECG peak is visible. This is the reason why ECG signal is recorded. In this way, neurologists can have a reference and do not confuse ECG peaks with epileptic seizures.

Eyes open

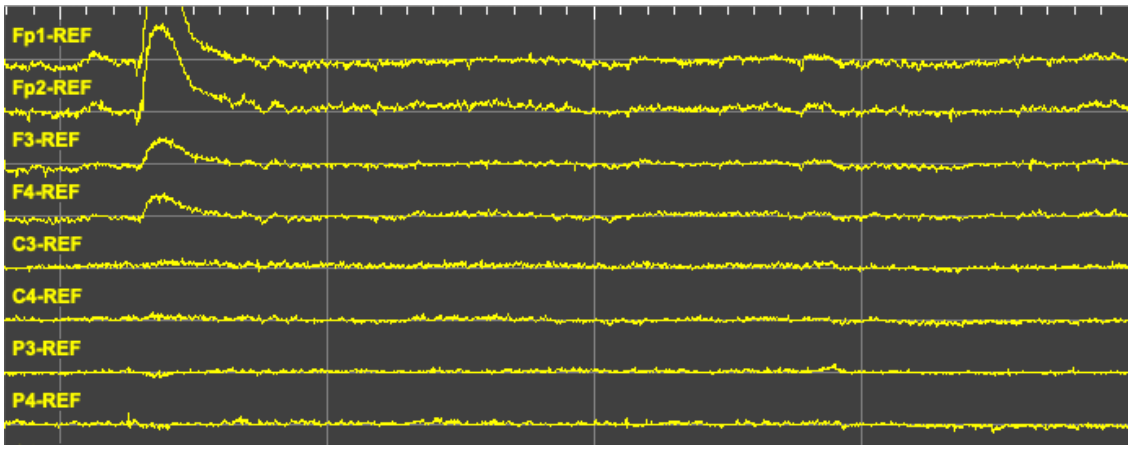


Figure 5.10: Visualization of EEG data for still open eyes in a resting state. The x-division is $1s$, the y-division is $100\mu V$. With eyes open, alpha waves disappear [59] [60]. Eyes blinking is visible especially on ”FP1” and ”FP2” channels. These potential changes are introduced by the corneo-retinal dipole (CRD) and the eyelid.

The above figure shows how EEG data appears when the participant is relaxed and keeps their eyes open. It can be seen that the alpha waves disappear. This fact is still the subject of debate among researchers. In the past much credence was given to the theory that the presence of alpha waves indicates synchronization of brain activities due to ”having nothing to do” [59] [60], and, on the opposite, stimulating primary sensory area generates low-amplitude, desynchronized EEG patterns in which the alpha rhythm is attenuated or blocked [59]. Now, many studies discredit this view, bringing more and more examples of cases where this is contradicted by experiments [61] [62] [63].

Another fact to take into account is that, when the eyes are open, blinking and eye movements are more likely to appear than when the eyes are closed. If for blinking the reason is clear, for the eyes movements more comments can be done. In fact, even though with eyes open it is easier than the participant moves to look around, it is also easy to instruct him (of course if cooperative) to stare at a fixed point, blocking all possibilities of movement. This is not possible when the eyes are closed, in which case the patient cannot focus on a target. It is in fact well-known [64] that when the eyes are closed, the presence of lateral eye movements increases, especially when the participant is tired or is falling asleep. For some patients, it is necessary to place a light mask over the eyelids to decrease these movements. The characteristics of blinking and eye movements are going to be explained in the next sections.

EOG - Eyes blinking and lateral eyes movement

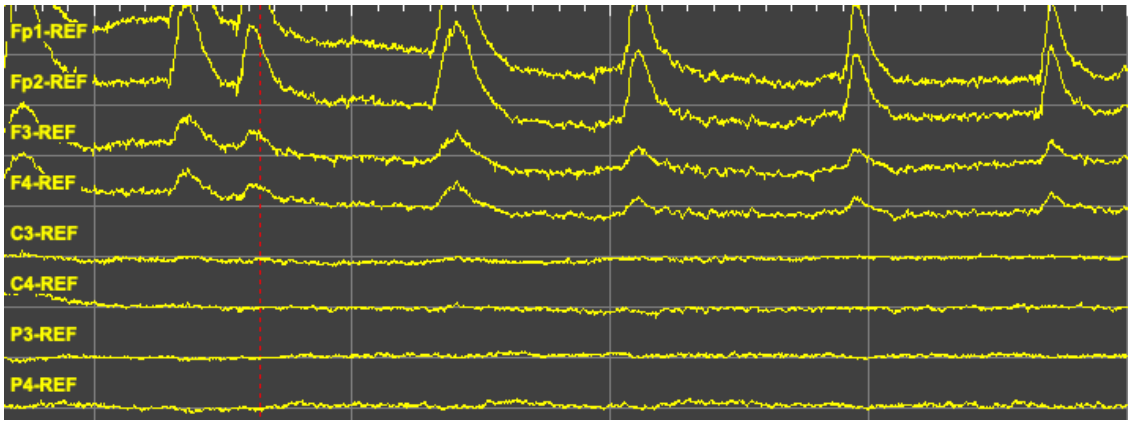


Figure 5.11: Visualization of EEG data for eyes blinking. The x-division is $1s$, the y-division is $100\mu V$. The participant is requested to repeatedly blink the eyes. The frequency is decided by the user, but between 0.5 and $1Hz$. The artifact is stronger on ”FP1” and ”FP2” channels because closest to the eyes. Then, the amplitude of the signal decreases distancing from the eyes region.

Figure 5.11 visualizes EEG data of a person that blinks the eyelids. As the eye moves, the corneo-retinal dipole (CRD) and eyelid introduce potential changes in the EEG activity [65]. In this case, the participant was asked to repeatedly blink the eyes, thus the frequent artifact. It is easy to see, and supported by literature [66] [67], that the blinking artifact shows the peak on the ”FP1” and ”FP2” channels (i.e. forehead) and that the power decreases moving away from the eyes. For example, channels ”F3” and ”F4” have decreased peaks, and ”C3” and ”C4” are even lower. The participants have been instructed to naturally blink their eyes and to not force the movement, leading otherwise to different effects [68].

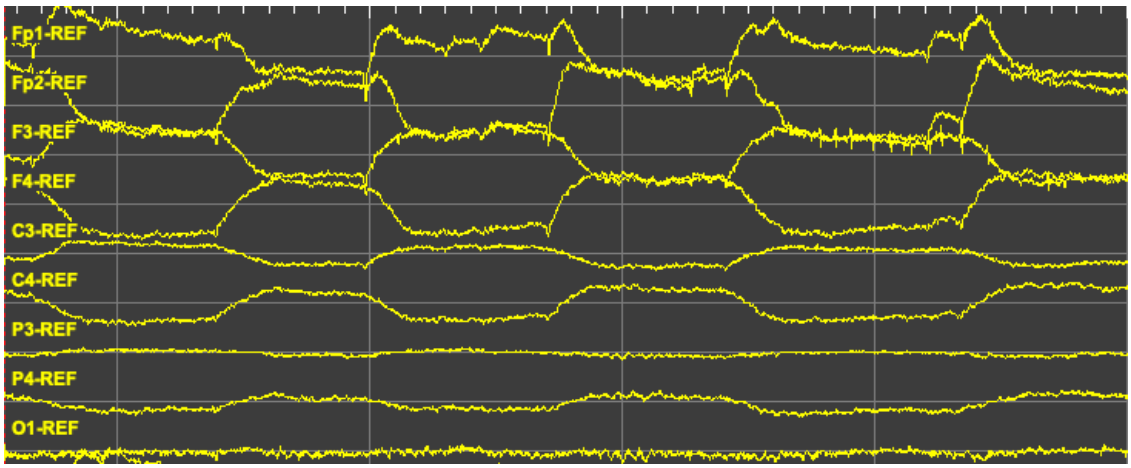


Figure 5.12: Visualization of EEG data for lateral eyes movements, with open eyes. The x-division is $1s$, the y-division is $100\mu V$. The participant was requested to keep the eye lids open and move the eyes horizontally, at a periodic rhythm. The corneo-retinal dipole induces even more diffuse artifacts then the eye blinks.

In Figure 5.12 the appearance of lateral eye movements is shown. The idea is the same as for the eyes blinking, the movement of the corneo-retinal dipole (CRD) induces potential changes in the EEG recordings [68]. As for the case before, the peak of the artifact appears on the frontal channels, reducing in intensity while moving away from the eyes (i.e. ”FP1”-”FP2” towards ”F3”-”F4” and then ”C3”-”C4”) [66] [67]. Also in this case, while it is easy to instruct a cooperative patient to not move the eyes when they are open, it is more difficult to control them when the eyelids are closed. Again, in certain circumstances, external solutions have to be taken, for example the eyes mask [69]. An example of this problem is shown below:

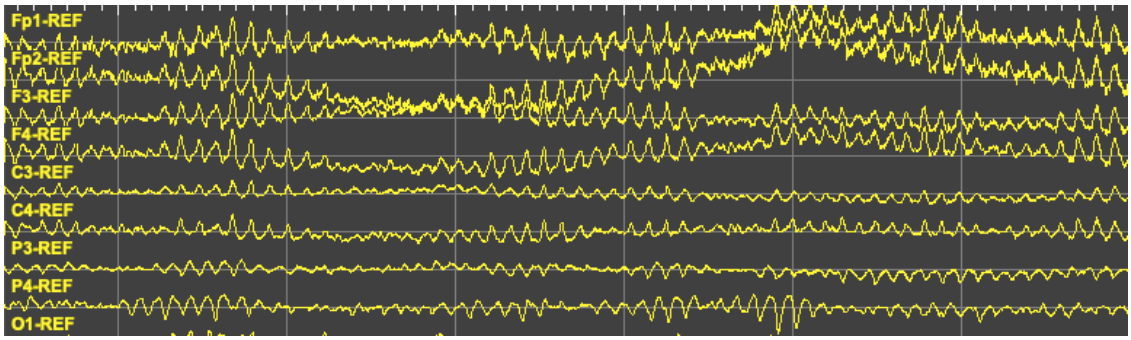


Figure 5.13: Visualization of EEG data for lateral eyes movements when closed eyes. The x and the y-division are as before. This time the eyes are closed and it can be seen the overlap of lateral eye movements and alpha waves. In this case, the lateral movements are unintentional, thus the rhythm is not forced.

In this case, it can be recognized that the eyes were closed because of the predominant alpha waves, but the shifts of the signal visible for example on ”FP1”, ”FP2”, ”F3”, and ”F4”, are caused by eye movements. This sometimes cannot be avoided.

EMG - Lateral head movement, chewing, jaw clenching and shivering

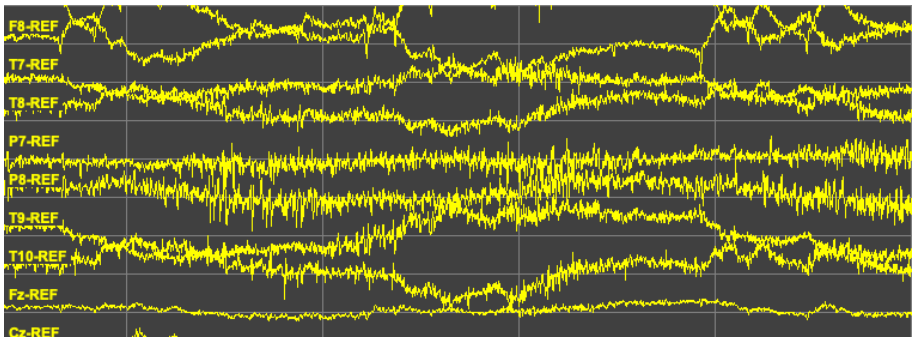


Figure 5.14: Visualization of EEG data for lateral head movements. x and y-divisions are $1s$ and $100\mu V$. The participant is instructed to move the head horizontally. The artifacts are due to the muscle movements and stress of the neck and shoulders, and from the movements of electrodes in the scalp.

In the above figure, it can be seen that an important noise is added to the signal. This is because an extreme number of muscles are involved and a lot of muscle tension is created [70]. The muscles are distant from the scalp, so the effect is less visible. It is generalized to all channels, but it is of particular strength on the temporal lobes [71].

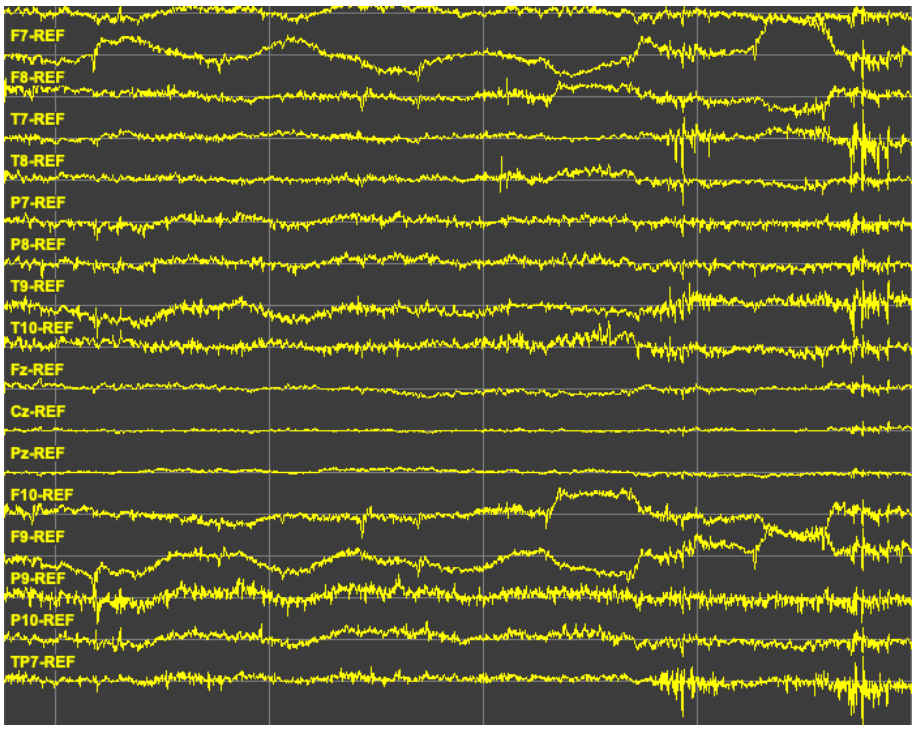


Figure 5.15: Visualization of EEG data for a person chewing. x and y-divisions are $1s$ and $100\mu V$. The participant is instructed to chew, without closing the mouth. In this case the artifacts are generalized in all channels, and are due to the muscle movements and stress of the jaw area.

Figure 5.15 visualizes the artifacts produced by a chewing action. Again, as skeletal muscles are involved, the noise is generalized and most focused on temporal lobes [66]. This artifact, even if rare, is of great importance, as it can cover the signal completely [69].

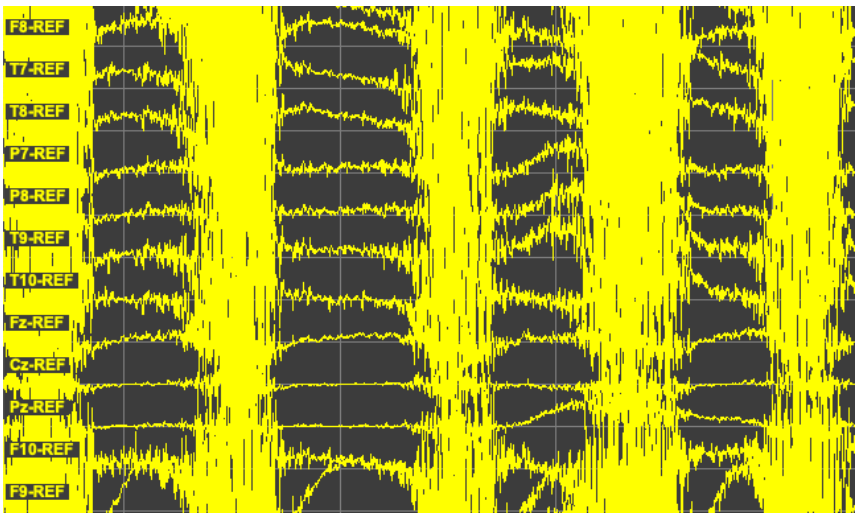


Figure 5.16: Visualization of EEG data for a person clenching the jaw. As for before, x-division is $1s$ and y-division is $100\mu V$. The artifact generated asking the participant to repeatedly clench the jaw is strong and generalized. In this case, the participant was instructed to clench the jaw hardly, but similar effect are visible already for low pressures [66].

Figure 5.16 shows the appearance of a clenching artifact. This artifact is of extreme power and is present on all channels. The intensity depends on the strength of the clench, but it is visible already at low pressures [66]. The signal is unstable also a little before and for some time after the action itself. The length of this artifact can be between a few hundred milliseconds to some seconds [66].

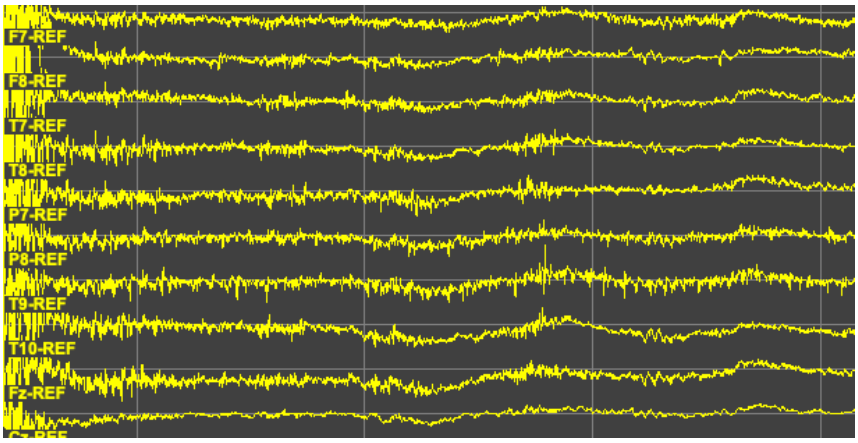


Figure 5.17: Visualization of EEG data for a person shivering. x-division and y-division are $1s$ and $100\mu V$. The shivering generates a generalized artifact due to muscles activity in the body, and electrode movements in the scalp.

In Figure 5.17 the shivering artifact is shown. In this case, the artifact is not of extreme strength but in general could be more intense. As for the other EMG artifacts, the noise is generalized to almost all channels. The artifact length is usually shorter than a second. In this case, the noise is longer because the participant was asked to simulate it by moving the body convulsively.

Electrode pops

In Figure 5.18 below, an example of an electrode pop artifact is shown:

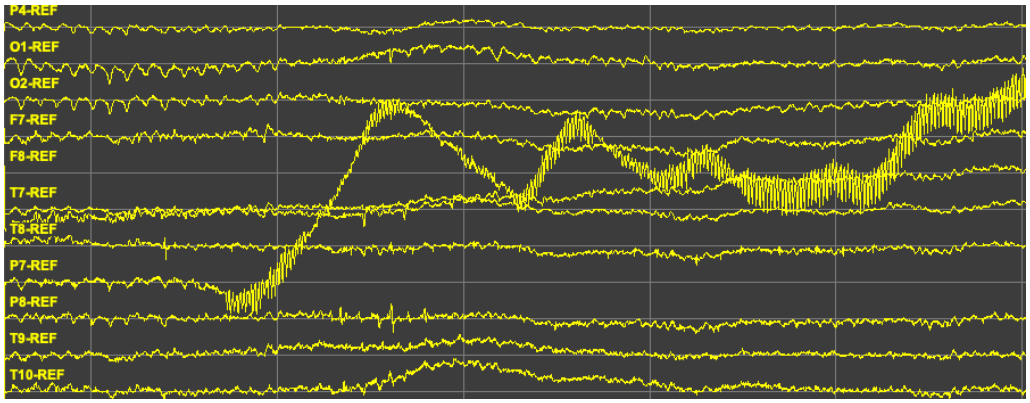


Figure 5.18: Visualization of EEG data for an electrode pop artifact on ”P7”. x-division is $1s$ and y-division is $100\mu V$. This artifact is characterized by a rapid change of a single channel, in the case of a channel-to-reference montage. It is normally generated by movements of the electrodes.

The electrode pop is usually easy to recognize and characterized by an abrupt change of one channel. In a channel-to-reference montage (i.e. as in this data), it can involve only one single channel, and it usually goes to normality after a few seconds. It can be caused by many factors as dryness or sweating skin, as well as rapid movements of the head or also for physical damage. The case is different if the electrode pop involves ”REF” or ”GND” channels. In fact, all voltage differences are computed with respect to ”REF” (i.e. channel-to-reference montage), while ”GND” is involved in a feedback-loop circuit to keep the ”REF” channel stable (DRL circuit [72]). Therefore, problems in ”REF” or ”GND” can affect all the other channels, as shown in the figure below:

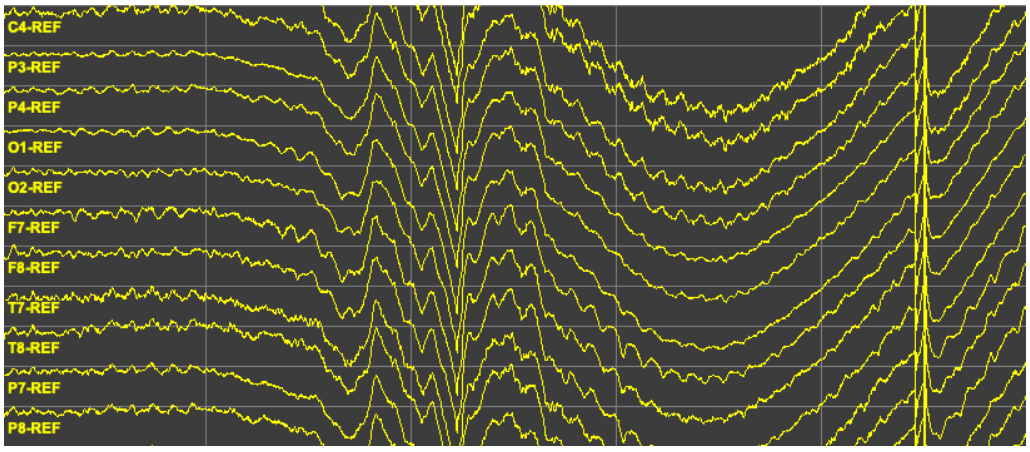
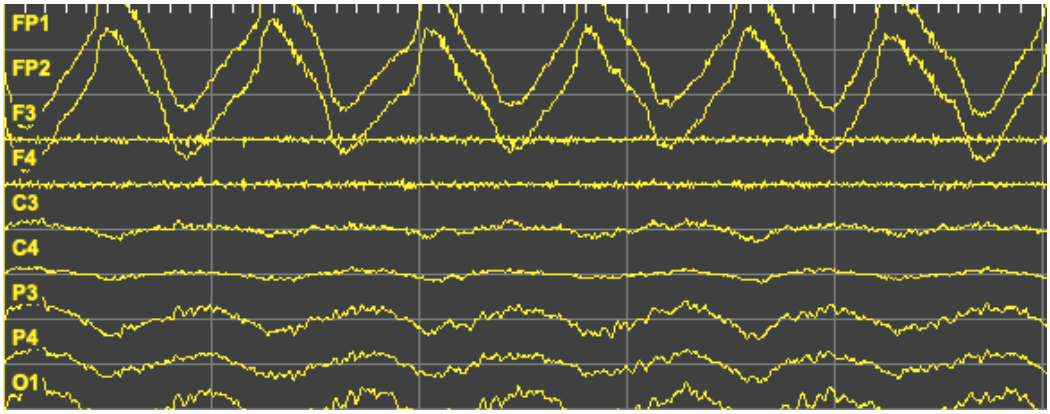
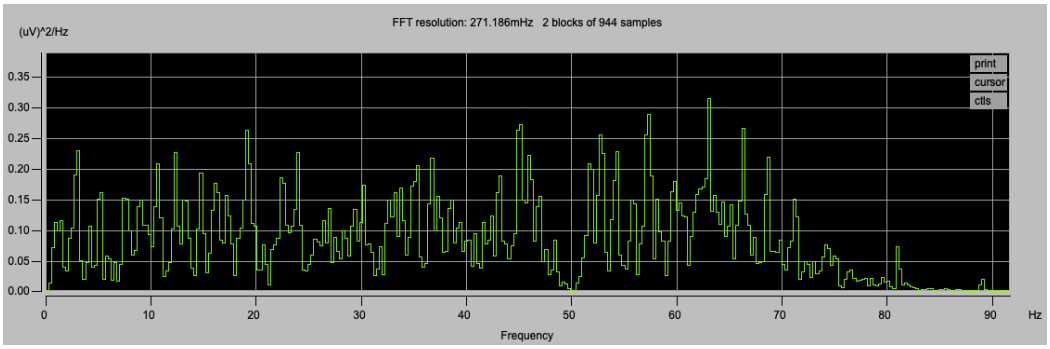


Figure 5.19: Visualization of EEG data for an electrode pop artifact on ”REF”. x-division is $1s$ and y-division is $100\mu V$. This is a particular case of electrode pop, because all channels are referenced in this montage to reference. So, if ”REF” shifts, all values of all channels are affected. The DRL circuit is designed for handling small variations in the ”REF” channels, and not for these events.

Technical problems
BC's amplifier is, as of writing, a beta device and some technical problems appeared in some recordings. In particular, the most common problems were that some channels were just recording noise (e.g. Figure 5.20) and that the ECG signal was spreading on all the channels (e.g. Figure 5.21).



(a) Example of a recording with just noise in "F3" and "F4" channels. x-division is $1s$ and y-division is $100\mu V$.



(b) Power spectrum histogram of the "F3" channel. It can be seen that the power of the signal is equally distributed over the entire range, with a negative peak at $50Hz$ and a decrease after $70Hz$ due to the notch and the bandpass filters always applied at the BC's recordings. A decrease also below $0.5Hz$ should be visible from the theory, but this particular visualization doesn't show it intuitively.

Figure 5.20: Visualization of recordings with noisy channels (a) and the power spectrum for one of those channels (b). The power spectrum of the noisy "F3" channel is almost uniformly distributed, showing the behaviour of white noise.

As it can be seen from Figure 5.20 (a) above, while the participant is blinking (see "FP1" and "FP2") the "F3" and "F4" channels are not showing any artifacts. In 5.20 (b), the power-frequency histogram is shown. It is easy to see that, instead of having a signal, just white noise is recorded, most probably due to the thermal energy of the system. The root reason is still under analysis from BC, and no formal solutions are given, even though the most probable source is on the splitter setup [38]. It is interesting to see the absence of the 50Hz and the decrease of power after 70Hz due to the filters applied for the visualization, as described before.

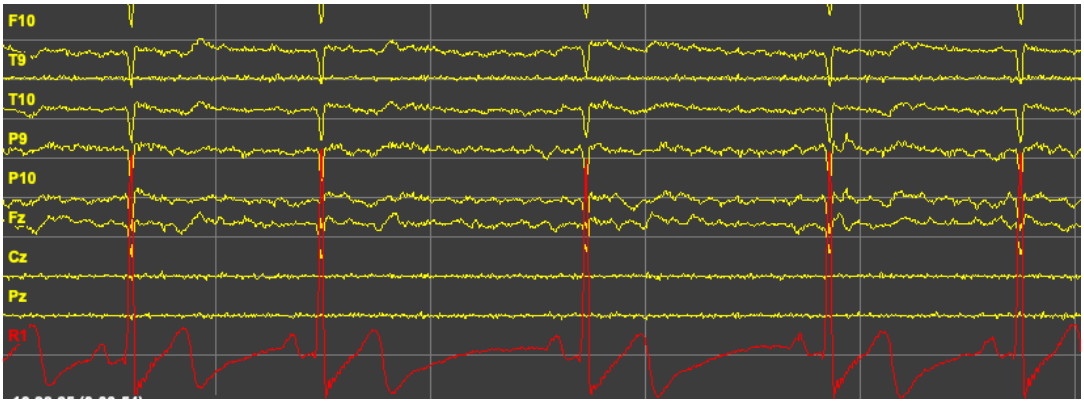


Figure 5.21: Visualization of EEG data for the ECG leakage. x-division is $1s$ and y-division is $100\mu V$. The peaks amplitude for the ECG "R1" channel has been resized to $500\mu V$. The ECG signal is spread over all the channels. This case differs from the one in Figure 5.9 because the ECG is present in all channels, and the amplitude of the artifact is independent from the location. Also, noisy channels are visualized, for example "T9", "Cz", and "Pz", that in fact are not affected by the leakage, as they are not acquiring any signal.

In Figure 5.21, the ECG leakage can be seen throughout the scan. Also this problem has still to be assessed and no explanations are available from the company at the moment.

5.3 Discussion and future work

Protocol

As doctors are interested in the resting period after the stimulations (see section 4.3, on the user needs) the best training and testing data would have been to have resting state recordings with labeled artifacts naturally present in the scan. This would have implied acquiring longer and more numerous sessions, and to have doctors label the recordings, affecting the limited budgets and time of the projects. For this reason, it has been optioned to instruct the participant to artificially create artifacts, not only for having samples of them but also for having a ground truth, as it was known when the artifact appeared. Side effects of the intentionality in the creation of artifacts (e.g. different behavior for the blinking, if forced [68]) have been assessed, to be as close as possible to natural appearances. This can affect and create biases for a detection machine learning algorithm. For example, the periodicity and consistency of the same artifact, created by an intentional repetition, can greatly affect parameters such as window size and generalization error. Moreover, the disproportion of artifacts/non-artifacts data with respect to a real case can affect performance measures based on unbalanced data [73].

In addition, for cooperative patients, epilepsy diagnosis is performed with eyes closed [47] and therefore the artifacts are different from the one recorded (e.g. alpha waves), with eyes open. On the other hand, for non-cooperative users, most of the time multimedia support is used to focus the user and calm him down [69], so eyes are kept open and the data is more similar to the one recorded. A better option, for this study, would have been to instruct some users to keep their eyes open and some to keep their eyes closed during the creation of the artifacts.

Moreover, the order of the tasks is the same for each participant. This can also lead to biases in the data [74], for example, because participants are more tired at the end of the protocol, leading, for example, to the mental arithmetic exercise creating fewer Ciganek waves than expected. Thus, the order of the exercises could have been randomized for less biased data.

Additionally, participants were healthy and relaxed, trained in the EEG recordings, and aware of the effects of their actions, and the technician was an expert. All conditions that differ from the project's real case scan. This is acceptable for a prior study, but, as future work, it should be extended to more real cases, for having a better understanding of the technical problems that can appear, and for better training the detection algorithms.

Data acquired

Some recordings from BC's device were affected by technical issues. While this created a loss of useful data, it also gave many opportunities for heuristics for sanity check options. This discussion will be enriched in future chapters. The correctness of the data has been assessed in parallel by the author and the technician.

The data acquisition took a considerable amount of time. This sacrificed resources for other parts of the project but gave extremely useful hands-on experience contributing to a better understanding of the system and of the related problems.

6 Riemannian Potato Field (RPF) and practical design

In this section, the steps used to design the best potato field and tune the parameters for BrainCapture's cap and data are described. As it will be shown, the potato field is specific for a certain setup.

6.1 Introduction

The Riemannian potato field (RPF) [49] is an online classification algorithm. It assumes that the variance of artifact segments of the EEG signal is greater than when not artifact. Under this assumption, it builds a system of comparison of signal epochs based on covariance matrices processed by Riemannian geometry [75], and rejection based on statistical tests [76]. The RPF is based on the previous Riemannian potato (RP) [77] algorithm, in which only one potato (i.e. frequency-spatial filter) is applied to the signal. The RPF extends RP by introducing the idea of potatoes focusing on each artifact class, and by combining them, through statistical tests, to an aggregate evaluation. In other words, the RPF can be seen as many RPs in parallel, with each RP single result contributing, using the right-tail Fisher's method [76] to the final assessment.

Geometry of Covariance Matrices

For a more in deep mathematical explanation of the properties of Riemannian geometry applied to covariance matrices, references to [75] and [78]. In the following, a brief introduction is given.

With $X \in \mathbb{R}^{C \times N}$ EEG signal epoch, with C channels/electrodes and N temporal samples, its covariance matrix $\Sigma \in \mathbb{R}^{C \times C}$ is estimated easily and fast [49] (assuming $N >> C$ [79], and the signal being centered, e.g., after band-pass filtering), as:

$$\Sigma = \frac{1}{N-1} X X^T \tag{6.1}$$

Where $N-1$ is used for the unbiased estimation version. For the reason that covariance matrices are symmetric positive definite (SPD), they are confined in a cone of the Euclidean subspace defined by the Cauchy-Schwarz inequality [49]. This subspace is shown to possess the properties of a Riemannian manifold M_C, of dimension $m = C(C+1)/2$ [75].

The affine-invariant Riemannian (AIR) distance between Σ_1 and Σ_2, and the geometric mean of $i = 1, ..., I$ matrices Σ_i (defined as the matrix that minimizes the sum of the square distances between Σ_is on the manifold) [80] can be computed just using the eigenvalue decomposition of the Σs involved [49]:

$$\Sigma = U diag(\lambda_1, ..., \lambda_C) U^T \tag{6.2}$$

where $\lambda_1, ..., \lambda_C$ are the eigenvalues and U the matrix of eigenvectors of Σ.

Principle of the RP

The RPF is an extension of the RP. It is so reasonable to start explaining the RP, as minimal construction of the RPF. Again, the following explanation is concise, the details can be found in the original paper [77].

The idea behind RP is to estimate a covariance matrix $\overline{\Sigma}$ of baseline (i.e. not artifacts) and a measure of dispersion (z-score). Then, the Riemannian distance between $\overline{\Sigma}$ and the covariance matrix at "time" t, Σ_t is computed and rejected if higher than an objective statistical criterion [49]. The criterion is based on the z-score for the particular covariance matrix Σ_t at time t:

$$z_t = \frac{d_t - \mu_{t-1}}{\sigma_{t-1}} \tag{6.3}$$

where the mean μ and the standard deviation σ are computed using the Riemannian distances between the previous matrices and the baseline matrix:

$$\mu_t = \frac{1}{t}\sum_{\tau=1}^{t} d_\tau \tag{6.4}$$

$$\sigma_t^2 = \frac{1}{t}\sum_{\tau=1}^{t} (d_\tau - \mu_\tau)^2 \tag{6.5}$$

"This z-score models the distribution of covariance matrices considered as clean and provides a z-score threshold z_{th} defining the hull of acceptability. The Riemannian potato then rejects epochs whose covariance matrices lie out of this region of acceptability" [49], as shown below:

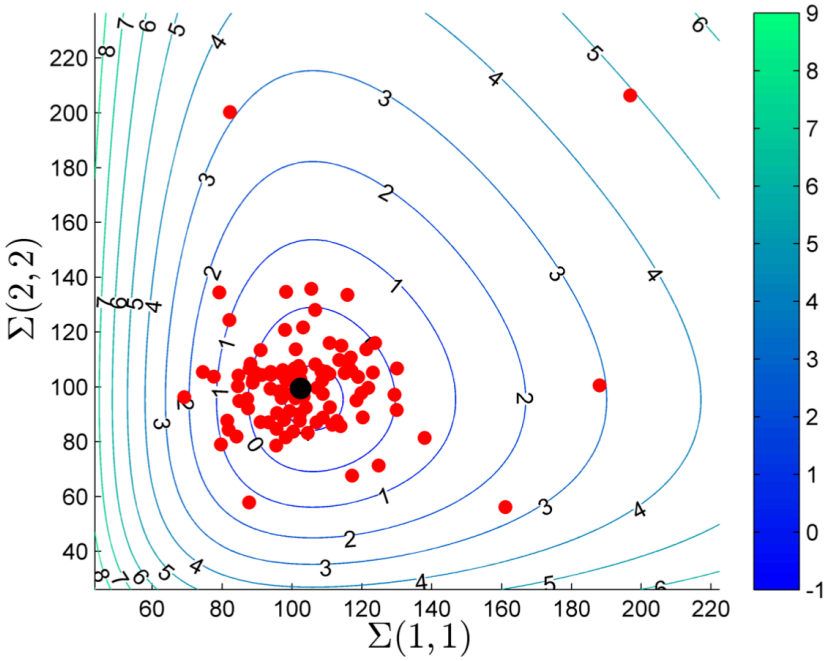


Figure 6.1: 2D projection of the z-score map of a Riemannian potato, for 100 simulated 2 × 2 matrices Σ (in red) and their reference matrix (in black). The colormap defines the z-score and a chosen isocontour $z_t h$ defines the potato. Image taken from [49].

On a standard implementation, the algorithm adapts to the data by updating its parameters (i.e. $\bar{\Sigma}_{t-1}$, μ_{t-1}, σ_{t-1}) whenever the present matrix (Σ_t) is free of artifacts (thus when $z_t < z_{th}$) following:

$$\bar{\Sigma}_t = \bar{\Sigma}_{t-1}^{\frac{1}{2}}(\bar{\Sigma}_{t-1}^{-\frac{1}{2}}\Sigma_t\bar{\Sigma}_{t-1}^{-\frac{1}{2}})^{\alpha}\bar{\Sigma}_{t-1}^{\frac{1}{2}} \tag{6.6}$$

where $\alpha \in [0, 1]$ defines the speed of adaptation or learning rate. This online update is equivalent to $\bar{\Sigma}_t = (1-\alpha)\Sigma_{t-1} + \alpha\Sigma_t$ in the Euclidean geometry [75]. And:

$$\mu_t^2 = (1-\beta)\mu_{t-1} + \beta d_t \tag{6.7}$$

$$\sigma_t^2 = (1-\beta)\sigma_{t-1}^2 + \beta(d_t - \mu_t)^2 \tag{6.8}$$

where the second learning rate β is usually chosen equal to α.

To summarize, the pseudo-code of the RP is shown below:

```
1:  function calibration( {Σ_t}_{t=1}^T ): returns  Σ̄, μ, σ
2:        Estimation of mean Σ̄ from {Σ_t}_{t=1}^T
3:        Estimation of the T distances to the reference Σ̄
4:        Estimation of the mean of distances μ with Eq. (6.4)
5:        Estimation of the standard deviation σ with Eq. (6.5)
6:
7:  function RiemannianPotato(Σ_t, T_init):
8:        initialization: 𝒯 ← ∅
9:        if t < T_init
10:             𝒯 ← 𝒯 ∪ {t}
11:       if t = T_init
12:             Σ̄_{T_init}, μ_{T_init}, σ_{T_init} ← calibration({Σ_t}_{t∈𝒯})
13:       if t > T_init
14:             Riemannian distance d_t ← δ_R(Σ_t, Σ̄_{t-1})
15:             Distance z-score z_t ← (d_t − μ_{t-1}) / σ_{t-1}
16:             if z_t ≤ z_th     //i.e. clean covariance matrix
17:                   Update Σ̄_t with Eq. (6.6)
18:                   Update μ_t with Eq. (6.7)
19:                   Update σ_t with Eq. (6.8)
```

Figure 6.2: Pseudo-code for the online Riemannian Potato algorithm

As explained above, the algorithm collects a certain number of covariance matrices (i.e. computes the covariance matrices of the epochs) and then calibrates the Potato (Figure 6.2, line 12), hence, it computes the mean covariance matrix $\bar{\Sigma}_{T_{init}}$ with the Riemannian mean, and it computes the average and the standard deviation of the distances with the baseline matrix (Figure 6.2, lines 1-5). Then, for each new matrix, it computes the Riemannian distance to the baseline matrix and converts it into the z-space (lines 13-15). If the z-score is lower than the threshold (i.e. the covariance matrix is clean), it updates the inner parameters (lines 16-19).

”Due to its popularity, the RP algorithm has been implemented in different languages like Python, Matlab, OpenViBE, and NeuroRT Studio” [49].

Principle of the RPF

The main problem of the RP is related to the curse of dimensionality. When an artifact generates variations on a channel, the distance between the actual covariance matrix and the baseline doesn't increase significantly if the dimension of the potato is too big (the problem is already appreciable for headsets with 8 channels [49]). Thus the RPF idea of defining and combining several potatoes of smaller dimensions is typically designed to capture artifacts affecting specific channels and frequency bands. Then, all z-scores resulting from the potatoes are combined on an aggregate p-value. Below, is a visualization of the idea:

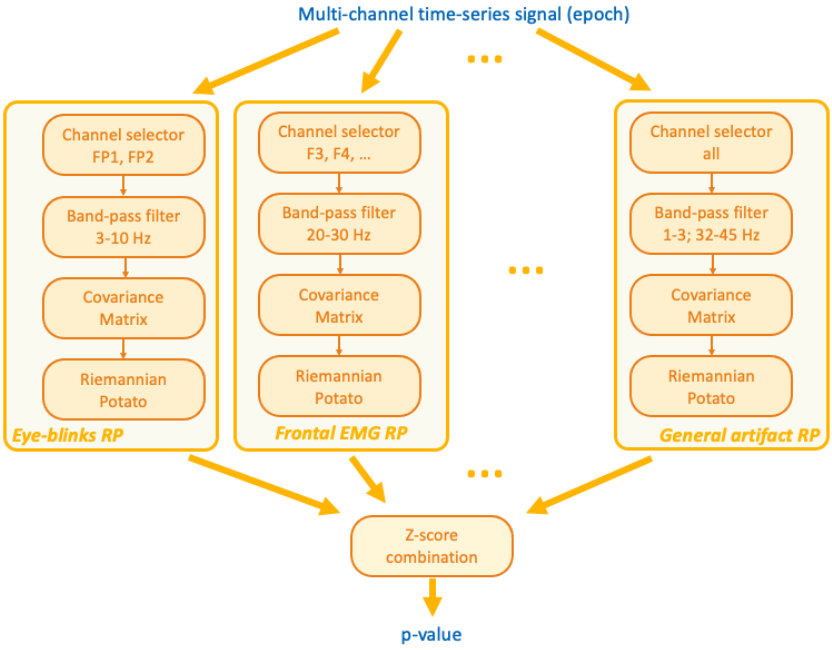


Figure 6.3: Visualization of the RPF algorithm. The idea is to apply the RPs after spatial-frequency filtering the signal, i.e. focusing on each particular artifact.

The final p-value p is obtained using the right-tail Fisher's method [76] on the p-values p_j from each potato, computed as:

$$p_j = 1 - cdf_{\mathcal{N}_{0,1}}(z_j) \tag{6.9}$$

where $cdf_{\mathcal{N}_{0,1}}$ is the cumulative distribution function of the normal distribution with mean 0 and standard deviation 1.

Fisher's method combines the p-values p_j as:

$$q = -2\sum_{j=1}^{J} \log(p_j), \quad q \sim \mathcal{X}^2_{2J} \tag{6.10}$$

where J is the number of potatoes in the RPF, and $\mathcal{X}^2_{2J}$ is the chi-square distribution with 2J degrees of freedom. The final p-value is thus:

$$p = 1 - cdf_{\mathcal{X}^2_{2J}}(q), \quad p \sim \mathcal{U}_{0,1} \tag{6.11}$$

where $\mathcal{U}_{0,1}$ is the uniform distribution between 0 and 1 [49].

6.2 Methods

The algorithm for the Riemannian potato field classifier consists of two major design parts: the field itself (i.e. spatial-frequency filters) and the hyperparameters for the different configurations, for example, the threshold distance after which a covariance matrix is considered artifact.

6.2.1 Design of the field

A potato field consists of a collection of spatial-frequency filters, each of them focused on a particular artifact or class of artifacts, and each designed for a particular setup. This means that, theoretically, all the possible sets of electrodes - with repetitions- and all the possible combinations of cutoff frequencies should be tested. Moreover, prior knowledge is provided by a rich artifacts literature [54] [66] [67] [70] [71] [81] [82] [20] [83]. This allows thinking about the design of the field more from a knowledge-based perspective with respect to a model-based optimization. The method chosen is to design some fields, collect them in a set and use the set as a discrete parameter for the following hyperparameter optimization.

From the system requirements chapter, it is known that the most important artifacts to focus on are the movement artifacts and the artifacts created by the stress in the jaw area. In addition, eye movements are also considered, even though less relevant. For this reason, the design will focus on those characteristics. It's important to highlight that the literature agrees to a certain extent on the topology of the artifacts, hence on the filters to use. Also, for this reason, it is important to try different combinations, because different indications are retrieved from different sources.

Artifacts can be physiological (originated because of the electrical activity of other body parts) or external artifacts (the sources of these artifacts are electronic gadgets, transmission lines, etc.) [82]

EMG

Electromyogram (EMG) contamination derives from contractions and movements of the -in particular- head muscles. EMG has a broad frequency distribution, but it has the greatest amplitude between 20Hz and 30Hz frontally and 40Hz and 80Hz temporally. [66] [71] [82]

EOG

Electrooculogram (EOG) records ocular activities. Generally, ocular activities are strong enough to be recorded along with EEG. Eyes movements and eyes blinking are primarily picked up by the frontal electrodes in the 3Hz to 10Hz frequency range. [81] [82]

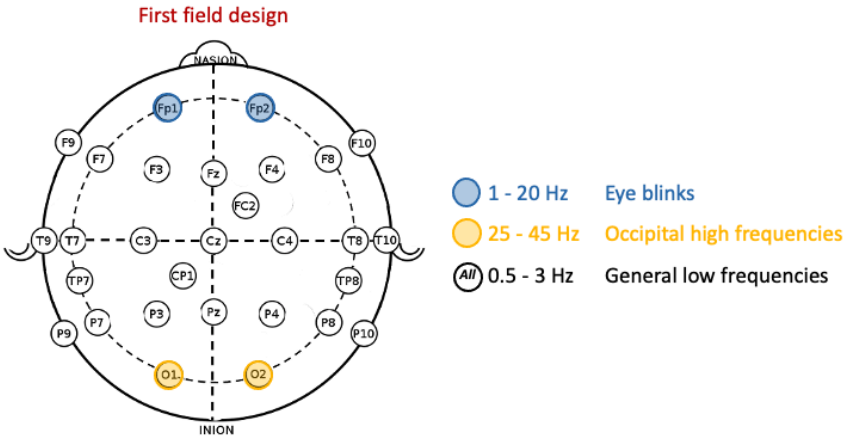
ECG

ECG is not of particular interest to this application. Firstly because ECG is not avoidable, secondly because doctors are trained to read the EEG over the ECG, and thirdly because also the ECG is recorded, making the removal of this noise in post-processing easy.

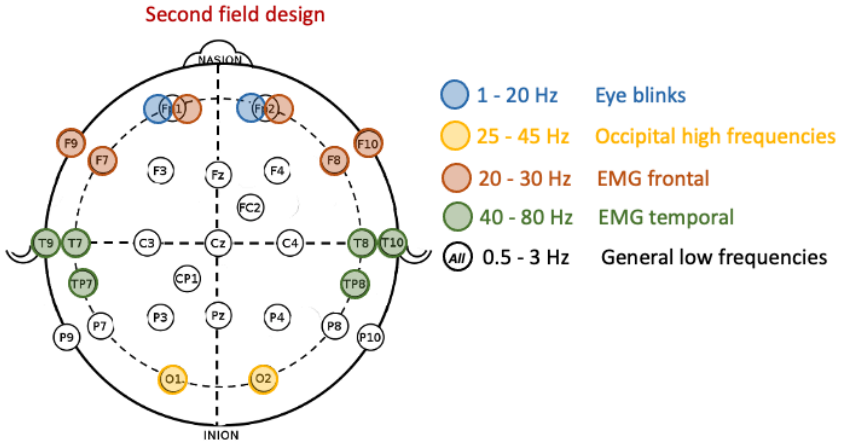
Other artifacts

The other most important artifact to care about is the electrode pop. Electrode pop can be caused by several conditions, among others heavy movements, sweating of the patient, and malfunctioning of the electrode. This phenomenon results in a sudden drop in the covariance of the target electrode with any other. Consequently, any two-by-two pairs of electrode band-pass filtered around 1 and 20 Hz would effectively detect these events. [49]

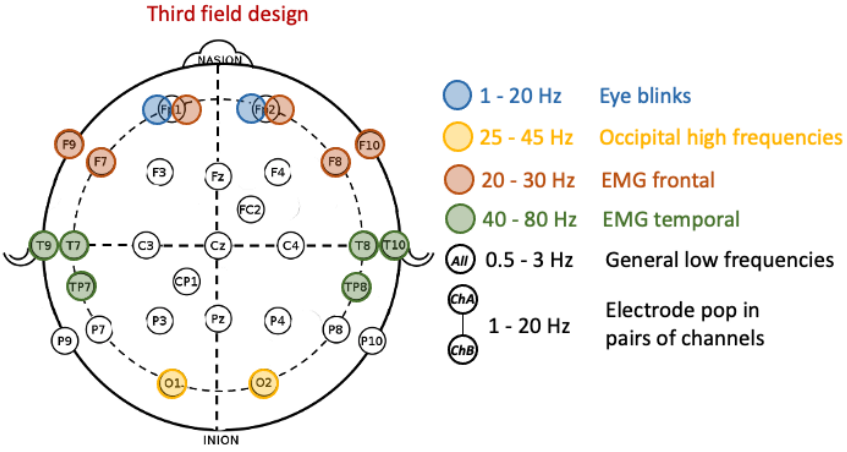
Therefore, three different configurations have been designed starting from the above considerations. In particular, below is a visualization of the different combinations chosen:



(a) Visualization of the first design for the potato field



(b) Visualization of the second design for the potato field



(c) Visualization of the third design for the potato field

Figure 6.4: Visualizations of the three potato field designs for the first iteration of parameters tuning

In Figure 6.4 the three designs for the potato field in the set for the first iteration are shown. As a reminder, a potato field is a set of spatial-frequency filters to be applied to the EEG signal, and then to perform the RP algorithm on each of them.

The spatial side of the filter is highlighted on the scalp figure. In more detail, one color is assigned to each potato, so channels colored differently belong to different potatoes, and channels of the same color belong to the same potato. The frequency part of the filter is visualized by writing the cutoff frequencies for each color (i.e. for each potato).

It can be seen that each field is an extension of the design above. In particular, in the potato in Figure 6.4 (a), the field is composed of three potatoes: one for the eye blinks, one for high frequencies artifacts in the occipital area, and the last that includes all channels and focuses on low frequencies artifacts. The design is not specific and it doesn't take EMG artifacts into consideration. The second design, in Figure 6.4 (b), introduces two additional potatoes to include respectively frontal and temporal EMG artifacts. Finally, Figure 6.4 (c) shows the third design, in which, in addition to the second, electrode pops artifacts are taken into account, by adding each pair of electrodes to the field.

For all designs, the biggest potato is the extremely unspecific focused on general low frequencies artifacts, that contain all 27 channels ("FC2" and "CP1" are GND and REF, not considered). In all the other cases, potatoes contain a maximum of 6 channels. Also, the first field is composed of 3 potatoes, the second field of 5, and the third of 24. Therefore, the first field is expected to be the fastest and the third the slowest.

6.2.2 Hyperparameter optimization

Other than the configuration of the field, the algorithm relies on four other main hyperparameters: the distance threshold z to consider a chunk of signal for a particular potato artifact, the probability threshold p for the Fisher's test for the global chunk artifact, the length l and the overlapping o of the epoch of data to test. In deep, the classifier requires many other hyperparameters: among others the alpha parameter α that controls for the learning rate, the threshold parameters for the test probabilities, and other heuristics used in computations (e.g. OAS method). All these parameters have been left out in the first optimization iteration for simplicity purposes.

Two hyperparameters techniques have been used in the process: the grid search and the bayesian optimization [84] [85]. The former was guided by the ease of conception and implementation. This technique returned important information on the time and memory spent for each configuration. This technique has been abandoned at an early stage of the process for bayesian optimization, faster, and more automated, and it gives the possibility to search on a continuous space. The algorithm used is implemented by the scikit-optimize package [86]. This decision was made because the number of configurations to examine forced the grid search to run for a not feasible amount of time. Therefore, the balance was between more iterations of the training (i.e. more tuning of the field) and more accuracy for each iteration (i.e. better tuning of the hyperparameters).

The RPF has been used for binary classification (i.e. 0: ”artifact epoch”, 1: ”clean epoch”) problem. For each EEG recording, every artifact chunk on a ”clean” exercise, and every ”artifact exercise” are annotated (see Chapter 5. Data collection). In each iteration of the bayesian optimizer, when the best configuration (i.e. combination of the parameters z, p, l, o, and field f) has been picked, the file is split into epochs of l seconds (with $l \in \mathcal{R}$) and o% of overlapping (with $o \in \mathcal{R}$ and $o \in [0, 100)$). Only epochs that are completely included in one annotated segment are kept and labeled with respect to the annotation (i.e. 0 if artifact, 1 if clean, and 2 if of no interest, for example when the epoch is not entirely included in a segment). In the figure below, an example is shown, and successively commented on:

Figure 6.5: Visualization of epochs labeling strategy

In Figure 6.5 a visualization of the method to label each epoch is shown. The signal with the annotations is given (see Chapter 5. Data Collection). In the first exercise, of resting state, an unexpected artifact appears (”unexpected” because it was not supposed to be in that exercise, and so it is annotated). Between each exercise, a break period is given to the participant to explain the next exercise and to rest. Then, another exercise starts (in the case above, a blinking exercise). ”Resting state”, ”eyes open”, ”eyes close”, and ”mathematical exercise” are considered ”clean” segments of data, if no artifacts are annotated within. ”Blinking”, ”lateral eye movement”, ”lateral head movement”, ”jaw clenching”, ”chewing”, and ”shivering” are always considered ”artifact” segments of data. Breaks in between exercises are considered segments of ”no interest”. Finally, data is split into epochs of l seconds (in the figure above, $l = 3.5s$) with o% of overlapping (in the figure, $o = 30\%$). As ground truth, each epoch is labeled as the respective segment if completely included, ”not interest” otherwise.

Thus, in Figure 6.5, epochs 1-4 are ”clean” because completely included in a ”clean” segment. For the same reason, epochs 6, 13, and 14 are labeled as ”artifact”, and 10, and 11 as ”no interest”. Lastly, epochs 5, 7, 8, 9, and 12 are not completely included in only one segment, thus, following the most conservative choice, are labeled as ”no interest”. In this way, the ground truth for each recording is built, and performance metrics (scores) can be measured for the algorithm.

The score used for the optimization is the Fbeta score, with $\beta = 2$, thus F2. This is because it is important to not lose artifact chunks, whereas it is acceptable to misclassify clean epochs. The F2 score favors recall over precision, thus considering false positives less relevant than false negatives.

As for the grid search, the running time has been inspected for each configuration, but it has not been used in the score for the optimization, keeping the two metrics separated.

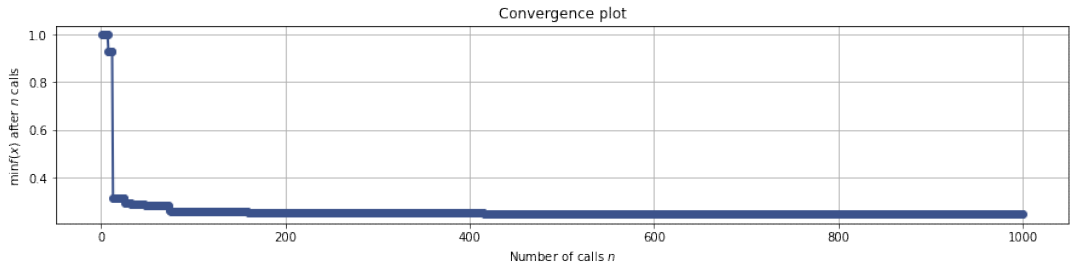
Different iterations have been performed for finding the best parameters and re-designing the potato field in a trial-and-error matter. Firstly, for each iteration, the convergence plot is analyzed for tuning the number of combinations to inspect in the following iteration. Secondly, a partial dependence plot (PDP) [87] of the hyperparameters (i.e. z, p, window duration l, overlapping o, and field f) is analyzed to understand the dependency of the score with respect to one hyperparameter and all couples of parameters, and to improve the ranges for the following iteration. Thirdly, a visualization of the data with ground truth and predicted labels is created to visually inspect the performances and try to find patterns of misclassification, to improve the field's designs. Finally, for the online purposes of the RPF, scatter plots of the running time with respect to each parameter of the RPF have been analyzed to better understand the relationship between hyperparameters and running time from an experimental point of view. Theoretically, for the RPF, "the cost of the algorithm is dominated by the computation of eigen decomposition required for i) the distance to the reference of each potato in the field, and ii) the update of each reference matrix. The complexity of an eigenvalue-eigenvector decomposition is cubic with the number of electrodes C and therefore is dominated by the largest potato of the RPF. This is a fast computation even for $C = 100$ on modern computers, however, in practice when C is large and large potatoes are used, dimensions can be reduced (by PCA), in order to keep the computational cost always very low" [49]. For this reason, in the design stage of the field, the size of the potatoes was kept small (i.e. below 8 [49]). The number of potatoes influences the time necessary because more potatoes mean more covariance matrices, more distance computations, and more predictions for each chunk of data. Initially, also memory usage was taken into account but, after some tests, it was clear that the space used by the algorithm was not relevant for any combinations, and removed.

The optimization covers different subjects' data for generalization purposes. Training the classifier with more subjects improves the robustness of the system. In particular, two subjects' data have been used. This number is not particularly representative, but the decision has been made for time constraints reasons.

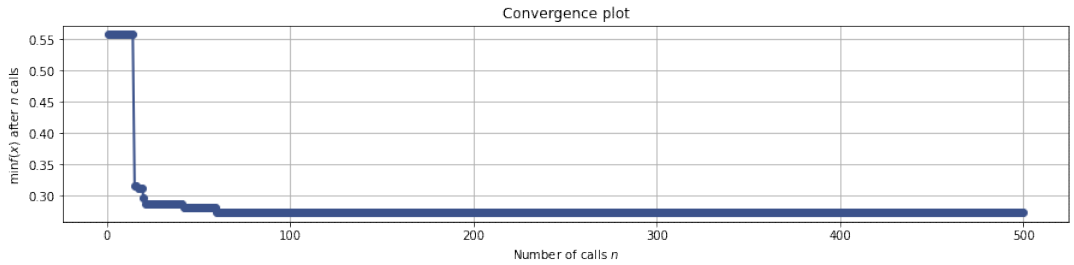
6.3 Results and conclusions

6.3.1 Convergence plots

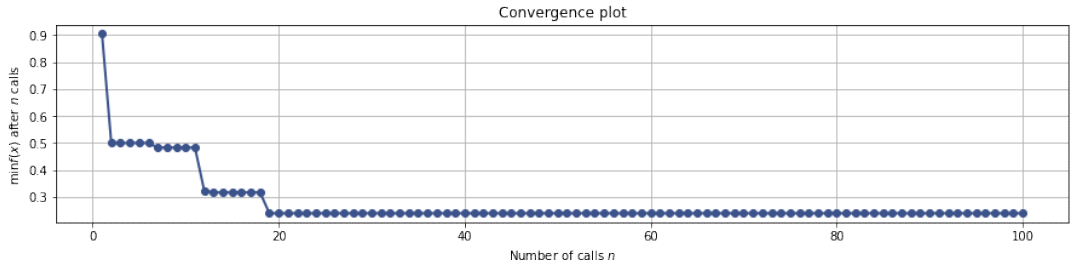
In the following, three convergence plots are shown respectively for the first, for a middle, and for the last iteration of parameters tuning. This is done to refine the number of combinations to analyze in the Bayesian optimization, in order to reduce the amount of time for each iteration. Note that, for constraints given by sklearn library, the optimizer can only minimize a target function [86]. Therefore, the function to minimize is $f(.) = 1 - F2(.)$, equivalent to maximize $f(.) = F2(.)$. For this reason, in the graphs below, the lowest the function value, the best the F2.



(a) Convergence plot for the first iteration of the parameters tuning



(b) Convergence plot for a middle iteration of the parameters tuning



(c) Convergence plot for the last iteration of the parameters tuning

Figure 6.6: Convergence plots for different stages of parameters tuning

In Figure 6.6 (a), it can be seen that 1000 calls have been performed, but after 415 the value of the function doesn't decrease. For this reason, the number of calls has been reduced in each iteration to 500 at an early stage, as shown in Figure 6.6 (b). Again, after some trials, it was clear that the performances didn't improve considerably even before 500 calls, whereas the training time is heavily affected by the number of combinations considered. For this reason, the number of calls has reduced to 100. That gave the opportunity to reduce the training time without affecting the final score, allowing for more tuning iterations.

6.3.2 Partial dependence plots (PDP)

In this section, the PDPs for the first and for last iterations are shown. ”The PDP shows the influence of each search-space dimension on the objective function. The diagonal shows the effect of a single dimension on the objective function, while the plots below the diagonal show the effect on the objective function when varying two dimensions. The Partial Dependence is calculated by averaging the objective value for a number of random samples in the search space while keeping one or two dimensions fixed at regular intervals. This averages out the effect of varying the other dimensions and shows the influence of one or two dimensions on the objective function. Also shown are small black dots for the points that were sampled during optimization. A red star indicates per default the best observed minimum” [88].

First iteration

Again, as for the convergence plot, the optimizer is minimizing $1 - F2$, so lower values mean better performances.

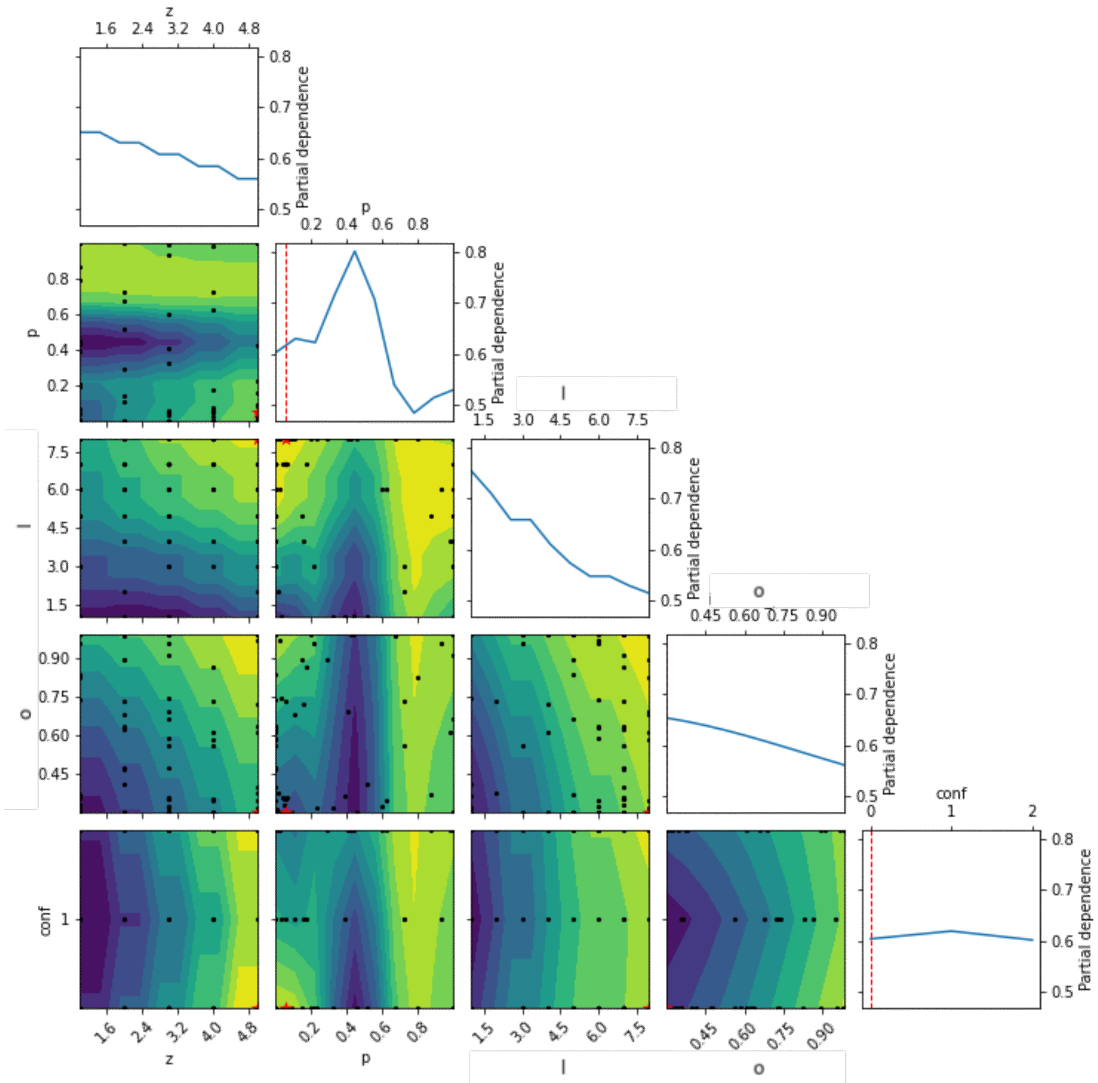


Figure 6.7: PDP for the first iteration of parameters tuning

The source space given to the Bayesian optimizer in the first iterations was:

$$
\begin{aligned}
z &\in [1, 5] \in \mathcal{N} \\
p &\in [1e-8, 1) \in \mathcal{R} \\
l &\in [1, 8] \in \mathcal{N} \\
o &\in [0.3, 1) \in \mathcal{R} \\
f &\in \{0, 1, 2\}
\end{aligned}
$$

In Figure 6.7 the partial dependences for the model parameters are shown. The source space for the first iteration has been inspired by the RPF paper and examples [49][89], for which a window size $l = 1s$, overlap $o = 50\%$, $p = 1e-4$, and $z = 2.5$ have been taken. For this reason, the ranges have been centered around those values.

Differently than expected, for each of the parameters but p, a value on the range extreme is chosen (shown by the red stars). This fact highlights the possibility that the ranges do not include the best values for the parameters, and that the performances would increase expanding the range in that direction, as it will show below.

For z, and l it can be seen that the graphs present steps. This is because the values belong to the $\mathcal{N}$ space so that only discrete values are evaluated. For this reason, it has to be careful with the extensions in the real values, for which no data points are available. f, on the other side, is a categorical parameter. In fact, it is a set containing different designs. Also in this case particular attention has to be taken to the extensions within the values. Finally, p and o belong to $\mathcal{R}$, and more uniform distributions of data points are available.

It is interesting to note that for p, the best model picked a sub-optimal value (indicated by the red line or star). This point is of great importance. From the couples of partial dependence graphs, it might seem that the variables are mutually independent. In fact, it can be seen that for each parameter, its single dimension dependence on the score (i.e. the diagonal plots) is almost exactly reported on the double dimensions dependence graphs, without any parameter influencing the behavior of the others in the final performances. If this was the case, the best model would have picked the value that individually minimizes $1 - F2(.)$. In contrast, the model that maximizes $F2(.)$ picked a sub-optimal solution for p alone, showing relationships among parameters of a higher degree than two.

The best model for the first iteration, as shown in Figure 6.9 in the next section, and highlighted in the PDP with red stars and red dashed lines, has parameters equal to $z = 5$, $p = 5.3e-2$, $l = 8$, $o = 30\%$, and first field. It means that the RPF, at this stage, prefers to have the longest and least overlapping epochs possible. Moreover, it tends to have the highest z threshold, so to accept covariance matrices distant from the baseline one, but to have a p similar to the example one. As the best z, l, and o are at the extremes of the ranges, those have been enlarged in the following iterations.

Last iteration

The PDP is reported, in Figure 6.8 below, for the best model in the last iteration of the tuning. As said before, the ranges for p, l, and z have been changed between iterations to always take more cases and to have a better distribution of the source space. In addition, z and l ranges have been converted to continuous intervals. The source space, in the last iteration, was:

$$
\begin{aligned}
z &\in [0, 100] \in \mathcal{R} \\
p &\in [0, 0.1] \in \mathcal{R} \\
l &\in [0, 60] \in \mathcal{R} \\
o &\in (0, 1) \in \mathcal{R} \\
f &\in \{0, 1, 2\}
\end{aligned}
$$

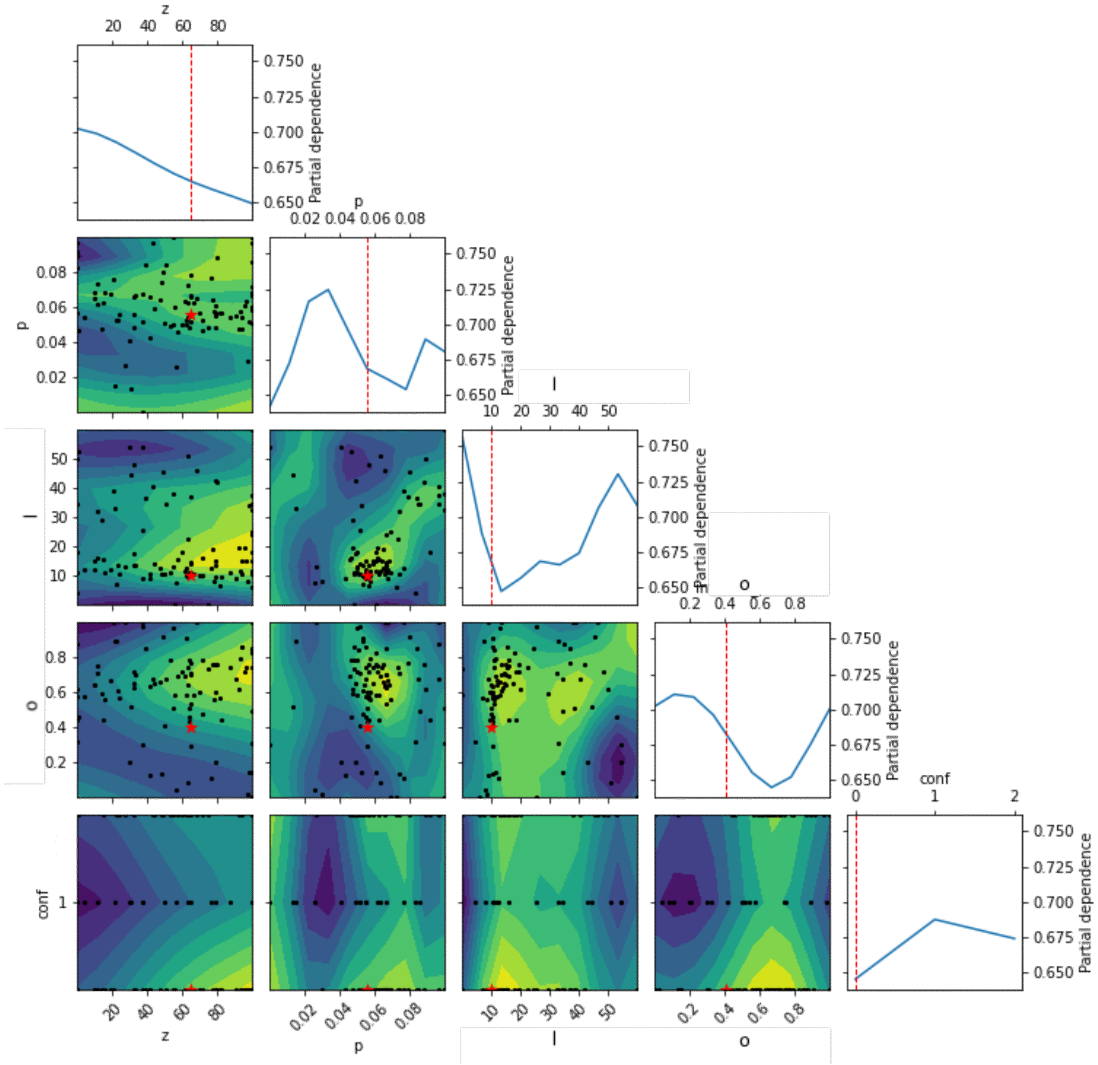


Figure 6.8: PDP for the last iteration of parameters tuning

In Figure 6.8 above, it can be seen that the graphs are more interdependent than before. In fact, for all parameters except f, sub-optimal solutions have been picked, highlighting a mutual dependence between the variables in the final F2 score. One possible explanation is that the ranges in which the variables were before didn't allow the parameters to influence each other.

One property to note is that for each continuous range (i.e. all but f parameter), the value that gives the best F2 score is in the between of the range, and not on the extremes. Even though this could be because of a local minimum in the range, it is a better situation than before. It is likely that the ranges are big enough to contain the global minimum of the score. This is also supported by considering that the best model, by the final score, doesn't have any global minimum for each standalone parameter (except for f). For example, looking at the z independent graph, the global minimum would be at 100, but the best model has $z \approx 65$. Moreover, for all the parameters but f, neither local minimum is chosen. On the other side, it does select the best f and close to the best l. It is also true that the difference between the best and the worst F2 is between 0.05 and 0.1 for each parameter. This puts the difference between the best and the worst model in a different perspective. In other words, the best model can choose sub-optimal single parameters because they depend also on other parameters, but also because the difference between the optimal and the sub-optimal choice is not too relevant. This could also be noted in the convergence plot in Figure 6.6 (c), for which many different combinations return a similar score. More in deep, three combinations give an F2 score closer than 2% with respect to the best model. These models picked combinations as follows:

- $F2 = 0.760$: $z = 65.04$, $p = 5.58 \cdot 10^{-2}$, $l = 10$, $o = 40.48$%, $f : 0$
- $F2 = 0.756$: $z = 77.70$, $p = 5.74 \cdot 10^{-2}$, $l = 8.10$, $o = 55.82$%, $f : 0$
- $F2 = 0.751$: $z = 32.85$, $p = 6.65 \cdot 10^{-2}$, $l = 13.53$, $o = 70.19$%, $f : 0$

As it can be noted, different combinations result in an almost-optimal score. An inverse relationship can be seen between z and l: the longer l, the smaller z. This can be explained by the fact that l controls the dimension of the signal epoch, or, in other words, how much information is contained in the respective covariance matrix. The data used tends to favor longer epochs because the artifacts are continuous and bounded in chunks. Therefore, for longer epochs, more artifact signal is contained and the covariance matrix results in a bigger distance to the baseline one. In addition, for clean segments, longer epochs average out small fluctuations of the signal, resulting in more stable covariance matrices, making them closer to the baseline. Hence, a smaller z of threshold is needed to distinguish clean and artifact epochs.

On the other hand, p is similar in each combination, and with the same order of magnitude as the one used in the pyRiemann library examples ($p = 0.01$ [89]).

It is also interesting to note that all the best models picked the first field design. Also, this fact results from the choice of the dataset. In fact, from the data, fewer artifacts are considered than the one initially prospected. For example, even though "electrode pop" is present on the dataset, it has been left out in the MVP (i.e. labeled as "not interest"), therefore there are no differences between the second and the third field. This is because, as shown before in Figure 6.4, the third field is the same as the second but with specific potatoes for the electrode pops and, as these artifacts are not present in the training set, the relative potatoes cannot contribute to the prediction, and on the opposite they dilute the probabilities of the other in the Fisher's method, worsening the specificity.

6.3.3 Ground truth against predicted labels

For a better understanding of the fields and the different parameters, thus to dig into the RPF black box, the graphs of the ground truth and the predicted labels by the RPF algorithm for each person are shown below. In particular, the graphs represent the different epochs as a sequence of dots (one dot for each epoch). Blue dots represent the ground truth labels, obtained as explained above in Figure 6.5, and yellow dots visualize the respective RPF predictions (note that ”no interest” epochs are not visualized). The graphs are split into different exercises.

First iteration

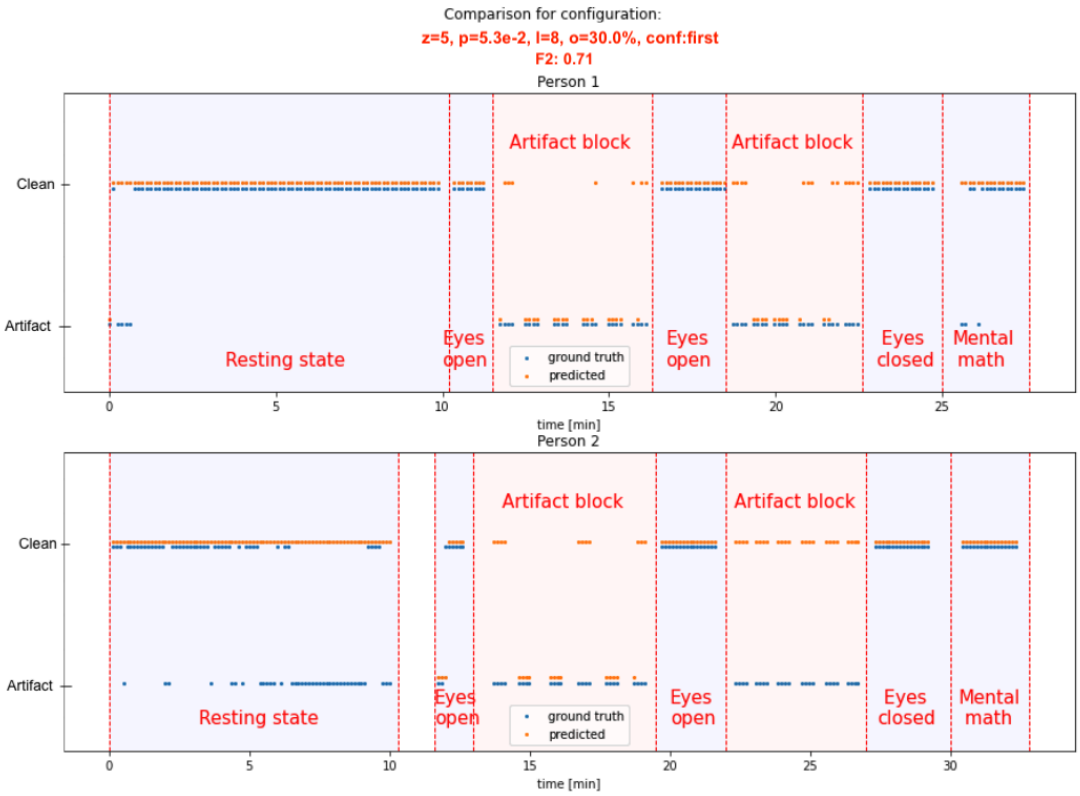
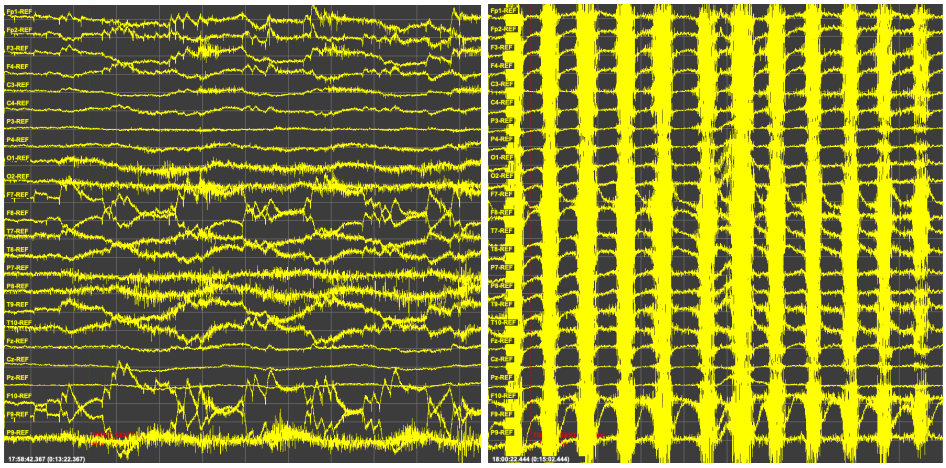


Figure 6.9: Comparison between predicted labels and ground truth for the best model by F2 score for the first iteration of the parameters tuning

From Figure 6.9 it can be seen that many epochs are misclassified both for the ”clean” and ”artifact” labels. In particular, for the first participant, ”clean” epochs are well predicted and some mistakes appear on the ”artifact” classification. In the second case, instead, both ”clean” and ”artifact” segments are problematic.

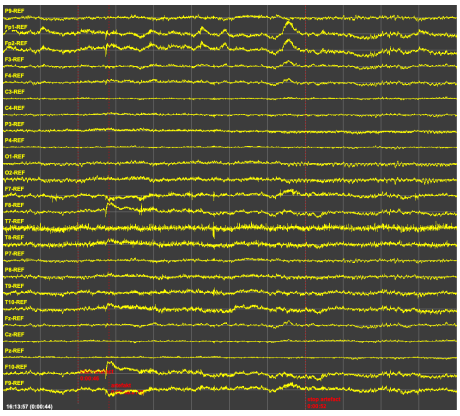
Whether the behavior of the first person is expected, the same cannot be stated for the second person. In fact, the F2 score tends to give more importance to artifact detection, thus it would be expected to have more false positives (i.e. clean epochs classified as artifacts) than false negatives (i.e. artifact epochs classified as clean).

To better understand this behavior, the differences in the artifacts between the first and the second person are visualized in the figures below:

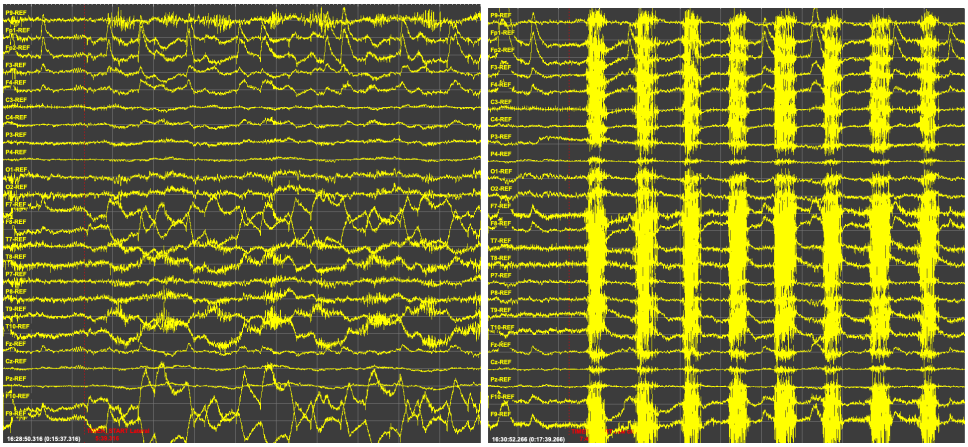


(a) Viz of head movement artifacts (b) Viz of jaw clenching

Figure 6.10: Visualization of two classes of artifacts - head movement in (a), and jaw clenching in (b) - for the first participant. x-division is $1s$ and y-division is 100



(a) Viz of artifacts in the resting state



(b) Viz of head movement artifacts (c) Viz of jaw clenching

Figure 6.11: Visualization of three classes of artifacts - eyes tension in the resting state in (a), head movement in (b), and jaw clenching in (c) - for the second participant. x-division is $1s$ and y-division is 100

The importance of the figures above is in the mutual differences. First of all, it can be seen, in Figure 6.11 (a) against (b) and (c), that the magnitude of the manually annotated artifacts within the resting state exercise is visually lower than the one for the artifacts exercises. Thus, it is easier to detect those artifacts than not the ones in the resting state, also with high values of z. This can explain why, for the second participant, none of the artifact epochs in the resting state exercise have been detected.

Secondly, it can be seen that the artifacts created by ”person 1”, in Figure 6.10 (a) and (b) have a greater magnitude of those created by ”person 2”, in Figure 6.11 (b) and (c), making their detection easier for the RPF classifier. This explains why for ”Person 1”, more artifact epochs are correctly classified with respect to ”Person 2”.

Finally, the F2 score is visualized, comparing it with a few baseline scores. In particular, three baselines are computed: a ”random” baseline, in which, for each epoch, a random (50/50) label is assigned, an ”artifact” baseline, in which all epochs are labeled as artifact, and a ”proportional” baseline, in which to each epoch, a random label is assigned with uniform distribution (”clean label”/”total epochs”). In this case, ”clean label”/”total epochs” = 0.61.

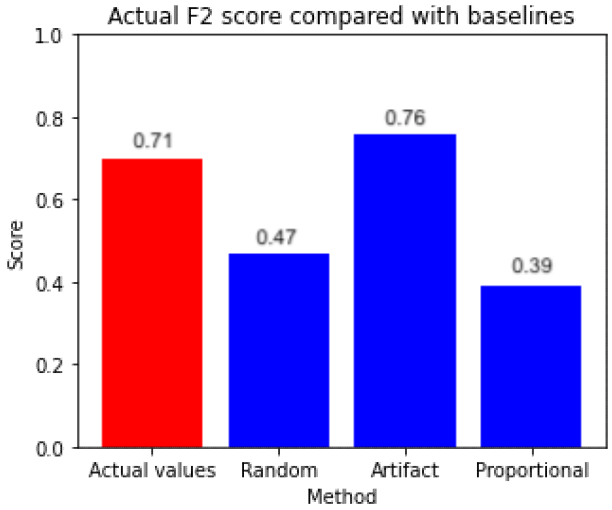


Figure 6.12: Real data F2 compared with different baselines for the best model for the first iteration of the parameters tuning

It can be seen in Figure 6.12 that the F2 score for the best model in the first iteration is 0.71. This value is better than the ones resulting from a random guess, in which F2 score, in this case, is 0.47, or from a proportional guess, with an F2 of 0.39. On the other hand, the performance is worse than labeling all epochs as artifact. If it would not make sense to consider all epochs not clean, the fact that the F2 score is higher than the model prediction highlights that the performances of the model are poor.

It is interesting to note the effect of the weight in the F2 measure. The proportional baseline score is lower than the random baseline because F2 weights recall (true positive/ground positive) are higher than precision (true positive/predicted positive). So it is better to spot all the artifact epochs (i.e. obtaining high recall) than retrieving only the correct ones 9i.e. obtaining high precision). In the random baselines, guessing ”clean” 50% of the time results in a better recall than not guessing it for $100 - 61 = 39\%$ of the time.

Last iteration

It is interesting to see the difference in the classification task between the best RPF model in the first and the last iteration of the hyperparameters tuning. For this reason, the same graph as before has been reported for the last iteration.

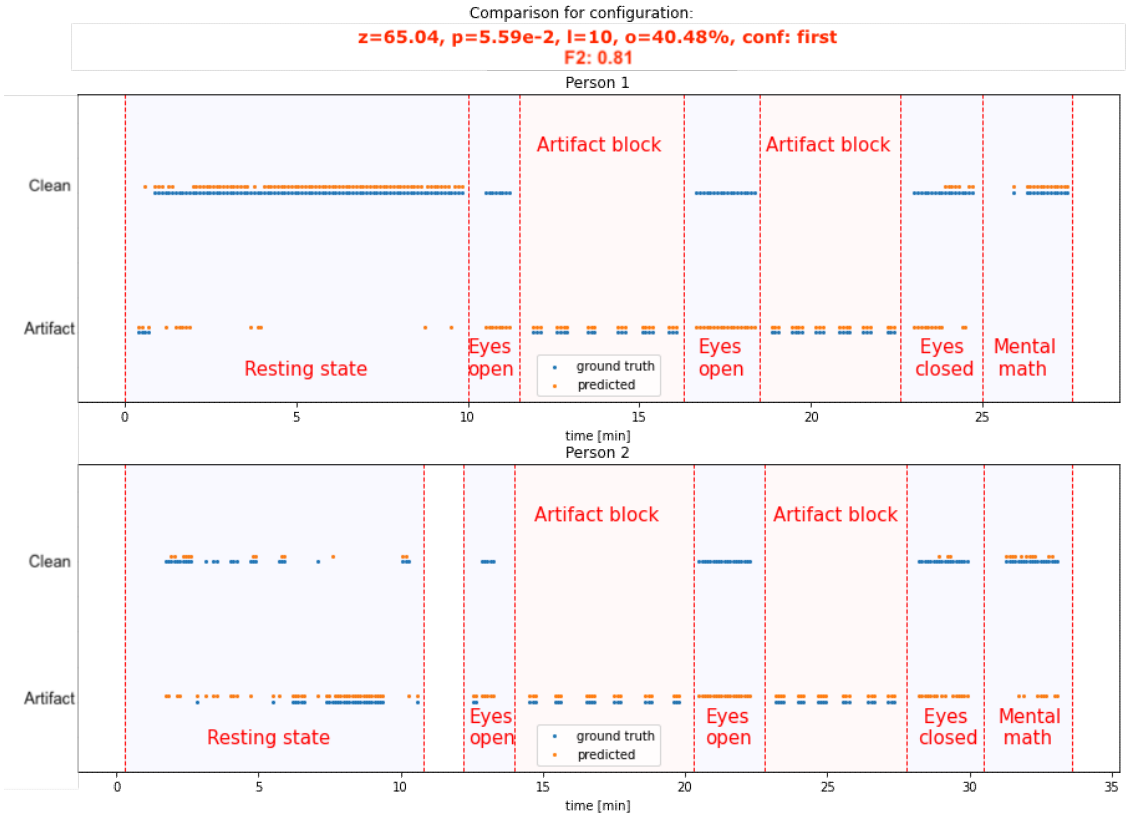


Figure 6.13: Comparison between predicted labels and ground truth for the best model by F2 score for the last iteration of the parameters tuning

Differently from the first iteration results, it can be seen, in Figure 6.13, that all epochs for the artifacts have been classified correctly. On the other side, many of the ”clean” epochs have been misclassified. Thus, the RPF is too sensitive to small variations of the signal.

In any case, the behavior this time is clearer than before. In fact, the classifier gives more importance to artifact detection (i.e. recall), to the detriment of precision, and, as expected, all artifact epochs are retrieved.

Finally, as before, the F2 score is compared with the three baselines and shown in Figure 6.14. This time, the proportion of ”clean epochs”/”total epochs” is 0.62.

It can be seen that this time, the F2 score for the trained model overtakes the baselines. This means that the model can actually retrieve more information from the data than any guessing. It can also be noted that the baselines are very similar to the case before, as expected. The differences with the previous case lie in the proportions between clear and total epochs. In fact, a slightly higher proportion results in a slightly lower F2 score, because the unbalance assumes higher importance. For example, in the ”proportional” baseline, more epochs are labeled as clean, on average, reducing the recall. On the ”artifact” baseline, having more clean epochs result in a lower precision.

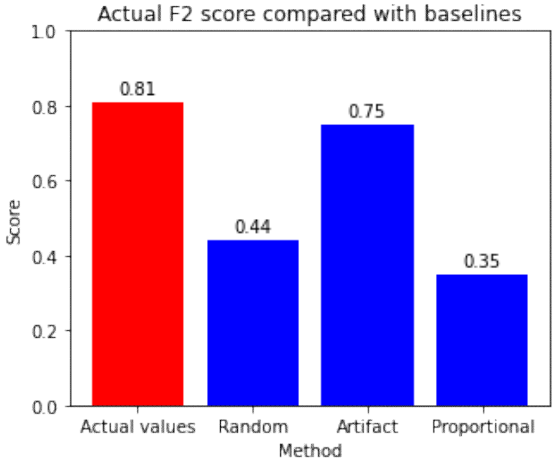


Figure 6.14: Real data F2 compared with different baselines for the best model for the last iteration of the parameters tuning

6.3.4 Running time analysis

In this last section, the results of the time analysis are shown. The relation between the running time (later rt) with each parameter is studied. Two experiments have been performed. The first by running the RPF algorithm over the following grids of parameters:

$$
\begin{aligned}
z &\in \{20, 40, 80\} \\
p &\in \{10^{-8}, 10^{-4}, 10^{-2}\} \\
l &\in \{1, 5, 10, 15, 20, 30\} \\
o &\in \{0.1, 0.25, 0.5, 0.75, 0.9\} \\
f &\in \{0, 1, 2\}
\end{aligned}
$$

The second, keeping all parameters constant to $z = 20$, $p = 10^{-2}$, $l = 10$, $o = 0.25$ and creating fields with $pt \in \mathcal{N}$ amount of potatoes and $ch \in \mathcal{N}$ amount of channels for each potato. With f_{pt_ch} the configuration in which the field has pt potatoes, each with ch channels, the RPF algorithm has been run over the grid of parameters:

$$
\begin{aligned}
z &= 20 \\
p &= 10^{-2} \\
l &= 10 \\
o &= 0.25 \\
f &\in \{f_{pt\_ch}, pt \in \{1, 2, 4, 8, 16, 32, 64\}, ch \in \{2, 4, 8, 16, 27\}\}
\end{aligned}
$$

Analysis of rt vs configuration

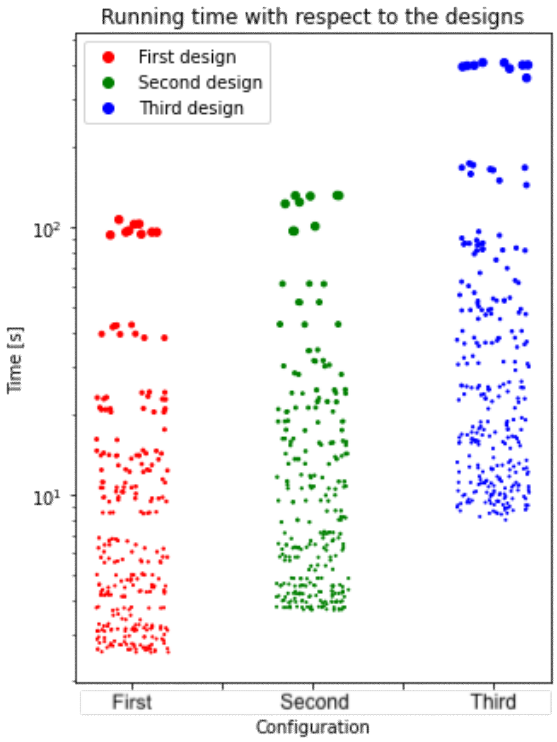


Figure 6.15: Scatter plots of the running time of the RPF with respect to f. Each call is represented by a point. The dimension of the marker is proportional to the number of epochs evaluated in that call.

In Figure 6.15 the rt of the RPF algorithm is scattered with respect to the three field designs. Each dot represents a call of the algorithm, and the size of the marker is proportional to the number of epochs (n) evaluated in the call. The exact n in each call is returned as a variable by the algorithm itself. Fast estimation of the number of epochs is given by $n \leq T/(l * o)$, where T is the total useful time in the recording, i.e. the total length of the segments not labeled as ”no interest”. This estimation is in general greater than the actual number of epochs because of the border effect between segments. When an epoch contains parts of two different segments are labeled as ”no interest” and not analyzed.

It can be seen that the rt is dependent both on f and on n. Keeping in mind that the y-axis is in logarithmic scale, it is easy to see that, for the third configuration, the average rt is higher than for the second, which is higher than for the first. Additionally, the calls with higher n (i.e. markers of bigger sizes) are associated also with higher running times for each configuration.

This was expected from the theory treated in the introduction of this chapter and in the quote reported in section 6.2.2 Hyperparameter optimization, for which the cost of the algorithm is dominated by the eigenvalue decomposition required for the distance to the reference of each potato in the field and the update of each reference matrix. In this case, the update of the reference matrix plays a minor role in the comparison, as it is plausible to assume that the number of clean epochs detected by each call is essentially the same overall, as also proven by the experimental data. In fact, on average, the markers of the same size and color are always in the same region, thus not much variability in time is given by the update itself, but more from the number of epochs to analyze.

The third configuration has 24 potatoes against the 5 of the second and the 3 of the first. Therefore, with the same number of epochs processed, the third design is expected to be more expensive than the second, in turn, slower than the first. Also, it is expected that the higher n, the higher the overall time, to analyze all the epochs. The figure above is given more as a qualitative starting point, a more in deep analysis with quantitative examples is going to be shown in a later stage.

Analysis of rt vs z and p

The plots of the two parameters to which the rt is independent are reported below in Figure 6.16. As before, the size of the markers is proportional to n, and the colors correspond to the design. z and p are considered categorical classes, thus the scale is not respected. This is not a problem because, as shown afterward, no information can be retrieved by the x-axis. In addition, horizontal jitter is added to better visualize the otherwise overlapping data points.

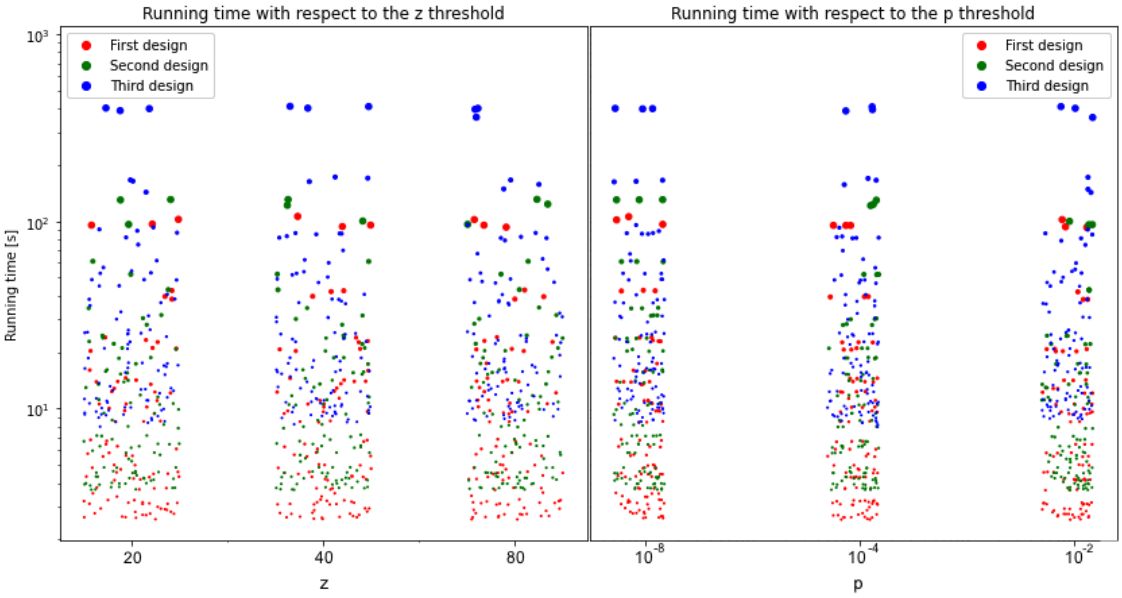


Figure 6.16: Scatter plots of the running time of the RPF with respect to z (on the left) and p (on the right). Each call is represented by a point. The dimension of the marker is proportional to the number of epochs evaluated in that call. Note that the values on the x-axis are not in scale.

In Figure 6.16 the plots of the rt with respect to z and p are visualized. Again, we have the information about the configuration and n, as blue bigger points are higher than blue smaller points, and blue points are higher than the green and red points with the same size.

New knowledge is brought by the fact that the distributions of rt are the same for all the values of z and p. The conclusion is that the running time, as expected, is independent of z and p.

This fact is less expected. In fact, from the theory, we know that the time to run the algorithm should be dependent also on the number of updates of the baseline matrix, which is dependent on the z and p values. In fact, for example, if $p = 1$, all epochs are considered clean and each should make the RPF baseline update, slowing the computation. On the other side, $p = 0$ should consider all epochs as artifacts and never update the matrix of baseline.

As can be seen by the experiments, though, none of these influence significantly the computation. No in deep analysis on this aspect are present in the paper from Barthelemy et al. [49].

Analysis of rt vs n

Finally, rt with respect to n is reported below. Firstly, a linear plot is shown to highlight the linear dependence between n and rt, then a log-log plot is shown to better visualize the data. Also in this case, horizontal jitter is added to clearer visualizations.

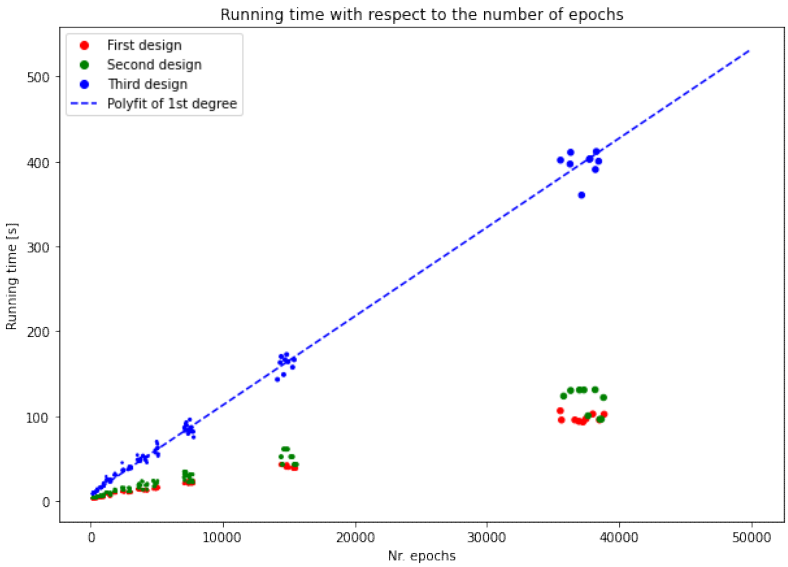


Figure 6.17: Scatter plots of the running time of the RPF with respect to n.

As expected, the rt depends heavily on the number of epochs n. In fact, when n increases, the average rt increases linearly, as shown in Figure 6.17. This visualization clusters the small values, so a log-log plot is reported below:

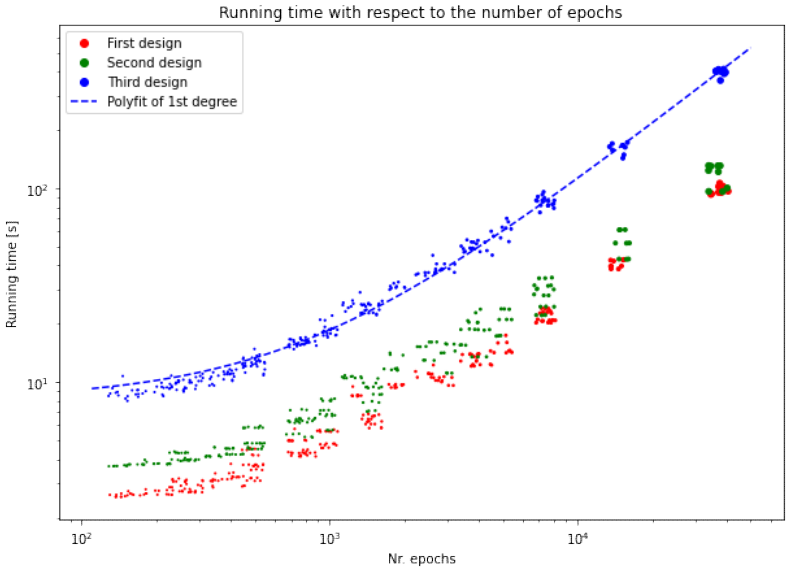


Figure 6.18: Scatter plots of the running time of the RPF with respect to n.

In the figure above, the small values of n are better visualized. For example, it is possible to note that for values of $n < 1000$, the rt is lower than the one expected from the fitting.

This means that there are some additional processes that slow down the computation when the number of epochs increases. Most probably this is due to the internal costs of the Python language.

Analysis of rt vs ch

In the following graph, the running times with respect to the number of channels for each potato are shown. In particular, both the total running time for the entire algorithm and the time for the covariance matrix computation is reported. Each data point is shown with a marker, for which its dimension is proportional to the number of potatoes in that field. In this way, it's more intuitive to compare fields with the same number of potatoes, but a different number of channels for each of them. A fit is run over the designs with the maximum number of potatoes, to show the relationship averaging out the contribution given by a different pt.

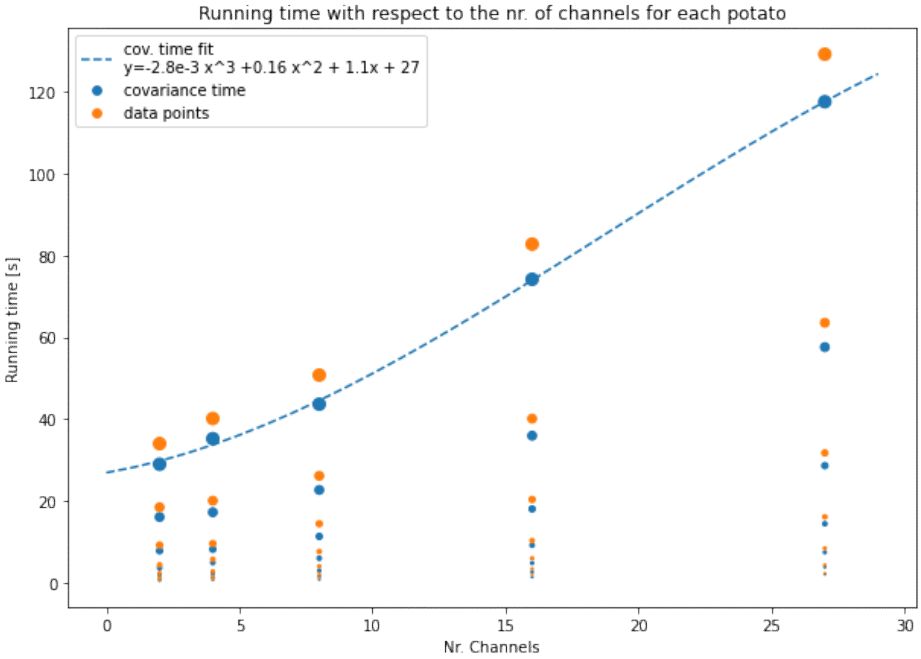


Figure 6.19: Scatter plots of the running time of the RPF with respect to ch. Each point represents a combination of pt and ch. The size of the marker is proportional to pt

As expected, the relationship is over than linear. More data points would be necessary for achieving a better estimation, but it is clear that the relationship is higher than the first degree. It is also rational that, with a parity of channels for each potato, a higher amount of potatoes result in a slower computation, as the algorithm needs to be run more times.

In addition, again predicted by the theory, the covariance time dominates the total running time of the algorithm. In fact, the computation of the covariance matrices is the heaviest task of the RPF. The proportions of computation time over total time, when the design has 64 potatoes, are $[85.18, 86.69, 87.99, 89.60, 91.01]$ respectively for $[2, 4, 8, 16, 32]$ channels for each potato, reflecting the predominance of the covariance matrices computation time. It is also interesting to note that the proportion increases with the number of channels. The computation of the matrices weighs increasingly more on the total time because that overtakes the corollary fixed setup costs of the practical algorithm from Python.

Analysis of rt vs pt

In the figure below, the running times with respect to the number of channels for each potato are shown. As before, both the total running time for the entire algorithm and the time for the covariance matrix computation are reported. Again as in the previous case, each data point is shown with a marker, but this time its dimension is proportional to the number of channels in that field. In this way, it's more intuitive to compare fields with the same number of channels for each potato, but a different number of potatoes. A fit is run over the designs with the maximum number of channels, to show the relationship averaging out the contribution given by different ch.

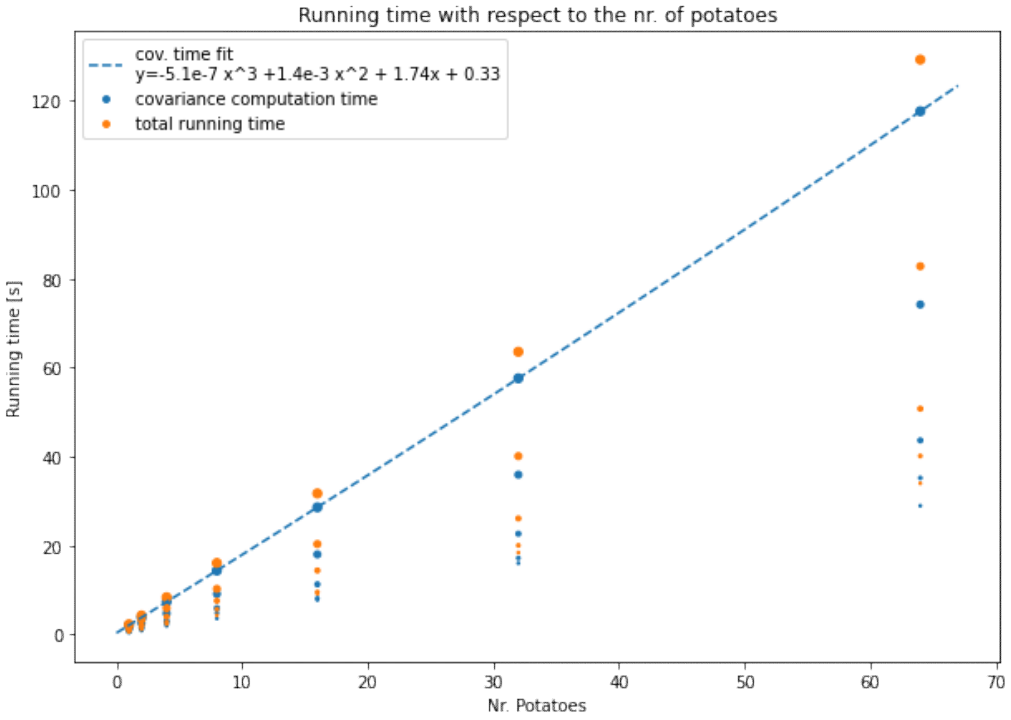


Figure 6.20: Scatter plots of the running time of the RPF with respect to pt. Each point represents a combination of pt and ch. The size of the marker is proportional to ch.

From Figure 6.20 above, it is clear that there is a linear relationship between the number of potatoes pt for a design and the running time of the algorithm when only designs with the same number of channels per potato are taken.

This was expected from the theory. In fact, the covariance computation time is proportional to the cube of the number of channels for the particular potato, but then the singular potato times are only added together. In this experimental setup in which all potatoes have the same number of channels, this results in a multiplication (i.e. summing the same time for a pt amount of time) hence in a linear relationship.

As for the previous graph, it can be seen that the covariance time dominates the total time. About 90% of the time is spent on the covariance matrix computation.

6.4 Discussion and future work

All the time analysis results with respect to the different parameters and designs options of the fields (i.e. amount of channels and number of potatoes) were expected from the theory and are comparable with the ones obtained in the main referenced papers [49][77].

On the other hand, if some parameters obtained are similar to the ones used in [49], for example, the p value (i.e. $p = 0.0058$ vs $p = 0.001$) and the o value (i.e. $o = 0.4$ vs $o = 0.5$), the other parameters (i.e. l and p) are very different. It is important to remember, though, the different methods used. In the paper from Barthelemy et al., z has been chosen from a statistical point of view, and l has been chosen arbitrarily from common sense. In this case, the choice of the parameters has been done from a hyperparameter tuning perspective, hence the parameters giving the best results have been selected. This difference has also been highlighted in the results and conclusions section, in which the inverse proportion between z and l is shown. In fact, usually, also the window size is taken into account in the z-score computation, and this relation is highlighted. In this case, instead, the computation of the z-score is independent of the sample size, and therefore, using a different window size l, also z needs to adapt.

It's not possible to compare the accuracy obtained in this project with other works, and this is because the reference paper focused on the comparisons among different calibrations and setups of the RPF, and only comparative performances were given. In addition, the design is specific for each headset, so no benchmarks were available for BrainCapture's data-specific case. Finally, most of the literature in the automatic EEG artifact detection uses the F2 score instead, to be independent of the usual dataset unbalance. In this case, instead, as the goal was driven by a real application and users' needs, the F2 score has been preferred to privilege false positives and to be sure not to miss any artifact epoch. In this context, an example of the different scores that would have been obtained for the best model by F2 score in the last iteration of hyperparameters tuning is shown in Figure 6.21. In particular, AUC, F1, and F2 scores are shown.

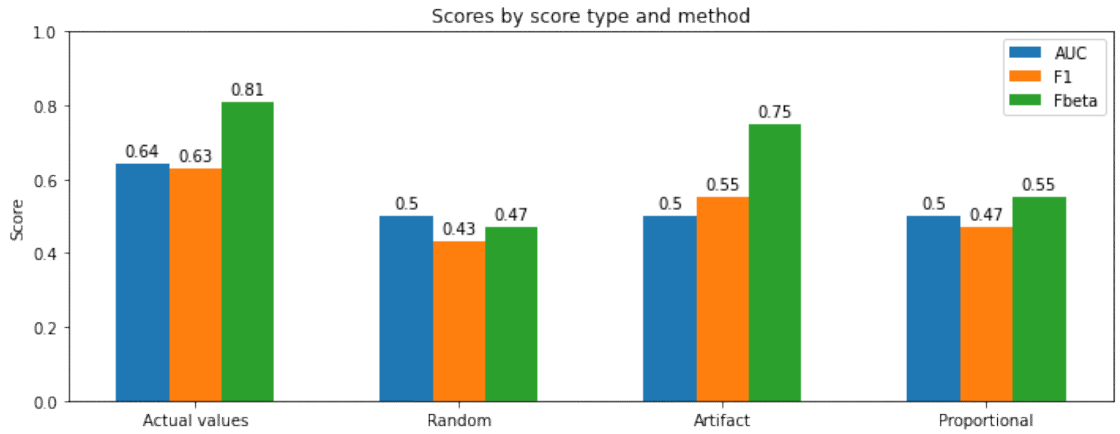


Figure 6.21: Comparison of different measurement scores (F1, F2, and ROC AUC). For each score, the prediction of the best model by F2 (called "Actual values") is compared with three baselines: random guess, each epoch labeled artifact, and a guess proportional to the "clean"-"artifact" epochs ratio in the recordings.

Obviously, the F2 score is the highest for the actual model, which has been optimized with that goal. It is interesting to note how it would have changed with the other methods. In fact, for the F1 score (that weights recall and precision equally), the score is lower because, in the final best model, many clean epochs are predicted as artifacts to not miss any artifact epochs. This decreases the precision, which has a more significant impact on the F1 score. For AUC the explanation is similar, as it is independent of the balance between the classes, and it is a measure of separability. Also, in this case, the high ratio of false positives decreases the final score. Exactly the same considerations can be made for the baselines. In particular, the ”artifact” baseline is higher for the F2 score because it weights the recall more, and the recall is of course maximum if all epochs are considered artifacts. For any measurement, the actual model performs better than the respective baselines.

The best outcomes from the referenced paper have been used for this project. Thus, the RPF instead of the RP and semi-online calibration have been used [49]. Exactly as in the paper, many other improvements can be implemented. For example, the fixed point Riemannian mean instead of the gradient, which should speed up the computations keeping the same accuracy, for a high amount of matrices. Also, all equations (6.3), (6.4), (6.5), (6.7), and (6.8) can be replaced with their geometric versions [49], more stable towards outliers.

It is important to remember that the design of the field is bound to the specific headset. In other words, a new set of spatial filters must be designed for each new cap. This fact is not problematic at this stage, and knowing the vision of the company neither in the next future. But it could become a problem if, for example, many caps and configurations are going to be used for different purposes by BrainCapture. In general, this is a problem for generability and transfer learning. In the following iterations, from electrode-based spatial filters, region-based spatial filters can be designed, with a translation map based on scalp locations of the electrodes.

Also, the transfer learning possibilities have been explored only briefly, and they were based on the generalization error given by training the algorithm in different subjects. In the future, it is fundamental to inspect and explore this aspect more, to understand to which extent it is possible to transfer the knowledge (i.e. hyperparameters and baseline matrix) to new subjects. In this use case, the transfer learning is helped by the calibration section, thus the semi-dynamic manner. In addition, in the standard EEG scan, the first exercises consist of the creation of artifacts to give the doctor a reference. This section can be used as ground truth in exactly the same way the training had been performed in this project.

An aspect that hasn’t been exploited but is of incredible potentiality for future iterations is the computation of the p-value. In fact, Fisher’s method combines all the potatoes giving them the same weight. A study of how different weights influence the importance attributed to each potato could be assessed, prioritizing certain artifacts over others.

In addition, the role of the calibration can be inspected in more detail. In fact, calibration is a fundamental aspect for the final performance. In the reference paper [49] a robust calibration is proposed, following a procedure standardly used. Hence, collecting the covariance matrices in a set and applying the first calibration, then using this first baseline matrix to reject artifact epochs from the initial set, and repeating the calibration with only the matrices remaining. This two-layers calibration can be easily implemented in the future.

The learning rate has not been taken into the analysis but it can be revealed of extreme importance. In fact, the alpha parameter has been set constant to 0.01. This means that, for each clean epoch, the baseline matrix shifts of 1% toward the new matrix, as well as the mean and the standard variation of the z statistic. This parameter can potentially change drastically the behavior of the classifier, updating the baseline to the drifts of the system. In fact, bigger values of α induce the system to be flexible and to follow the natural systematic changes of the signal as baseline drifts due to physiological changes (e.g. sweating, electrode movements,...), or psychological differences due to fatigue (e.g. inducing more eyes movements), attention state (e.g. fewer alpha waves), and relaxation (e.g. initial stress due to the new environment and tests). On the other hand, smaller values of α keep the system more stable and prevent it to adapt to repetitive artifacts in the signal. For all these reasons, this parameter has to be deeply analyzed in the future.

Finally, it has to be remembered that the RPF algorithm has been trained and generalized only on two recordings, thus the results can be improved by annotating and inserting in the pipeline more files.

7 RPF implementation for embedded systems

7.1 Methods

In this section, the methods used for implementing and testing the code in Python and Java for the RPF algorithm, and in Kotlin for the integration of the algorithm in the Android mobile app are described.

7.1.1 Python programming - usage of the pyRiemann library

Python examples of the Potato field were referenced in the paper by Quentin Barthèlemy et al [49]. In particular, the pyRiemann [90] python library was used. The library pyRiemann implements several distance measurements, clustering, and linear algebra methods based on the Riemannian geometry, used for building up the functions necessary for the algorithm. The examples given have been modified to comply with the necessities for the field design and the tuning of the parameters.
The code developed can be found published on GitHub [91]. In particular, with respect to the example found in the API gallery examples, [89], only the Riemannian potato field (RPF) method was kept, leaving out the Riemannian potato (RP) method. This is because, in the first iteration of the project, only the RPF was of interest. In addition, the visualization of the real-time simulation has been removed, and the F2 measurement test has been implemented instead. Moreover, all the implementations for the automatic tuning of the parameters - i.e. testing all the different configurations - have been written from scratch. All the pipeline has been automatized.

7.1.2 Java programming - translation and refactoring of the python code and libraries

As the main objective of the thesis was to test the feasibility of the quality feature, it was necessary to re-implement the Python libraries and the RPF algorithm into Java language, to make it runnable on an embedded platform - in particular, the Android system. As a second step, the algorithm will need to cooperate with the Kotlin language. For this reason, a considerable part of the project has been focused on the translation of Python code.

After a thorough examination of the program and all the libraries in Python, a list of functionalities necessary for the logic of the algorithm was drawn up. The basic operations that the program must complete are, first of all, the computation of covariance matrices, normalized covariance matrices (useful for neglecting signal power), and regularised covariance matrices (for stability reasons).

Secondly, basic functions of linear algebra and matrix calculus such as multiplication, summation, power elevation, inversion, exponential, logarithm, and extraction of eigenvalues of matrices are needed to manipulate the fundamental objects in elaboration. In addition, it is necessary to have signal processing functionalities, in particular, to be able to filter the signal.

Furthermore, the logical building blocks on which the entire theory is based are the operations of Riemann geometry on the spaces of symmetric positive definite (SPD) matrices - the covariance matrices - such as the Riemann distance, the Riemann mean, the Euclidean mean, and geodesics.

Finally, some statistical tools are needed, such as the cumulative density function for gaussian and chi-square distributions.

Most of the operations above have been translated into static functions for usability and ease of implementation, and to align with the Kotlin language, which prefers standalone functions. In any case, some classes have been introduced in the code to give structure to the program, such as the class ”Potato” and ”PotatoField” which contain definitions of the structural elements of the algorithm. In particular, an object of the ”PotatoField” class maintains a list of ”Potato” objects, each of them stores the properties of the potato - cut off frequencies and electrodes of interests (i.e. frequency and spatial filtering) - and supporting functions as ”fit”, ”partial_fit” and ”predict_probability” for the classification of the different signal’s portions.

In Figure 7.1 below, the container classes and their functions are visualized.

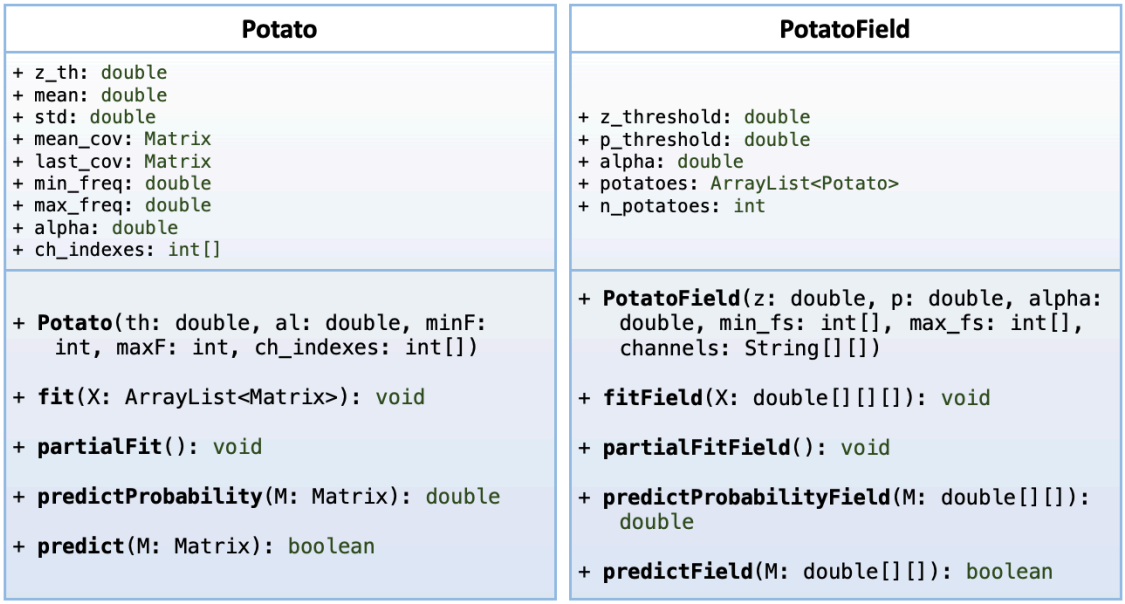


Figure 7.1: Visualization of the container classes Potato and Potato Field. ”+”: public attribute or method.

It can be seen, from the figure, that the classes implement the main functionalities of the RP and RPF algorithms. The PotatoField class relies on the Potato one. In the mobile application, only methods of the PotatoField class are going to be called. In particular, at the initialization of the app, an instance of the RPF is created with the constructor. After collecting the epochs for the calibration, the ”fitField” method is called to initialize the field. Then the ”predictField” method is called for each new epoch, and the ”partialFitField” updates the potatoes if the epoch is clean. Internally, the PotatoField instance calls the respective Potato methods for each potato of the field, hence implementing the algorithms shown in the previous chapter.

In the following Figure 7.2, the auxiliary classes for the methods above are shown. All these methods are called by the container classes to filter the incoming signal, compute the different versions of covariance matrices, apply the Riemannian operations such as mean and distance, compute the z-score of an epoch, and combine all of them together to obtain the p-value.

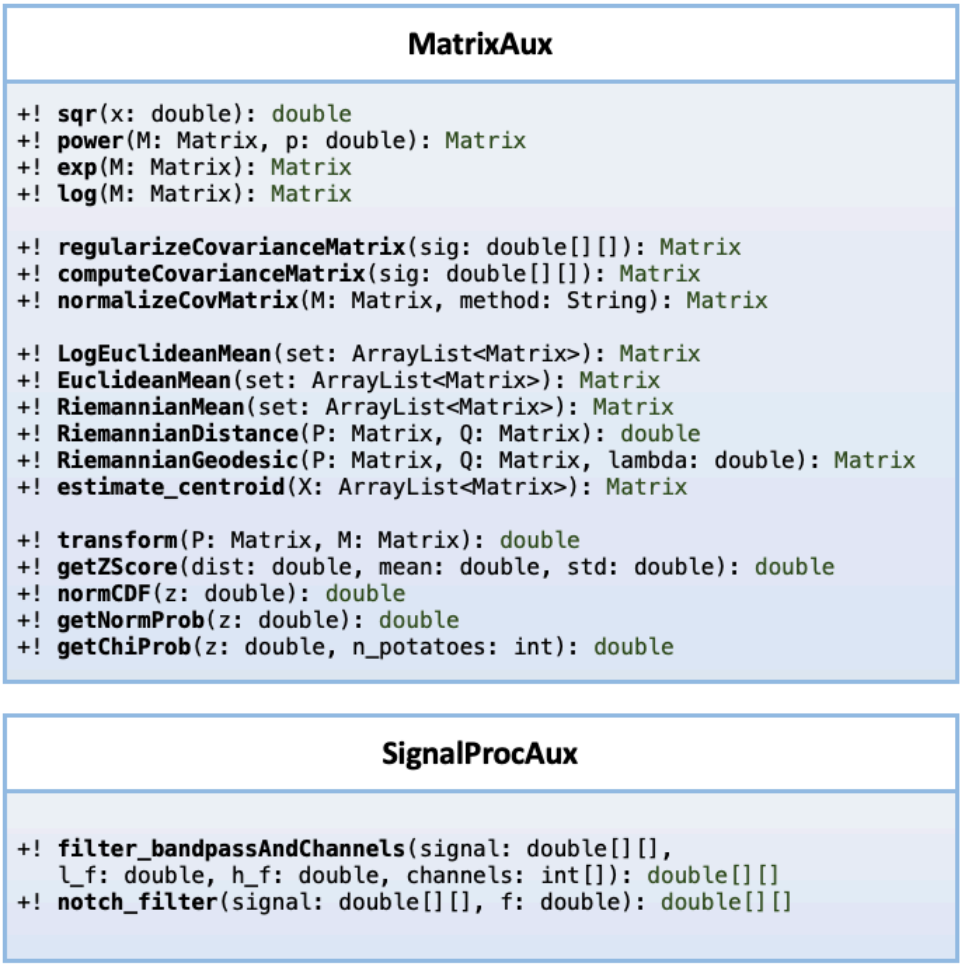


Figure 7.2: Visualization of the auxiliary classes for the RPF algorithm. ”+!”: public and static method.

7.1.3 Tests - correctness of the Java functions

Parallel to the writing of the code parts, various tests were written and executed to verify that all the functionalities were correct and that the results were in line with those obtained in Python. The tests were divided into five sections: basic matrix operations, creation of covariance matrices, Riemann geometry operations, bandpass filtering, and container class tests.

Basic matrix operations

For this group of operations, the tests consist of creating random valid and not-valid matrices in Python, copying them on Java, and running the functions on each matrix, checking that then the results are the same in both versions of the code.

Creation of the covariance matrices

For this group of tests, raw data have been injected into the methods for computing the covariance matrices, then the resulting matrices have been element-wise compared. In particular, standard covariance matrices, normalized covariance matrices, and regularized covariance matrices creation methods have been tested.

Riemannian geometry operations

For this set of tests, the covariance matrices created above have been used to compute the Riemannian distance, the Riemannian mean, the Euclidean mean, and the Riemannian geodesic among SPD matrices always both in Python and in the implemented Java code. Also in this case the results (matrices of values) have been compared element-wise.

Bandpass filtering

In this case, some raw signals have been filtered in Python and in Java, then the frequency domain has been examined to verify that the distribution is the same.

Container class tests

Finally, the entire code has been tested by running the Python and Java implementations of the RPF algorithm in parallel and checking that the probabilities of detecting the artifacts, for each section of the signal, are the same.

In Figure 7.3, the test class is visualized.

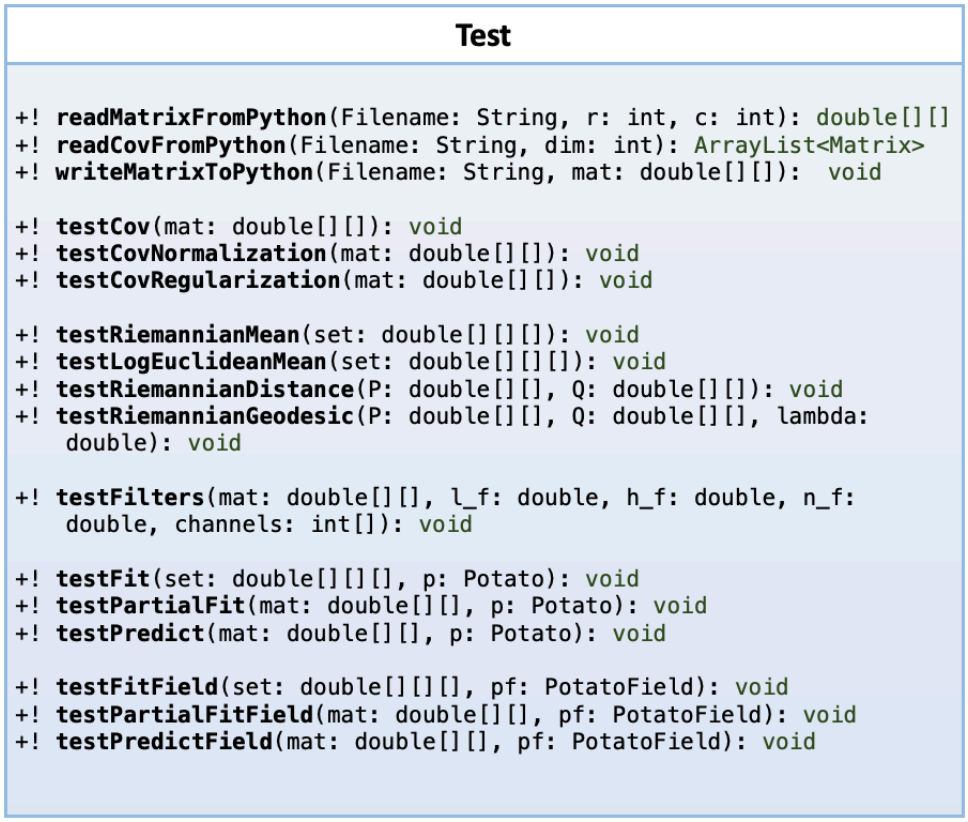


Figure 7.3: Visualization of the test class for the RPF algorithm. ”+!”: public and static method.

It can be seen that all the methods of the other classes are tested. Three first three functions are auxiliary and needed to share the results between Python and Java, to check that the results are the same. This is done in Python for ease of use. The following chunk focuses on testing the computation of the covariance matrices, i.e. normal, normalized, and regularized. Then the different Riemannian functions are tested, verifying that the returned matrices are the same as Python. The filtering is tested with ”testFilters”. Finally, the container classes PotatoField and Potato are tested with the last set of methods.

7.2 Results and conclusions

This section reports the results obtained for the tests outlined above.

7.2.1 Tests - correctness of the Java functions

Basic matrix operations

Each resulting matrix has been compared element-wise after applying the same operation on the same initial matrices. After some debugging iterations, the resulting matrices are the same. The elements of the matrices have been truncated for precision issues in the exchanging of data between Java and Python.

Creation of the covariance matrices

The testing of the creation of the covariance matrices and the debugging iterations took a considerable amount of time. If for the base computation a built-in method was provided in the library [92], the same cannot be stated for the normalized and the regularized covariance matrices. For the normalization, two methods have been implemented and tested: covariance-based and trace-based [93]. Both methods are useful when, in the particular potato, the power of the signal is not relevant, e.g. when a low-frequency artifact on the entire scalp is searched. Furthermore, the regularization has been performed via the Oracle method (OAS) [94]. In the design stage, both Ledoit-Wolf [95] and OAS methods have been analyzed. Eventually, the OAS coefficient has been preferred over the Ledoit-Wolf because convergence is significantly faster and more accurate because it is of easier implementation and comprehension and because it requires less computation time and power [94][96][97]. Regularization is used for stability purposes. In fact, in the computation of the Riemannian mean, many invertions and power elevations of matrices are required, and those led to numerical instabilities that can be solved by the shrinkage. The shrinkage consists of stabilizing the values in the matrix diagonal, thus re-centering the most extreme values. All tests led to the same results for all the beginning matrices. Thus the functions have been proved correct.

Riemannian geometry operations

The implementations of these functions (i.e. Riemannian distance, Riemannian mean, Euclidean mean, and Riemannian geodesic) have been inspired by [49][77][75] and by the Python library code [90]. Also, in this case, all the results (either matrices or values) have been element-wise compared. After several iterations, the implementations have been proven correct as before. The implementation of the Riemannian mean required particular efforts. In particular, as explained above, the regularization method for the covariance matrices has been introduced to compensate for the numerical instabilities due to invertion operations.

Bandpass filtering

In Figure 7.4, the result of the Butterworth bandpass filter is shown for the same signal, processed with Python and with Java methods.

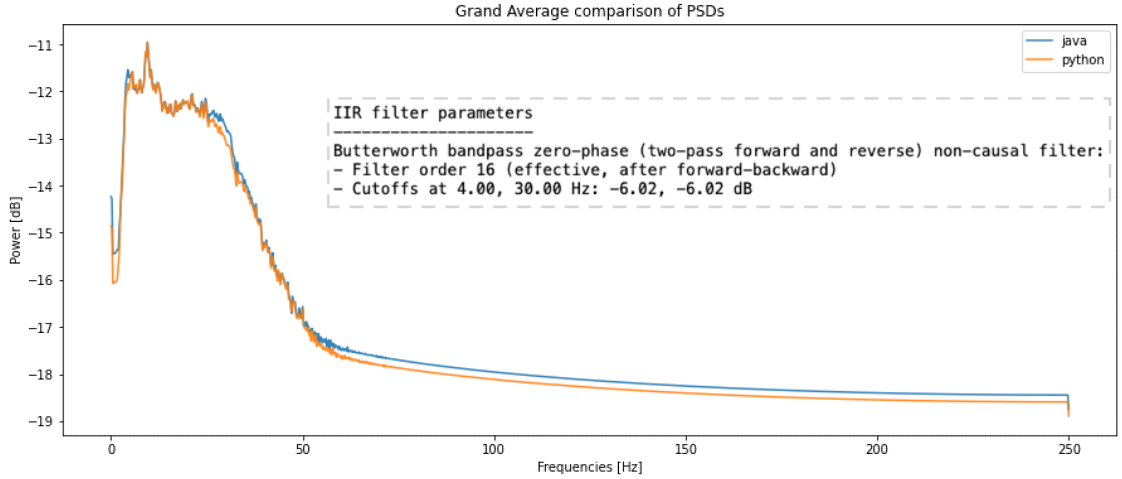


Figure 7.4: Visualization of the grand average PSD resulting from filtering the same signal with the same bandpass filter in Python and in Java.

It can be seen that the results are not exactly the same. This was not expected, as the methods are from the standard libraries, and no sources of differences are mentioned in the documentation.

In particular, the greatest differences are notable for frequencies below 3Hz and above 50Hz. This reflects a difference in the suppression of the signal outside of the cutoff regions. The most probable reason for this difference is due to inner parameters that have not been equalized between the two methods.

To dig deeper into the problem, the implementation libraries of the two methods have been analyzed. The methods, though, referred to other methods, each of them in different libraries, resulting in broad and tangled research. Eventually, for time constraints, it has been decided to postpone this analysis for the following iterations of the project, as the difference is not relevant for MVP purposes. The fact that this difference is not relevant will be shown in the probability predictions section, below.

Container classes

Both implementations gave the same results. The probabilities given from the two algorithms are, apart from numerical rounding, the same. Small differences in the probabilities, but not in the artifact detection, are introduced with the filtering. This was expected as it has been shown above that the filtering introduces small differences in the data.

7.3 Discussion and future work

Not many decisions have been made in this chapter that could have had a relevant impact on the outcome of the project. In other words, because this chapter treated the refactoring of the algorithm in Java language, and only the correctness of the implementation was of interest, all the relevant design decisions had already been taken in the previous sections. Therefore, not many discussion inputs can be observed. In particular, standard good practices have been adopted for the Java implementation, and no further work could have been done within the purpose of this project.

In the following iterations of the project, the code should be refactored and optimized under memory usage, CPU load, and running time. For example, the fixed point Riemannian mean method could be implemented. In addition, the code could be commented on in a more descriptive manner, and a stable version could be published for external debugging and usage, as no Java version of the RPF is available up to now.

8 Mobile app quality feature prototype

In the chapter, the prototyping of the quality feature for the mobile application is explained. In particular, for the MVP, the visualization of the progress bar has been implemented.

8.1 Introduction

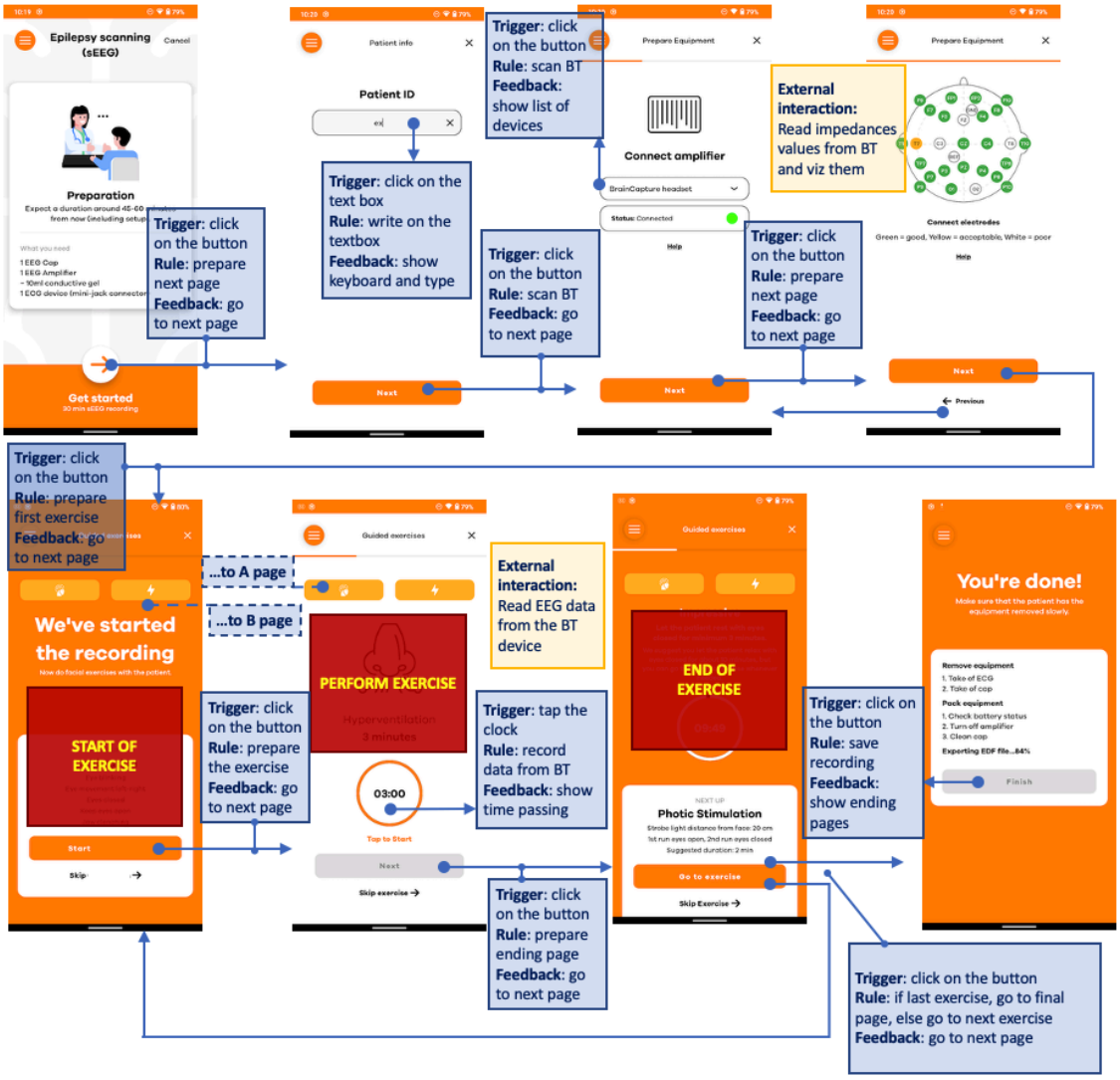
To test the feasibility of the online quality control feature in the mobile application, the feature has been implemented. Having the feature running returned important knowledge on the technical aspect side, as well as the user experience. The feature needed to cooperate with the already existing application. The purpose of the app is to guide non-expert technicians through the standard EEG scan, providing the best information for acquiring good-quality recordings. The goal of the quality feature is indeed on this side, i.e. improving the overall quality by prolonging the scan of an amount of time at least equal to the useless data recorded, thus, the epochs labeled as artifacts by the RPF algorithm.

For the MVP, an idea of the initial requirements was given by the results of chapter 4 Users' needs and system requirements. In fact, in the interviews performed with expert and non-expert technicians, as well as from the usability tests provided, the need for a progress bar that can show the parts of the clean data against poor quality data was found.

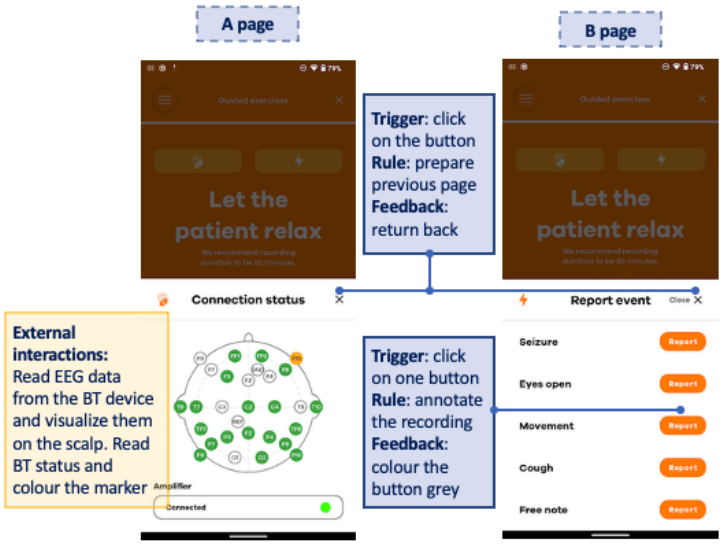
The general and high-level workflow of the existing app is overall linear and shown on the next page in the wireframe in Figure 8.1. The vision for the existing mobile application is to guide the operator through all the steps to perform an EEG scan that can be used by a doctor to diagnose epilepsy. In other words, doctors need artifact-free segments in which the EEG brain signal is readable. It is better if these segments are consecutive and long, rather than fragmented. From the welcoming page, in Figure 8.1(a) in the top-left corner, the nurse is guided through the registration of the patient and the connection to the Bluetooth amplifier. From there, the first quality check is provided by the feature implemented in the top-right corner. The application reads the impedance measured for each electrode (as a proportion of 128Hz signal power read by the electrodes against the one sent by GND [98]) and visualizes the quality of the connection with a color scale: green - good, orange - poor, white - insufficient.

In the second row of screens in Figure 8.1(a), it is assumed that everything is functioning and well connected. The application starts saving the incoming EEG signal from the Bluetooth connection. A customizable set of exercises is proposed. This is a recursive stage, in which the operator instructs the patient on the next exercise (bottom-left screen), starts the timer for the exercise, and visually checks that the patient doesn't create artifacts. When the timer stops, it can go to the end of the exercise to continue with the next one. Each exercise can be skipped. At the beginning and at the end of each exercise, an annotation is assigned to the signal, to make the following reading by the doctor easier. When the set of exercises is concluded, the uploading page is shown (bottom-right screen), and the recording is saved.

At every moment during the recording, the operator can check that the device is connected and that all the channels are at a proper impedance level. In addition, the operator can annotate visually recognized artifacts, like talking, snoozing, or movements. These two functions are provided by the screens in Figure 8.1 (b), accessible at any moment with the two buttons in the exercise screens.



(a) Linear flow of the app



(b) Additional branches for data inspection

Figure 8.1: Wireframe visualization of the existing mobile application, updated to September 2022.

The quality assessment, in the existing application, is given by the impedance feature, and by the manually annotated events that the operator can detect. For a better track of the artifacts, and for an estimation of the time to prolong the recording to have a standard time (i.e. twenty minutes) of a good quality recording, the quality feature has been implemented.

In Chapter 4 Users' needs and system requirements, the necessity for an intuitive way to visualize the artifact segments of EEG signal (i.e. the progress bar for the MVP) has been interiorized. In Chapter 6 Riemannian Potato Field (RPF) and practical design, the online and embedded friendly method for artifact detection has been studied and tuned for BrainCapture data, analysing also the performances. In chapter 7 RPF implementation for embedded systems, the RPF algorithm has been converted into Java language, to make it compatible with an android operating system portable device.

In this chapter, the libraries written are exploited to implement the quality feature and to cooperate with the existing application.

8.2 Methods

8.2.1 Android development

The structure of the RPF algorithm has been implemented in Kotlin, the language in which the existing app is developed. Kotlin is compatible with native Java classes and methods, so the libraries can be applied with no further work. Java libraries have been written from the beginning with the practical application idea, so they have been designed and implemented to handle in the correct way the data received. Thus, only minor details are going to be corrected on the library's side.

From the application functional side, some functions will be modified or added to include the RPF algorithm. In particular, ”startRecording” is the function called when the exercises start and that emits the impedance values acquired from the BT flow. This function is going to be modified because, differently from the case before in which the EEG data were just saved to the recording, now they are going to be read and evaluated with the RPF. Therefore, ”startRecording” initializes a new function called ”startEvaluateEEG” to generate a flow of quality objects. In more detail, this function gets the EEG samples from ”startRecording”, it buffers them into chunks of data, and passes them to a new function ”getChunkQuality”. ”getChunkQuality” receives these chunks of data and it applies the RPF algorithm to evaluate their quality. The results of ”getChunkQuality” are given as input to the smoothing function ”getSegmentQuality”, which applies an averaging method to get a smoother visualization. Its result is finally emitted in a new flow by ”startEvaluateEEG”. Finally, the quality objects are collected by the ”collectQualityFlow” and saved in a mutable list. The prolonging estimated time is computed from the number of chunks labeled as artifacts.

Thirdly, from the user interface side, a new button that opens a newly designed and implemented screen will be added to the exercise screens. In this new screen, a label with the prolonging time estimation is visualized, as well as the progress bar. The bar is going to be initially designed as a sequence of rectangles, each colored with the respective segment quality. This last interface object is going to be bound through the ”BindingAdapter” named ”segmentQuality”. This will bind the color of each segment of the bar with the respective value in the mutable list.

The entire process is visualized in Figure 8.2 below.

```
1:  fun getSegmentQuality(segment): return qualityObject
2:      if nr. clean chunks in segment > 2*t/3
3:          return green
4:      else if nr. clean chunks in segment > t/3
5:          return orange
6:      else
7:          return red
8:
9:  fun getChunkQuality(chunk): return boolean
10:     return rpfClassification(chunk)
11:
12: fun startEvaluateEEG(EEGsamples): emit qualityObjects
13:     initialization: chunk ← empty buffer , segment ← queue[t elem.]
14:     chunk ← n consecutive EEGsamples
15:     calibration: rpfCalibration(first c chunks)
16:     for each chunk
17:         segment ← getChunkQuality(chunk)
18:     emit getSegmentQuality(segment)
19:
20: fun startRecording(collect EEGsamples):
21:     ... //other not relevant instructions
22:     startEvaluateEEG(EEGsamples)
23:
24: list barSegments
25: fun collectQualityFlow(collect qualityObject):
26:     barSegments ← qualityObject
27:
28: binder segmentQuality:
29:     for each segment in barSegments
30:         colour the UI segment respectively
```

Figure 8.2: Visualization of the flow for the modified or added functions for the quality feature.

8.2.2 User Interface validation - A/B tests

In order to validate the visualization of the progress bar, A/B tests have been prepared on the user interface. In the following, the different tests are shown.

Amount of quality levels

An A/b test has been designed to test how many levels to present to the EEG operator. In particular, whether using three or two levels, e.g. green - good quality, orange - medium, and red - bad; or green - good quality, red - bad quality. The two different designs are shown below:

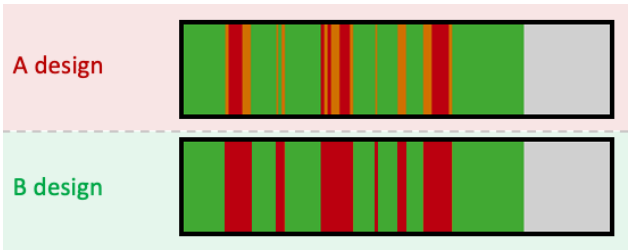


Figure 8.3: Visualization of the A/B test for the number of quality levels.

The test was in a semi-structured questionnaire manner. In particular, the level of comprehension of the colors as well as the specificity of the data shown have been planned to be tested. Due to time constraints, this test had not been performed.

Feature accessing mode

First of all, an A/B test has been designed to check the preferred method to access the new feature. In particular, design A shows three buttons on the exercise screens, and the middle one directly opens the feature screen. Design B, instead, shows two buttons, and the left one brings to a multitab screen opened on the impedance tab, in which the operator can switch to the quality tab. The two different designs are shown in the figure below.

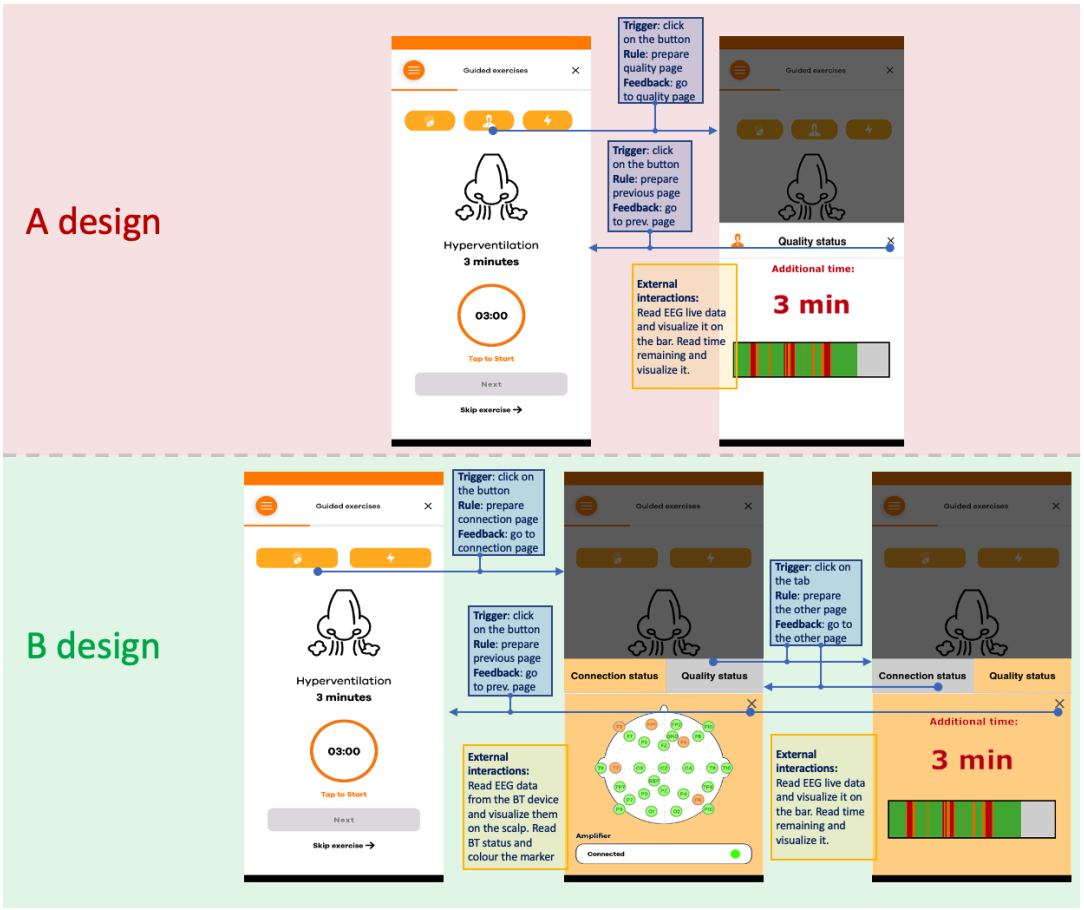


Figure 8.4: Visualization of the A/B test for the feature accessing flow.

Also, this test has not been performed for time constraints. In this case, the time for reaching the quality status screen would have been measured, after asking the operator to check for the quality of the recording. In addition, a semi-structured questionnaire would have asked for the intuitiveness of the options.

8.2.3 Correctness and performance testing

After the android development, the feature has been tested with simulated data to verify that the results are consistent with the predictions given by Python. In particular, the same recording has been analyzed with the Python algorithm and given as simulated data to the Android algorithm. Then, the prediction of each epoch has been compared element-wise.

Secondly, the performance of the algorithm has been measured. In particular, memory allocation, CPU load, and battery drain have been profiled with the Android Studio profiler with and without the feature running, to compare the differences in the performances. The profilation has been performed during the first exercise, to remove other variables.

8.3 Results and conclusions

8.3.1 Android development

The resulting pseudocode is shown in Figure 8.2 in the methods. The final code cannot be published for the company policy, as the feature is integrated into the entire application.

In Figure 8.5 below, the UI for the new quality feature is shown.

The bar is updated every time a new segment is recorded, and the quality is visualized with the three colors. The timer shows how long the recording is estimated to be prolonged to reach twenty minutes of good-quality data scan.

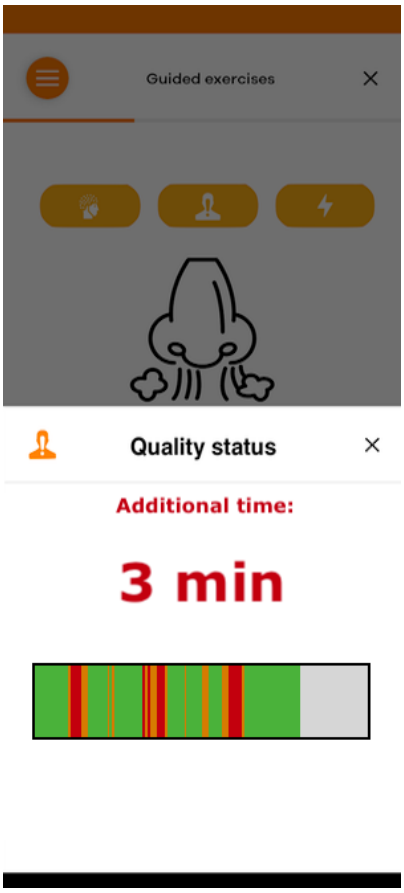


Figure 8.5: UI of the new quality feature, reachable from the middle button present in every exercise screen.

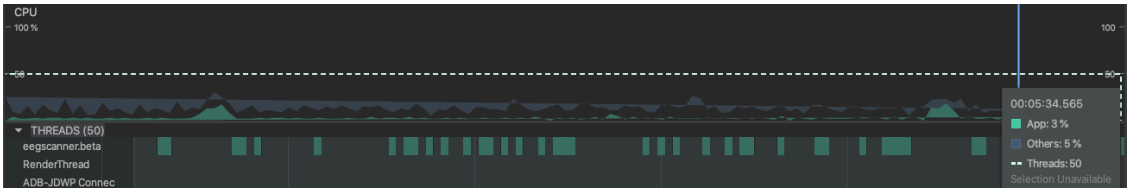
8.3.2 Correctness and performance testing

In the following, the results for the correctness of the implementation and the comparison between the performances of the app with and without the new features are shown.

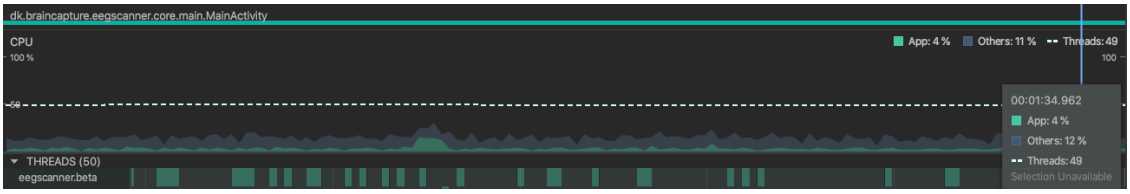
Correctness

The outcome of the algorithm in Java and in Python has been compared. The Java code adds the extra step of averaging epochs for having smoother segments in the protocol. After filtering the Python results with the same method, the segments are equal element-wise, proving the correctness in a general case.

CPU load



(a) Visualization of the CPU load without the quality feature



(b) Visualization of the CPU load with the quality feature running

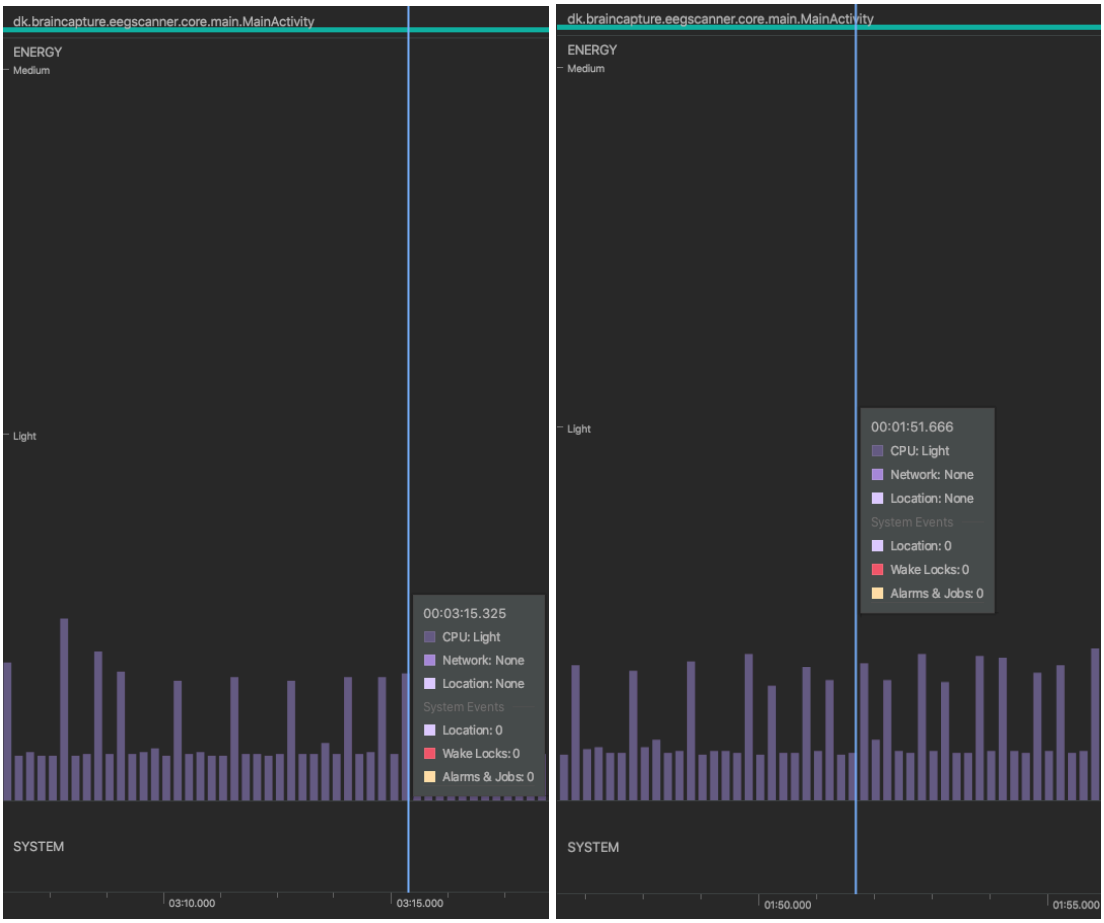
Figure 8.6: Visualizations of the CPU load for the app with and without the quality feature. In both cases, the profilation has been performed during the first exercise.

As it can be seen in Figure 8.6, the difference between the two versions of the application is negligible. In fact in Figure 8.6(a), it can be seen that the CPU load without the quality feature is 3%, whereas in (b), it can be seen that the quality feature adds 1% of CPU pressure, to reach a total of 4% of load.

This behavior was expected, as the method was chosen to be computational light.

In addition, it can be seen that routine background applications weights more on the CPU than not the application itself. This means that its entire contribution is inexpensive.

Battery consumption



(a) Visualization of the battery consumption without the quality feature

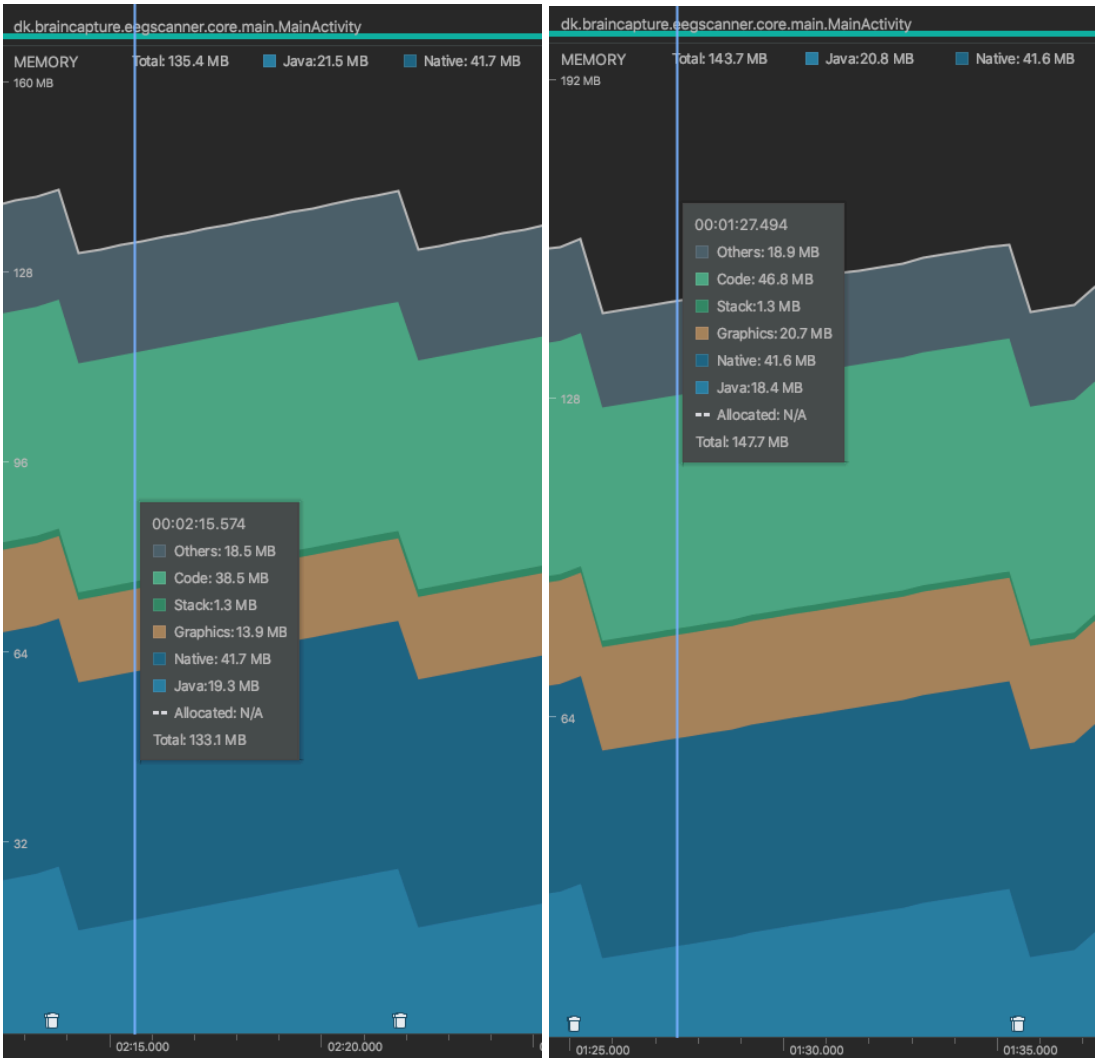
(b) Visualization of the battery consumption with the quality feature running

Figure 8.7: Visualizations of the battery consumption for the app with and without the quality feature. In both cases, the profilation has been performed during the first exercise.

The difference between Figure 8.7(a) and (b) (i.e. without (a) and with (b) the quality feature) is minimal, as expected from the fact that the difference of CPU load is not relevant.

In fact, even though CPU pressure and power consumption are not proportional in general [99], in this case, the CPU load difference is a good indication that the battery pressure is not going to change heavily because all the other relevant factors as screen time and RAM usage are constant in the two versions.

Memory pressure



(a) Visualization of the memory pressure without the quality feature

(b) Visualization of the memory pressure with the quality feature running

Figure 8.8: Visualizations of the memory pressure for the app with and without the quality feature. In both cases, the profilation has been performed during the first exercise.

In Figure 8.8, the memory used by the application is shown for the case in which the quality feature is disabled (a) and the one in which is running (b). Note that the scale for the two cases is not the same. In this case, the difference is small but relevant. In fact, removing the constant items as ”Java”, ”Native”, and ”Others”, the differences lie in ”Graphics” ($+9.6\%$), in which the new feature adds one screen with the progression bar, and ”Code” ($+18\%$), for the new functions added. On the other hand, ”Stack”, i.e. the LIFO stack of objects, is constant. This means that the potato field does not impact memory relevantly.

It is important to note that the memory pressure has not been optimized in the design of the algorithm, as it was out of the scope of the project. So these results can be improved in the following iterations.

8.4 Discussion and future work

The results reinforce what was already known from the theory, thus the overload added by this feature is minimal. The goal for this first iteration was to study the feasibility of the system, and not to optimize it, thus the final additional load can only improve.

This chapter is open to an immense amount of future work. First of all, the two UI tests could be performed and implemented, and all UI aspects (e.g. information shown, interactions with the users, colors...) should be validated. Secondly, more performance tests could be executed in a more systematic and continuous way. Thirdly, the entire smoothing method should be validated through experiments to examine the best granularity to return to the operator. Finally, at this stage, everything is ready to test the capacity of the system to support the EEG operators and to improve the overall quality of the recorded files. It is believed that these tests will return a huge amount of knowledge to guide the work toward the second iteration of the project.

9 Discussions and future work

The following chapter will cover the discussions and future work of the project. As mentioned in the Methods Overview chapter, the entire process followed a holistic and multidisciplinary technique, in which each component contributes to validating, giving feedback, and improving each other. In all the chapters, the discussions and future work sections focused on their specific topics. In this chapter, instead, the mutual influences are discussed, and the general future work is highlighted.

In this top-down setup, the users' needs chapter introduced the constraints and requirements for the system. The knowledge that the feature had to run on limited computational resources and in a real-time matter directed the research on light-weight detection algorithms, preferred to heavier but possibly more accurate ones. This choice can be argued in different ways. Only to cite one example, the training takes more time for machine and deep learning classifiers, but once the model is trained, the personal experience suggests that the prediction can be fast enough and computationally inexpensive for a mobile environment, thus combining accuracy with computational power. In addition, new frameworks such as TensorFlow Light [100] have been released and can be used to run deep learning models on an android device. Several reasons have turned away from this solution. First- EEG signals are intrinsically different for users and sessions affecting the re-usability and generalization of models [101], thus making transfer learning of fundamental importance. From the Riemannian geometry study [78], it was known that Riemannian based algorithms are performing transfer learning particularly due to Riemannian geometric properties [75]. In addition, the basis of the RPF algorithm was already available in Python, speeding up the initial stages and giving the opportunity to test it and assess the feasibility in a real application. Also, RPF presents the outcome in an intuitive manner, thus facilitating the work of visualizing and interpreting the results. Although this point does not seem of much importance at this stage of the project, it will be central to future iterations, where, instead of having a binary classifier, each artifact can be separately classified and the operator can focus on each one, augmenting specificity. This aspect is not only important for the UX and UI, but also for the design of the field. In fact, a multi-class classifier can return extremely useful insights on which artifacts are mostly mis-classified, giving the opportunity to improve the field to better fit the data. In this context, it is also known from the users' needs chapter that not all artifacts have the same importance and that for example, eyes blinking are less important than head movements. The RPF fits this aspect quite well. In fact, in this iteration, in the evaluation of the final p-value, the same importance is assigned to each potato, but in the next iterations, it would be possible to adjust the different weights or to give complete control, on which artifacts to keep and which to ignore, to the operator, thus increasing customization.

Over the scope of this first stage of the thesis, the RPF has not been compared to other algorithms in an experimental manner. From the 'understanding' phase, though, it is known that several researchers are studying different methods. These can be compared through the framework built in the prototype chapter. In fact, the "return rpfClassification(chunk)" in line 10 of the pseudo-code in Figure 8.2, called by the "getChunkQuality" function, can be replaced by any algorithm that accepts an epoch as input and returns a binary label. In this way, the performances of any new method can be experimentally profiled. In addition, data collected in parallel from the Filadelfia device has been used to tune the RPF. Filadelfia data is less noisy because analogue filters in the Natus amplifier process the

EEG signal. An analysis of the differences in the parameters tuning and in the performance of RPF for BrainCapture's data is of interest in the future.

The data collected is of considerable value not only for all the projects that requested them but also for future ones. It is of great importance to understand the impact of the protocol used in the classifier algorithms. The data is strongly biased around the artifacts. In particular, it takes the proportion of artifacts to the extreme, losing the specificity of the real-case scenario. In a standard EEG scan, the proportion of clean epochs is markedly higher than the artifact one. Moreover, depending on the class of artifact, artifacts appear randomly during the recording and for a shorter period of time than the one presented in the protocol used. These two factors can influence both the parameter values of the RPF and the performance scores. On one hand, the window length and overlap (and, indirectly, also z and p thresholds) are influenced by the consecutive representation of the same artifact. The RPF is convenient to take a long window length because it encapsulates more of the artifact variance, and it is easier to separate it from a clean epoch (thus influencing also z and p). On the other hand, the resulting balance between the two classes made it easier to apply the F2 score, which would have been greatly affected by data unbalancing. It is also true that this balance would be lost in any case moving to a multi-artifact classifier, as in that case each class of artifacts would be unbalanced with the "clean" class. In that eventuality, data should be re-balanced removing a proportion of the "clean" epochs from the training set or augmenting the artifacts classes. Otherwise, metrics not dependent on the balancing can be adopted, such as the ROC AUC score. Using the AUC score, though, the bigger importance given to the artifact class would be lost for better separability. But it is known from the users' needs, that in this case recall is more important than separability. In addition, it should be always kept in mind all the practical variables that can influence the choices, such as time constraints. Having a protocol in which the artifacts are controlled, allows for annotating recordings in a faster way and with less dependence on experts than in a standard recording. This allows non-expert neurologists to create the ground truth, and in the few weeks available for the project it gave the opportunity to insert it in the pipeline. Finally, the compartmentalization in the protocol is also reflected in the calibration exercises at the beginning of the standard EEG scan. That stage can become of fundamental importance for the performances of any algorithm, because, from the beginning and in an individualized manner, examples of any class of artifacts are available before starting any recording.

The calibration is one aspect that can weigh enormously on the entire analysis. Deep investigations on the calibration stage were out of the scope of this project. A two-step robust calibration technique has already been discussed in the RPF discussions section. On top of that, past recordings can be used to create a generalized baseline matrix, used in the first step of the robust calibration to remove the outliers. Another huge branch to inspect could be the advantages brought by using the standardised protocol. In fact, the artifact exercises at the beginning, useful for the neurologists to have a reference when they read the scans, can return more information than the one used now. A future exploration can evolve by applying Riemannian clustering methods like Mean-shift or K-mean [102] to the signal. This could mean having one baseline for each class of artifacts and clustering the incoming epochs with the closest baseline.

Just the surface of the UX design and prototyping has been scratched. In this part of the project, immense work can be done to improve all the other parts. For example, it is known from the results of the users' needs chapter, that the patients can be particularly stressed in the first part of the recording, affecting the calibration section. Simple heuristics on the standard variance of EEG clean signals can be used to start with the exercises only when it is recorded with proper quality. Secondly, a bunch of heuristics can be used before and during the recording to give feedback to the operators. For example, $50Hz$ noise level can be constantly measured to make sure that the GND and REF electrodes are well connected (practical eventuality, found in data acquired with an old version of the BrainCapture's device in Bhutan [103]). In addition, the entire smoothing process to pass from epoch quality to segment quality has just been sketched. From the interviews with the doctors and the EEG technicians, it is clear that is not relevant to alert the operator for small and sparse chunks of artifact signal. For this reason, a smoothing function can be designed to summarise the information into bigger segments to visualise in the progress bar. This part strictly depends on the window size parameter returned by the hyperparameters tuning.

Finally, the outcomes of this project allow for testing the actual improvements in the quality of the signals acquired, by the difference in the data acquired with the quality feature activated or not.

10 Conclusions

The feasibility of an online EEG artifact detection algorithm, implemented on an embedded system, has been successfully researched in this analysis. The conclusions of the entire project are sketched in this chapter.

In particular, after an initial **introduction** to the problem and insight into the analysis, a brief overview of the **state of the art** of solutions has been summarized, showing that only a few methods are suitable for the use case.

Different documents previously collected by the company have been studied to **empathise** with the **users' needs**. Further analysis was conducted through experiments and semi-structured questionnaires to doctors, expert EEG technicians, and non-expert volunteers. In addition, several **data collection** sessions have been performed in a neurological hospital, to immerse into the environment and get hands-on experience. The theoretical knowledge on EEG scan reading has been acquired from doctors, understanding there is a need to clean data to visualize and recognize seizure patterns. From experienced technicians and recording sessions, different practical expedients to obtain clean scans have been interiorised. A list of good practices to avoid the generation of artifacts has been compiled, reserving particular importance to the relaxation part. Finally, the best methods to support non-expert technicians have been assessed by the experience gained by working beside them.

The problem and the **system requirements** have been **defined** and then shown using the lean canvas tool. Subsequently, solutions have been **ideated** and presented through a user story map. The outcomes are that the direct users (i.e. non-expert operators) need an intuitive system that supports them during the recording of the data. This support is going to be implemented as a new feature in the existing mobile app from BrainCapture, showing a real-time quality-based color-coded progress bar. In addition, an estimation of the duration to prolong the recording is visualized on the same screen. The **ideation** process acceded to the selection of the **Riemannian Potato Field (RPF)** for the **design** of the real-time artifact detector. It has been tuned firstly by designing different fields and then applying a hyperparameter optimizer, maximizing the F2 score obtained by comparing the predicted labels of the epochs with the ground truth retrieved from the events annotated in the scans collected. The results obtained have been analysed to understand the mutual influences between parameters and to analyse execution time with a view to real-time implementation.

To test the feasibility, in response to the problem statement, the RPF has been **implemented for embedded systems**, i.e. coded into Java language. The **mobile application quality feature** has also been **prototyped** and incorporated into the existing android application. The **tests** conducted consisted of comparing the profiled CPU load, memory pressure, and energy consumption of the mobile application in which the feature is activated and in which it is deactivated.

The results prove that this feature is feasible to run online and on a mobile device. In addition, an F2 score of 0.81 has been obtained from the data used. Finally, the process is developed with a top-down strategy, always keeping the users' needs as the focal point. Thus, the proposed solutions address all the practical questions posed by the research problem successfully.

Bibliography

[1] World healt Organization. *Epilepsy*. 9 February 2022. URL: https://www.who.int/news-room/fact-sheets/detail/epilepsy.

[2] Robert S Fisher et al. “ILAE official report: a practical clinical definition of epilepsy”. In: *Epilepsia* 55.4 (2014), pp. 475–482.

[3] Ethan M Goldberg and Douglas A Coulter. “Mechanisms of epileptogenesis: a convergence on neural circuit dysfunction”. In: *Nature Reviews Neuroscience* 14.5 (2013), pp. 337–349.

[4] Lara V Marcuse, Madeline C Fields, and Ji Yeoun Jenna Yoo. *Rowan’s Primer of EEG E-Book*. Elsevier Health Sciences, 2015.

[5] Riitta Hari and Aina Puce. *MEG-EEG Primer*. Oxford University Press, 2017.

[6] Mona Sazgar and Michael G Young. “EEG artifacts”. In: *Absolute epilepsy and EEG rotation review*. Springer, 2019, pp. 149–162.

[7] Paul Loomis. *Grant application*. unpublished. 2022.

[8] BrainCapture. *BrainCapture - Epilepsy*. 2022. URL: https://braincapture.dk.

[9] BrainCapture. *BrainCapture*. 2022. URL: https://braincapture.dk.

[10] Luoise Klok. personal communication. July 2022.

[11] Paul Loomis. personal communication. Sept. 2022.

[12] Radu Gatej. personal communication. July 2022.

[13] Natus. *Natus website*. 2020. URL: https://natus.com (visited on 10/15/2022).

[14] Cadwell. *Cadwell website*. 2019. URL: https://www.cadwell.com (visited on 10/15/2022).

[15] Compumedics. *Compumedics website*. 2021. URL: https://www.compumedics.com.au/en/ (visited on 10/15/2022).

[16] zeto. *zeto website*. 2022. URL: https://zeto-inc.com (visited on 10/15/2022).

[17] brainproducts. *brainproducts website*. 2022. URL: https://www.brainproducts.com (visited on 10/15/2022).

[18] Alessandro Liberati et al. “The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration”. In: *Journal of clinical epidemiology* 62.10 (2009), e1–e34.

[19] Matthew J Page et al. “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews”. In: *Systematic reviews* 10.1 (2021), pp. 1–11.

[20] Malik Muhammad Naeem Mannan, Muhammad Ahmad Kamran, and Myung Yung Jeong. “Identification and removal of physiological artifacts from electroencephalogram signals: A review”. In: *Ieee Access* 6 (2018), pp. 30630–30652.

[21] Xiao Jiang, Gui-Bin Bian, and Zean Tian. “Removal of artifacts from EEG signals: a review”. In: *Sensors* 19.5 (2019), p. 987.

[22] Chi Qin Lai et al. “Artifacts and noise removal for electroencephalogram (EEG): A literature review”. In: *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. IEEE. 2018, pp. 326–332.

[23] Shaibal Barua and Shahina Begum. “A review on machine learning algorithms in handling EEG artifacts”. In: *The Swedish AI Society (SAIS) Workshop SAIS, 14, 22-23 May 2014, Stockholm, Sweden*. 2014.

[24] Md Kafiul Islam, Amir Rastegarnia, and Zhi Yang. “Methods for artifact detection and removal from scalp EEG: A review”. In: *Neurophysiologie Clinique/Clinical Neurophysiology* 46.4-5 (2016), pp. 287–305.

[25] Laura Frølich, Tobias S Andersen, and Morten Mørup. “Classification of independent components of EEG into multiple artifact classes”. In: *Psychophysiology* 52.1 (2015), pp. 32–45.
[26] V Krishnaveni et al. “Automatic identification and removal of ocular artifacts from EEG using wavelet transform”. In: *Measurement science review* 6.4 (2006), pp. 45–57.
[27] Vernon Lawhern, W David Hairston, and Kay Robbins. “DETECT: A MATLAB toolbox for event detection and identification in time series, with applications to artifact detection in EEG signals”. In: *PloS one* 8.4 (2013), e62944.
[28] Demetres Kostas, Stephane Aroca-Ouellette, and Frank Rudzicz. “BENDR: using transformers and a contrastive self-supervised learning task to learn from massive amounts of EEG data”. In: *arXiv preprint arXiv:2101.12037* (2021).
[29] Niels Birbaumer et al. “Neurofeedback and brain–computer interface: clinical applications”. In: *International review of neurobiology* 86 (2009), pp. 107–117.
[30] Arnaud Delorme and Scott Makeig. “EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis”. In: *Journal of neuroscience methods* 134.1 (2004), pp. 9–21.
[31] Arnaud Delorme et al. “EEGLAB, SIFT, NFT, BCILAB, and ERICA: new tools for advanced EEG processing”. In: *Computational intelligence and neuroscience* 2011 (2011).
[32] Christian Andreas Kothe and Scott Makeig. “BCILAB: a platform for brain–computer interface development”. In: *Journal of neural engineering* 10.5 (2013), p. 056014.
[33] Alois Schlögl and Clemens Brunner. “BioSig: a free and open source software library for BCI research”. In: *Computer* 41.10 (2008), pp. 44–50.
[34] Yann Renard et al. “Openvibe: An open-source software platform to design, test, and use brain–computer interfaces in real and virtual environments”. In: *Presence* 19.1 (2010), pp. 35–53.
[35] Hasso Plattner. “An introduction to design thinking”. In: *Iinstitute of Design at Stanford* (2013), pp. 1–15.
[36] Interaction design foundation. *The 5 Stages in the Design Thinking Process*. 2022. URL: https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process.
[37] Luoise Klok. personal communication. Sept. 2022.
[38] Radu Gatej. personal communication. Sept. 2022.
[39] Tue Lehn-Schioler. personal communication. June 2022.
[40] BrainCapture. “BC1- User requirements - working file-2.pdf”. Not Published. 2021.
[41] TeleEEG. “Top Technical Tips - ’In a nutshell’”. Private document. 2018.
[42] teleeeg. *teleeeg website*. 2022. URL: https://www.teleeeg.org (visited on 10/15/2022).
[43] Filadelfia Hospital. *Filadelfia Hospital website*. 2022. URL: https://www.filadelfia.dk (visited on 10/15/2022).
[44] Iman Chelhi. *User tests Qualitative Data*. 2022. URL: https://miro.com/app/board/uXjVPSTPzh0=/?share%5C_link%5C_id=19416785169.
[45] Mullen Steve. *An Introduction to Lean Canvas*. 2022. URL: https://medium.com/@steve_mullen/an-introduction-to-lean-canvas-5c17c469d3e0 (visited on 10/15/2022).
[46] Jeff Patton. *User Story Mapping*. Beijing: O’Reilly, 2014. ISBN: 978-1-4919-0490-9. URL: https://www.safaribooksonline.com/library/view/user-story-mapping/9781491904893/.
[47] Günther Deuschl. “Recommendations for the practice of clinical neurophysiology”. In: *guidelines of the International Federation of Clinical Neurophysiology* (1999).

[48] Wikipedia. *High impedance*. 2018. URL: https://en.wikipedia.org/wiki/High_impedance.
[49] Quentin Barthélemy et al. "The Riemannian potato field: a tool for online signal quality index of EEG". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.2 (2019), pp. 244–255.
[50] Sandor Beniczky. personal communications. June 25, 2022.
[51] Tobias Andersen. personal communications. June 25, 2022.
[52] Natus. *Nicolet ICU Monitor*. 2022. URL: https://www.cephalon.eu/products/eeg-erp/aeeg/#1505386051622-cb88454c-3f2b2a7e-6a20 (visited on 10/15/2022).
[53] Wikipedia. *10–20 system (EEG)*. 2022. URL: https://en.wikipedia.org/wiki/10%E2%80%9320_system_(EEG) (visited on 10/15/2022).
[54] Pål G Larsson, Orvar Eeg-Olofsson, and Göran Lantz. "Alpha frequency estimation in patients with epilepsy". In: *Clinical EEG and Neuroscience* 43.2 (2012), pp. 97–104.
[55] Mila Halgren et al. "The generation and propagation of the human alpha rhythm". In: *Proceedings of the National Academy of Sciences* 116.47 (2019), pp. 23772–23782.
[56] Radu Gatej. personal communication. Nov. 2022.
[57] N Takahashi et al. "Frontal midline theta rhythm in young healthy adults". In: *Clinical Electroencephalography* 28.1 (1997), pp. 49–54.
[58] EEGpedia. *Midline theta rhythm (Ciganek)*. 2018. URL: http://www.eegpedia.org/index.php?title=Midline_theta_rhythm_(Ciganek).
[59] Gert Pfurtscheller, A Stancak Jr, and Ch Neuper. "Event-related synchronization (ERS) in the alpha band—an electrophysiological correlate of cortical idling: a review". In: *International journal of psychophysiology* 24.1-2 (1996), pp. 39–46.
[60] Edgar Douglas Adrian and Brian HC Matthews. "The Berger rhythm: potential changes from the occipital lobes in man". In: *Brain* 57.4 (1934), pp. 355–385.
[61] Simon P Kelly et al. "Increases in alpha oscillatory power reflect an active retinotopic mechanism for distracter suppression during sustained visuospatial attention". In: *Journal of neurophysiology* 95.6 (2006), pp. 3844–3851.
[62] Tonia A Rihs, Christoph M Michel, and Gregor Thut. "Mechanisms of selective inhibition in visual spatial attention are indexed by α-band EEG synchronization". In: *European Journal of Neuroscience* 25.2 (2007), pp. 603–610.
[63] Gregor Thut et al. "α-Band electroencephalographic activity over occipital cortex indexes visuospatial attention bias and predicts visual target detection". In: *Journal of Neuroscience* 26.37 (2006), pp. 9494–9502.
[64] Lawrence J Hirsch and Richard Brenner. *Atlas of EEG in critical care*. John Wiley & Sons, 2011.
[65] Reinmar J Kobler et al. "Corneo-retinal-dipole and eyelid-related eye artifacts can be corrected offline and online in electroencephalographic and magnetoencephalographic signals". In: *NeuroImage* 218 (2020), p. 117000.
[66] Irina I Goncharova et al. "EMG contamination of EEG: spectral and topographical characteristics". In: *Clinical neurophysiology* 114.9 (2003), pp. 1580–1593.
[67] Alexander J Shackman et al. "Electromyogenic artifacts and electroencephalographic inferences". In: *Brain topography* 22.1 (2009), pp. 7–12.
[68] Masaki Iwasaki et al. "Effects of eyelid closure, blinks, and eye movements on the electroencephalogram". In: *Clinical Neurophysiology* 116.4 (2005), pp. 878–885.
[69] Rebekka Lindholm Ronne Hansen. personal communications. Sept.–Dec. 2022.

[70] Dario Farina, Roberto Merletti, and Roger M Enoka. “The extraction of neural strategies from the surface EMG: an update”. In: *Journal of Applied Physiology* 117.11 (2014), pp. 1215–1230.
[71] Junshui Ma et al. “Muscle artifacts in multichannel EEG: characteristics and reduction”. In: *Clinical neurophysiology* 123.8 (2012), pp. 1676–1686.
[72] Bruce B Winter and John G Webster. “Driven-right-leg circuit design”. In: *IEEE Transactions on biomedical engineering* 1 (1983), pp. 62–66.
[73] Barbara Kitchenham. “A procedure for analyzing unbalanced datasets”. In: *IEEE transactions on Software Engineering* 24.4 (1998), pp. 278–301.
[74] KP Suresh. “An overview of randomization techniques: an unbiased assessment of outcome in clinical research”. In: *Journal of human reproductive sciences* 4.1 (2011), p. 8.
[75] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. “Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review”. In: *Brain-Computer Interfaces* 4.3 (2017), pp. 155–174.
[76] Ronald Aylmer Fisher. “Statistical methods for research workers”. In: *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.
[77] Alexandre Barachant, Anton Andreev, and Marco Congedo. “The Riemannian Potato: an automatic and adaptive artifact detection method for online experiments using Riemannian geometry”. In: *TOBI Workshop IV*. 2013, pp. 19–20.
[78] Giovanni Grego. *Geodesics in the Riemannian Geometry for EEG Artifact detection*. 2022. URL: https://github.com/GiovanniGr/RiemannianGeometrySpecialCourse/blob/main/SpecialCourseRiemann-2.pdf.
[79] Emmanuel K Kalunga et al. “Online SSVEP-based BCI using Riemannian geometry”. In: *Neurocomputing* 191 (2016), pp. 55–68.
[80] Maher Moakher. “A differential geometric approach to the geometric mean of symmetric positive-definite matrices”. In: *SIAM journal on matrix analysis and applications* 26.3 (2005), pp. 735–747.
[81] Jose Antonio Urigüen and Begoña Garcia-Zapirain. *EEG artifact removal—state-of-the-art and guidelines*. 2015.
[82] Avinash Tandle et al. “Classification of artefacts in EEG signal recordings and EOG artefact removal using EOG subtraction”. In: *Commun Appl Electron* 4 (2016), pp. 12–9.
[83] Mehrdad Fatourechi et al. “EMG and EOG artifacts in brain computer interface systems: A survey”. In: *Clinical neurophysiology* 118.3 (2007), pp. 480–494.
[84] Matthias Feurer and Frank Hutter. *Hyperparameter optimization*. 2019.
[85] Li Yang and Abdallah Shami. *On hyperparameter optimization of machine learning algorithms: Theory and practice*. 2020.
[86] Contributors. *skopt.gp$_{minimize}$*. 2022. URL: https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html#skopt.gp_minimize.
[87] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.
[88] Contributors. *skopt.plots.plot$_{objective}$*. 2022. URL: https://scikit-optimize.github.io/stable/modules/generated/skopt.plots.plot_objective.html#skopt.plots.plot_objective.
[89] Barthélemy Quentin. *Online Artifact Detection with Riemannian Potato Field*. 2022. URL: https://pyriemann.readthedocs.io/en/latest/auto_examples/artifacts/plot_detect_riemannian_potato_field_EEG.html#sphx-glr-auto-examples-artifacts-plot-detect-riemannian-potato-field-eeg-py.
[90] Quentin Barthèlemy and pyRiemann 0.3 Contributors. “pyRiemann”. In: (2022).

[91] Giovanni Grego. *EmbeddedRPF*. 2022. URL: https://github.com/GiovanniGr/embeddedRPF.
[92] Contributors. *Class Covariance*. 2022. URL: https://commons.apache.org/proper/commons-math/javadocs/api-3.6.1/org/apache/commons/math3/stat/correlation/Covariance.html.
[93] Contributors. *PyRiemann/Covariance.py*. 2022. URL: https://github.com/pyRiemann/pyRiemann/blob/master/pyriemann/utils/covariance.py.
[94] Yilun Chen et al. "Shrinkage algorithms for MMSE covariance estimation". In: *IEEE Transactions on Signal Processing* 58.10 (2010), pp. 5016–5029.
[95] Olivier Ledoit and Michael Wolf. "A well-conditioned estimator for large-dimensional covariance matrices". In: *Journal of multivariate analysis* 88.2 (2004), pp. 365–411.
[96] Scikit-learn. *Shrinkage covariance estimation: LedoitWolf vs OAS and max-likelihood*. 2022. URL: https://scikit-learn.org/stable/auto_examples/covariance/plot_covariance_estimation.html#sphx-glr-auto-examples-covariance-plot-covariance-estimation-py.
[97] Scikit-learn. *Ledoit-Wolf vs OAS estimation*. 2022. URL: https://scikit-learn.org/stable/auto_examples/covariance/plot_lw_vs_oas.html#sphx-glr-auto-examples-covariance-plot-lw-vs-oas-py.
[98] BrainCapture. "BC2 - tech specifications". Not Published. 2020.
[99] iBug. *Relationship between CPU Usage and Battery Usage*. 2022. URL: https://android.stackexchange.com/questions/167045/relationship-between-cpu-usage-and-battery-usage.
[100] TensorFlow. *Deploy machine learning models on mobile and edge devices*. 2022. URL: https://www.tensorflow.org/lite.
[101] Zitong Wan et al. "A review on transfer learning in EEG signal analysis". In: *Neurocomputing* 421 (2021), pp. 1–14.
[102] Raghav Subbarao and Peter Meer. "Nonlinear mean shift over Riemannian manifolds". In: *International journal of computer vision* 84.1 (2009), pp. 1–20.
[103] BrainCapture. "Bhutan scans quality assessment". Not Published. 2022.

A BC relevant extract of the document on ”User requirements”

	so that I can perform the *(relates to guidance for the operator of the device and putting the focus on the patient not the equipment, technical issues etc.)*
RR	As a patient I need my personal data to be accessible only to relevant people involved in my diagnosis and treatment like my GP or neurologist because I want to protect my privacy.
	As a patient I need a precise diagnosis so that I can get the correct treatment that will help me live a normal life. *(relates to the quality of EEG recording and patient info given to the EEG reader)*
BR	As a patient I need a low cost EEG scanning and reading process since I have to pay out of pocket if I don't have insurance. (business requirement)

GP/Neurologist requirements = Referring and diagnosing doctor

	As a GP/neurologist I need to be able to refer all my epileptic patients to an EEG scan (neonates are out of scope for BC1) so that I can help all patients to the best of my ability. *(relates to e.g. cap sizes and app UI)*
	As a GP/neurologist I need a quality report from the EEG reader so that I can diagnose the patients and prescribe the right medicine.

EEG operator

	The overall process
	As the EEG operator I need to be able to perform the standard EEG scanning process on patients so that I am in compliance with the medical procedure of scanning for epilepsy.
	As the EEG operator I need to be able to use the BC1 on all patient groups so that I can help them all to get an EEG recording. *(this relates to the cap/electrodes and the app UI (questions, guidance and flexibility)*

	As the EEG operator I need the BC1 to be easy to use so that I don't need special training or expert knowledge to operate it. I need to be able to operate the BC1 just by reading the User Guide.
	As the EEG operator I need the user guide to be easy to understand so that I can quickly learn how to operate the BC1 and start scanning without further training time and costs.
	As an EEG operator I need the device to be small and mobile so that I can take it to the location of the patient since the patient can sometimes be prohibited in coming to the clinic (e.g. sick people, people living in rural areas etc)
	As an EEG operator I need to be able to do EEG scans in rural areas where there is no wifi so patients who live remotely can still get diagnosed and treated.
	As an EEG operator I need to be able to do EEG scans in rural areas where there is no or poor power supply so patients who live remotely can still get diagnosed and treated.
	As an EEG operator I need the BC1 to be robust so that I can bring it to rural areas and so that noncooperative patients don't break it e.g. by pulling the amplifier to the floor by moving around or pulling the cap/electrodes cables.
	As an EEG operator I want the BC1 to look visually appealing and as little "medical" as possible so that it induces trust and calm in patients (e.g. children and patients with mental illnesses)
	Install and setup phase - phone
	As an EEG operator I need to be able to use a phone that is easily accessible for me, so that I don't need to buy a new phone to download and operate the app.
	As an EEG operator I need it to be easy to find and install the app on my phone. (or another person who does it)
	As an EEG operator I need it to be easy to set up the phone and the app (if any initial set up is needed) to start doing EEG recordings so that I don't make mistakes or get stuck in the process.
	Prepare equipment phase
CAP	As an EEG operator I need enough cap sizes to cover the range of my patient's head circumference so that I can do EEG recordings on all my patients.
CAP	As an EEG operator I need it to be quick and easy to identify which cap size is right for my patient so that I don't waste time on picking the wrong cap and stress the patient by trying to fit a wrong sized cap.
CAP	As an EEG operator I need it to be easy to place the cap correctly on the patient's head so that I don't place it askew and get reading from wrong electrode positions.

CAP	As an EEG operator I need it to be intuitive and quick to connect the cap to the amp so that I don’t make mistakes and cannot start an EEG recording.
	As an EEG operator I need it to be easy to place ECG on the patient so that I don’t stress the patient too much with the risk of inducing uncooperativeness.
	As an EEG operator I need it to be intuitive and quick to connect the ECG to the amp so that I don’t make mistakes and cannot start an EEG recording.
	As an EEG operator I need it to be intuitive how to operate the amplifier (turn it on and off, charge it etc.) so that I don’t make mistakes.
	As an EEG operator I need it to be intuitive and easy to connect the app with the amplifier so that I don’t make mistakes and get stuck or connect to the wrong amplifier.
CAP	As an EEG operator I need it to be quick and easy to gel up the electrodes so that the process is as short as possible for the patient and not stress the patient unnecessarily.
CAP	As an EEG operator I need it to be easy to identify if the cap electrodes are connected or not so that I don’t do mistakes like apply more gel and create bridges.
CAP	As an EEG operator I need the cap to be difficult to move or take off once it is placed correctly on the patient's head so that a non-cooperative patient does not manipulate the cap and diminish the quality of the EEG recording.
CAP	As an EEG operator I need the cap to be robust so that my patient doesn't break it during recording by e.g. scratching the head, pulling the cords etc.
	EEG recording phase
	As an EEG operator I need to collect the patient information that the EEG reader needs to interpret the EEG recording.
RA	As an EEG operator I need to tie myself (my ID) to the EEG recording so that I can identify my recordings from other operators' recordings.
	As an EEG operator I need to add patient identification to the EEG recording so that the EEG reader and the diagnosing doctor can identify the right person.
	As an EEG operator I need to be able to identify which recordings I have done so that if I have an interrupted recording that I need to finish later, I don’t open my colleagues recordings and add notes etc. to the wrong recording.
	As an EEG operator I need the app UI to be easy to use so that I don’t make mistakes that could ruin the recording or the interpretation.
	As an EEG operator I need to be well guided through the actual EEG recording process so that I know what to do and when to do it. (relates to the sequence of HV and PS, contraindications etc.)
	As an EEG operator I need it to be clear which EEG recording type I choose (free form or standard EEG), so that I don’t choose the wrong one.

	As an EEG operator I need it to be easy to evaluate how the recording session went and send the right information to the EEG reader so that the EEG reader can interpret the EEG recording correctly.
	As an EEG operator I need to be able to add free notes to the recording in the moment so that I can give any information to the EEG reader that I believe is important. (e.g. patient needed a glass of water, patient was crying etc.)
	As an EEG operator I need to be able to adjust the recording process (in the moment) to fit the individual patient so that I adapt to the physical and mental capabilities of the patient. (app UI needs to be flexible: skip, navigate back and forward, come back, finish etc.)
	As an EEG operator I need to be able to finish the started process (from the pre-interview to the en evaluation) at any point so that I can stop causing any more distress to a non-cooperative.
	As an EEG operator I need it to be easy to identify if an electrode that has already been connected loses its connectivity during recording so that I can take appropriate actions to connect it again.
	As an EEG operator I need to be able to report events that are important information for the EEG reader, so that it is easier for the EEG reader to interpret the EEG recording. (relates to seizures, coughs, movements etc.)
	EEG recordings - sending and handling
	As an EEG operator I need it to be easy to send the EEG recording to the EEG reader so that the diagnosing process can continue as soon as possible.
	As an EEG operator I need to be able to re-send an EEG recording in case of interruption from power cut's or similar so that the diagnosing process can continue as soon as possible.
	As an EEG operator I need to be able to access and send an EEG recording that has been taken in a rural area with no wi/fi or power network at a later stage so that the diagnosing process can continue.
	As an EEG operator I need only the finished recordings to be sent to the EEG reader. Interrupted recordings should be stored on the phone and sent when I have finished them at a later stage or accepted to send an unfinished recording. (e.g. interrupted recordings due to non-cooperative patients where notes need to be added later)
	As an EEG operator I need to be able to free up space on my phone so that I can continue doing EEG recordings.
	Cleaning phase

CAP	As an EEG operator I need it to be easy to take the cap off the patient's head so that it does not give discomfort to the patient.
CAP	As an EEG operator I need the cap/electrodes to be robust so that I don't break them during dismounting from the patient and cleaning.
CAP	As an EEG operator I need it to be easy to disconnect the cap from the amplifier so that I easily can move on to clean the cap.
	As an EEG operator I need it to be easy to disconnect ECG from the amplifier so that I can put my equipment away.
CAP	As an EEG operator I need the cap to be easy to clean, so that I don't risk using a "dirty" cap on a new patient causing the impedances to be high or contaminating the patient.
CAP	As an EEG operator I need to be able to clean the cap/electrodes with water and normal cleaning agents so that I don't need to buy special, expensive cleaning agents that might be hard to get.
CAP	As an EEG operator I need the cap to be easy to hang (or otherwise place) to dry so that it dries fast and in an orderly way.
CAP	As an EEG operator I need the cap to dry fast so that I can quickly use it again with a new patient and do as many scans as needed per day.
CAP	As an EEG operator I need the cap to be robust against the soap agents I use. (relates to the cap mesh etc.)
	As an EEG operator I need it to be easy to clean the ECG so that I can use it right away with another patient.
	As an EEG operator I need to be able to swipe off the amplifier with commonly used cleaning agents/wipes.
	Maintenance
	As an EEG operator I need to be able to see when the amplifier battery needs charging so that I don't start an EEG scan and run out of battery midway interrupting the scan..
	As an EEG operator I need to be able to see when the amplifier is fully changed so that I can stop the charging and use it for EEG scans.
	As an EEG operator I need to be able to see the battery level if I go to rural areas with poor power supply, so that I know if I have enough battery to do the needed EEG scans.
CAP	As an EEG operator I need it to be easy to identify the cap lifespan (electrodes and mesh etc.) so that I don't continue to use a cap that doesn't perform optimally anymore.
	Storage phase

	As an EEG operator I need the device to be easy to store in the clinic so that it doesn't take up too much space, does require special protection or hanging racks etc.
	As an EEG operator I need the individual device to be easy to distinguish from other devices (if I have purchased multiple devices), so that I can easily pick the right one.

FREE RECORDING MODE

	As an EEG operator I need to be able to perform an EEG scan of any length not tied to the standard EEG process or other specific medical processes so that I do a recording for purposes free of choice.
	As an EEG operator I need to be able to collect patient data to the free format EEG so that the EEG reader can interpret the recording correctly.
	As an EEG operator I need to be able to add free notes about the free format EEG recording explaining the purpose of my recording.

The EEG clinic

	As the EEG clinic owner I need the device to be low cost so that I can sell EEG scans at a reasonable price that patients can afford to pay for.
	As the EEG clinic owner I need to be able to use the device with a minimum of extra costs of cotton swabs, syringes, conductive gel etc. so that I can sell EEG scans at a price that patients can afford and or maximize profit.
	As the EEG clinic owner I need to be able to use the device with an easily accessible conductive gel so that I don't run out of it and can't do EEG scans.
	As the EEG clinic I need fast delivery of my new bought BC1 product in order to start helping my patients.
	See comments from Lisbeth….

The neurophysiologist/EEG reader

	As the EEG reader I need to receive high quality EEG recordings so that I can interpret the EEG correctly and send a quality concluding report to the diagnosing doctor. (relates to ease of use, impedance levels, patient information, supporting notes etc.)
	As an EEG reader I need to be able to open the EEG recording (+ patient data) in a program accessible to me so that I can access the data.

	As an EEG reader I need to be able to send my EEG report to the diagnosing doctor so that the patient can get the right diagnosis and treatment.

Information sharing platform - (Cloud)

	EEG reader
UR-XX	As an EEG reader I need to be able to access the cloud with a personal login so that no one else can access the EEG recordings that I am handling.
	As an EEG reader I need to be able to see if there are any new EEG recordings that I need to interpret so that I can get to work and help the patient getting a diagnosis soon.
	As an EEG reader I need to be able to see from which clinic the EEG recording is sent from so that I know who I am collaborating with and can clarify any issues arising.
	As an EEG reader I need to be able to download the EEG recordings to my reading program in order to interpret the EEG.
	As an EEG reader I need to be able to upload my interpretation report and address it to the diagnosing doctor so that the patient gets diagnosed.
	As an EEG reader I need to be able to see old recordings and interpretation reports so that I can get an overview of what I have done, e.g. to check against my payment.
	As an EEG reader I need recordings and interpretation reports to be tied to the individual patient so that I can see how many recordings have been done for each patient and go back and check previous recordings and compare with new recordings for that particular patient.
	As an EEG reader I need to have an overview of the date of the recordings and interpretation reports so that I can easily see the recording history for each patient without having to open the recordings/interpretation reports. *(EEG reader can see e.g. that the last recording was done three years ago etc.)*
	As an EEG reader I need a payment system to be connected to the cloud so that I can get paid for the EEG reports that I have done.
	Diagnosing doctor
	As the diagnosing doctor I need to be able to access the cloud with a personal login so that no one else can access the EEG recordings and reports that are connected to my clinic.
	As the diagnosing doctor I need to be able to see if there are any new EEG reports that I need to read so that I can diagnose the patient and start treatment.
	As the diagnosing doctor I need to be able to see which EEG reader has done the interpretation report so that I know who I am collaborating with and can clarify any

	issues arising.
	As the diagnosing doctor I need to be able to download the EEG report to my computer, open and read it so that I can diagnose the patient and start treatment.
	As the diagnosing doctor I need to be able to find the old EEG recordings and interpretation reports for the individual patients so that I can get an overview of the patient's EEG history. This helps me in diagnosing and evaluating if the medicine is working.
	As the diagnosing doctor I need to have relevant patient and EEG information displayed in the cloud so that I can easily get an overview of the patient's recordings, i.e. patients name or ID, recording date, report date, no. of recordings etc.
	As a diagnosing doctor I would like to be able to attach comments to a patient's profile/EEG reports in the cloud in order to have a quick overview of essential information, like chosen medicine etc.
	As a diagnosing doctor (GP/neurologist) I need to be able to create profiles of my EEG operators in the cloud so that they can be assigned with a login to the mobile app and upload EEG scans.
	As a diagnosing doctor I need to be able to deactivate or delete profiles of my EEG operators in the cloud if they are e.g. no longer employed with me. (this should not delete any EEG recordings made by the EEG operator)
	As the diagnosing doctor I need to be able to delete patients from the cloud system if the person is no longer my patient.
	As a diagnosing doctor I need to be able to delete EEG recordings or reports connected to a patient so that I can clear out fault recordings, very old recordings etc. so that they do not disturb my overview.
	Clinic/admin
	As a clinic I need to be able to create a profile in the cloud for each of my diagnosing doctors (GP/neurologist) so that they have an account that they can log in to and find recordings and reports.
	As a clinic I need to have stats of my diagnosing doctors and their work to have an overview of how well my clinic is doing.
	As a clinic I need to be able to delete my diagnosing doctors from the cloud if e.g. they are no longer employed at my clinic.
	As a clinic I need to be able to delete my profile completely from the cloud if I no longer need the BrainCapture services.

B Already available interview with an EEG doctor

Date: Tue 9th November
Participants: Steve Coates (TeleEEG), Charlotte Stow (TeleEEG), Paul Loomis (BrainCapture), Louise Klok (BrainCapture)

Interview with Dr Coates on EEG reading

Q1: Which patient information is important in order to be able to read an EEG received remote?
Name, age, sex, conscoius state of the patient (sleep, drowsy, awake).
Family history of epilepsy (yes/no?)
How often the patient has seizures.
Nature of seizure. Helps to focus on where we will find a seizure and what sort (helpful in planning the recording/tests or reading the EEG or both?)
Recovery period.
Other neurological problems
Diabetes
Cardiovascular problems (why?)
- Also look at standard guidance.

We get EEG's where performa has been filled out by the person doing the EEG. (In EEG reading programs). Tick-boxes (state of the patient etc.) and free hand (past history, nature of seizure etc.).

Q2: Is there important information about how the recording session went?
Cooperative, non-cooperative, sedated.

A good technician will explicitly say "hyperventilation was performed well for 3 min". Any less than that they will also give an explanation. How well and how long. The same with photo stimulation, they will say how well the patient tolerated it.
Not all do, but they should.
Every EEG should be annotated: Eyes closed, eyes open, patient coughed, patient had a seizure. Sometimes it doesn't happen nearly enough though.

(It could be possible to make something like "comment on the session using these phrases" and then have pre-defines choices)

Q3: What is a standard recording? (Hyper, photo stimulation etc.?)
Routine adult EEG is 20-30 min. Depending on the patient's tolerance. And includes hyperventilation and photo stimulation.
Neonate it's 1 hour.

Q4: When do artefacts become too much to read the EEG?
I would not try to quantify something like that.

We train technicians to comment on artefacts in the EEG. But they don't always. All interpreters can recognize artefacts and tell the difference between artefacts and epileptic signals. There are some gray areas but usually.

Eye movements are not really a problem. Eye lit flitter can sometimes last the whole EEG, but you can still read the EEG. Eye movement artefacts don't matter that much. Interpreter should still be able to read what's in the EEG. I don't know when you would say it's too much.

EMG artefacts (clenching jaws etc.), that's usually only temporary. A Childs EEG is usually full of EMG artefacts and all you can see is small gaps in between of background activity. So to specify, now it's too much, it's hard to do.

In rare cases we have to decline reading an EEG because there's not enough normal background to pick out. If electrode impedance is not down there is often a lot of noise.
Moving artifacts are usually temp, especially in adults. Eye movement artefact we can see, they are not a problem.

An experienced reader can look past the artefacts to the background.

Idea: Electrode impedance artefacts, EMG and movement are all high freq. maybe look at that with the SW. If there is too much in this period of time reject the EEG, or if it's too much within the epox prompt the technician.

Q5: Initial test and checking montages in the beginning of a recording, what is the technician checking for?
Not all clinics do this in their protocol. Most don't. They just get on with the recording.

Usually good practice to change montages during recording to look at various areas of the scalp. But a lot of EEG machines don't have this capability so they don't do that.

Montages: ref and bipolar.
Bipolar: see if there is a focus of epilepsy activity and where it is.
Ref: used to size and responses. You might pick up an epileptic signal that is small in Bipolar montage but big in referential montage. But you don't know where it is.

There is no fixed protocol to change montages. Because the interpreter can always change the montages.
A reason for having a protocol for checking various montages in the beginning could be to check that the electrodes are in the right area. But you would expect from a technician that the electrodes are in the right place.

In the UK we don't do this procedure of checking montages in the beginning of a recording.

***END**

C Already available interview with an EEG technician

Date: Wed 10th November 2021
Participants: Anne Clarke (TeleEEG), Charlotte Stow (TeleEEG), Louise Klok (BrainCapture)

Interview with Anne Clarke (EEG technician) on EEG reading

Q1: Which patient information is important in order to be able to read an EEG received remote?
Basic demographics: Age, clinical history as to why they want the EEG, description of the events, last known event - if 1-2 hours ago the EEG will be affected by it.

Button is needed for annotating if a patient seems to have a seizure.

Interpreter would like to know why we are seeing the patient? When did they have seizures, how often, stiffness in limbs, vomiting.

Q2: Is there important information about how the recording session went?
If it's a child, were they asleep, awake, sedated.
Certain medications will influence the EEG. Could it be done by a drop down menu? Yes, there are probably 2-3 meds you could mention and then "other".

Children: It's kind of obvious in the EEG when they are moving around. It would be good to have annotations that could be pressed saying "movement", "restless". Some artefacts can be awkward and could make an interpreter uncertain if it's a genuine seizure or an artefact.

Eye closure in a young child is in the lap of the Gods. Hard to get a child to calm down. Let's get a readable EEG rather than worry about the details on what they are doing.

Q3: What is a standard recording? (Hyper, photo stimulation etc.?)
Standard: Hyperventilation for 3 minutes. If less <3min it is noted/annotated. Patient could have an absent seizure and then not respond. Annotation: "not responding".

Hyperventilation is done not far into the recording. Not towards the end. Because after hyper you are more likely to see abnormalities. So hyper after maybe the first 5 min.

We normally do photic towards the end.

Does it make sense to have a button to start/stop any kind of other provocations? Yes, sometimes you might do a second hyperventilation, or (reflect epilepsy) doing crosswords etc.

Q4: When do artefacts become too much to read the EEG?
The important thing is that you can measure the impedance. That is a huge way of reducing a lot of artefacts. Often it is poor contact that will lead to lots of artefacts.
Secondly is movements.

In children moving around you will get it (artefacts), so the person recording needs to be trained to calm children down. Distract them.

It's a common thing to ask a patient to relax their jaws or stop chewing. Because you get bursts of EMG activity going through.

When do you know an artefact is too much? It's just something you recognize from seeing the EEG. I suppose it's when you can't detect certain frequencies over certain channels.
If you have to put a limit on it, let's say 2 seconds.

Q5: Initial test and checking montages in the beginning of a recording, what is the technician checking for?
They are doing that to generate artefacts to be able to recognize the artefacts in the patient. I tend to do that sometimes with patients doing 24 hours home recording because you can't see the patient. So when they come back you can recognize the artefacts because certain artefacts can mimic epileptic signals so it's only when you recognize it by eye that you know what you are looking at.

Gel bridges are quite obvious. Then you just have a flat line.

It's not like the old days when it was on paper then you had to change montages. But you don't have to do that now when it is all digital.

END

D Further interviews with EEG technicians

Date: [redacted]
Participants: [redacted] (Filadelfia Hospital), [redacted] (Filadelfia Hospital), [redacted] (Filadelfia Hospital), Giovanni Grego (BrainCapture)

Interview with experienced technicians on EEG quality assessment

Q1: Which are the artifacts that most affect the reading?

Participant 1:
Answer:

- Eye movement. Some patients with closed eyes can't relax their eyes (they move from side to side, something they can't always control), This can impact the EEG special in the frontal region, but can usually also effect the other channels too. So what we do in this case, is that we take a small sand bag, made for the eyes, and we lay it over the eyes, this usually relaxes the eyes so we don't see artefacts.
- Muscle artefact from the jaw and face. We see this a lot, when the patients do not relax their faces. Most people are tense in their face without even knowing, so I will normally tell my patients (in the beginning of the scan) to relax their faces and maybe to open their mouth just a little bit so that their teeth do not touches.
- Movement in general. Example talking, moving their head and body.
- 50 Hz noise. Usually from devises that are connected to electric power for example the socket (the electricity output from the wall, I'm not sure of the name) and the patient hold the devise ore is near it.

Participant 2:
Answer… Muscle artefacts.

Participant 3:
Movement artifacts, especially in small children, cognitive handicapped adults

Q2: What do you usually keep under control when you take a scan? What do you look at on the waveforms?

Participant 1:
Answer:

- Right placement of the electrodes.
- Knowing our equipment works (no broken electrodes ore cables, and so on).
- Impedance are fine.
- Telling the patient to relax. If possible I make the patient relax in a bed without talking, telling them to relax all muscles in their face.
- No devises or some kind of electricity near the patient. Don't wanna have 50 hz noise.

- Closed eyes. During the EEG scan, I want the patient to have closed eyes as much as possible. Because the EEG is so much more readable and many rhythms are only shown with closed eyes.

What do you look at on the waveforms? Answer:

- "PDR" (posterior dominant rhythm - an Alfa rhythm over the occipital region (usually O1 and O2).
- Eye blinking, both open and closed eyes.
- Epileptic pattern
- Normal non epileptic pattern.

Participant 2:
Answer…Depends on the age and how the patients "normal" background looks.

Participant 3:
I suspect artifacts (motoric) especiallz when there is a extreme high amplitude, very sharp-waveforms

Other artifacts looking at the ECG bar/video

Q3: What do you do if something goes wrong, if you see there is a lot of noise?

Participant 1:
Answer:

- Check impedances. If they are bad, I will jell up and scrub, to be sure they are fine.
- Ask the patient to relax and lay/sit still. Maybe they are moving or tense in their face.
- Check the electrode is the impedance still bad maybe it's because its broken, and need to be changed.
- Check the head box. Is the electrodes pressed in right, no gab ore poor connection.
- Check the cables. Are they too twisted or are they well connected.
- 50 Hz noise. Check if there is any devises ore some kind of cable/power near the patient.
- Eye movement. As mentioned before some patients can have a lot of eye movement, in that cases we have a small sand bag we can lay over the patient's eyes.
- Protocol and montage. Check if the protocol and montages is right, did anything change by mistake.

Participant 2:
Answer… Try to find out why. Is there good impedanse. Is the any loose wires ore chargingkabels near by.

Participant 3:
Order a new EEG, evt. with sleep and melatonin
Try to describe the quite parts of the EEG and conclude only with insecurity due to noise

Q4: What statistics and information would you like to have to understand how the quality of the scan is, if you cannot have access to the waveforms?
(Examples if no ideas: percentage of quality in real-time, percentage of the quality in the last minutes, visualization of the quality in real-time (colors), where the problems come from, progression bar, visual artifacts notes…)

Participant 1:
Answer:

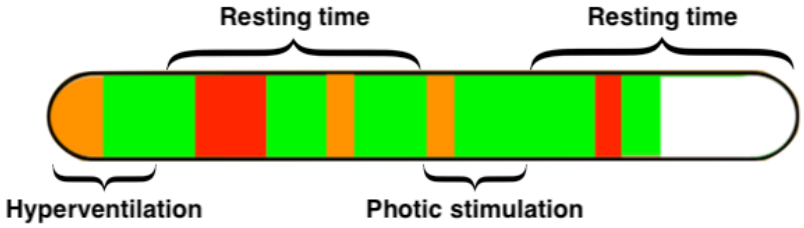
- I would like to know if the impedances are fine.
- It would also be nice to know if there is a lot of "data-noise" in "real time". So I could maybe go through a check list to make sure I get the best scan as possible. Maybe something like the steps I mentions above in Q3.

Participant 2:
Answer…I don't understand the question.

Participant 3:
where the problems come from > Yes this would be nice

visual artifacts notes > Yes please

Q5: Would a progression bar to see the quality of data over time be useful? (E.g. I see there is a lot of red in one section and I decide to prolong that section for having better data)



Participant 1:
Answer:

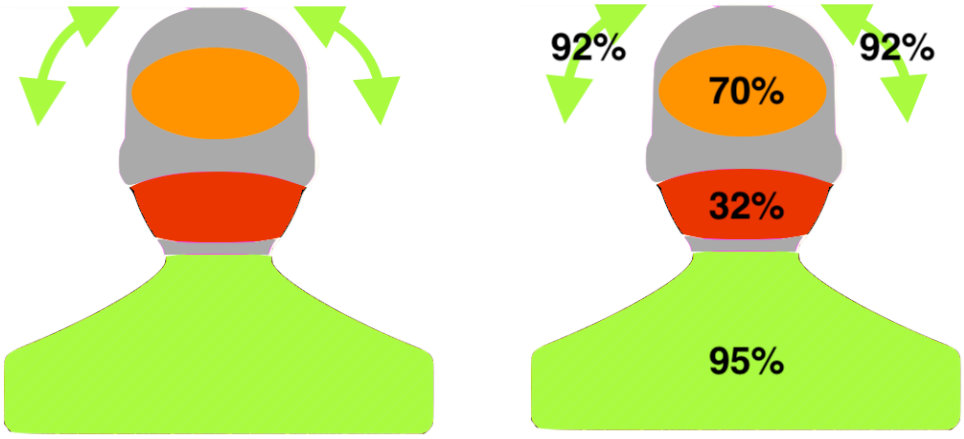
- Yes, I actually think this would be a great idea if you can't read the EEG! Sometimes we will also prolong the scan if we know the scan have been affected with noise, artifact and so on.

Participant 2:
Answer… I don't understand the question.

Participant 3:
Answer…
Yes, this would be nice, but most of the time our recordings are really good
and otherwise if the patient is moving a lot most of the time > the whole EEG i disturbed

It would not be necessary for a std EEG = 30min because I can screen the EEG fast
Would help for recordings with more 1 hour

Q6: Would a half-bust with colored sections be useful? To see, at the moment, if there are artifacts generated by which parts of the body.



Participant 1:
Answer:

- Hmm I don't know if I would need it.

Participant 2:
Answer… Maybe it could help. But I depends why there is artefacts.

Participant 3:
No, would give me extra information, since I am experienced in spotting and locate artifacts

Q7: Would a notification when the data is of bad quality be useful? If yes, do you think an audio alarm could be a problem for the patient?

Participant 1:
Answer:

- Yes, if it was me running a scan without EEG I would like to be informed if the scan is suddenly doing bad. But is the notification to the patient ore the staff? Because I think the staff should be there to help and assist the patient on what to do if an alarm sound goes on.
  About an audio alarm, I think it could work well if the patient beforehand is explained about the sound what it means and what to do if they hear it and maybe with some help fron the staff. I also think the alarm should be gentle and not “choking” ore with a lot of volume.

Participant 2:
Answer…That might be use full. The most important thing is good impedance

Participant 3:
During recording this would be nice, if the technician is as experienced as in our center it is not necessary
An audio alarm would disturb th recording and stress most of the patients even more

Q8: Would a notification when the patient is falling asleep useful? If yes, how do you understand from the data that the participant is sleeping?

Participant 1:
Answer:

- First, I wanna mention that its not bad if a patient falls asleep. It’s actually something we really often wanna see in the proses of finding out if the patient have epilepsy or not. Example when we get a new patient to Filadelfia we will do 2 scans, one “standard scan” where the patient is awake, and one “sleep scan” where the patient is sleeping. At other hospitals, I know they often tell the patient to try sleep in the end of a “standard scan”. The reason for this is that 20-30 % of people with epilepsy will only have seizures during sleep.
- But a notification when the patient falls asleep would be a great idea, because sometimes you can also have very sleepy patients there example will fall asleep during hyperventilation or other tasks we want them to do. Sometimes, we must remind the patient to stay awake.
  Another thought on why it could be a good idea, would be to maybe prolong the scan a

bit? Because who knows maybe that patient is one of those who only shows epileptic pattern during sleep. Just a thought.

- How do I know if the patient is sleeping by looking at data.
  We look for different sleep patterns, there will tell us if the patient is sleeping or not.
  So people are sleeping in cycles, we have 4 sleep stages in one cycle and they repeat them self again and again.
  For each stage, we see different sleep pattern on the EEG. Example the first sign of sleep will be that the amplitude of the waves over O1 and O2 (that we call PDR (Posterior dominant rhythm) will be flat. You cannot see any waves, just a flat line. At the same time the eyes will begin flutter from side to side (lateral). It shows a very clear pattern in the EEG (look like eye flutter artefacts in the frontal and pre temporal regen).
  Also over CZ you will see something we call “vertex sharp waves” it’s a normal sleep pattern, where you will see a delta wave with a very high amplitude and often look very sharp.
  These are the first 3 signs that a patient is sleeping. Throughout the cycle, you will see many other different sleep patterns that will change during each stage.

Participant 2:
Answer…There will be Wertex in central midtline.

Participant 3:
No technician/me should be able to spot this
interupted/disappeared backgound/diffuse slowing consistent for more than one epoche > pointing towards N1
Not always difficult to tell since many patients in stnadard EEG tend to shift between drowsiness/awake state

Q9: Would a notification when the data is continuously really noisy or when the data is constantly null be useful? (For example when there is a false positive in the impedance reading due to the hairs.)

Participant 1:
Answer:

- Yes I think that would be a great idea. Like I answer in Q7, some kind of notification would be nice, so you can be sure the scan will be as good as possible.
  I think what I would like to have doing a scan without the EEG, would first be some kind of *check list before the scan begin*, to make sure my starting point is right. Then I would like *some kind of visual “real time” picture of the impedances maybe in red/green color*, so I always know the connection is fine. Then I think the idea of *a progression bar* would be nice, also with colors. Where I can see if the date have been too noisy and maybe if it

could show me if the patient is sleeping could be cool too.
Then some kind of *notification if the data is bad* will be very useful to, in that case you can have some kind of “check list” ready, to go throw to solve the problem.

Participant 2:
Answer… I don’t understand the question.

Participant 3:
I am not sure if you mean during recording or describing
During recording and describing af visual signal would be useful
Not an audio signal, see answer Q7

E Further interviews with non-expert operators

Interview on EEG quality assessment with non-experts

Q1: What do you think is the quality of the data you took? How much a doctor can understand from the data you took?

Answer… (From 1 to 6, where 1 is very bad, and 6 is very good)

Participant 1:
5
All electrodes were green. The patient was relaxed. I annotated the artifacts.

Participant 2:
4
We were in a noisy environment, we don't have details but just the big picture of what was happening.

Participant 3:
5
Some electrodes were yellow. The patient didn't move.

Q2: How sure are you about the previous answer? Why? What went wrong or good?

Answer… (From 1 to 6, where 1 is "I answered randomly", and 6 is "I'm very confident")

Participant 1:
3
a- I am not sure. I just followed the points.
b - Nothing, everything worked smoothly.

Participant 2:
2
a - Some things were assumed to be correct (impedance, gelling…) but no other information is given, I do not know about EEG.
b - Nothing went wrong. It was quick, all green things fast. It was not easy to relax if someone tells you to relax, because relaxing is abstract and personal.

Participant 3:
1
a- I have no idea of the data. I did not have any information about it. The user didn't move but some electrodes were yellow. I don't know how much it can affect.
b- Some electrodes were yellow

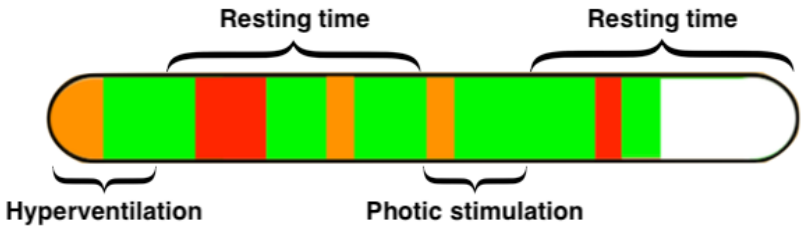
Q3: What would give you a better understanding of the quality of the recording? (E.g. someone tells you the final quality; or someone tells you each time something is going wrong; or every time you want, you can ask for the quality in that moment; other ideas…)

Answer…
Participant 1:
At the end (or also during the scan) a doctor tells me how good the data is.

Participant 2:
A professional tells me what the quality is.
A list of important priorities.

Participant 3:
Yes, if someone tells me the quality of course I can know how good the scan was. No I don't want any other information.

Q4: Knowing that the important aspect of a scan is to collect 20 minutes of good quality data, do you think it would help to have a bar that tells you how long it is of good quality? Why?
(Note: Green means: in this period the quality was good; yellow means: quality was enough; red means that the quality was not enough)

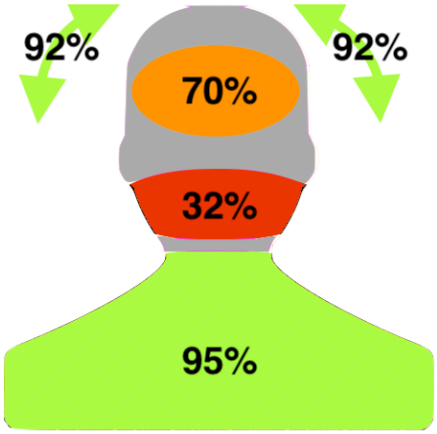


Answer…
Participant 1:
Yes, perfect! With a clock that says how much time I miss.

Participant 2:
Yes, it would be very nice.
Do we need something more? No.

Participant 3:
Yes with this I would easily assess the quality, and understand if I have to take more data

Q4: Would help you to have a visualization of why the quality is bad and from where the problems come, like in the example below?
(Note: Green means: in this area the user is not creating artifacts; yellow means: quality is enough; red means that the quality is not enough because of artifacts created in this area)



Answer…
Participant 1:
I think it is great. I can understand where is the problem from

Participant 2:
No I think it is confusing, I do not understand. As said before I think the bar is enough

Participant 3:
Yes it can help but I would prefer to follow the list without this.

F Protocol used for data recording

Protocol for Filadelfia Scans

- Set the environment up (connections of the electrodes to Natus, splitter…);
- Make the participant sign the consent form;
- Explain the procedure:
  - lying on the bed for the resting state;
  - artefacts (eye movements, head movements) plus arithmetic exercise;
  - electrode pops by touching the head;
- Gel the participant up;
- Start the recording: name the file “FIL_XX.edf” with “XX” the user_id given by BrainCapture;
- Protocol:

Task	Starting time	Duration	Notes
Resting State - Eyes closed, just lie and relax	0:30	10 min	At the beginning of the task, open and close the eyes slowly for 5 times.
Eyes Open	10:30	1 min	Keep the gaze on a fixed point
Eyes Blink	11:30	30 sec	Don’t force the blink, just open/close without forcing
Lateral Eye movements	12:30	30 sec	Keep the head straight and move only the eyes
Lateral Head Movement	13:30	30 sec	Keep the body straight. Move the eyes with the head
Chewing	14:30	30 sec	Without clenching too hard
Jaw Clenching	15:30	30 sec	Keep teeth in contact and clench/release the jaw
Shivering	16:30	30 sec	Make many and

			small fast movements with the body
Eyes opened	17:30	2 min	
Eyes Blink	20:30	30 sec	Don’t force the blink, just open/close without forcing
Lateral Eye movements	21:30	30 sec	Keep the head straight and move only the eyes
Lateral Head Movement	22:30	30 sec	Keep the body straight. Move the eyes with the head
Chewing	23:30	30 sec	Without clenching too hard
Jaw Clenching	24:30	30 sec	Keep teeth in contact and clench/release the jaw
Shivering	25:30	30 sec	Make many and small fast movements with the body
Eyes closed	26:30	2 min	In the meanwhile, explain the next exercise
Mental arithmetic*	28:30	2 min	Perform a focusing task. Keep your eyes closed.

Electrode pops**	30:30	4 min	Touch/move/remove some electrodes.

*In the mental arithmetic task, the participant is asked to count down mentally, starting from 200, by steps of 7; (i.e. 200, 193, 186, 179, 172, 165, 158…) and to notify the technician when he/she finishes. If the track is lost, start again from 200. Stop the participant after 2 min. N.B. the participant has to count mentally and with the eyes closed, without speaking.

**Electrode pop task consists in touching or moving or removing an electrode to study certain types of noise in the channels. In particular: gently move/touch in order “FP1”, “P3” and “P8” each for 5 sec, then wait for 1 min; then pull/remove in order “F4” and “T7” and wait for 1 min. Finally, touch “REF” for 5 sec, wait 1 min, then pull/remove “REF” and wait for 1 min.

Save the file as “FIL_XX.edf” where “XX” is the user_id given by BrainCapture.