Open University of Cyprus

Faculty of Pure and Applied Sciences

Postgraduate (Master's) Programme of *Cognitive Systems*

Postgraduate (Master's) Dissertation

Adaptive e-Learning EEG-based Environment

Amalia Chrysanthakopoulou

Supervisor
Foivos Mylonas

December 2022

Open University of Cyprus

Faculty of Pure and Applied Sciences

Postgraduate (Master's) Programme of *Cognitive Systems*

Postgraduate (Master's) Dissertation

Adaptive e-Learning EEG-based Environment

Amalia Chrysanthakopoulou

Supervisor
Foivos Mylonas

The present Postgraduate (Master's) Dissertation was submitted in partial fulfilment of the requirements for the postgraduate degree in Cognitive Systems Faculty of Pure and Applied Sciences of the Open University of Cyprus.

December 2022

BLANK PAGE

Summary

The present dissertation goal was to research the development of an adaptive e-learning web application environment based on the user's mental states and provide a positive user experience. The mental state that was selected was that of relaxation and concentration. These states will be calculated using a low-cost EEG device that the user will wear throughout his/her interaction with the environment. These mental states will correspond to the *Available* and *Busy* statuses of the system and will be changed automatically by the system to match with the detected mental states. Also, these status changes will also enable/disable system Notifications accordingly. Then a system evaluation will be deployed using 10 participants. The evaluation showed that such a system would indeed have a positive user experience.

Keywords: adaptive and interactive systems, e-learning, EEG, machine learning, brain-computer interfaces, mental states, relaxation, concentration.

Περίληψη

Η παρούσα διπλωματική εργασία είχε ως στόχο να ερευνήσει την ανάπτυξη ενός adaptive e-learning web περιβάλλοντος βασιζόμενο στην νοητική κατάσταση του χρήστη και να δώσει ένα θετικό user experience. Από τις διάφορες νοητικές καταστάσεις επιλέχθηκαν αυτές της χαλάρωσης και της συγκέντρωσης. Αυτές οι καταστάσεις υπολογίζονται χρησιμοποιώντας μια συσκευή ηλεκτροεγκεφαλογράφου χαμηλού κόστους όπου θα φοράει ο χρήστης καθ' όλη του την αλληλεπίδραση με το σύστημα. Αυτές οι νοητικές καταστάσεις ανταποκρίνονται στα αντίστοιχα statuses *Available* και *Busy* του συστήματος και θα αλλάζουν αυτόματα από το σύστημα έτσι ώστε να αντιστοιχούν με τις νοητικές καταστάσεις του χρήση όπου υπολογίστηκαν προηγουμένως. Ακόμα, αυτή η εναλλαγή του status θα ενεργοποιεί/απενεργοποιεί και τις ειδοποιήσεις του συστήματος. Στην συνέχεια, θα πραγματοποιηθεί η αξιολόγηση του συστήματος από 10 χρήστες. Αυτή η αξιολόγηση έδειξε ότι ένα τέτοιο σύστημα μπορεί όντως να δώσει ένα θετικό user experience στον χρήστη.

Acknowledgments

I would like to thank the Slack community of NeuroTechX and Brainflow that helped me out in some technical questions. Also, I would like to thank my partner and my friends for supporting me and tolerating me throughout the whole year that I was emerged in this dissertation. Last but not least, I would like to thank my supervisor for accepting this topic and for our collaboration.

Contents

Chapter 1
Introduction

E-learning systems are now gaining a huge popularity due to COVID-19 era where everything is forced to go online. Traditionally, e-learning systems were designed using the so called 'one-size-fits-all' approach. The learning content and system functionality was the same for all students. This is problematic for the learning process though: every student has individual differences and needs. With the rise of adaptive systems, the learning content, presentation, navigation, or system functionality can now be adapted based on the users' needs or preferences and increase the learning outcomes (Hammad et al., 2018).

Creating and maintaining a user model is necessary for the system to adapt and can be accomplished explicitly, or implicitly (Papatheocharous et al., 2014). Learning is a cognitive process that relies on the student's current internal states like emotions, mental engagement, mental workload etc. Mental engagement is the level of someone's alertness and mental workload is the mental effort that is put on a task (Chaouachi & Frasson, 2012). Electroencephalography (EEG) appears to be the most accurate physiological method to estimate someone's mental state, with a temporal resolution of milliseconds (Chaouachi et al., 2015). As a result, there is now a growing number of studies that have used EEG in adaptive learning environments in order to improve user experience, performance, or both.

1.1 Purpose and necessity of the research

EEG is a non-invasive method to detect the electrical activity of the brain when neurons fire in synchrony (Light et al., 2010). EEG devices used to be cumbersome in the past and were used for medical or research purposes, with a difficult setup and high cost. Nowadays, these devices are getting cheaper and more accessible. These are even used commercially for games and neurofeedback. Still, no EEG device is used in any actual e-learning system.

The purpose of this research is to create an adaptive e-learning web application environment that adapts its functionality based on the learner's mental state and provide a positive user experience. Mental state is an umbrella term for: attention/engagement, cognitive workload, affective state, mental fatigue, emotions (Gerjets et al., 2014), relaxation, concentration etc (Frasson & Chalfoun, 2010). The mental state of the user will be calculated from EEG physiological data using an EEG device that he/she will be wearing while using the system. The specific adaptation that was chosen will be discussed. Then, an evaluation of the system will be deployed.

As mentioned before, due to COVID-19 outbreak, most schools and universities were forced to deploy online lessons. This sudden and huge step towards online learning makes it imperative to keep students motivated and maximize their learning performance or/and user experience when using these online platforms to support their learning.

With this research we also aim to contribute to the few relevant studies available for real-time adaptations in e-learning systems using EEG. We noticed that most research papers were not focusing on web applications architecture. Also, we found many EEG classification studies but only few of them used these classifiers in a realistic environment with adaptations. Also, all the problems that were faced at the design/development process will be discussed. With this we wish to make future implementations of such systems less intimidating and easier. Finally, with the use of a low-cost and commercially available EEG device in this system we make a step forward to make these systems more accessible in the future.

The structure of this dissertation is as follows: First, the interdisciplinary background is discussed: brain fundamentals, EEG, Brain Computer Interfaces, adaptive systems, e-learning systems, machine learning etc. Then, the architecture and the components of the created system is presented. Last, the evaluation of the system is presented and discussed leading to the conclusion of the dissertation.

Chapter 2
Background

2.1 Brain Fundamentals

To better understand EEG a brief introduction to the electrophysiology of the human brain will be presented. The human brain consists of three main parts: the cerebrum, the cerebellum and the brainstem. The cerebrum is the area that mainly controls high-level functions such as complex thinking. The cerebellum is responsible for balance and muscles coordination while brainstem controls involuntary functions such as breathing, heart and hormone regulation (Sanei & Chambers, 2007).

The cerebrum can be further divided into four main lobes, each one with a different main function. The frontal lobe for example is responsible for problem solving, emotions, movement, and speech. The parietal lobe is also involved in problem solving but also in pain and taste. The temporal lobe is responsible for hearing and memory, and finally the occipital lobe for seeing (Kamel & Malik, 2014). Also, the cerebral cortex is a convoluted surface layer of neural tissue at the cerebrum (Sanei & Chambers, 2007). We will see later on that the location plays an important role in EEG tasks.

Inside these structures around 100 billion of nerve cells called neurons are contained. Each neuron consists of its cell body, an axon that relays information to other neurons and dendrites that receive information from other neurons. Each neuron is connected to around 10.000 other neurons through their dendrites and pass information to other neurons in an area called the synapse using neurotransmitters (Sanei & Chambers, 2007).

The neurons have electrochemical properties and at their resting state they are polarized at -70mV, called the resting potential or membrane potential. If the membrane potential exceeds a threshold of -55mV then an action potential is triggered through the axon lasting approximately 1ms. Neurotransmitters are released at the synapse making the next neuron to depolarize (excitatory post-synaptic potential, EPSP) and the extracellular space to be more negative or hyperpolarize (inhibitory post-synaptic potential, IPSP). IPSPs and EPSPs are summed up temporally making the membrane potential more or less likely to create an action potential (Kirschstein & Köhling, 2009; Tivadar & Murray, 2019).

2.2 Measurement

As already discussed, Electroencephalography (EEG) is a non-invasive method for measuring the electrical activity of the brain when large population of neurons fire in synchrony. This activity is detected by electrodes placed on the scalp (Light et al., 2010). EEG was first used in humans by Hans Berger in 1920 and after some years it was mainly used for neurological condition or brain function assessments such as seizures (Kamel & Malik, 2014). The recording of the brain activity is called an electroencephalogram which is also abbreviated as "EEG" (Li et al., 2020).

There are some conditions that must be met so that EEG can be successfully measured. First, a large population of neurons must fire in synchrony and more specifically around 10.000 to 50.000 neurons (Li et al., 2020). Second, these neurons must be close enough to the scalp so that their activity can be grasped by the electrodes. Last, only a specific orientation and type of neurons can be measured and specifically parallel pyramidal neurons, otherwise their activity will be canceled out. This means that mainly the activity of the cerebral cortex can be measured that has all the above characteristics (Sanei & Chambers, 2007). Nonetheless, not all electrical neuronal activities can be detected by EEG. Due to the nature of pyramidal neurons, only EPSPs and IPSPs create a dipole that can be measured from the scalp when post-synaptic potential is summed up (Jackson & Bolger, 2014; Kirschstein & Köhling, 2009; Tivadar & Murray, 2019). Another reason is due to their longer duration than that of action potentials (Kirschstein &

Köhling, 2009). This activity varies with time, and it can be measured almost instantly right after occurring. Hence, EEG has an excellent temporal resolution (Hajare & Kadam, 2021). On the other hand, we cannot safely distinguish the neural location of the signal because the activity is a sum of all EPSPs and IPSPs from many different sources, and their dipole influences the charge in many different directions. Thus, EEG has a very low spatial resolution, regardless of the number of electrodes used (Xia & Hu, 2019).

Regarding this activity, there are two types of EEG activity: The spontaneous activity of the brain, called spontaneous EEG and evoked potentials (EPs) where the brain activity is associated with a specific event (psychological or physical) (Lu & Hu, 2019). Steady-state visual evoked potentials (SSVEPs) and P300 are two very popular examples of EPs. SSVEPs is the brain activity measured at the occipital lobe when a visual stimulus like a flashing light is repeating itself at a specific frequency. The P300 is a positive potential detected 300ms after an odd stimulus is presented among regular ones (oddball paradigm) (Portillo-Lara et al., 2021).

In all the above cases, when an event occurs there are small voltage differences in the EEG signal that can be extracted by repeating the stimulus in a precise time-locked manner, and averaging all the trials (Teplan, 2002). This technique is very popular among researchers, and it is called Event Related Potentials (ERPs) (Lu & Hu, 2019). In this dissertation we will be focusing on spontaneous EEG as this is more relative to grasping a user's mental state.

2.3 Sensors

There are several items that are needed to measure EEG. A set of electrodes to grasp the neuronal electrical activity from the scalp, an analog-to-digital (A/D) converter that converts the analog signal to a digital one, an amplifier to magnify the signal so that it can be successfully digitized by the A/D converter and a recording unit / PC to store this information (Teplan, 2002).

So, using the most minimal setting, three electrodes are needed for the measurement: A signal (or *active*) electrode, a reference electrode, and a ground electrode. The actual activity is the difference in potential between the active and the reference electrode, that's why the reference electrode should be positioned at a place with 0 voltage, even though this is practically impossible. Finally, the ground electrode is needed for power line noise removal (Xia & Hu, 2019).

Luckily, there are many differences between modern EEG devices and older ones. EEG devices do not longer require long electrode wires that must be connected to the recording devices. Also, these devices used to be separate devices that the user had to keep somewhere. Today, the EEG devices are small enough to be worn directly in the head, have integrated electrodes without external wires and send the signal wirelessly to nearby devices real-time. This is highly important as the signal quality is compromised from the length of the wire and even more from the wired being moved or tangled (Casson, 2019).

Usually, a conductive substance, such as a gel, is applied between the electrode and the scalp to reduce the contact resistance (impedance) between skin-electrode and to improve the signal quality. These electrodes that require gel are called *wet* electrodes, they are usually made from silver (Ag) and silver chloride (AgCl) and have a shape of a cup / disc. Wet electrodes have the following disadvantages: Gel application requires skin preparation (skin cleaning) that is time consuming (di Flumeri et al., 2019). This procedure may cause skin irritation, infection (Teplan, 2002) or allergies (Hajare & Kadam, 2021) and it is generally uncomfortable as these products are sticky. Moreover, if the gel is spread to nearby electrodes, it may cause them to short. Also, when time passes by the gel may dry up, causing impedance instability that can influence the recordings. Lastly, impedance must be checked to be under certain values that also takes time (di Flumeri et al., 2019). Even after the EEG recording is finished, the gel removal is difficult, time consuming and usually requires hair wash (Zander et al., 2011). Obviously, the application of this conductive gel must be made for each electrode that can be up to 256 electrodes. Fortunately, a much smaller number of electrodes is generally preferred (Xia & Hu, 2019). As a result, a specialist is needed to take care of the above steps (Casson, 2019). Regardless of all the drawbacks mentioned, wet electrodes are considered the gold-standard in clinical or research settings because they provide high signal quality (Portillo-Lara et al., 2021).

Instead, one can use *dry* electrodes which are directly contacted with the skin, with no conductive substances in between. As a result, the drawbacks related to the gel do not longer apply. In this case, the time needed for the electrode setup is reduced significantly. At hairy parts, the contact between the electrodes and the skin is ensured by the pins of the electrode that can successfully sit behind the hair (Casson, 2019). Therefore, no hair wash is needed after EEG usage and no irritation or allergy is caused by this procedure. All of the above make dry electrodes more appropriate for usage outside of the lab, in real-world scenarios where the user could use the device with no technical help.

There are times, however, that the EEG recording is contaminated, and the signal does not come from the brain. Usually, this signal has a higher amplitude and a different shape than that of a normal brain signal. This contamination is called artifacts (Teplan, 2002) and someone must be very careful about accidentally taking into account artifacts at his/her further analysis. There are two types of artifacts: physiological and non-physiological. Physiological artifacts may be caused by the heart pulse, by breathing, sweating, blinking or from eye movements in general, by muscle contractions like movement in general or tongue movement, talking or chewing (Louis et al., 2016). Non-physiological artifacts may be caused by AC power line noise, low battery of the EEG device, electrode or wire movements, wire connectivity issues or shortages, excess quantity or drying of the paste or gel. Detecting these artifacts can be accomplished manually (offline) by the researcher or automatically (online) by the system. Finally, using specific electrodes for detecting eye or muscle movement and cardiac pulse further helps detecting these physiological artifacts (Teplan, 2002).

At the clinical and research settings a standardized position of electrodes is required. For this reason, a system for electrode placement was proposed in 1947 by H.H. Jasper. More specifically, this system uses 10% or 20% interelectrode distance from standard skull landmarks such as nasion-inion (anterior-posterior plane) and left-right preauricular points (lateral plane). Also, depending on the lobes that they correspond to these positions are prefixed with: *Fp* (Fronto-polar), *F* (Frontal), *P* (Parietal), *O* (Occipital), *T* (Temporal). Additionally, there is also the *C* (Central) point that is named after the central sulcus. These names are followed by a number. Odd numbers are used for left hemisphere and even numbers for the right hemisphere. For example, O1 means the first electrode position of the occipital area at the left hemisphere. This system originally used 19 points with 2 extra points at the earlobes A1 and A2 where *A* is taken from the word *auricular* (Klem et al., 1999).

Finally, there are also two alternative methods for measuring brain activity that unlike EEG, are invasive ones. They both require a surgical procedure to place electrodes/array of electrodes inside or above the cortex itself. The latter is called Electrocorticogram (ECoG). Due to their close distance to the signal source, their signal quality is much higher compared to that of EEG and they even have the potential to measure the activity of a single neuron. Therefore, they present an excellent spatial and temporal resolution, but at the same time they have some serious drawbacks. Apart from the surgical procedure itself, problems may arise involving foreign body reaction / infection, stability of the electrodes, small area covered and inability to move electrodes to another area. For this reason, they are only used by people with epilepsy, tetraplegia, or other disabilities (Abdulkader et al., 2015).

2.4 EEG Devices

There are many available EEG devices with a varying number, type and placement of electrodes, sampling rate, recommended application and price. Additionally, some devices also provide an extra set of sensors that measure heart rate activity, muscle activity or eye movements (Soufineyestani et al., 2020). Low-cost devices that use 1 to 4 channels include: NeuroSky MindWave Mobile 2, Muse 2 and MyndPlay Myndband, all of which have dry electrodes and cost under $300. These devices are commercial in the field of neurofeedback. Due to their type of electrodes these devices have a quick setup and amongst them, Muse is the most popular choice. In contrast, expensive devices include: Emotiv Epoc X, OpenBCI, Neuroelectrics Enobio, etc that have many more channels and they are more research-based. Their cost ranges from $850 to even $25000 (Portillo-Lara et al., 2021).

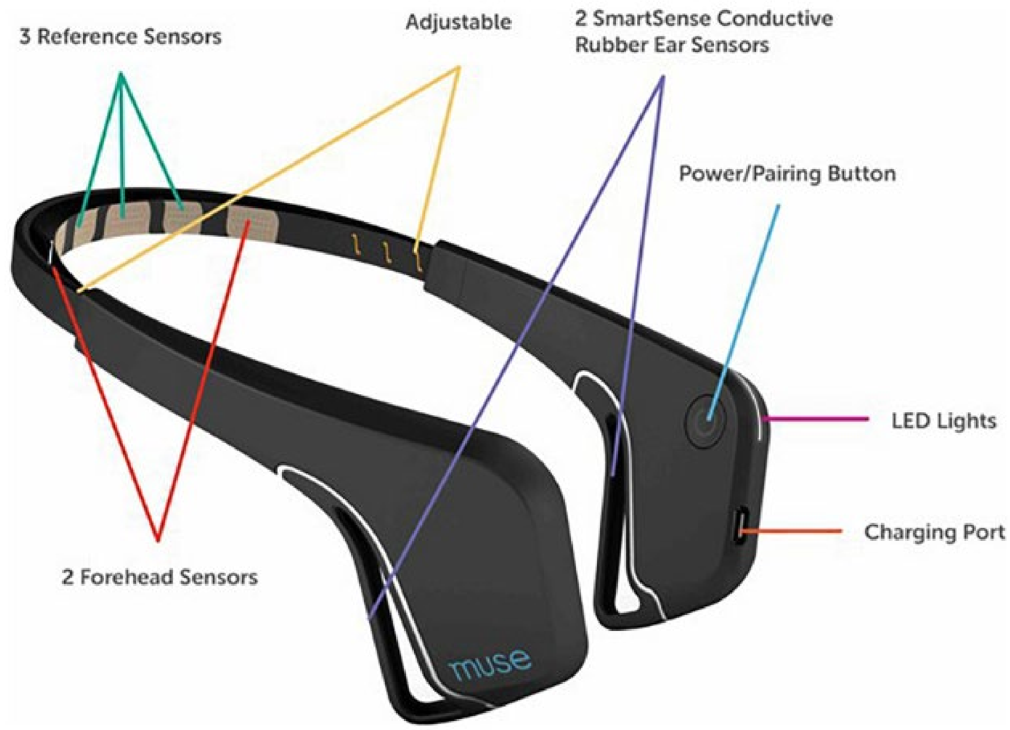

Figure 1: Muse 2016 device (Krigolson et al., 2021).

Muse is a wearable headband by InterAxon Inc. that has 2 frontal electrodes at the position AF7, AF8 and two temporal electrodes at TP9 and TP10 (Krigolson et al., 2021). It has 3 reference electrodes at the center and also includes a 3-axis accelerometer (Abujelala et al., 2016) and a heart-rate sensor (Sawangjai et al., 2020) with a streaming rate of 256Hz through Bluetooth protocol. Muse has 3 versions: Muse (released in 2014), Muse 2 (released in 2016), (Sawangjai et al., 2020) and Muse S (Soufineyestani et al., 2020) and its battery lasts for 5 hours

(Sawangjai et al., 2020). Muse's application is in the field of neurofeedback that acts as an aid for meditation and has its own commercial Android/iOs app for this purpose (Sawangjai et al., 2020). However, its data can also be collected outside of this application (Sawangjai et al., 2020), making it an excellent candidate for research or custom applications due to its low cost and quick setup. Besides, research also shows that Muse is a reliable and accurate tool that can be used in ERP and EEG experiments successfully (Krigolson et al., 2021).

Another EEG device is OpenBCI, which is an open-source device that started as a low-cost Kickstarter campaign. It consists of a printed circuit board (PCB) and a software for visualizing and collecting data. An extra set of electrodes (dry or wet) must be purchased and placed in the head in the desirable location by the user (Sawangjai et al., 2020). Several years after its launch their e-shop included some bundles with several different solutions based on each user's needs and budget. For example, 'All-in-one EEG electrode cap started kit' bundle includes: 16-channel OpenBCI board, Ag/AgCl electrodes and an electrode cup that costs $2449.99 (OpenBCI, n.d.-a). But, due to customizations needed and extra products that must also be purchased, it is more appropriate for users with engineering background, making it less a popular choice (Sawangjai et al., 2020).

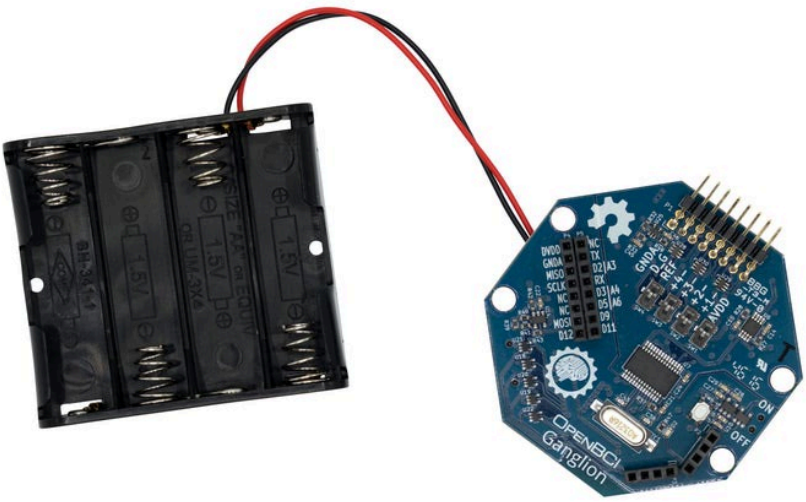

Figure 2: OpenBCI 4-channel Ganglion board (OpenBCI, n.d.-b).

2.5 Applications, Brain-Computer Interfaces

There are many different applications where EEG can be used by scientists, clinicians, engineers, simple end-users etc. Generally, we can distinguish the applications in some broad categories like medical, research and Brain-Computer Interfaces (BCI).

At the medical field applications include detection of abnormal brain functioning or cognitive impairments. This could be seizures, strokes, traumatic brain injuries, Parkinson's disease, attention deficit hyperactivity disorder (ADHD), sleep disorders, language disorders, autism, anxiety, post-traumatic stress disorder (PTSD), memory impairments, coma (Soufineyestani et al., 2020), brain tumors (Bera, 2021), etc. Moreover, in some cases it may even predict seizures (Slimen et al., 2020) or strokes (Kaur et al., 2022).

At the research field scientists utilize EEG to better understand how the brain works when performing tasks under different emotional or mental states such as stress, drowsiness, fatigue, suicidal states, insight, etc. These tasks may include surgeries, driving/piloting, working, drinking, decision-making, learning, etc (Soufineyestani et al., 2020). Engagement and workload are also important factors that can be measured using EEG. These are mostly used in learning contexts due to their correlation with the learning outcomes (Khedher et al., 2019).

Brain Computer Interfaces (BCIs) are systems that provide a communication channel between one's brain with a software or hardware (Portillo-Lara et al., 2021) and it is the most popular application area of EEG (Soufineyestani et al., 2020), called EEG-based BCIs (*eBCIs*) (Portillo-Lara et al., 2021). BCIs can be distincted based on their invasiveness into invasive and non-invasive and based on the brain signal type into active, passive and reactive BCIs.

Active BCIs require the user to intentionally do something that exerts specific EEG patterns that can be recognized by the system. The ultimate goal is to control a device by the proper analysis of the EEG data. This recognized EEG pattern could be motor imagery, mental arithmetic, etc (Portillo-Lara et al., 2021). Using motor imagery, users of such systems imagine to move their limbs without actually performing any motor output (Portillo-Lara et al., 2021). This signal is then processed, and the users are able to control mechanical parts like robotic arms using their brain only. This is particularly useful for people with motor disabilities. Moreover, people with speech impairments, are able to communicate with the outside world (Soufineyestani et al., 2020) using the so-called "spellers" where they choose letters/words using their brain only (Thompson, 2019). Of course, the response time is much lower than that of the input devices (mouse, keyboard etc) that would be used instead (Shishkin, 2022), the accuracy can be low and user training is needed that can take much effort and time (Portillo-Lara et al., 2021).

Passive BCIs on the other hand, do not require the user to do anything in particular. Therefore, passive BCIs are mainly used to grasp the user's current mental states (attention, fatigue etc) (Portillo-Lara et al., 2021) and it is mostly used by healthy individuals (Zander & Kothe, 2011). The goal is to send user information to the system without any extra user effort (implicitly), making these systems a great candidate for adaptive systems (Zander & Kothe, 2011). Finally, reactive BCIs rely on external stimulus that are known to generate very specific brain responses such as P300 EP (Douibi et al., 2021), in order to control something (Zander & Kothe, 2011).

It should be noted here that there is a phenomenon related with the training of the user to use active BCIs. This term is called BCI illiteracy, which is the inability of 15-30% of the users to control a system (Vidaurre & Blankertz, 2010) even though they have received the same training as successful individuals (Thompson, 2019). More specifically, one can be unsuccessful using a specific recognizable pattern (for example motor imagery) but successful in another. BCI illiteracy is not always considered a permanent situation. Differences in BCI training and in actual use of the system such as mood, attention, caffeine consumption and sleep may result in the user's illiteracy. However, this term is criticized by many researchers as the

BCI design may be more to blame than the user's inability. Therefore, individualized BCIs may be the solution to this problem (Thompson, 2019).

BCIs have applications in medical, research, security, education, marketing (Portillo-Lara et al., 2021), gaming, art, Augmented / Virtual Reality (AR/VR) and Autonomous-driving vehicles fields (Shishkin, 2022). In BCI applications, EEG data is processed real-time (online) by the system to achieve its goal. Some examples include: device control, mental states (workload / fatigue / stress) monitoring for operator's safety, learning adaptive systems based on workload or vigilance (Douibi et al., 2021), gaming with or without the use of VR/AR (Shishkin, 2022), user identification, rehabilitation, decision-making examination when purchasing, neurofeedback (Portillo-Lara et al., 2021). Neurofeedback is a form of biofeedback where the user gets an immediate auditory or visual response based on the desired brain activity (Teplan, 2002) that lead to specific behavioral responses. Neurofeedback has many applications, for example treatment of attention deficit hyperactivity disorder (ADHD) or seizure regulation where the user implicitly learns to avoid brain activity that leads to seizures (Portillo-Lara et al., 2021), meditation aid as mentioned earlier, etc.

2.6 Neural Oscillations

The brain's electrical activity described earlier shows a rhythmic pattern called brain oscillations or brainwaves that are classified into five major frequencies. Slow frequency waves such as alpha (α), theta (θ) and delta (δ) and high frequency waves such as beta (β) and gamma (γ). Some of them are more present in specific brain regions than others (Kamel & Malik, 2014). Also, the lowest or highest ends of those frequencies are not always consistent among researchers usually having 1-2 Hz deviation.

Generally, each wave is associated with specific cognitive states even though the wave's frequency is only a part of this puzzle. The amplitude, phase or coherence may also play their role (Herrmann et al., 2016).

Delta waves are the lowest waves ranging from 0.5 to 4Hz. In contrast, the amplitudes of these waves are the highest and these waves are associated with deep sleep (Kamel & Malik, 2014). For awake individuals, it is associated with cortical plasticity and are indicators of cognitive processing, as shown in the P300 in ERP experiments (Malik & Amin, 2017).
Theta waves range from 4-8Hz and are associated with meditation and drowsiness and these are mostly evident at children. However, having high delta activity may indicate brain disorders (Kamel & Malik, 2014). Also, theta waves are associated with attention and memory processes (Malik & Amin, 2017).

Alpha waves range from 8-13Hz, they are more dominant in the occipital lobe and are associated with creativity, wakefulness, relaxation, having the eyes closed (Kamel & Malik, 2014), memory and attention (Herrmann et al., 2016). Additionally, it is shown that alpha waves are present when someone inhibits task-irrelevant information. Beta waves range from 14-26 Hz, they are more present in the frontal and central regions and are associated with active attention and thinking, problem solving (Kamel & Malik, 2014), motion tasks or tasks with sensorimotor interaction (Herrmann et al., 2016).
Gamma waves range from 25Hz and up, they are associated with cortical activation, attention, working memory, long-term memory and conscious perception. Gamma waves can be easily confounded with muscle artifacts, that's why research on those is limited comparing to other brainwaves (Malik & Amin, 2017).

2.7 EEG Processing Pipeline

EEG-based BCIs must translate the signal to desired commands or classes (in our case, mental states). This is a complex and difficult task and in order to do so, the signal must first be processed and analyzed with the help of machine learning (Lotte, 2014). This includes the following sub-tasks: EEG signal acquisition, preprocessing, feature extraction and classification.

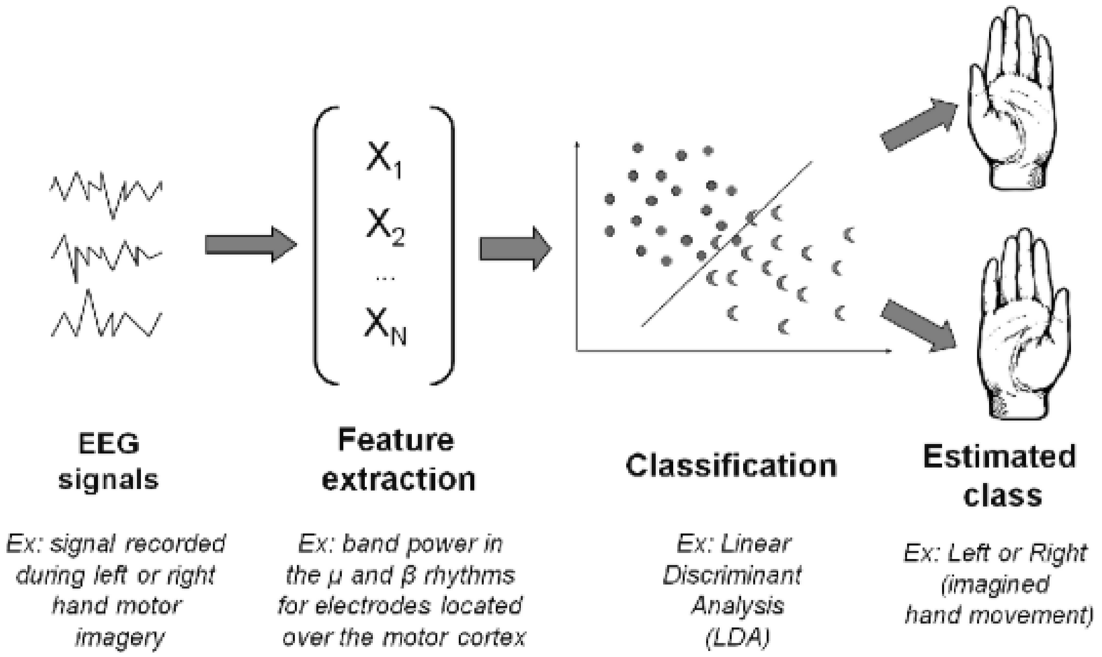

Figure 3: BCI signal processing pipeline for motor-imagery BCI (Lotte, 2014).

The signal acquisition is made through the EEG device and the recording device as described earlier. EEG signal preprocessing removes noise and artifacts in order to increase signal quality. This is usually done with signal filtering, data segmentation, removal of bad segments or more advanced techniques (Li et al., 2020). Signal filtering is done by applying low-pass, high-pass, band-pass or band-stop filters to the signal. Low / high pass filters keep the signal below/above a certain frequency and attenuate the rest of signal. Band pass / band stop filters keep / attenuate the signal in a specific range of frequencies (Peng, 2019). The most common filtering scenario is to apply a filter in order to remove frequencies below 0.1 Hz and also remove the power line noise, which is 50Hz in Asia and Europe, and 60Hz in Unites States (Li et al., 2020). While someone may apply a low-pass of 30Hz for example, there is still the need to filter power line noise because it could be extremely powerful that could still contaminate the signal (Li et al., 2020).

As mentioned earlier, another part of preprocessing is data segmentation. Data segmentation is a standard procedure at ERP experiments due to their nature. The continuous EEG recording must be separated into smaller segments that start with the stimulus onset (Peng, 2019). On the other hand, even though at resting-state potentials EEG there is no specific event happening, there is still a recommendation to segment the data into smaller pieces (Li et al., 2020). These segments are often called windows (Candra et al., 2015). Unfortunately, there is no standard window size. Scientists use different sizes starting from 1s (Chaouachi & Frasson, 2012), to 40s (Wang et al., 2012), that usually overlap by 50% (Li et al., 2020). Additionally, if a data segment is detected to have a lot of artifacts it can be discarded manually or automatically with advanced techniques such as Independent Component Analysis (ICA) that can detect eye blinks or muscle movements (Li et al., 2020).

After preprocessing is completed, the processed signal that describes best what we want to recognize called *features*, must be collected into a vector called *feature vector* (Lotte, 2014). However, the features must be selected wisely as more features will lead to computational complexity (Tambe & Khachane, 2017). Specifically for EEG, the Power Spectral Density (PSD) is the most frequent and important feature. This is the power distribution of the signal amongst several frequencies. In other words, it is the transformation of the signal from time-domain to frequency-domain (Zhang, 2019). The frequencies that are included are part of the brain oscillations and more specifically they depend on the task at hand. For example, You (You, 2021) used the frequencies from 9-43Hz (with 4Hz bin) as it lead to better classification results when classifying relaxation/concentration mental states, whereas for motor imagery classification the frequencies from 8Hz to 24Hz are more common (Lotte, 2014). Therefore, the feature vector from the concentration/relaxation example that was described is *number of channels* **x** *feature* for each EEG window so if we have a 4-channel EEG device and divide the frequencies of 9-43Hz into a 4Hz bin (band width) the result would be a *4***x***8* dimensions for each window.

Machine learning is the ideal tool to decode information from such high dimensions because statistical models are not capable of doing so (Li et al., 2020) and classifiers learn to identify the appropriate class from a feature vector (supervised learning). So, the next step after having a feature vector is to define a class (for example *concentration* and *relaxation*), split the available dataset into training set and test set and train the selected ML algorithm with the labeled training set. Finally, using the test set someone is able to evaluate the accuracy of the algorithm (Li et al., 2020). So, overall, the pipeline is the following: Acquire enough labeled raw EEG data for all desirable classes (*concentration*, *relaxation*). Preprocess the EEG signal, find the features (so calculate the PSD) and form a feature vector. Then, select the appropriate ML algorithm. Split the dataset into training set and test set, train the data with the training set and evaluate its performance with the test set. If the results are acceptable, use this ML classifier to the BCI system. The algorithm will be able to predict the class of a new dataset. More details about machine learning will be presented in the following chapter.

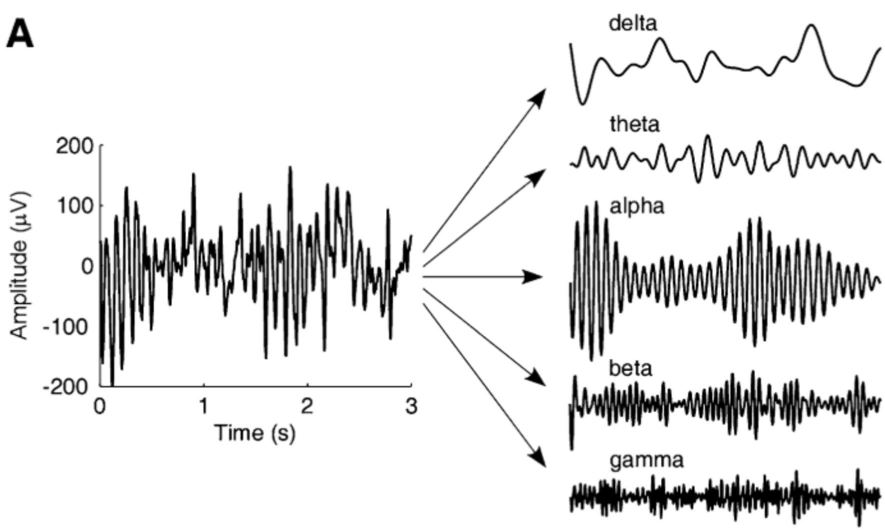

Figure 4: Signal in time-domain and it's representation in delta, theta, alpha, beta and gamma frequencies (Zhang, 2019).

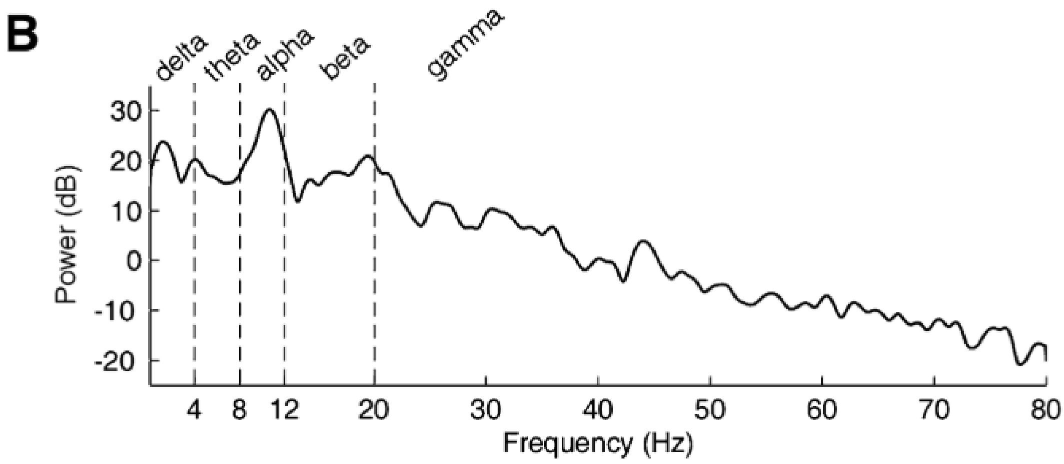

Figure 5: The spectrum of the same signal. Alpha is the most powerful frequency (Zhang, 2019).

2.8 Machine Learning

The relationship between inputs and outputs of data can be successfully created from mathematical models in some simple occasions. This means that a prediction can be made possible when given an input. However, this is not directly possible when the complexity of this relationship is high. Machine Learning (ML) is a set of techniques that can automatically create a mathematical model in such complex scenarios (Baştanlar & Özuysal, 2014). In other words, ML can learn without being specifically programmed to with rules or instructions (Sarker, 2021). This is accomplished using large amounts of data called training data, so recent advances in data storing was a huge contribution to ML (Baştanlar & Özuysal, 2014). This training data includes a feature vector as described earlier, which is a vector containing the most salient information about the data at hand. These features can be numerical, categorical, or ordinal (values with an order). However, one must not use too many features because it could lead to several issues such as performance costs, etc (*curse of dimensionality*). For this reason, there are some available techniques to minimize the feature length (*dimensionality reduction*). (Badillo et al., 2020).

ML algorithms can be distinguished in the following categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (Sarker, 2021) based on the availability of the output data (Baştanlar & Özuysal, 2014). At supervised learning, inputs are always followed by outputs and common tasks include classification and regression. Classification predicts discrete output values based on data inputs (Sarker, 2021) while regression will map the inputs to a continuous output variable (Baştanlar & Özuysal, 2014). On the other hand, unsupervised learning does not need outputs. Its goal is to find groups of similar items. Common tasks include clustering, density estimation, feature learning, dimensionality reduction etc (Sarker, 2021). For example, finding social accounts based on similar activity (Baştanlar & Özuysal, 2014). Semi-supervised is a hybrid learning category between supervised and unsupervised learning as it uses pairs of inputs-outputs, but some outputs can be omitted. Last, reinforcement learning is based on rewards or penalties that are used to improve the systems efficiency (Sarker, 2021). In this chapter we will focus on supervised learning algorithms used specifically for classification as this is the systems goal.

Supervised learning algorithms have the ability to evaluate their performance using various data-splitting techniques. Usually, the collected data is divided in a training set and a validation set. The training set creates models with multiple ML parameters that are tried on the validation set in order to find the optimum ones. The best model is then selected and tried on another set called the test set that calculates its performance. The performance can be affected by the total data length, overlapping data and the algorithm selected itself (Xu & Goodacre, 2018).

Such evaluation is highly important because ML algorithms may fall into one of the following problems: overfitting or underfitting. Overfitting is when the algorithm performs very well when evaluated on the training data, but its predictions are not accurate on data that the algorithm has never seen (Xu & Goodacre, 2018). This means that the algorithm is not able to generalize. Underfitting on the other hand, is when the created model is too simple to make the appropriate mappings between inputs and outputs (Baştanlar & Özuysal, 2014).

Naïve Bayes is a simple supervised learning algorithm based on Bayes Theorem that uses conditional probability. It creates a probability table that is updated with the training data. The reason why it is called "Naïve" is due to the assumption that every feature is independent from another. Due to its simplicity, other ML algorithms may perform better than Naïve Bayes. However, it can be used for clustering or classification of two or more classes, it has good performance, and it does not need a large dataset (Ray, 2019). One of its common usages is text categorization (Hosseini et al., 2021) but it can also be used in EEG tasks such as emotion recognition, seizure detection and motor imagery (Saeidi et al., 2021).

K-Nearest Neighbors (KNN) is a supervised learning algorithm that is used for classification and regression (Hosseini et al., 2021). It depicts all training data in the n-dimensional space and tries to classify unseen data using Euclidean distance function from the k nearest neighbors of each point based on their similarity (Sarker, 2021). The number k is selected based on the amount of the dataset (Hosseini et al., 2021), from 1 to 20 (Sciaraffa et al., 2019). However, the selection of the number k is a difficult task (Sarker, 2021) and its accuracy depends on the quality of the data. Also, if the feature length is kept low its computational cost is also low (Sciaraffa et al., 2019). It is successfully used in EEG tasks with high accuracy (Saeidi et al., 2021).

Random Forest is an ensemble supervised learning algorithm used for classification and regression. It creates a set of decision trees during training and takes all created trees into account to make its decision using bagging operation (Hosseini et al., 2021) and output randomization. When all individual decision trees are created, a majority voting decides the final result. Random Forest is accurate and can make use of large amounts of data (al Amrani et al., 2018). Therefore, it performs better than other ML algorithms in EEG tasks that include a large dataset (Saeidi et al., 2021). Moreover, is works really well on smaller datasets also, due to its robustness to overfitting. Finally, its parameters do not need many adjustments (Badillo et al., 2020).

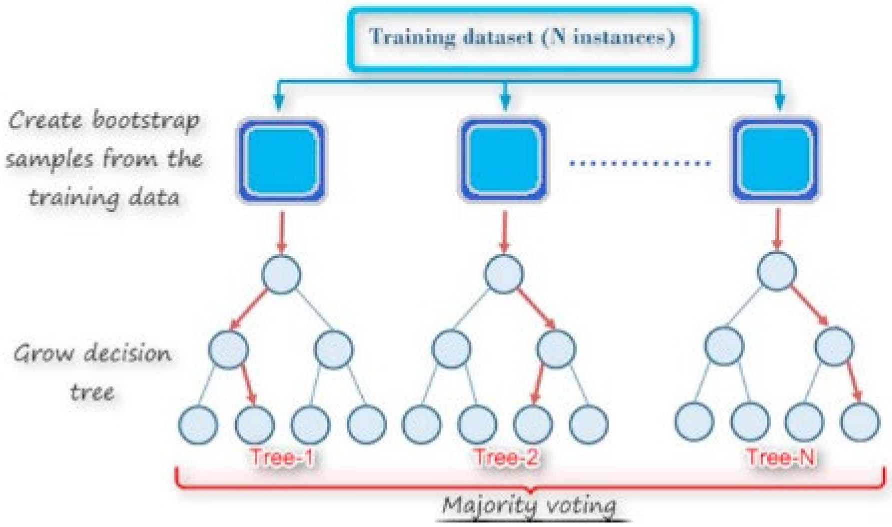

Figure 6: Random Forest (al Amrani et al., 2018)

Support Vector Machine (SVM) is a supervised algorithm created at the 1990s that is being used in binary classification and regression tasks. This algorithm works by plotting all the data points in n-dimensions and tries to find the hyper-plane that separates these two classes with the largest possible margin (al Amrani et al., 2018). This separation can be linear or non-linear. Non-linear separation is possible due to kernel functions that transform the data into higher dimensions that make the separation possible (Baştanlar & Özuysal, 2014). SVM is the most frequent and accurate ML algorithm used in EEG tasks, possibly due to its simplicity and its high generalization. However, its parameters must be optimized because its performance is highly dependent on it (Saeidi et al., 2021). This is a difficult task in EEG data due to the non-stationary characteristics of EEG (Aggarwal & Chugh, 2022).

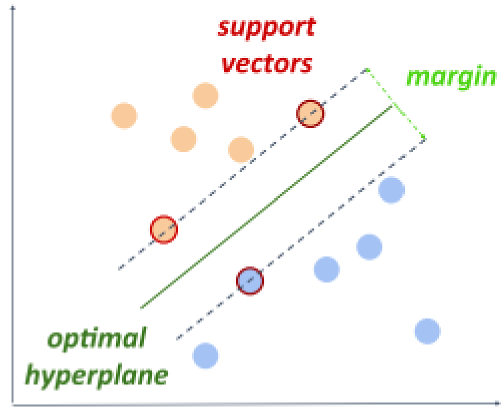

Figure 7: Support Vector Machine (Badillo et al., 2020)

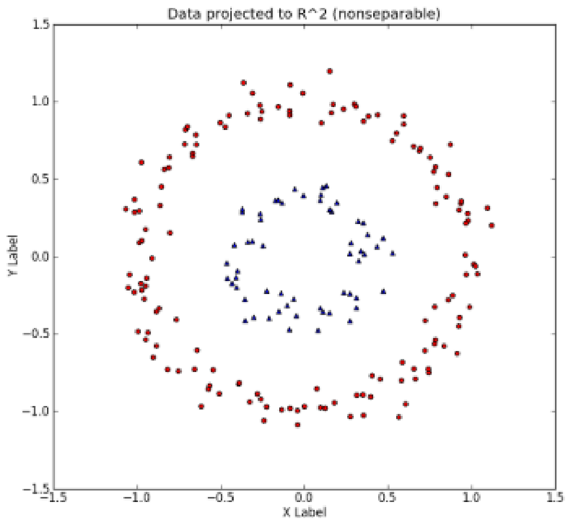

Figure 8: Non-separable data (Hosseini et al., 2021)

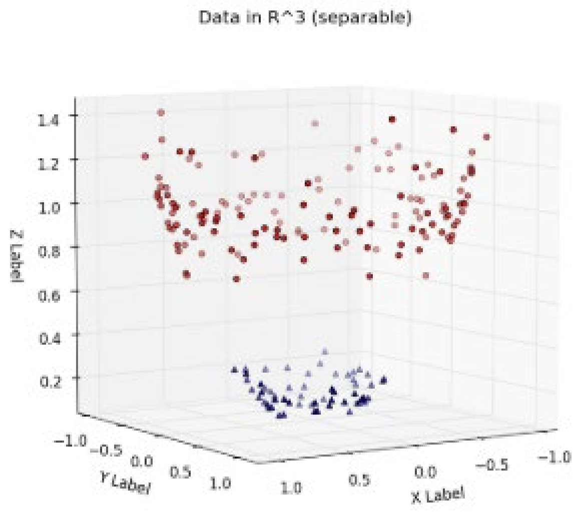

Figure 9: The same data transformed to a higher dimension (separable). (Hosseini et al., 2021)

2.9 E-Learning

E-learning is a board term "describing any type of learning that depends on or is enhanced by online communication using the latest information and communication technologies" (Nagy, 2017). E-learning is much more flexible than conventional learning at classrooms and can be performed asynchronously at any time the student chooses to. This saves time and transportation costs that may apply. Also, there is a huge collection of available courses that the user can select from and follow at his/her own pace, offered in an interactive environment using rich multimedia educational material (Nagy, 2017). Additionally, there are e-learning environments where there is also a teacher performing synchronous learning (Klašnja-Milićević et al., 2017). Universities or schools use e-learning systems to exchange educational material or assignments, perform tests and inform the students about their progress. E-learning is also used in domains outside of education, for example it is used in companies for the training of their employees (Klašnja-Milićević et al., 2017). Finally, e-learning is considered low-cost compared to traditional learning because there is no need to buy textbooks that can be obsolete after a short period of time. In contrast, the e-learning content is updated regularly (Hammad et al., 2018). Nonetheless, there is lack of interaction between the teacher and the students leading to student's isolation (Hammad et al., 2018), lack of motivation and lack of immediate feedback from the students (Al-Nafjan & Aldayel, 2022).

Regarding asynchronous learning, there are also Massive Open Online Courses (MOOC) that are self-paced and open for thousands of people to attend to. They also share a common educational material (Ceron et al., 2021). The most popular ones are Coursera, EdX, FutureLearn, Open EdX and Moodle (Ceron et al., 2021). In some of those, there is an active community where students are asking questions and receive feedback (Russell et al., 2013) in forums or there is a chat functionality available in public rooms or privately (Coetzee et al., 2014). The main disadvantage of MOOCs is their high drop rate that may be due to poor quality of the course, lack of effective time-management of the students, lack of prerequisite knowledge for attending the course or for reasons that were already mentioned generally in e-learning contexts (Ceron et al., 2021).

2.10 Adaptive and Interactive Systems

Most systems are designed to meet some characteristics that the majority of users would need, commonly called the 'one-size-fits-all'. However, each user has unique characteristics, preferences, interests, and emotions that affect his/her interaction with the system. Adaptive and interactive systems are systems that have the ability to change their content, functionality or user interface based on collected data about the user. The goal is to increase the system's functionality and provide a positive user experience due to personalization (Papatheocharous et al., 2014). An example of such systems is that of Google, Bing and Amazon that personalize the search results or provide personalized recommendation to its users (Papatheocharous et al., 2014).

In order to design such system, a user model must be created and maintained (Germanakos & Belk, 2016) using tools such as data mining and machine learning (Bezold & Minker, 2011). Then, adaptation mechanisms decide exactly what will be adapted. There are many different algorithms for deciding that: rule-based filtering, collaboration filtering, content-based filtering etc. Rule-based filtering is the simplest approach of using rules for this decision (Germanakos & Belk, 2016).

A user model represents relevant information about the user, able to make personalization possible (Germanakos & Belk, 2016). This information could be static, about the user himself/herself such as gender, date of birth etc or dynamic information such as knowledge, interests, goals, background and individual traits. On the other hand, information about the user's context can be his/her location, device or device type, interaction history, etc. This information can be provided by the user explicitly (questionnaires, psychometric tests) or inferred implicitly by the system (Germanakos & Belk, 2016). For example, in a web system a log file of all user actions is kept. So, a user model can be constructed by observing the user's interactions and creating a high-level meaning out of these sequences. This could be a user preference (Bezold & Minker, 2011) or interest, the level of knowledge or learning goal in an e-learning system, personalized user authentication, etc (Papatheocharous et al., 2014). The creation of an implicit user model is a much more challenging task but also preferrable as no further user-interaction is needed that could potentially increase the user's cognitive load. Of course, due to the complexity of creating this model, the model could be incorrect (Germanakos & Belk, 2016). Additionally, the user's physiological data can also be included in the creation of the user-model by using external sensors that are connected to the system (Bezold & Minker, 2011).

There are specific factors that contribute more to the user's positive experience and the overall usability of the system for each domain, and according to Germanakos & Belk (Germanakos & Belk, 2016), the user's personality, emotions and cognition may contribute to all domains having a human-computer interaction.

Specifically for the learning domain, cognitive processes that support learning such as creative thinking, inspiration, concentration, and motivation are affected by user's emotions (Chaouachi & Frasson, 2010). And even though the recognition of the student's emotions may be an easy task for conventional learning environments where teachers can adapt their teaching accordingly, this is not the case in an e-learning system (Nandi et al., 2021). Accordingly, other issues of e-learning versus conventional learning are lack of motivation, lack of immediate feedback about the difficulty of the educational material (Al-Nafjan & Aldayel, 2022), or mental states of the students (Chaouachi & Frasson, 2012).

Therefore, we conclude that an immediate feedback about the user's mental states could lead to increased user and learning experience and potentially provide better learning performance. This feedback, according to the system type, could be fed back into the system itself in order to adapt, or it could inform the teacher to better support learning.

The aforementioned human factors can be elicited implicitly with the use of external sensors. More specifically, EEG devices are a popular solution for providing unobtrusive and continuous data about the users' mental states. Also, compared to other physiological sensors such as electrocardiographic activity (ECG), neural-based ones (EEG, fNIRS) are considered an even better choice (Mühl et al., 2014). However, mental states elicitation is a complex task that needs expert knowledge in many different domains such as: sensor technology, signal processing, neurophysiology, experimental psychology, systems design, engineering, and advanced machine learning algorithms (Brouwer et al., 2015).

2.11 Related Research

2.11.1 Adaptive E-Learning Systems

Generally, there are many different categories of related papers. Mental states is the first and most important one. Then, the application of use such as e-learning, driving or aviation for example. Also, there are two broad categories based on the

method of EEG analysis: the papers that use machine learning or some that use only statistical methods, and the ones that use adaptations or not. Finally, the device that is used is very important. Papers using commercial devices of 1-4 channels is totally different than the ones that use research or medical EEG devices of up to 64 channels because they have very different capabilities than the first ones.
To this time, no related work is found that satisfy the above criteria: the creation of an e-learning system that provides adaptations on the system functionality based on the mental states of concentration/relaxion with online EEG processing using ML algorithms and EEG devices of 1-4 channels, so less relevant studies will be presented.

Chaouachi and others (Chaouachi et al., 2015) created an online adaptive system using a 14-channel EEG device and mental states of attention (using engagement index (Pope et al., 1995)) and workload. The goal was to keep the students in a positive state in order to increase the learning experience and outcomes. The adaptation was on the learning activity: the system got to choose the most relevant learning activity from either a problem-solving task or a worked example. The system was compared to a version of the system that did not consider the mental state adaptation rules and showed that it successfully achieved its goal.

Walter and others (Walter et al., 2017) created an online adaptive learning environment based on the user's workload, using a 28-channel EEG device and ML algorithms. The goal was to improve learning performance. The adaptation was on the learning material difficulty. The system was compared to error-based adaptive environment and showed that it successfully achieved its goal.

Hu and Kuo (Hu & Kuo, 2017) used 5-channel EEG device to record data while students watched an educational video. This data was correlated to the exam scores of students with a rate of 83%. The authors propose that this correlation could be used in an adaptive system that could propose the number of times a video should be watched to increase the learning performance.

Lin and Kao (Lin & Kao, 2018) investigated the mental states (mental effort/confusion levels) classification performance of a system using online YouTube videos and a 1-channel EEG device. Their goal was to help future e-learning systems to inform the users about their mental state so that they could improve their learning experience or performance and at the same time inform instructors as well so that they could alter the learning material accordingly. The accuracy of the ML classifier reached 95% (Lin & Kao, 2018).

Kosmyna and Maes (Kosmyna & Maes, 2019) created a prototype which recorded student's EEG data and calculated their engagement index (Pope et al., 1995) online. 1-channel EEG device was used and provided haptic vibration feedback through a given wearable scarf device when engagement levels were low. Its goal was to increase the student's engagement and performance that was indeed, achieved.

Liu and Ardakani (Liu & Ardakani, 2022) proposed an e-learning adaptive system based on the student's emotions using a 14-channel EEG device. The adaptation was on the learning content (funny videos vs learning material) and its goal was to increase the learning performance, engagement and satisfaction compared to a traditional e-learning environment. However, only satisfaction was improved.

2.11.2 Concentrated and Relaxed mental states

A study of Edla and others (Edla et al., 2018) showed that mental states of attention and meditation can be correctly classified with an accuracy of 75%, using 1-channel EEG device and Random Forest ML algorithm. The attention task was to try to solve a mathematical problem, for one minute. The meditation task was to try and relax by closing their eyes for one minute. The same conclusion comes from another study of Ricker and others (Richer et al., 2018) using 4-channel Muse EEG device. At focus tasks subjects were instructed to perform mental arithmetic, dictation and 'Where's Waldo'. At relax tasks, they were asked just to relax by watching some nature scenes but not close their eyes. Relax and focus tasks were alternated with a total training time of 3x3 minute for each class.

Moreover, a study by Bird (Bird et al., 2018) successfully classified 3 mental states of relaxing, concentrating and neutral, using a 4-channel Muse EEG device and Random Forest ML algorithm. At the relax task, subjects were instructed to relax while listening to a low-tempo music with sound effects, but not close their eyes. The neutral task was the same as the relaxed task but with no stimuli. At the concentrating task, participants were instructed to follow a ball that was hidden under one of the three cups that were presented on the screen and then switched. Total training time of 1 minute for each class and the accuracy was 87%.

Another study of You (You, 2021) used 2-channel EEG device to classify relaxation / concentration mental states with an accuracy of 80% using SVM ML algorithm. The concentration task was a reverse 5-digit span, meaning that a 5-digit number was presented aloud, and the participant should silently recite the digits of that number in reverse. At the relaxing task participants were asked just to relax. Both tasks were performed with eyes closed to avoid artifacts. The total training time was 20x10 seconds for each class.

From all of the above studies we can conclude that it is feasible to classify concentration / relaxation mental states using an EEG device of 1-4 channels. More specifically, using Random Forest or SVM ML algorithm with a relatively small total training time of less than 18 minutes. It is also important to note that the EEG devices that were used in those studies included at least 1 frontal electrode as frontal lobes are known to be associated with problem-solving tasks (Channon, 2004). Finally, many studies used the same EEG device as the one that was chosen for this dissertation (Muse).

Chapter 3
Implementation

In the current study, a prototype web adaptive system was developed, called *EEG Learn*. It can also be classified as a passive BCI, and it simulates an e-learning environment where users are participants in online lectures and are consumers of learning content in general. This system could be a Massive Open Online Course (MOOC) or a system used by schools or universities to complement or replace their lectures. This system is designed to adapt its functionality based on the user's mental states using Muse 2, a low-cost EEG device.

More specifically, a status functionality was developed. This is commonly used in online communication systems such as skype, slack etc and indicates whether a user is available. Usually, this status is changed explicitly by the user among several other statuses (*Online, Busy*, *Away* etc) or implicitly by the system when the user is idle (*Away*). However, the *Busy* indication is a status that cannot be changed implicitly. In the current implementation, the status of the user is automatically changed based on his/her mental state. The selected mental states were: *Relaxing* and *Concentrating* that correspond to *Available* and *Busy* statuses. Besides, these mental states were more suitable for detection using the specific EEG device.

The structure of this chapter is the following: first the developed system will be presented. Then, all its components and architecture will be presented and discussed in detail.

3.1 User Interface and Functionalities

First, the user has the ability to register an account with a username and password and then log into the system with these credentials. The web application has the following links/topics: *Introduction*, *Device Tutorial*, *Prepare Device*, *Signal Quality*, *Train*, *Predict* and *Attributions*. The following functionalities are also available: Notifications, Connectivity, Settings, Logout and Status. A full presentation of the system is available at the appendix.

Initially, the user is instructed to go through the first link *Introduction* and read the instructions that are presented there. From there, the system will guide the user as to what to do next. At the *Device Tutorial*, a video on how to wear the Muse device is presented. Then, at the *Prepare Device* the user is asked to turn on the device, wear it and pair it with the system. When this is done successfully, the user moves on to the next step, *Signal Quality,* where valuable information is presented about the importance of the signal quality. At that point, the user can test how good the signal is for each of the four electrodes of the EEG device. If the signal quality is good for all four electrodes, the user is prompted to move on to the *Train* step and from there to the *Predict* step. Specifically, for *Prepare Device* and *Signal Quality* the user is suggested to ask for extra guidance on how to wear the device and check its signal because this is the most important factor of the system's success. However, if needed, the user can also ask for extra guidance at any step.

The training procedure is the most important part of the system. Two task types are presented in an alternated order for one minute and each one is expected to induce a specific mental state. Each relaxing task presents a picture (nature photos mostly) and asks the user to relax with eyes open while listening to a soothing music track with no lyrics. Each concentrating task presents two consecutive shapes asking the user to count how many circles/squares/rectangles are contained in the given shape. There are 6 sessions for each task type with a total training time of 12 minutes. The music and shapes are different from each other.

Task Type	Task	Length	Metric
Relaxation	Listen to a music track	60	Seconds
Concentration	Count embedded shape	30	Seconds
Concentration	Count embedded shape	30	Seconds
Relaxation	Listen to a music track	60	Seconds
Concentration	Count embedded shape	30	Seconds
Concentration	Count embedded shape	30	Seconds
Relaxation	Listen to a music track	60	Seconds
Concentration	Count embedded shape	30	Seconds
Concentration	Count embedded shape	30	Seconds
Relaxation	Listen to a music track	60	Seconds
Concentration	Count embedded shape	30	Seconds
Concentration	Count embedded shape	30	Seconds
Relaxation	Listen to a music track	60	Seconds
Concentration	Count embedded shape	30	Seconds
Concentration	Count embedded shape	30	Seconds
Relaxation	Listen to a music track	60	Seconds
Concentration	Count embedded shape	30	Seconds
Concentration	Count embedded shape	30	Seconds
	Total Time	12	Minutes

Table 1: Training

After the user moves on to the *predict* procedure, he/she is asked to select one task type of his/her choice in order to induce the according mental state and test it. For this reason, both task types are presented at the same time on the screen. If the user chooses to follow the relax task, then he/she can ignore the shape and try to relax with the relaxing photo and music. If the user chooses to follow the concentrating task, then he/she can mute the music (with the proper icon or using windows mixer) and count the shapes. A splitter between these tasks is also available so that the user can easily hide the irrelevant task. The user is advised to use it.

There are four predict sessions lasting for 30 seconds each, and when each session ends the user mental state is calculated and presented on the screen. This calculation uses each user's training set, so it is subject-specific. Then, the adaptation takes place, and the user status changes automatically (*Available* or *Busy*). Feedback from the user is asked as to whether the prediction was correct. At the same time, at *Available* status all system notifications are enabled and at

Busy status, notifications are disabled. When all four predict sessions end, the user is prompted to repeat the predict phase to better validate the predictions.

Additionally, if for any reason at the train/predict step an error occurs, then the user can repeat the current step instead of repeating the whole procedure. Finally, when the predict phase is completed the system thanks the user for his/her participation and the evaluation questionnaire is given.

Finally, at the *Settings* functionality, the user is able to change the EEG device between a test device and two versions of the Muse 2 device. The first version of Muse uses native Bluetooth connectivity and the second one needs a BLED112 USB dongle for Bluetooth Smart / Bluetooth Low Energy communication. The second one was selected by default. The test device was used only for development purposes.

3.2 System Analysis

At this system, detailed instructions and the whole User Interface was created so that it is user-friendly and supportive for the user-student. Each step was strictly organized into small steps in order to make the user feel secure and comfortable.

The connectivity functionality is the one responsible for informing the user if there is any connectivity issue (EEG device pairing issue, backend communication issue, eeg-client communication issue). Without it, the user could complete the training without success if any error occurred that would frustrate the user.

A simple Notifications functionality was developed where the user is notified for 3 basic events: the user logging into the system (a welcome) and completing the train and predict phases. This functionality was only developed to show that these notifications would be disabled at the *Busy* status.

The Status functionality that was developed emulates *Busy* and *Available* statuses of communication applications like skype or slack. In an e-learning system this would indicate that the user is concentrated in his/her studying and does not wish to communicate with anyone else using a chat for example. Moreover, this change of status also disables or enables all notifications without having to explicitly change the status as constantly changing the status can be cumbersome.

The factors that contributed heavily to the decision of using the Muse device against others are the following: The Muse 2 headband is a low-cost EEG device that can be easily purchased. Its ease of use and its availability in Europe are important. For this reason, Muse 2 is quite popular choice among the BCI field and this also contributes to having a large community, libraries or content in the internet for this topic. Also, even though Muse 2 was the only device tested, the system also has the potential to work with approximately 20 other EEG devices as well with a small change in the code, using the *Settings* functionality.

Furthermore, the participants were intentionally not asked to minimize their eye blinking, even though it is known to be an artifact. The reason is that the blink rate is associated with tasks of different level of difficulty, and this could be actually used as an advantage to successfully classify relaxation and concentration mental states (Bird et al., 2018). Also, if the users were instructed to do so, it could lead to their distress. Finally, they were asked to have their eyes open because in a real environment they would not close their eyes while navigating through the system.

At the predict phase, the two task types were presented side by side with a splitter in between. This was implemented in such a way so that the user would feel more in control. This way the user could select any mental state he/she wishes to. Also, if the algorithm would choose the task type instead of the user, then he/she may not feel trust in the algorithm. Additionally, the splitter usage was suggested to hide the irrelevant task. The reason was that as already mentioned, EEG signal is easily contaminated with eye movements so all tasks should always be presented in the same way. But, if the two tasks were presented side by side, user's eye movements could contaminate the signal. However, by using the splitter the task would always be in the same position.

Generally, the adaptations that were selected in this system were functional ones. The reason was that it would be easily noticed and evaluated by the user. Adapting the learning content would be a much more difficult choice and the user would not necessarily know that the learning content has changed among several others. Also, that would make the implementation of the system much more complicated as the learning content should be selected wisely by the adaptation algorithm. That would exceed the dissertation's goal and also add an extra burden of finding plenty of learning content to choose from. Moreover, the user would need much more time in order to actually study the learning content and see the adapted content.

Finally, the EEG data was collected (train) and analyzed (predict) online. No sophisticated signal processing techniques were used as they are more time-consuming and some of them need offline processing. The thinking was to use the simplest technique that would get reasonable results. The ML algorithm and parameters were chosen based first on other research papers and some tests. As for feature selection, only PSD was used as it is the most common feature (Al-Nafjan & Aldayel, 2022).

3.2.1 System Architecture

The system contains three basic components: a backend component (runs on the server), a web-client component (compiled at the server, run on the client) and an eeg-client component (runs on the client).

The backend component is responsible for all the communication between other components, for saving the raw EEG data to the Database (MySQL), for predicting the user's mental state (mental state algorithm) and returning it to the user. It is written in python using the Flask framework. Two technologies were used for the communication between server and clients. WebAPI, a classic unidirectional Application Programming Interface (API) (Ed-douibi et al., 2017) and WebSockets, a very fast bi-directional communication protocol (Srinivasan et al., 2013). WebAPI was used for operations such as saving or retrieving EEG data and WebSockets for operations like checking EEG connectivity, returning asynchronous error messages from the eeg device, start /stop recording of the eeg data and so on. There operations require a must faster protocol than WebAPI, that's why WebSockets were used. Another reason was that only WebSockets could open a communication channel between the world of JavaScript (web-client) and Python (eeg-client), due to the bi-directional ability of the protocol.

The web-client (written in Angular JavaScript framework) contains the user interface. The text, images, music tracks etc. The eeg-client is a python script that the user must run on his/her PC in order to establish a communication channel between the EEG device and the backend. This is responsible for fetching and sending the raw data from the EEG device to the backend when it is asked for. The EEG data is obtained using BrainFlow (Parfenov, 2022), a python library that can fetch, parse and even analyze EEG / EMG / ECG data from a list of available devices using a unified API.

The database that is used (MySQL) saves all necessary information unencrypted, such as the user's id, username, password, and email. This information is needed for the authentication mechanism to work (user register/login). System supported EEG devices are saved under the *device* table. The field *id* is the device id that is needed so that brainflow library can accurately recognize and connect to this device. Also, the name and the sampling rate of the device is stored at that table. This table comes predefined with 2 records for the Muse 2 EEG device: one version uses the BLED 112 USB dongle (Muse 2 with dongle), and one version that uses native Bluetooth (Muse 2). There is also one record for the test EEG device that was used for testing reasons. If more records were inserted at that table with their corresponding brainflow id, then the user would be able to use more EEG devices that are currently supported by brainflow. At the registration of a new user, a new record is saved at *user_setting* that connects this user (*user_id*) with the default EEG device, the Muse 2 (*device_id*). If this device is the BLED USB dongle version, the dongle port will also be needed (*dongle_port)*. However, the user can change the device from User Settings and/or set the current dongle port (COM7, COM9, etc) as seen by his/her computer settings.

id	name	sampling_rate
-1	Test device	250
22	Muse 2 (with dongle)	256
38	Muse 2	256

Table 2: Device table's predefined data.

The available mental states are saved under the table *mental_state* and come predefined with 3 records: *Unknown* with id = -1, *Relaxed* with id = 0 and *Concentrated* with id = 1. These records are useful for internal code organization (*eeg*.clasification_class, *user_mental_state*.mental_state_id, etc).

id	name	create_time
-1	Unknown	24/8/2022 21:55
0	Relaxed	24/8/2022 21:55
1	Concentrated	24/8/2022 21:55

Table 3: mental_state table's predefined data.

User's train and predict EEG data are saved under the tables: *eeg* and *user_eeg_train,* or *eeg* and *user_eeg_predict* accordingly. The *eeg* table has the following fields: *block_id*, a unique guid that references the EEG block. *brainflow_id* is an integer that is used internally by brainflow library. *electrode_1, electrode_2, electrode_3, electrode_4* are the raw EEG numeric values for each electrode. *classification_class* describes the mental state that is to be classified. Their values (-1, 0, 1) are already mentioned in the *mental_state* table. Finally, *brainflow_unix_time* is the timestamp of the recording in unix format.

So, when the user trains the algorithm, the tables *eeg* and *user_eeg_train* are saved with raw EEG data. The table *user_eeg_train* connects the data from the *eeg* table to a user (*user_id*) and with the device that was used at the time of the recording (*device_id*). The device is crucial to be included in the database at that point because every device has a different sampling rate that affects further calculations. The *device_id* is fetched from the table *user_setting* and the sampling rate is fetched table *device*. This raw data must be saved and processed so that it can be used as a ML feature and predict the user's mental states.

When the user runs the predict phase, the tables *eeg* and *user_eeg_predict* are saved with raw EEG data. The table *user_eeg_predict* connects the data from the *eeg* table with the same fields that were mentioned above, but with unknown (-1) *classification_class*. When the system calculates this block's mental state, this prediction is saved under the table *user_mental_state* for a specific user *(user_id)*, with the calculated mental state-prediction *(mental_state_id)*, for this specific block *(block_id)*. Based on the user's feedback, if this prediction was correct, the field *is_correct* will be true. This field is useful for debugging or any future statistical calculation. Finally, the latest record of a user's mental state is used from the web-client to change the indication of Available/Busy and to enable/disable Notifications.

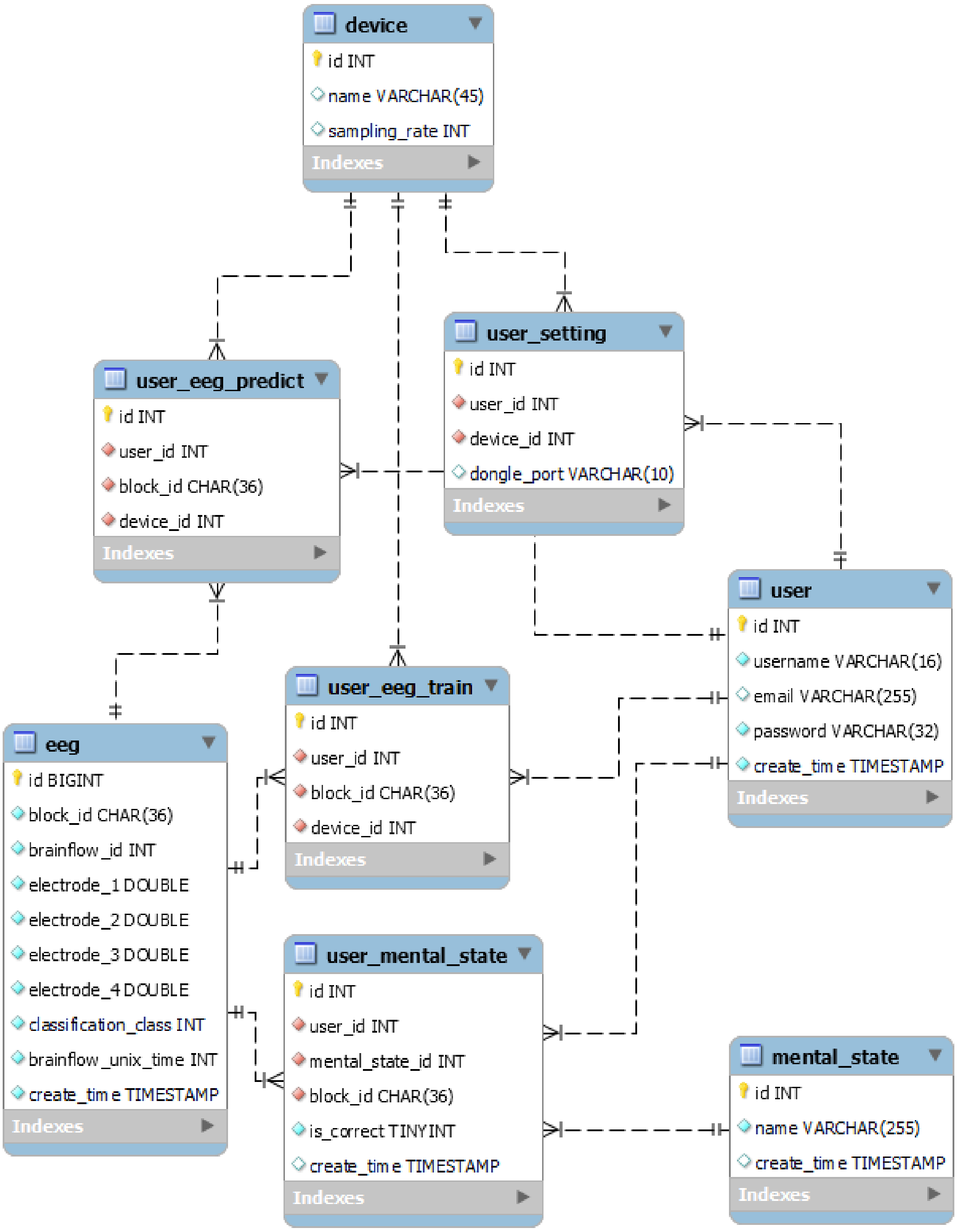

Figure 10: Database diagram.

3.2.2 Training flow

At the train phase the web-client will send a request to the backend using WebSockets ("eeg-start-training"). The backend will transfer this command to the eeg-client so that the device can start the recording of the raw EEG data. At this point the web-client will present the relaxing/concentrating task to the user. When the task is completed, a new WebSocket event will be emitted from the web-client ('eeg-stop-training') to the backend and then to eeg-client so that the EEG device can stop its recording and send the raw data back to the backend ('send_eeg_data()') and eventually be saved to the database.

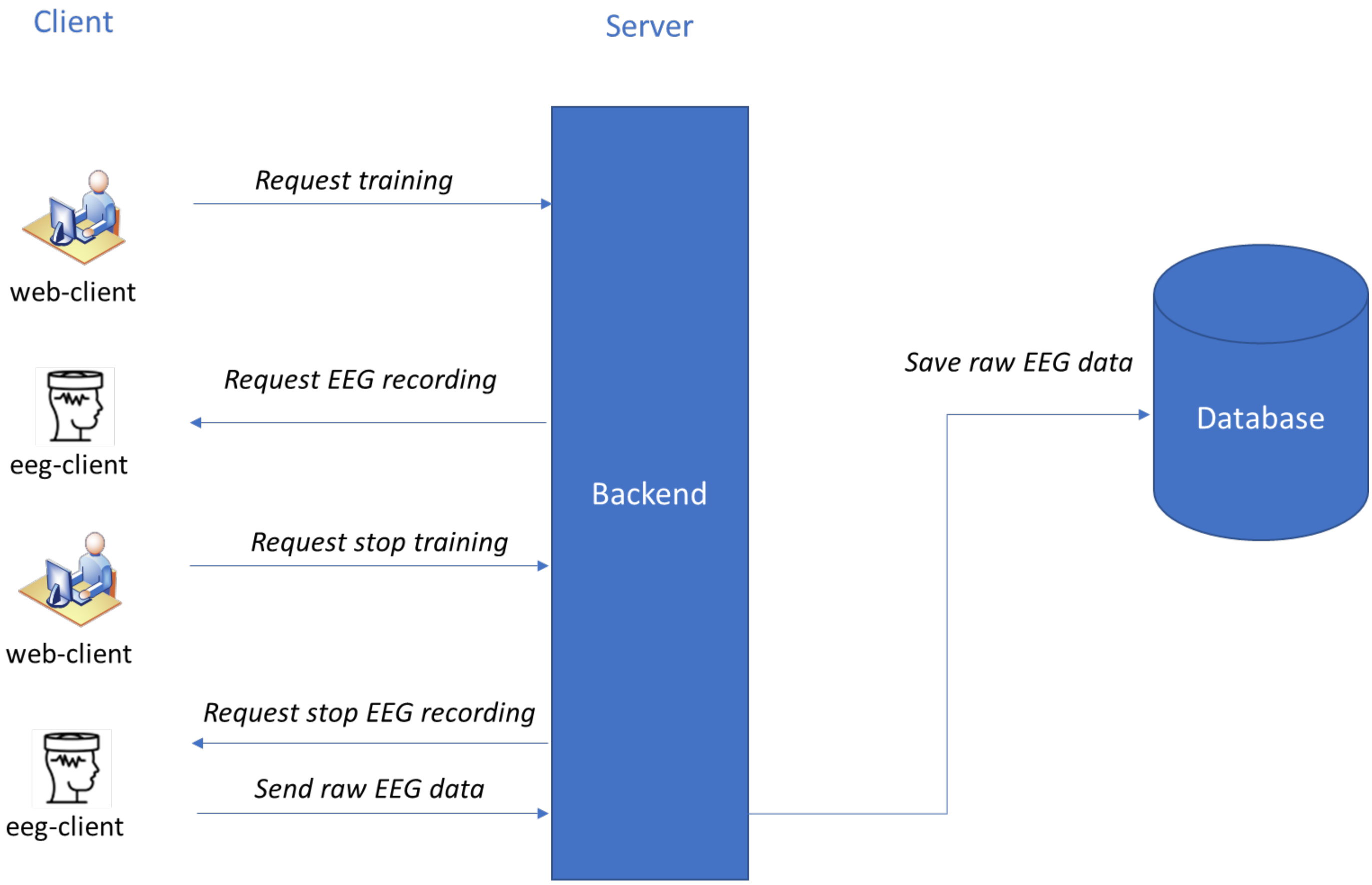

Figure 11: Training flow.

eeg-client: The event of start / stop training will change the global variable "predicting" to false and "streaming" to true so that the function "eeg_stream()" can know when to start/stop the recording of the EEG data. This recording is done using *brainflow's functions board.start_stream(), board.get_board_data(), board.stop_stream(). When the recording is completed, for debugging reasons the data is also saved in a csv files and finally the function send_eeg_data() is called so that the data are saved to the database using a WebAPI call.*

```
@sio.on('eeg-start-training')
async def on_training(stringObject):
    print('I received to Start Training')
    global streaming
    if streaming is not True:
        print(stringObject)
        data = json.loads(stringObject)
        block_id = data["blockId"]
        classifier = data["classifier"]

        await start_training(block_id, classifier)
    else:
        print('Already streaming so not doing anything')


@sio.on('eeg-stop-training')
async def on_message():
    print('I received to Stop Training')
    await stop_training_or_predicting()



async def start_training(block_id, classifier):
    print("At Start Training")
    global predicting
    global streaming
    predicting = False
    streaming = True
    await eeg_stream(block_id, False, classifier)


async def eeg_stream(block_id, is_predict, current_classifier):
    print('at eeg_stream, block id: %s' %block_id)
    print('current_classifier: %s' %current_classifier)
    global board_id, streaming
    lock.acquire()
    time_index = BoardShim.get_timestamp_channel(board_id)
    # Add eeg_channels + timestamp + markers
    time_eeg_channels = [e for i, e in enumerate(EEG_CHANNELS)]
    # insert sample/package number
```

```
    package_num_channel = BoardShim.get_package_num_channel(board_id)
    time_eeg_channels.insert(package_num_channel,package_num_channel)
    # insert timestamp channel
    time_eeg_channels.append(time_index)

    global board
    try:
        if board is None or (board is not None and board.is_prepared()
== False):
            print('Board is not prepared! Exiting..')
            raise Exception("Board is not prepared! Exiting..")
        BoardShim.log_message(LogLevels.LEVEL_DEBUG.value, 'Start
streaming..')
        board.start_stream(499999)

        while(streaming):
            await asyncio.sleep(0.1)

        print('Getting overall board data..')
        all_board_data = board.get_board_data()
        board.stop_stream()

        signal_length = all_board_data.shape[1]
        print('Signal length is %s' %signal_length)
        global sampling_rate
        # Check signal length to make sure you got 30/60 seconds of
data depending on the classifier
        if(current_classifier == 0 and signal_length < (sampling_rate *
60) - 700):
            raise Exception("Signal length is less than 60 seconds
(%s)! Please check Muse connectivity!" %(signal_length / sampling_rate)
)
        if(current_classifier == 1 and signal_length < (sampling_rate *
30) - 700):
            raise Exception("Signal length is less than 30 seconds
(%s)! Please check Muse connectivity!" %(signal_length /
sampling_rate))
        if(is_predict and signal_length < (sampling_rate * 30) - 700):
            raise Exception("Signal length is less than 30 seconds
(%s)! Please check Muse connectivity!" %(signal_length /
sampling_rate))

        # Get only EEG and time channels
        all_board_data = all_board_data[time_eeg_channels, :]

        label_column = None
        # block response instead of current classifier
        # append class name next to it
```

```
        print('current_classifier: %s' %current_classifier)
        if(current_classifier == 0):
            # append column with '0'
            label_column = np.zeros((1,
len(np.transpose(all_board_data))),dtype=int)
        elif(current_classifier == 1):
            # append column with '1'
            label_column = np.ones((1,
len(np.transpose(all_board_data))),dtype=int)
        else: # append column with '-1' when label is unknown
            label_column = np.full((1,
len(np.transpose(all_board_data))), -1, dtype=int)


        print('Adding data...')
        all_board_data = np.append(all_board_data, label_column,
axis=0)
        print('Writing raw csv...')
        folder = 'train'
        if(is_predict == True):
            folder = 'predict'
        filename = './raw/' + folder + '/' + str(block_id) +
'.csv'
        DataFilter.write_file(all_board_data, filename, 'a')  # use 'a'
for append mode

        print('Written')
        print('Will send_eeg_data')
        # Send EEG data to backend service
        await send_eeg_data(all_board_data, is_predict, block_id)

        print('will emit eeg-completed-streaming')
        await sio.emit('eeg-completed-streaming', block_id)
        print('Completed!')
    except Exception as error:
        print(error)
        errorMessage = error.args
        await sio.emit('eeg-streaming-error', 'eeg_stream error: %s'
%errorMessage)
    finally:
        lock.release()
        print('after lock.release')
        # Stop indication of streaming
        streaming = False

async def send_eeg_data(eeg_data, is_predict, block_id):
    global token
    # Numpy is not serializable so convert it into a list
```

```
    print('Posting stream to server...')
    url = SERVER_URL + '/SaveEEG?block_id=' + str(block_id) + 
'&predict=' + str(is_predict)
    print(url)

    response = requests.post(url, json=eeg_data.tolist(), 
headers={'Authorization': 'Bearer %s' %token})
    # Raise an error for 4xx or 5xx responses
    #response.raise_for_status()
    if response.status_code > 400 and response.status_code < 600:
        data = response.json()
        errorMessage = data.get('message')
        raise Exception(errorMessage)
    print('Done!')
```

3.2.3 Predict flow

At the predict phase the web-client will send a request to the backend using WebSockets ('eeg-start-predicting'). The backend will transfer this command to the eeg-client so that the device can start the recording of the raw EEG data. At this point the web-client will present both the relaxing and the concentrating tasks to the user. When the task is completed, a new WebSocket event will be emitted from the web-client to the backend ('eeg-stop-predicting') so that the EEG device can stop its recording and send the raw data back to the backend and eventually be saved to the database. Finally, web-client will send a new request to the backend to get the prediction (WebAPI call 'PredictBlock'). The backend will calculate the mental state from the saved raw EEG data from the database and return the result to the user.

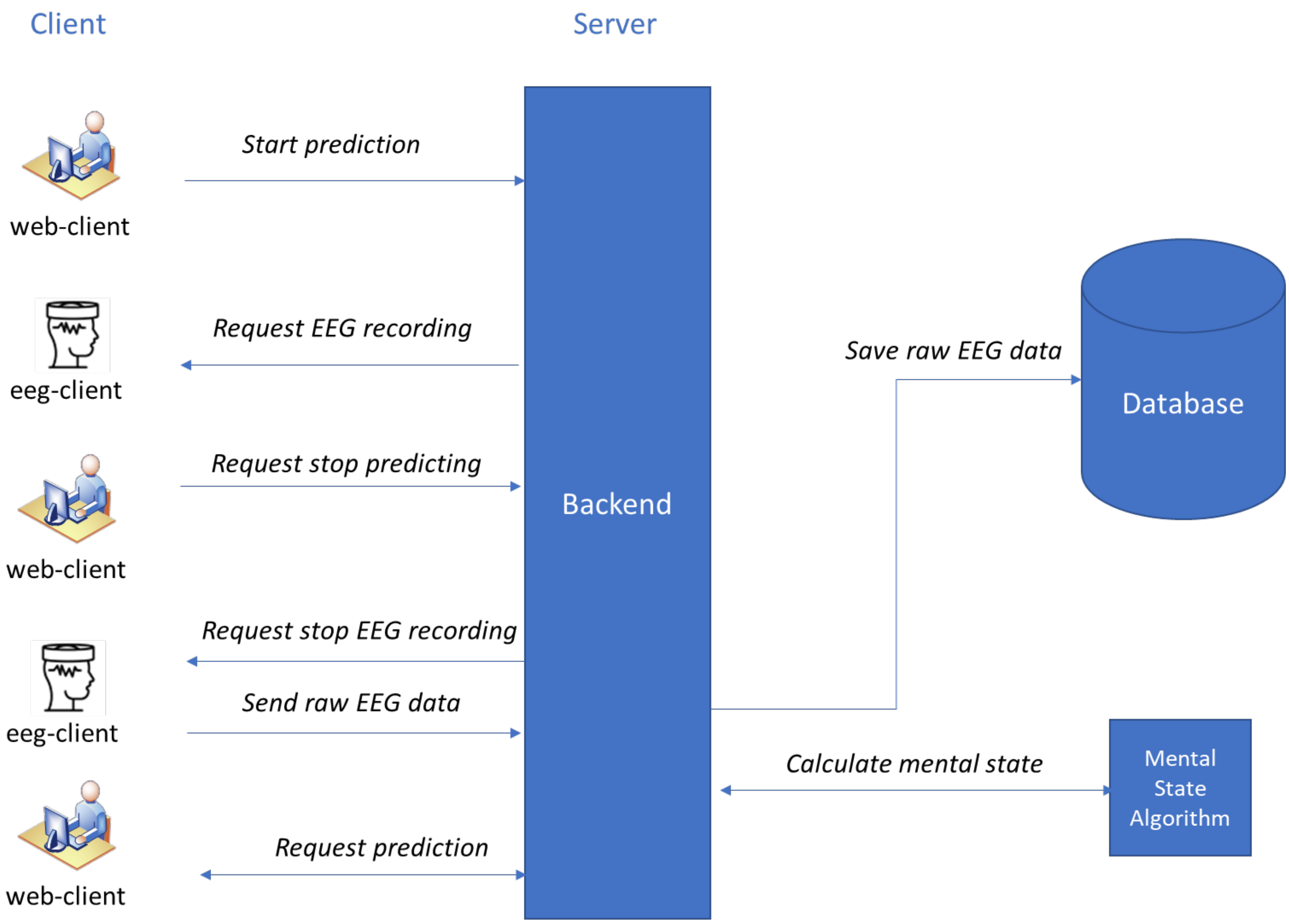

Figure 12: Predict flow.

The mental state algorithm uses BrainFlow's functions to add filters to the signal. More specifically, a bandpass between 7 and 59Hz is added and a bandstop between 48-52Hz is applied in order to prevent Europe's noise line (Peng, 2019). Then, the Power Spectrum Density is calculated for each EEG channel using Welch's method with a hamming windowing function based on (Shoorangiz et al., 2021). Then, the relative band powers are calculated with 1 Hz band width from 7-59Hz (alpha, beta, gamma frequencies) slightly tweaked from You's study (You, 2021) that used 9-43Hz with 4 Hz band width and use this result as a feature in the classification. This is calculated for every 7 seconds of the recorded EEG data with 50% overlap. This window of 7 seconds was selected arbitrarily due to the longer duration of the recording than that of You's study. The 50% overlap was used as it is generally more common (Apicella et al., 2022; Chaouachi & Frasson, 2012; Tremmel et al., 2019). Eventually, Random Forest algorithm used this feature to calculate the prediction. Random forest's specific parameters (n_estimators=100, criterion="entropy", max_depth=10, min_samples_split=6) were taken from Bellamy's study (Bellamy, 2021). Random Forest was selected because it was used when classifying concentration/relaxation on Edla's (Edla et al., 2018) and Birds' study (Bird et al., 2018).

eeg-client: The event of start / stop predicting will change the global variable "predicting" to true and "streaming" to true so that the function "eeg_stream()" can know when to start/stop the recording of the EEG data. This is the same function for train/predict phase.

```
@sio.on('eeg-start-predicting')
async def on_predicting(block_id):
    print('I received to Start predicting')
    print('Block id: %s' %block_id)
    global streaming

    if streaming is not True:
        await start_predicting(block_id)
    else:
        print('Already streaming so not doing anything')
```

```
async def start_predicting(block_id):
    print("At Start Predicting")

    global streaming
    global predicting
    predicting = True
    streaming = True
    await eeg_stream(block_id, True, None)


@sio.on('eeg-stop-predicting')
async def on_message():
    print('I received to Stop Predicting')
    await stop_training_or_predicting()


async def stop_training_or_predicting():
    print('At Stop Training/Predicting')
    global streaming
    global predicting
    streaming = False
    predicting = False
```

backend's predict WebAPI call: Given the block id and the current user, fetch current EEG device sampling rate (from database), get all training data of this user (from database), get the predict data (from database) of this block, make sure there are least 9 seconds of predict data and call 'predict_block_from_db()' function of mental state algorithm to predict the mental state.

```
@app.route('/PredictBlock', methods=['GET'])
@cross_origin(origin='*')
@token_required
def on_predict_block(current_user_id):
  print('on_predict_block')
  print('Current user: %s' %current_user_id)

  prediction = -1 # Unknown
  train_data_array = []
  predict_data = []
  predict_sampling_rate = 0
  train_sampling_rate = 0

  args = request.args
```

```
  block_id = args.get("block_id")

  if(block_id is None):
    raise InvalidAPIUsage("Block Id not found!")

  try:
    with closing(mysql.connection.cursor()) as cursor:
      print('Fetching predict sampling rate..')
      # Get training data sampling rate
      cursor.execute("SELECT DISTINCT(sampling_rate) FROM
user_eeg_predict INNER JOIN device ON device.id =
user_eeg_predict.device_id WHERE user_id = %s AND block_id = %s",
(current_user_id,block_id))
      predict_sampling_rate_sql = cursor.fetchall()
      if predict_sampling_rate_sql is None or
len(predict_sampling_rate_sql) == 0:
        raise InvalidAPIUsage("Could not found prediction record!")

      if len(predict_sampling_rate_sql) == 1:
        predict_sampling_rate = predict_sampling_rate_sql[0]
        print('Predict sampling rate: %s' %predict_sampling_rate)
      else:
        raise InvalidAPIUsage("Predict sampling rate has many different
rates!")

      # Tuple to int
      predict_sampling_rate = int(predict_sampling_rate[0])

      print('Fetching predict data..')
      cursor.execute("SELECT eeg.* FROM user_eeg_predict INNER JOIN eeg
ON user_eeg_predict.block_id = eeg.block_id WHERE user_id = %s AND
user_eeg_predict.block_id = %s ORDER BY eeg.id ASC",
(current_user_id,block_id))

      # Fetch one record and return result
      predict_eeg_data_sql = cursor.fetchall()
      row_headers=[x[0] for x in cursor.description] #this will extract
row headers
      train_rows = []
      if predict_eeg_data_sql is not None and len(predict_eeg_data_sql)
> 0:
        for eeg_data in predict_eeg_data_sql:
          eeg_train_item_data = dict(zip(row_headers,eeg_data))
          train_rows.append([eeg_train_item_data.get('brainflow_id'),
eeg_train_item_data.get('electrode_1'),
eeg_train_item_data.get('electrode_2'),
eeg_train_item_data.get('electrode_3'),
eeg_train_item_data.get('electrode_4'),
```

```
eeg_train_item_data.get('brainflow_unix_time'),
eeg_train_item_data.get('classification_class')])

      predict_data = np.array([x for x in train_rows]).transpose()
      print('Fetching train sampling rate..')
      # Get training data sampling rate
      cursor.execute("SELECT DISTINCT(device.sampling_rate) FROM
user_eeg_train INNER JOIN eeg ON user_eeg_train.block_id = eeg.block_id
INNER JOIN device ON device.id = user_eeg_train.device_id WHERE user_id
= %s", (current_user_id,))
      train_sampling_rate_sql = cursor.fetchall()

      if train_sampling_rate_sql is None or
len(train_sampling_rate_sql) == 0:
        raise InvalidAPIUsage("Could not found train record!")

      if len(train_sampling_rate_sql) == 1:
        train_sampling_rate = train_sampling_rate_sql[0]
        print('Train sampling rate: %s' %train_sampling_rate)
      else:
        raise InvalidAPIUsage("Train sampling rate has many different
rates!")

      # Tuple to int
      train_sampling_rate = int(train_sampling_rate[0])

      print('Fetching train data..')
      # Get all train blocks for this user
      cursor.execute("SELECT block_id FROM user_eeg_train WHERE user_id
= %s", (current_user_id,))
      # Fetch one record and return result
      train_eeg_blocks_sql = cursor.fetchall()

      if train_eeg_blocks_sql is None or len(train_eeg_blocks_sql) ==
0:
        raise InvalidAPIUsage("Could not found user_eeg_train
records!")

      for train_eeg_block in train_eeg_blocks_sql:
        cursor.execute("SELECT * FROM eeg WHERE block_id = %s ORDER BY
eeg.id ASC", (train_eeg_block[0],))

        # Get all training EEG data!
        eeg_data_sql = cursor.fetchall()
        row_headers=[x[0] for x in cursor.description] #this will
extract row headers

        if eeg_data_sql is None or len(eeg_data_sql) == 0:
```

```
          raise InvalidAPIUsage("Could not found eeg records!")

        predict_rows = []
        for eeg_data in eeg_data_sql:
          eeg_item_data = dict(zip(row_headers,eeg_data))
          row = [eeg_item_data.get('brainflow_id'),
eeg_item_data.get('electrode_1'), eeg_item_data.get('electrode_2'),
eeg_item_data.get('electrode_3'), eeg_item_data.get('electrode_4'),
eeg_item_data.get('brainflow_unix_time'),
eeg_item_data.get('classification_class')]
          predict_rows.append(row)

        nd_array = np.array([x for x in predict_rows]).transpose()
        train_data_array.append(nd_array)
      print('Fetched %s rows of train data' %len(train_data_array))
      print(predict_data.shape[1])
      if predict_data.shape[1] < 2000: # 9 seconds of predict data
        raise InvalidAPIUsage("At least 9 seconds of predict data are
required!")

      print('Will run predict_block_from_db..')
      prediction = predict_block_from_db(train_data_array,
train_sampling_rate, predict_data, predict_sampling_rate)

    print('Prediction: %s' %prediction)
  except InvalidAPIUsage as error:
    print('InvalidAPIUsage occured: %s' %error)
    raise
  except Exception as error:
    print('An Error occured: %s' %error)
    raise InvalidAPIUsage(error.args)

  return make_response(jsonify(prediction), 201)
```

mental state algorithm: Prepare the train and predict data. Remove all rows that have zeroes in all 4 channels (function remove_amplitude_thresholds). Also, remove the first 5 seconds of the predict data because the user may need some extra time to decide which task type to follow (concentration or relaxation).

```
def predict_block_from_db(train_data, train_streaming_rate,
predict_data, predict_streaming_rate):
    prediction = -1
```

```
    train_data = prepare_train_data_db(train_data,
train_streaming_rate)
    new_data = prepare_new_data_db(predict_data,
predict_streaming_rate)
    prediction = predict_rf(train_data, new_data, False)
    # turn numpy.int64 to plain int
    return int(prediction)




def prepare_train_data_db(all_train_data, sampling_rate):
    try:
        dataset_x = list()
        dataset_y = list()
        for data in all_train_data:

            if (REMOVE_AMPLITUDES == True):
                data = remove_amplitude_thresholds(data)

            cur_pos = sampling_rate
            length = data.shape[1]
            while cur_pos + int(WINDOW_SIZE * sampling_rate) < length:

                data_in_window = data[DATA_INDEX, cur_pos:cur_pos +
int(WINDOW_SIZE * sampling_rate)]
                feature_vector = extract_features(data_in_window,
sampling_rate)
                dataset_x.append(feature_vector)
                # Fetch data in that window and check the it's
classifier.
                # I am supposing here that for every window the
classifier will be the same
                classifier = int(data_in_window[CLASSIFICATION_INDEX,
0])
                dataset_y.append(classifier)

                cur_pos = cur_pos + int(WINDOW_SIZE * OVERLAPS *
sampling_rate)
        return dataset_x, dataset_y
    except Exception as error:
        print('prepare_train_data_db: %s' %error)
        raise


def prepare_new_data_db(data, sampling_rate):
    try:
        dataset_x = list()
```

```
        if (REMOVE_AMPLITUDES == True):
            data = remove_amplitude_thresholds(data)

            cur_pos = sampling_rate * 5 # start after 5 seconds just to
be sure everything is ok
            length = data.shape[1]
            while cur_pos + int(WINDOW_SIZE * sampling_rate) < length:
                data_in_window = data[DATA_INDEX, cur_pos:cur_pos +
int(WINDOW_SIZE * sampling_rate)]
                feature_vector = extract_features(data_in_window,
sampling_rate)
                dataset_x.append(feature_vector)

                cur_pos = cur_pos + int(WINDOW_SIZE * OVERLAPS *
sampling_rate)
        return dataset_x
    except Exception as error:
        print('prepare_new_data_db: %s' %error)
        raise

def remove_amplitude_thresholds(data):
    # Remove if all EEG channels are 0
    data = data[:, ~np.all(data[1:4,:] == 0, axis=0)]
    res_data = np.copy(data, order='C')

    return res_data
```

mental state algorithm: Convert the list of features to a np array:

```
def extract_features(data_in_window, sampling_rate):
    band_features = extract_band_features(data_in_window, EEG_CHANNELS,
sampling_rate, True)
    feature_vector = np.array(band_features)

    return feature_vector
```

mental state algorithm: Use basic signal processing filtering (BrainFlow) and get relative band powers as a list of features:

```
def extract_band_features(data, eeg_channels, sampling_rate,
applyFilters):
```

```
    nfft = DataFilter.get_nearest_power_of_two(sampling_rate)
    features = list()
    for channel in enumerate(eeg_channels):
        if(applyFilters):
            # Band stop  48-52 Hz
            DataFilter.perform_bandstop(data[channel], sampling_rate,
48.0, 52.0, 4, FilterTypes.BUTTERWORTH.value, 0)
            # Band pass 7 - 59 Hz
            DataFilter.perform_bandpass(data[channel], sampling_rate,
7.0, 59.0, 4, FilterTypes.BUTTERWORTH.value, 0)

        psd = DataFilter.get_psd_welch(data[channel], nfft, nfft // 2,
sampling_rate, WindowOperations.HAMMING.value)
        sum = DataFilter.get_band_power(psd, 7, 59)

        for x in range(7, 59):
            freq = x
            abs = DataFilter.get_band_power(psd, freq, freq + 1)
            features.append(abs / sum)
            freq = x + 1
    return features
```

mental state algorithm: Predict for every slice of data (window) using random forest and choose the prediction that is more dominant:

```
def predict_rf(train_data, new_data, report_accuracy):
    classifier = RandomForestClassifier(n_estimators=100,
criterion="entropy", max_depth=10, max_features="auto",
min_samples_split=6, random_state=137)
    if(report_accuracy):
        predict_accuracy(classifier, 'Random Forest', train_data)

    return predict(classifier, 'Random Forest', train_data, new_data)

def predict(model, modelName, train_data, new_data):
    model.fit(train_data[0], train_data[1])

    y_pred = model.predict(new_data)
    y = choose_best_prediction(y_pred)

    return y

def choose_best_prediction(prediction):
    print(prediction)
```

```python
output = None
if prediction is not None:
    counts = np.bincount(prediction)
    output = np.argmax(counts)
    print(output)

return output
```

Chapter 4
Evaluation

4.1 Evaluation Settings

This dissertation was evaluated based on the User Experience (UX), which is considered one of the most valuable factors of a product's / system's quality. UX is the user's emotional and psychological responses that results from the product's use (Díaz-Oreiro et al., 2019). So, a pleasant and organized User Interface, is as important as the system's main idea, performance, and functionality.

For this evaluation 10 users were asked to participate voluntarily and use the system that was already deployed in a local computer. Users were not asked to control their sleep or avoid caffeine or alcohol consumption. For each user an account was created by the dissertation writer that was already logged in. Before using the system, the users were informed about dissertations goal, adaptive systems, EEG, and the tasks that they should perform through the system. No demo was provided. The only additional help that was provided was at the steps of preparing, wearing and signal quality as this was the most important part of the system to ensure a proper train and predict phase. More specifically, the user was asked to follow the instructions of the platform and do the following: read the instructions – video, prepare and wear the EEG device and check its signal quality. Finally, train and test (predict) the algorithm. This corresponded to all chapters/links that were included in the system. When all tasks were completed, the user was asked to fill in a questionnaire. This step was very important to be performed immediately after the use of the system so that the user could successfully express his/her immediate response. The first section of the questionnaire contained demographic questions such as age, gender, occupation etc. The second section contained questions about the platform's functionalities to ensure that all the aforementioned tasks were completed successfully. Finally, the third section contained the User Experience Questionnaire (UEQ), which is one of the most popular standardized questionnaires of UX (Díaz-Oreiro et al., 2019). It contains 26 questions in randomized order with 7-stage scales, starting from -3 (most negative) to +3 (most positive), with 0 being a neutral answer. Half of the questions have a positive meaning and half of them a negative one. It is shown to have good construct validity of scales and high scale consistency (Schrepp, 2019).

UEQ measures UX in 6 different scales: Attractiveness, efficiency, perspicuity, dependability, stimulation, and novelty. Attractiveness is how pleasant the product is. Efficiency is how easily and fast the user accomplished the system's tasks. Perspicuity is how easy it is for the user to learn and understand the product. Dependability is how secure and predictable (in a positive sense) the product is felt. Stimulation is how interesting and motivating the product is. Finally, novelty is how creative and innovative the product is (Schrepp, 2019).

4.2 Results

All ten users that participated in this evaluation used the system and evaluated it successfully. Approximately, 1 hour was needed for the whole process. Six users were female and four were male.

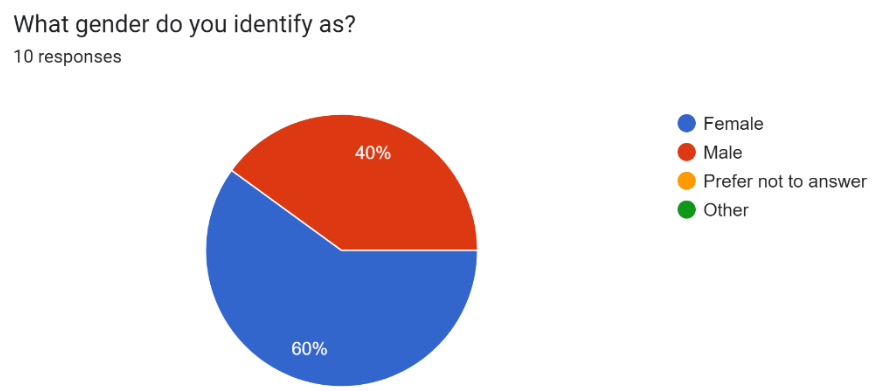

Figure 13: Gender.

Most of the participants were in the age of 30-39 years old (60%), 20% of them were 20-29 years old, 10% of them were 40-49 years old and 10% were 50+ years old. All participants native language was Greek.

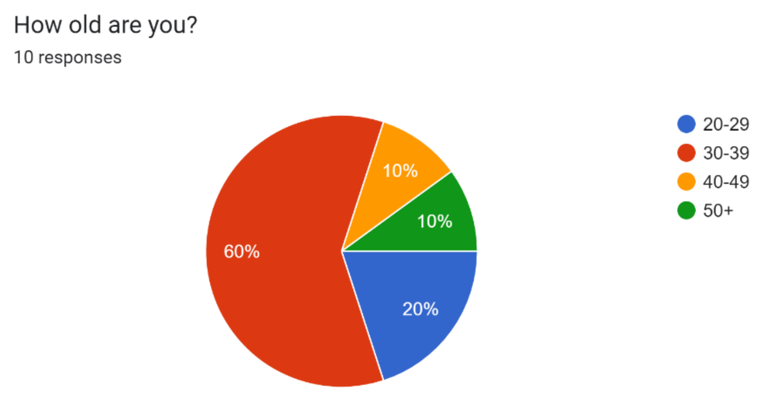

Figure 14: Age.

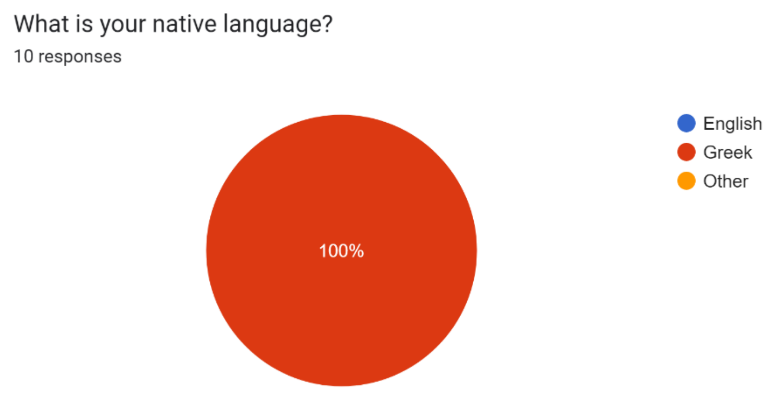

Figure 15: Native Language.

The 70% of the participants had a bachelor's degree and 30% of them had a master's degree. More specifically, 30% of them had a Computer Science degree, 20% of them had a medical-related degree, 20% of them had an engineering degree, 10% of them had an education-related degree, 10% of them had an economics degree, and 10% of them was in other area.

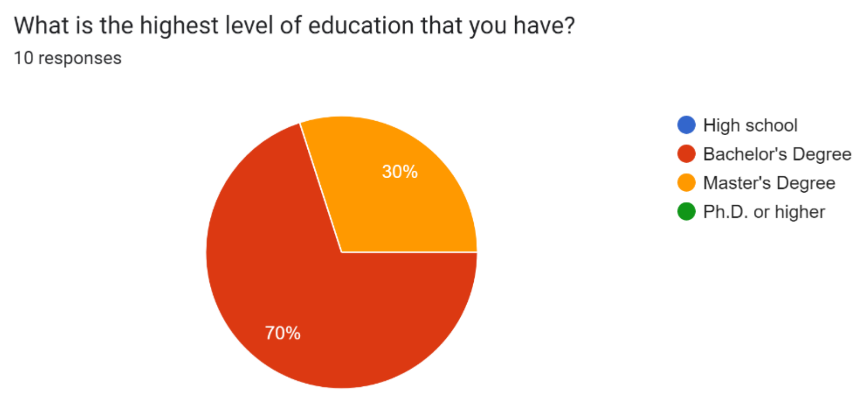

Figure 16: Education level.

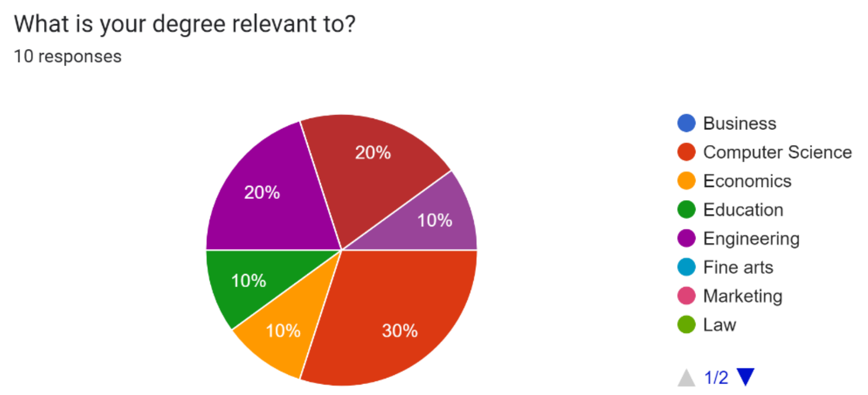

Figure 17: Degree area.

Almost all participants (90%) had a full-time job, and one participant (10%) had retired.

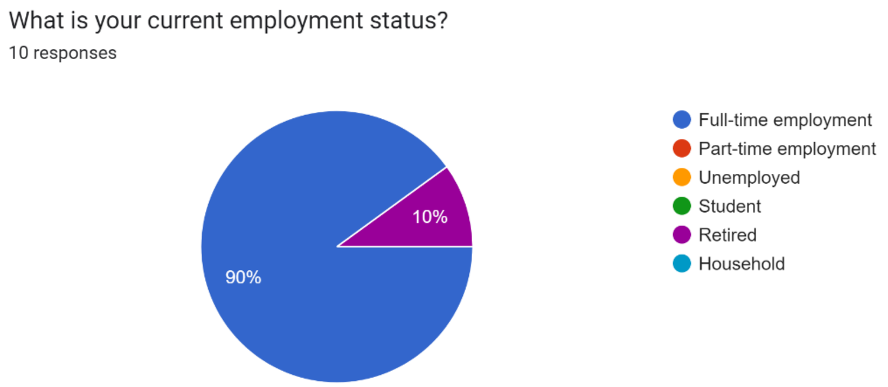

Figure 18: Employment status.

The 80% of the participants were married, and only the 10% of them had children.

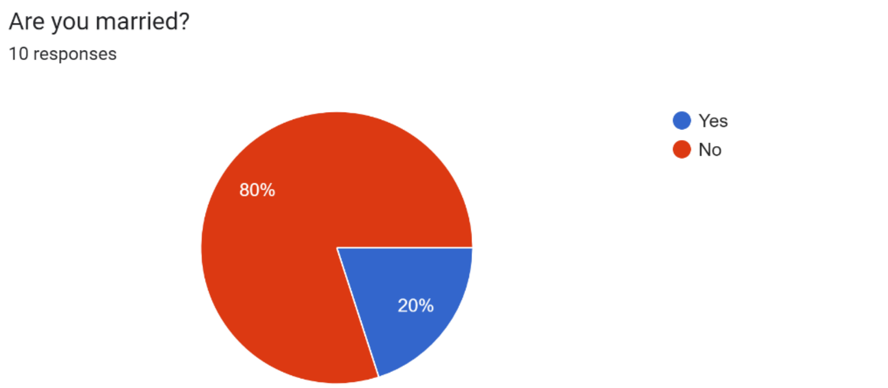

Figure 19: Marital Status.

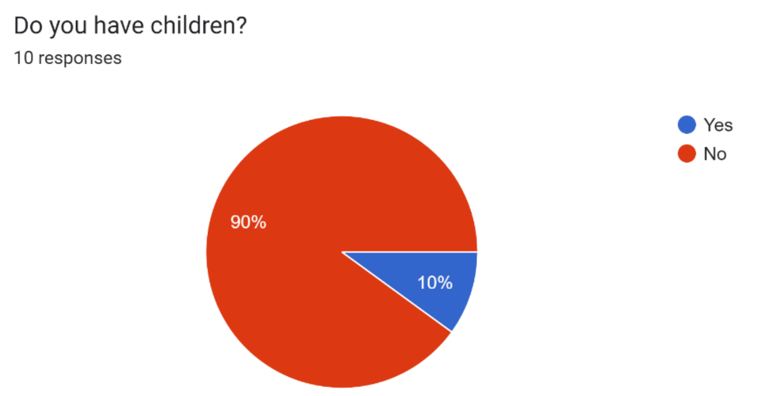

Figure 20: Children.

All participants were familiar with technology (100%), and 80% of them is using wearable devices.

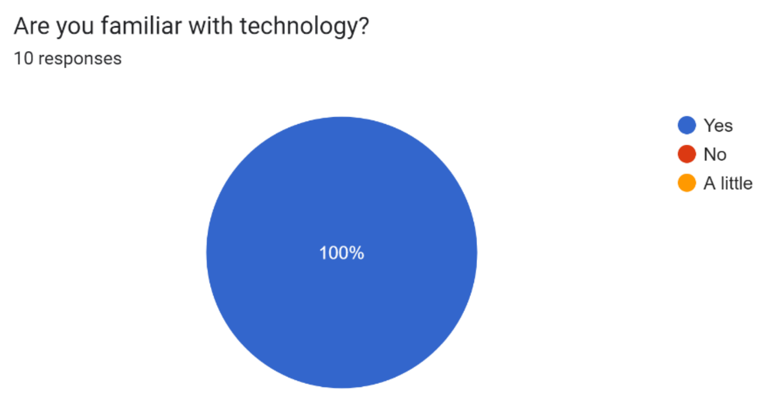

Figure 21: Technology familiarity.

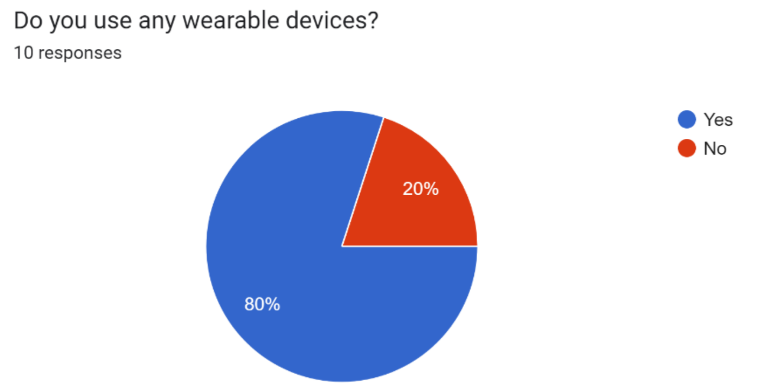

Figure 22: Wearable devices.

Half of the participants uses e-learning platforms frequently, 30% of them not that frequently, and 20% of them have never used online learning platforms. Additionally, only 30% of the users were already familiar with the concept of EEG.

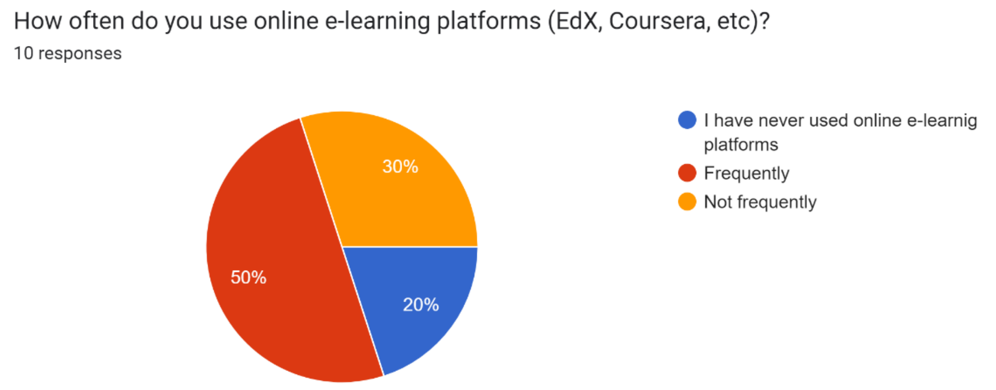

Figure 23: E-learning platforms.

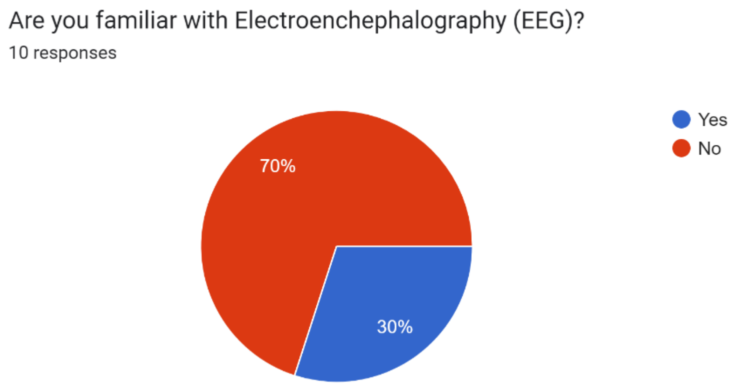

Figure 24: EEG familiarity.

All participants considered that the instructions inside the system were complete (90% strongly agreed, 10% agreed).

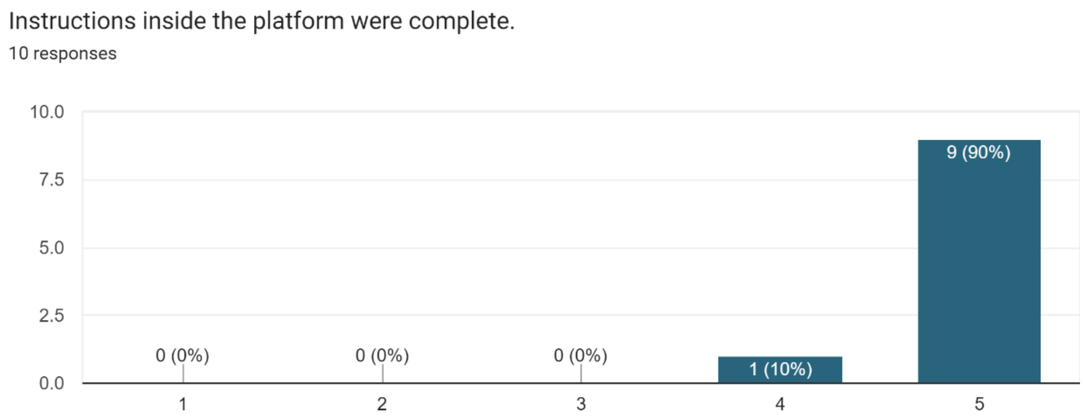

Figure 25: Instructions.

All participants agreed that the EEG device was prepared successfully (70% strongly agreed, 30% agreed).

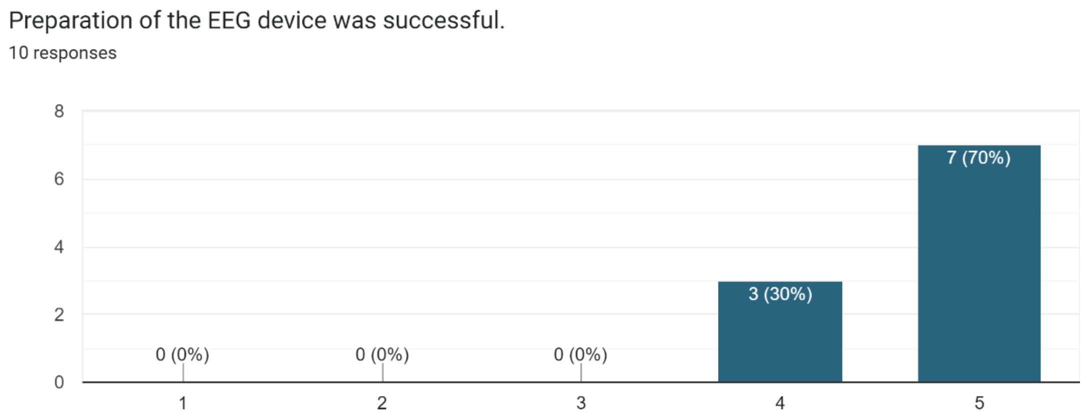

Figure 26: Preparation of the EEG device.

All participants agreed that they performed a successful signal quality check. (80% strongly agreed, 20% agreed).

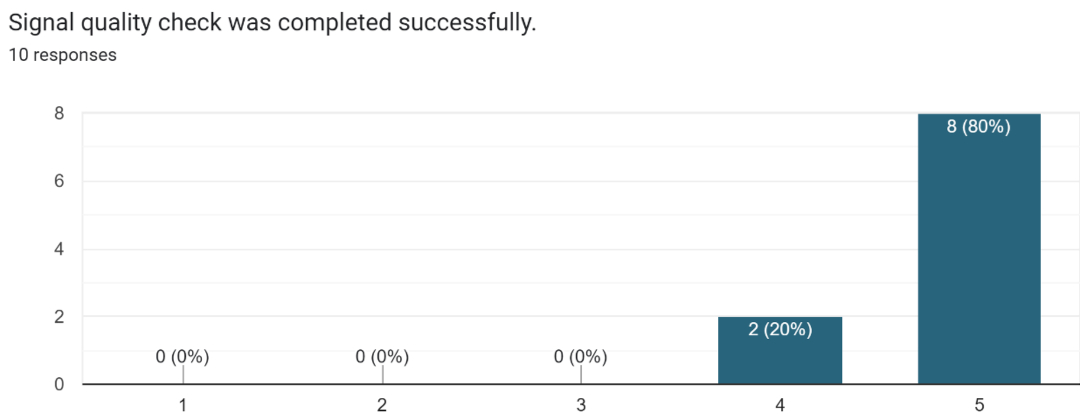

Figure 27: Signal Quality.

All participants completed training and predict phase successfully (90% strongly agreed, 10% agreed).

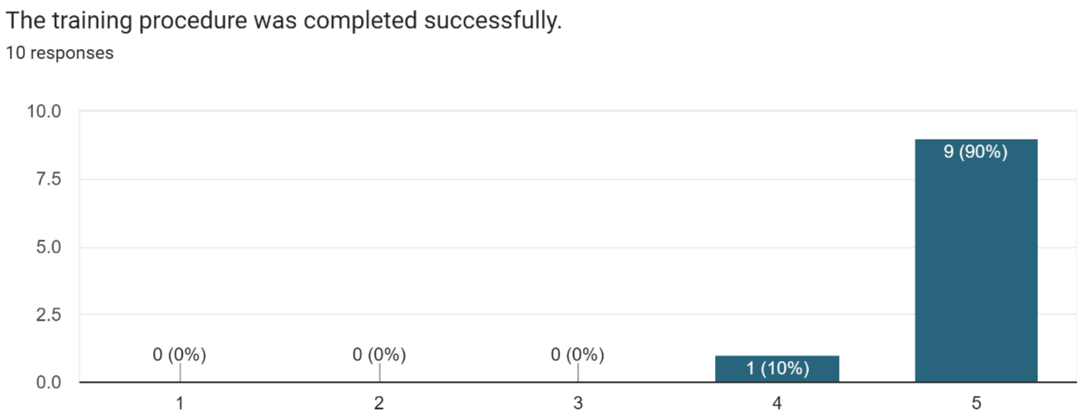

Figure 28: Training procedure.

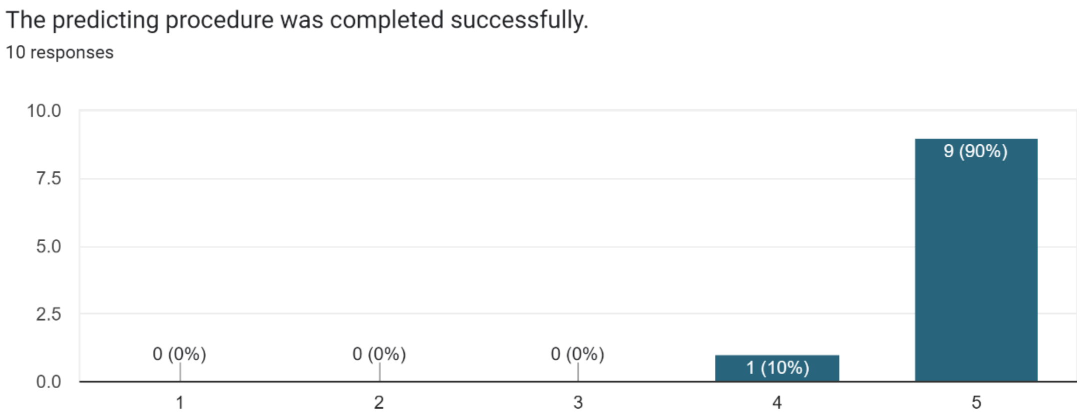

Figure 29: Predict procedure.

The 40% of the participants highly agreed that the predictions made by the system were accurate. The 30% of the participants agreed to that also. The 20% of them were neutral about their response and the 10% of the participants did not agree that the predictions were accurate.

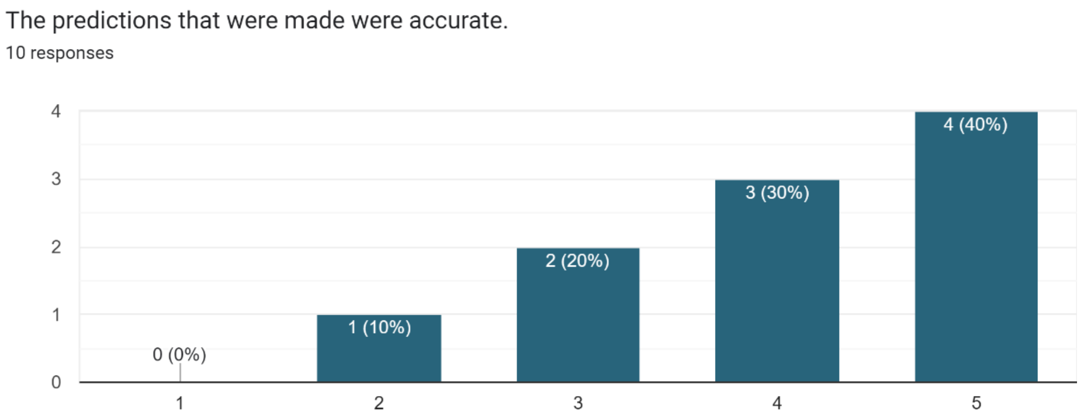

Figure 30: Predictions.

Most of the users agreed that any error that occurred in the system was clear (60% strongly agreed, 10% agreed). The 20% of the participants were neutral about their response and the 10% of them responded that the error was not clear.

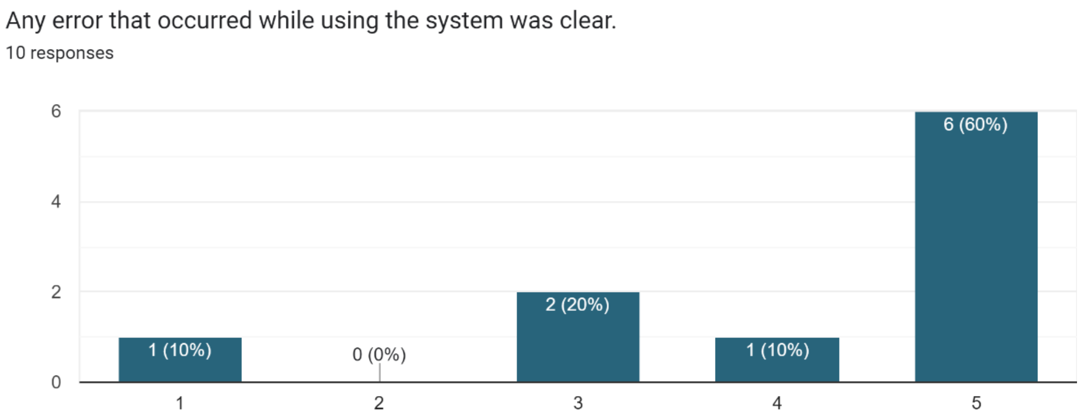

Figure 31: Errors.

Most of the users found that the adaptations of the system would be practical in a real environment (60% strongly agreed, 30% agreed). The 10% of the users did not find them practical.

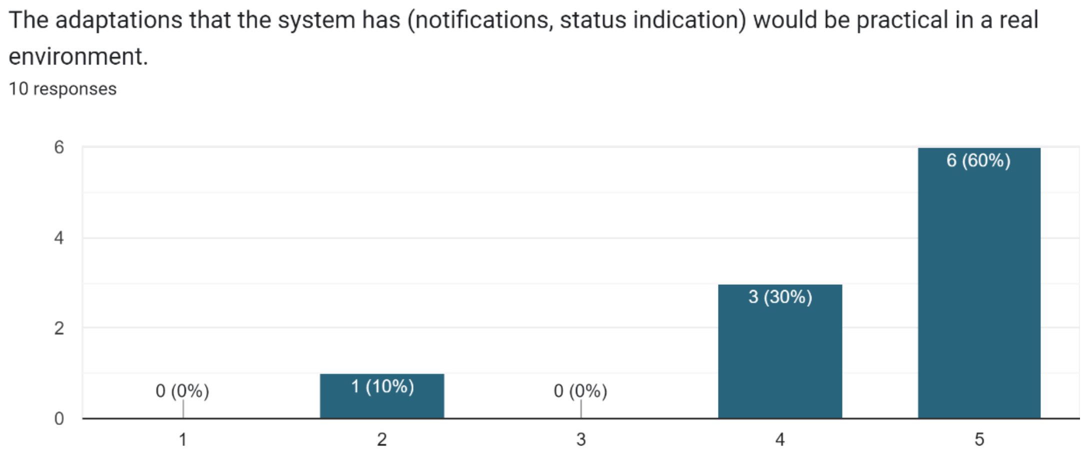

Figure 32: Practical adaptations.

Most of the participants agreed that the EEG device was comfortable (50% strongly agreed, 30% agreed). The 20% of the users disagreed.

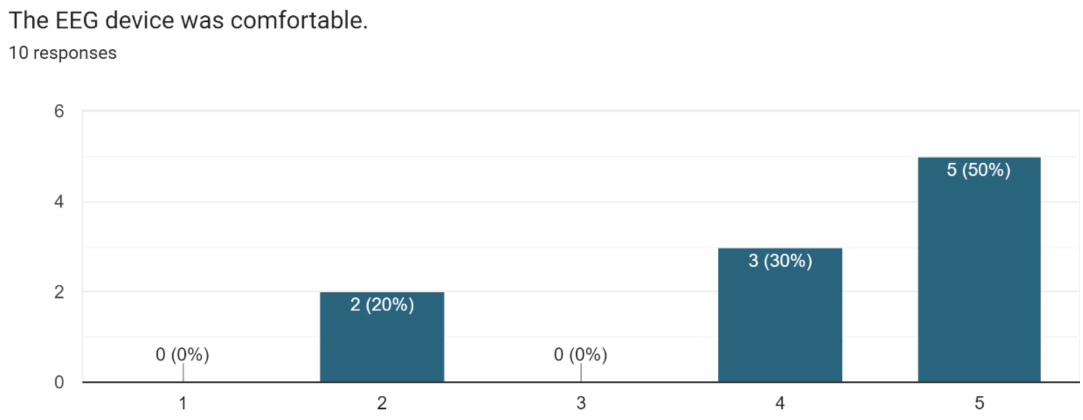

Figure 33: EEG device comfort.

Finally, most of the participants claimed that they would use a system such as the one created in this dissertation (40% strongly agreed, 40% agreed). The 20% of them were neutral about this.

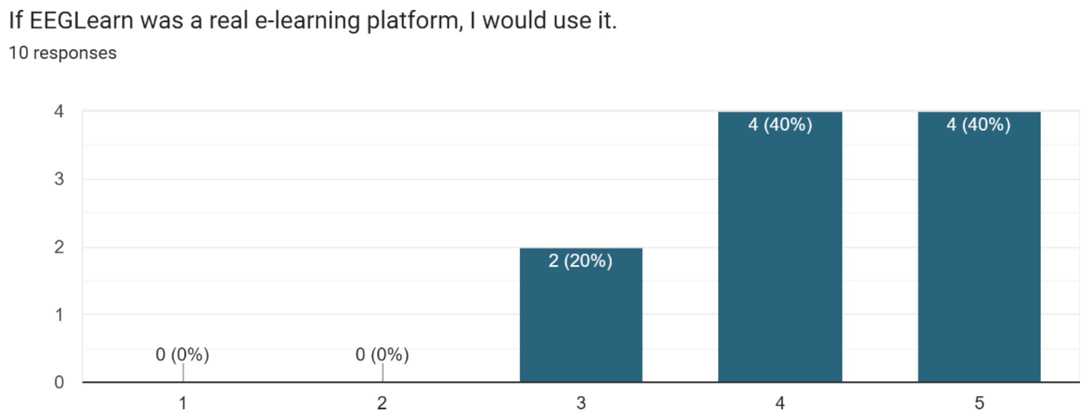

Figure 34: Real use.

The results of the UEQ are the following:

Item	Mean	Variance	Std. Dev.	Left	Right	Scale
1	2,3	0,7	0,8	annoying	enjoyable	Attractiveness
2	2,6	0,5	0,7	not understandable	understandable	Perspicuity
3	2,1	1,0	1,0	creative	dull	Novelty
4	2,4	1,2	1,1	easy to learn	difficult to learn	Perspicuity
5	2,2	1,1	1,0	valuable	inferior	Stimulation
6	2,1	0,8	0,9	boring	exciting	Stimulation
7	2,7	0,2	0,5	not interesting	interesting	Stimulation
8	1,7	1,3	1,2	unpredictable	predictable	Dependability
9	2,2	0,8	0,9	fast	slow	Efficiency
10	2,0	2,0	1,4	inventive	conventional	Novelty
11	2,5	0,5	0,7	obstructive	supportive	Dependability
12	2,6	0,5	0,7	good	bad	Attractiveness
13	2,6	0,7	0,8	complicated	easy	Perspicuity
14	2,7	0,2	0,5	unlikable	pleasing	Attractiveness
15	2,0	1,3	1,2	usual	leading edge	Novelty
16	2,6	0,3	0,5	unpleasant	pleasant	Attractiveness
17	2,6	0,9	1,0	secure	not secure	Dependability
18	2,8	0,2	0,4	motivating	demotivating	Stimulation
19	2,4	1,6	1,3	meets expectations	does not meet expectations	Dependability
20	2,1	1,2	1,1	inefficient	efficient	Efficiency
21	2,7	0,5	0,7	clear	confusing	Perspicuity
22	2,4	0,7	0,8	impractical	practical	Efficiency
23	2,7	0,9	0,9	organized	cluttered	Efficiency
24	2,8	0,2	0,4	attractive	unattractive	Attractiveness
25	3,0	0,0	0,0	friendly	unfriendly	Attractiveness
26	2,6	0,9	1,0	conservative	innovative	Novelty

UEQ Scales (Mean and Variance)		
Attractiveness	2,667	0,12
Perspicuity	2,575	0,60
Efficiency	2,350	0,54
Dependability	2,300	0,76
Stimulation	2,450	0,25
Novelty	2,175	0,54

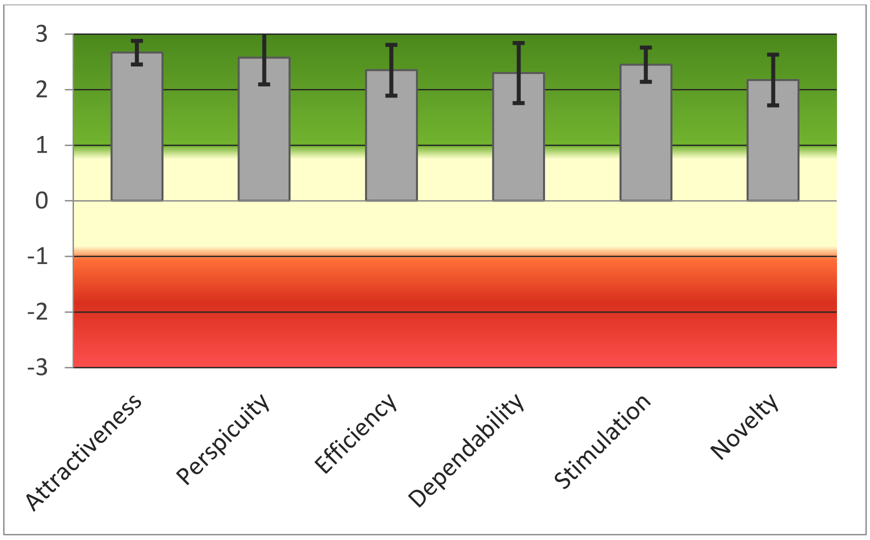

Figure 35: UEQ Scales.

Pragmatic and Hedonic Quality	
Attractiveness	2,67
Pragmatic Quality	2,41
Hedonic Quality	2,31

All items received only positive responses on average (>0). Attractiveness was the scale that received the higher score (2,66), followed by perspicuity (2,57), stimulation (2,45), efficiency (2,35), dependability (2,3), and novelty (2,17). These scales can also be grouped into pragmatic quality which is task-related (perspicuity, efficiency, dependability) and hedonic quality (stimulation, novelty). Based on these qualities, attractiveness scored 2,67, pragmatic quality was 2,41 and hedonic quality 2,31.

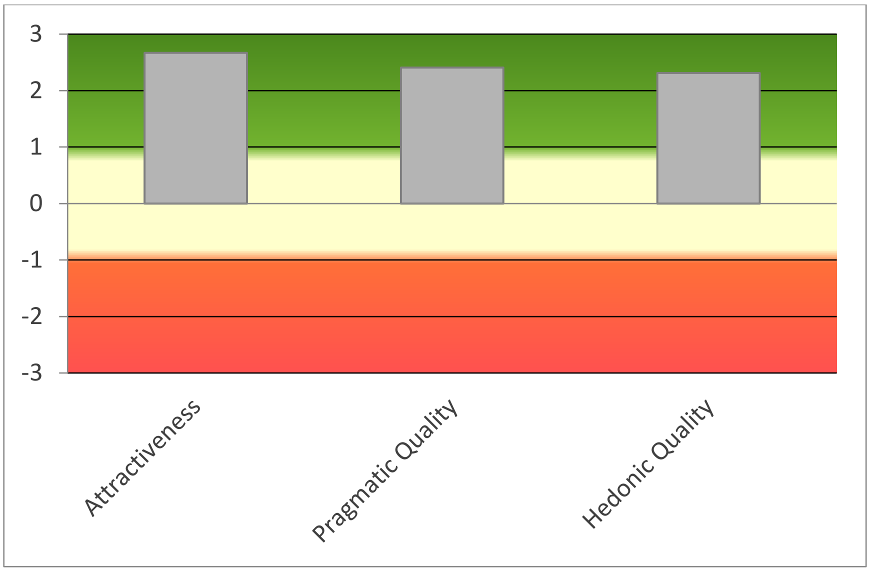

Figure 36: UEQ: Pragmatic and Hedonic Quality.

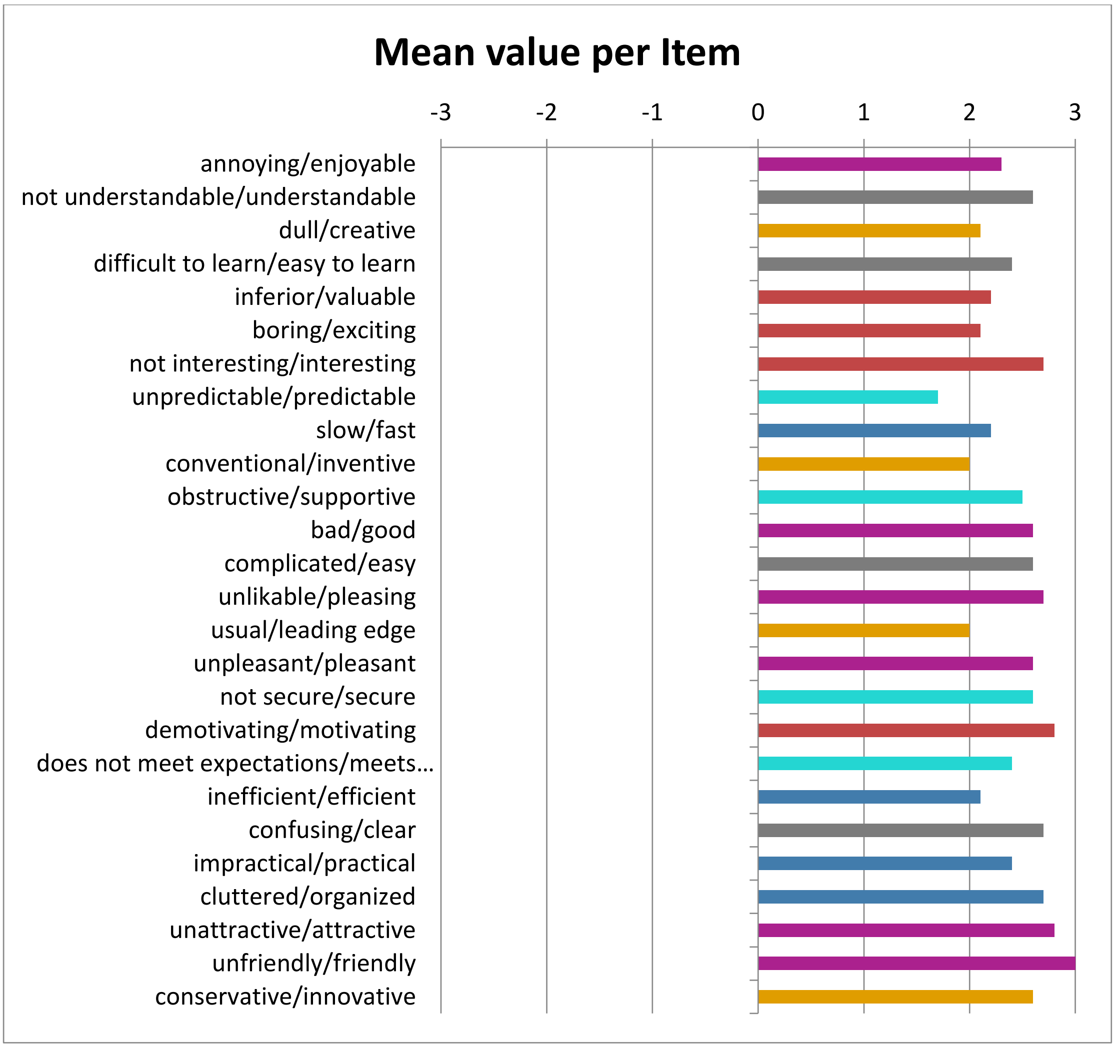

Figure 37: UEQ: Mean per item.

Chapter 5
Discussion

5.1 Synopsis

Generally, all sub-tasks of the system were successfully performed by all users. This is the most important part because User Experience cannot be positive if a system cannot do what it claims to be doing. Additionally, all participants stated that they found the adaptations useful and that they would use such a system if it was real. Also, one of the drawbacks of EEG devices is that these are uncomfortable. However, only 20% of the participants found the specific device that was used uncomfortable which is a very positive result.

More specifically, 70% of the participants strongly agreed that the EEG device was prepared successfully. This percentage was not 100% because 1-2 disconnections occurred between the EEG device and the system that forced the participants to prepare the device again. This disconnection showed an error in the screen that for the 10% of the participants was not clear. The diversity of the rest responses regarding the errors may be due to the fact that no errors occurred to them while using the system. Moreover, even though this disconnection may occurred once for each participant, no one seemed to be bothered about this because the evaluation was overall positive. More about this disconnection issue is discussed in the challenges chapter.

Furthermore, regarding the accuracy of the predictions, it must be noted that the algorithm was only tested in one person who did not take part in the evaluation of the system (the writer of this dissertation), with 70% accuracy. Therefore, it is a huge success to see that this percentage matches the average of the participants as well, without performing any more tests to other people and without using any advanced signal techniques for artifact removal. Nonetheless, 20% of the participants were neutral about the accuracy of the predictions and one participant stated that he/she had inaccurate results. This could be because of their difficulty to relax such that this relaxation could be depicted in their brainwaves. Alternatively, it could be because participants were not asked to avoid caffeine or alcohol consumption or control their sleep even though most research studies that can successfully predict mental states have a strict experimental protocol.

To sum up, sufficient instructions, successful signal quality check, accurate predictions, practical adaptations, a comfortable low-cost EEG device and the unexpectedly positive results from the UEQ indicate that such a system would indeed keep an overall positive user experience. This could also mean that such a system would be valuable to be integrated in a real-world e-learning platform.

5.2 Challenges

In general, BCIs that use EEG devices present quite a few challenges and most of these are the challenges of the EEG devices themselves. The first one is low signal to noise ratio (Al-Nafjan & Aldayel, 2022) and the presence of EEG artifacts as already mentioned, that contaminate the signal. For this reason, participants of this study were asked to minimize their movement and try not to squeeze their teeth or clench their jaw, as far as possible. However, this may not be easy to ask or applicable in a real e-learning environment.

Furthermore, a signal quality check is required to detect artifacts, but no ready-to-use signal quality algorithm was available on the internet. Even worse, the signal quality can change at any time due to user movement or other conditions, so a continuous signal quality check is preferred, but it requires a lot of time to be implemented manually. In this system, a basic signal quality check procedure was used before collecting any EEG data. Due to no available algorithm for this purpose, a custom algorithm was created based on trial-and-error. More specifically, if the absolute band power of 48-52 Hz (the frequencies near the power line noise) exceeds a specific threshold, then this channel is characterized as noisy. However, continuous signal quality check was not integrated.

Another important challenge is that there is no standard protocol for predicting mental states using EEG yet. Every study uses different electrode locations, different signal processing techniques, different training data length and durations, different ML algorithms and features (Katmah et al., 2021) according to the system's application (Edla et al., 2018). This makes it really difficult to implement such a system.

Another challenge was the selection of the EEG device to use. OpenBCI and Muse 2 devices were already purchased by the writer of the dissertation some years back for other reasons. Both are quite popular with a big community and libraries. However, OpenBCI's setup is much more cumbersome than that of Muse. Also, many additional problems were encountered in OpenBCI such as some broken electrodes (leading to strange raw EEG values of 1000-3000 μV) that contributed to the decision to use Muse. Moreover, finding the proper libraries in order to fetch the raw EEG data was another important challenge. Initially, a JavaScript library called 'musejs' (Shaked, 2021) was evaluated. This library is using Web Bluetooth protocol that directly connects the Muse device with the web browser making their connection trivial. Using musejs, the user only had to enable Bluetooth and then the web browser would find and connect to the device. However, constant disconnections made this library impossible to use. Also, 'musejs' is a project that is no longer maintained and the only device that is supported is the Muse device.

Then, brainflow library was evaluated. One of its strongest points is that it has a common API for many different EEG devices. This makes it possible to use different EEG devices with the same API. Also, this project is currently in active development and supported by OpenBCI. But even though it seemed to be the perfect choice, there was no JavaScript binding for the brainflow library. This practically means that a web client environment cannot directly communicate with the EEG device. This was only possible through python (or some other supported by brainflow languages such as C#). That's why WebSockets protocol was used, so that the web-client could talk to the backend, and the backend to the EEG device using a script that is run on the user's PC (eeg-client), in python language. However, that made the flow of the system much more complicated and much more difficult to develop. But, according to brainflow's development roadmap, JavaScript binding is on their future plans so this issue should be resolved in the near future.

Finally, the exact mental state to choose from was itself a challenge. Initially, engagement or workload were chosen. The reason was that there was much more research on those topics. However, there was an important detail: all of those studies were using devices with an average of 31 electrodes that could be placed freely in any location desirable. That would be impossible with the Muse device. Additionally, the training time for workload was much longer (40 minutes (Miklody et al., 2017; Walter et al., 2017)) than that of concentration/relaxation that would also be cumbersome. Moreover, the testing of such mental state would also be difficult due to the training time that would be needed each time the code changed. Finally, most studies for workload train and test on the same tasks so that makes workload generalization ability limited to a few studies (Kakkos et al., 2021).

5.3 Limitations and Future Work

Even though the system was positively evaluated from the users there are some known limitations of the system. For example, the mental state of *Relaxation* may not be an exact match of the *Available* status. Available students would not only be the ones that are relaxed, but the ones that are either relaxed or neutral. However, *Neutral* state was not successfully classified. Also, it is unclear whether the *Concentrated* state that the algorithm learned is actually a *Mental Calculation* state. Ideally, all concentrated students should be in a *Concentrated* state and therefore in a *Busy* status and not only the ones that are mentally calculating something. More time is be needed to clarify those issues.

Also, there were times where the EEG device would disconnect from the system. The reasons for this are not yet clear. It could be a problem of the brainflow library, a problem of Muse 2 EEG device, Bluetooth or a hardware/driver problem of the current PC that ran the project.

Additionally, even though battery life is stated to last for 5 hours in Muse 2 device, in reality the battery drops after approximately 1 hour of use, which is not an issue for meditation. But for a potential student using such system, 1 hour could be quite a small duration. Moreover, even though most participants did not find the EEG device uncomfortable to wear, the actual use of the system was approximately half an hour. This duration may not be representative of an actual use of such system so a longer duration may indeed be felt uncomfortable. Nonetheless, these issues could be solved in the future where new EEG devices will be cheaper, more comfortable, and more reliable.

Furthermore, we must not forget that Muse 2 EEG device that was used was designed for meditation neurofeedback. Therefore, it's signal quality may drop when someone wears glasses because the glasses block the proper placement of the device. This, of course, is not an issue when someone is meditating because the glasses are no longer needed. In an e-learning environment this could be an issue though. Similarly, in meditation the user is not expected to move, blink or somehow generate artifacts, as opposed to an e-learning environment where the user can do so more easily. Additionally, for a productive system, constant quality check would be needed to ensure a good signal quality, that was not implemented in this system.

Also, due to limited time, multi-user support was not fully implemented between the eeg-client and the web-client. More effort is needed there, so that WebSockets can talk only to specific clients and not to every client that is listening at that time. However, if JavaScript binding of brainflow is completed, this step would no longer be necessary as well as the whole existence of eeg-client itself. It must also be noted that the users did not get in touch with the installation process of the eeg-client. The eeg-client was already set up for them. This step was omitted, because all the users that tested the system would have to come either way to a specific location where the system was installed and the EEG device would be available. Also, as already mentioned, eeg-client will no longer be necessary when JavaScript binding of brainflow is completed. In that case, the user would just have to enable the Bluetooth connection to connect his/her EEG device with the web environment.

Finally, the fact that this system uses a subject-specific classifier with predefined parameters (7-59Hz with 1Hz bin, 7 second window and specific Random Forest parameters) that were not studied earlier on those participants may not be the best approach. These specific parameters were tested and evaluated only on one user (that did not participate in the evaluation) due to limited time. However, the current parameters that were used resulted in better accuracy results than the ones presented in the literature. Moreover, in a real system it would be cumbersome for the students not only to spend 12 minutes to train the algorithm but also retrain it every now and then. This retraining would be needed because the of the non-stationarity of the signal, the user's fatigue, or the different levels of the signal quality between the training and the actual use (Qu et al., 2022). Moreover, different affective contexts between training and using could potentially affect the accuracy of the classifier (Mühl et al., 2014). Also, as stated earlier, strict caffeine, alcohol or sleep protocol were not asked from the participants to allow for a more flexible usage but that could also have an effect to the prediction accuracy. So, a more elaborate classifier that could generalize across subjects and time using new algorithms and more advanced methods could be used instead (Brouwer et al., 2015; Mühl et al., 2014).

Therefore, such generalized classifier that would be tested across time and subjects would be an ideal future goal. Furthermore, to add some sophisticated filtering and artifact removal algorithms and add more features at the classification algorithm to achieve an even better prediction accuracy. Also, to investigate neutral state and clarify whether concentrated state applies for tasks that do not require mental calculation. Finally, to encrypt the EEG data and the user's information that is now saved in plain text in the database. That would increase the security of the system.

5.4 Conclusion

In this study an adaptive e-learning environment was created. It used a popular and comfortable low-cost EEG device that could implicitly grasp the user's current mental state and provide useful system adaptations that could support the user's learning. It used an efficient, attractive, and friendly environment leading to an overall positive user experience. In the future, it is expected that the cost of the EEG devices will be even lower. Also, with technology and algorithm advancements the EEG devices limitations such as artifacts could be minimized. That could make the usage of such devices common and trivial. Adaptive e-learning systems with implicit adaptations that can make use of such devices would be a huge advantage as learners would want to support their learning achieving as much personalization possible. Generally, using EEG-based adaptations is a step towards the future. After all, Elon Musk's and Facebook's future plans is to develop a Brain-Computer Interface that could successfully read people's minds ("Business Standard," 2019).

Appendix A
System Presentation

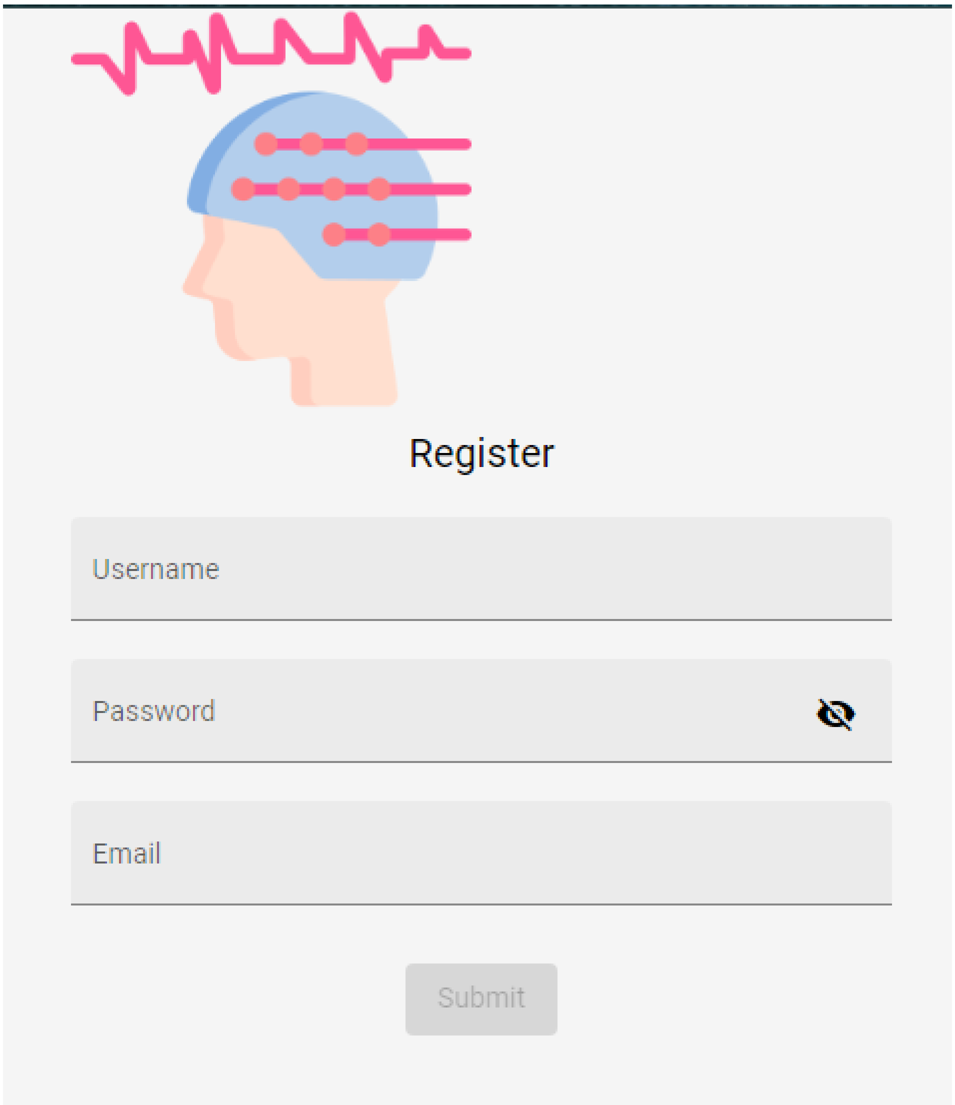

Figure 38: User Registration.

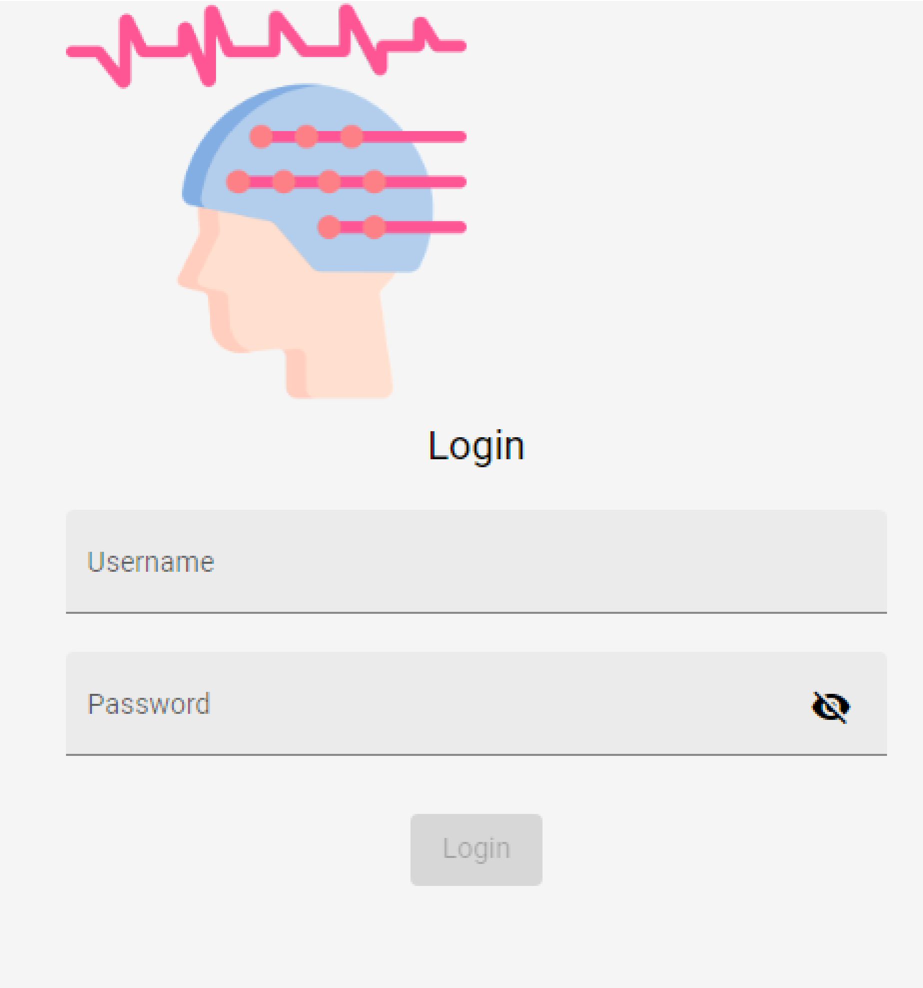

Figure 39: User login.

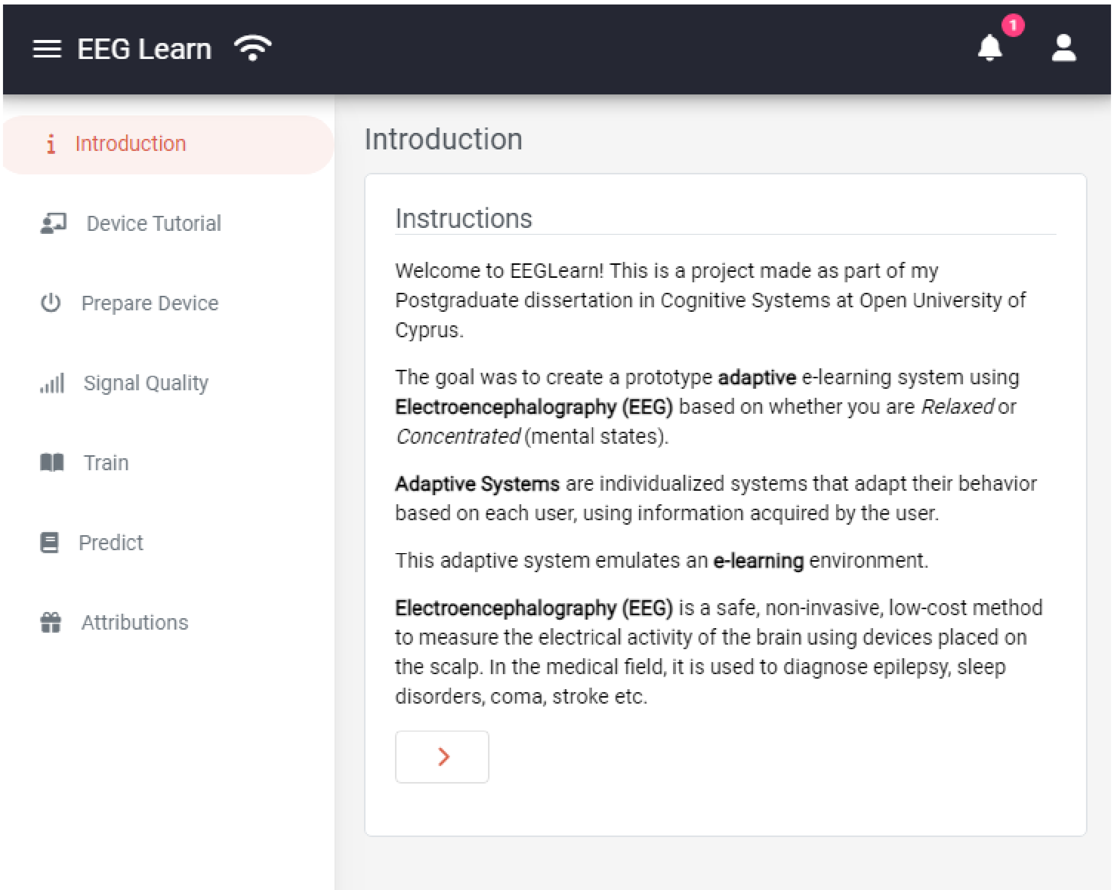

Figure 40: Instructions.

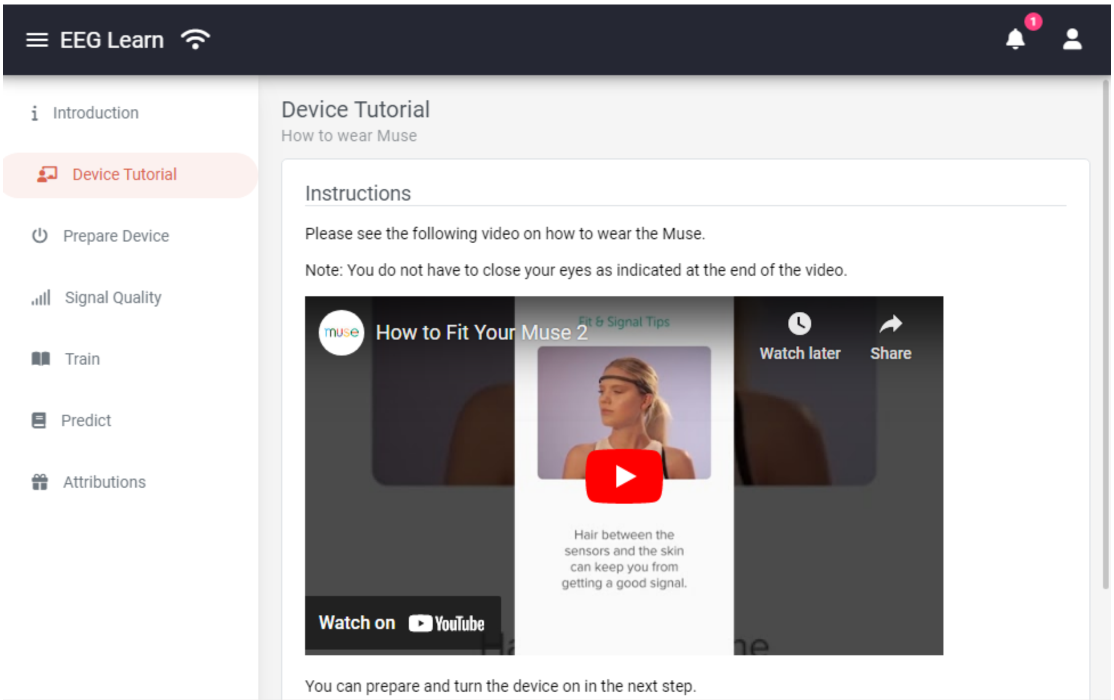

Figure 41: Muse 2 Tutorial.

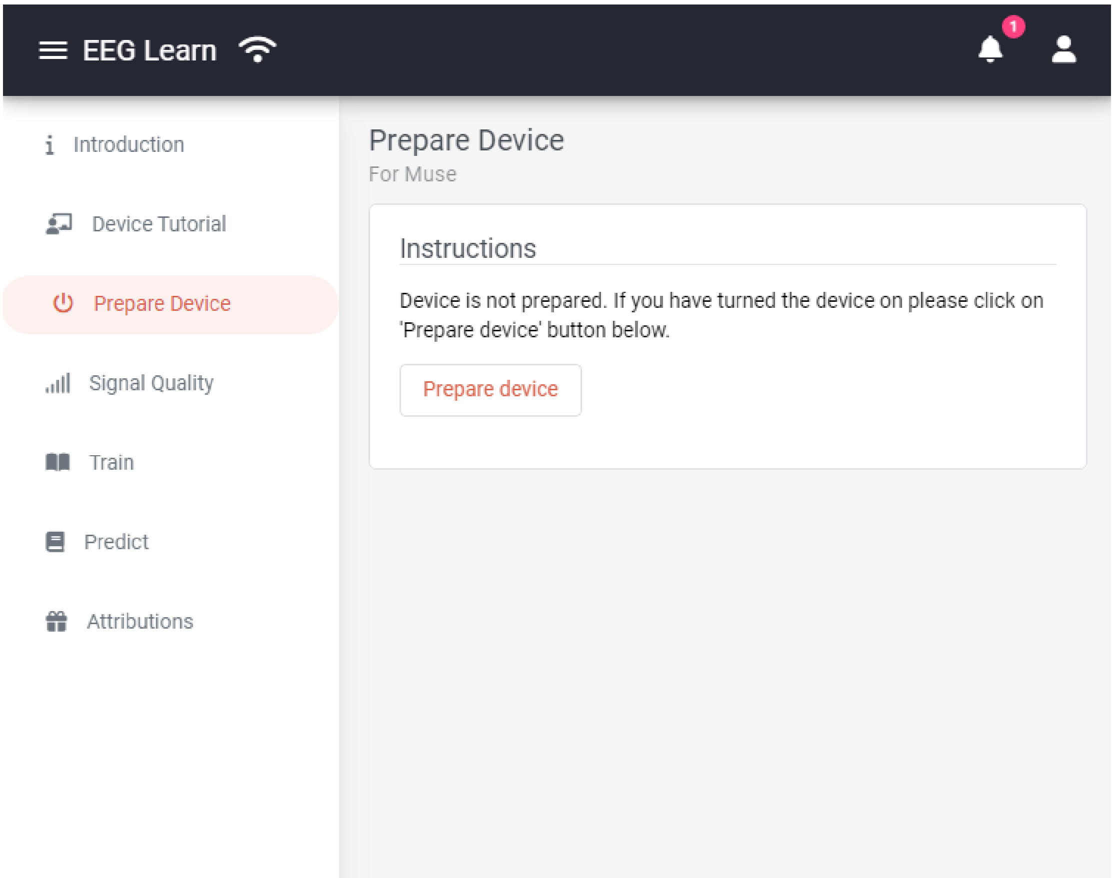

Figure 42: Prepare EEG device.

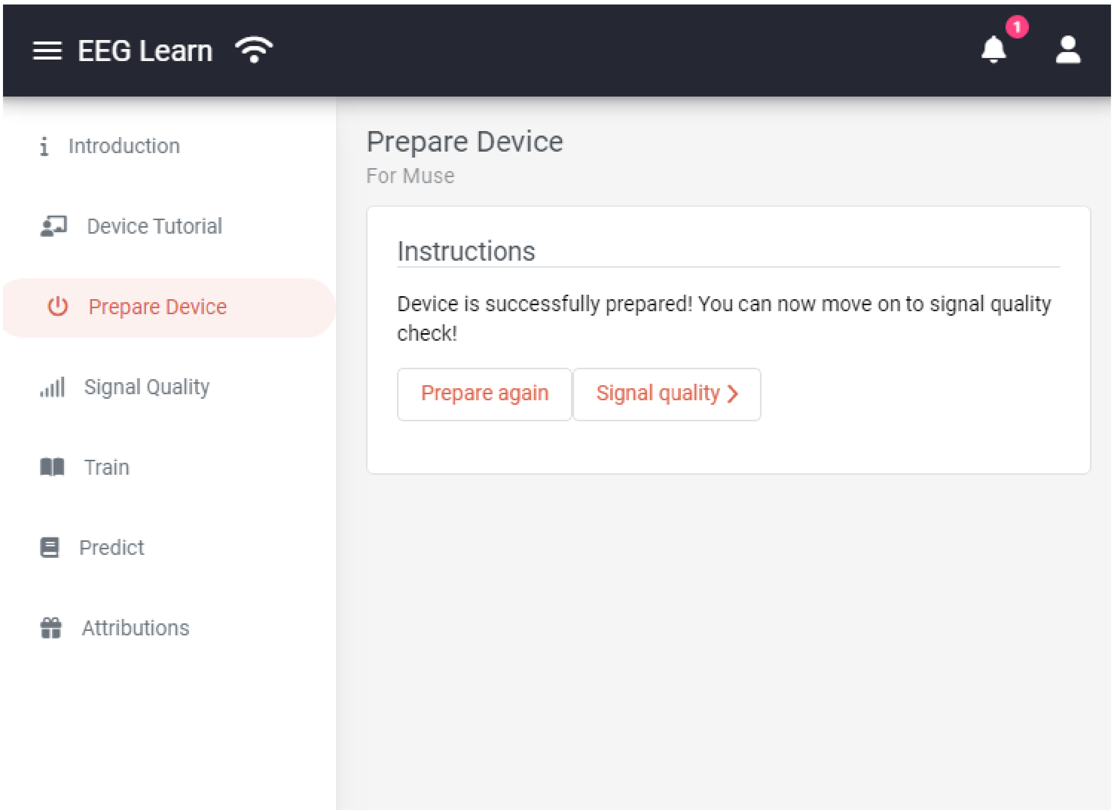

Figure 43: Successful EEG device preparation.

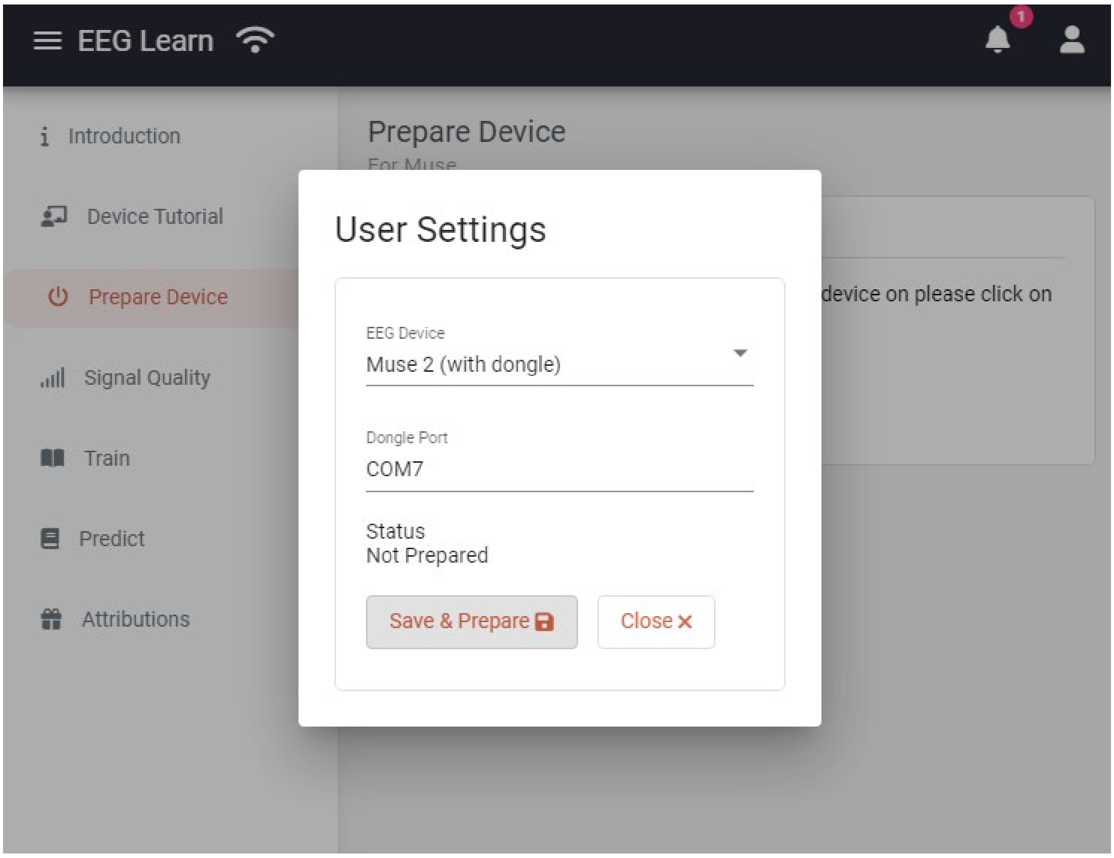

Figure 44: User Settings: Change EEG device.

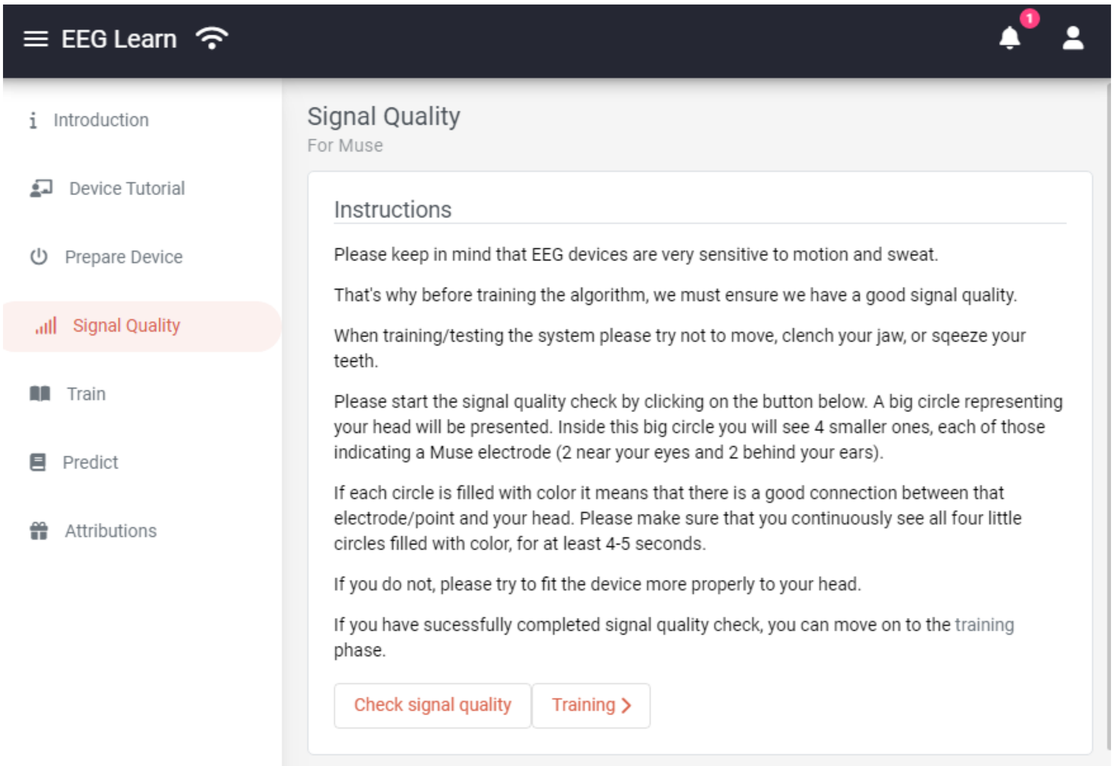

Figure 45: Signal Quality instructions.

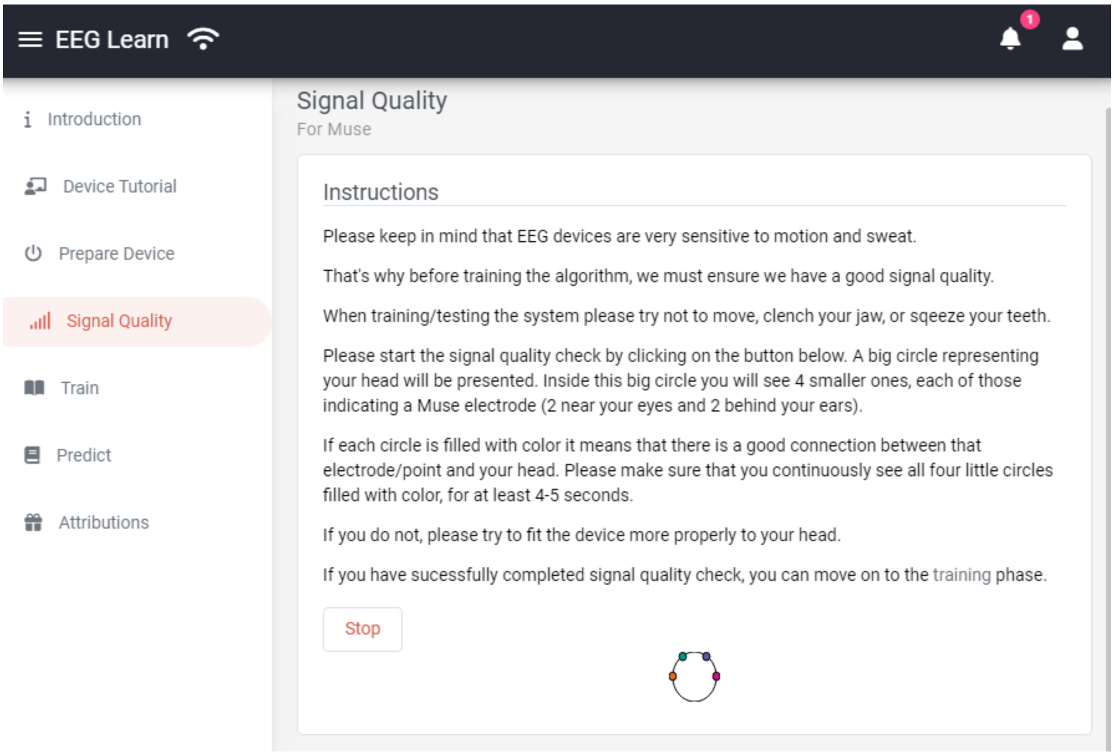

Figure 46: Signal Quality check. All electrodes have good signal.

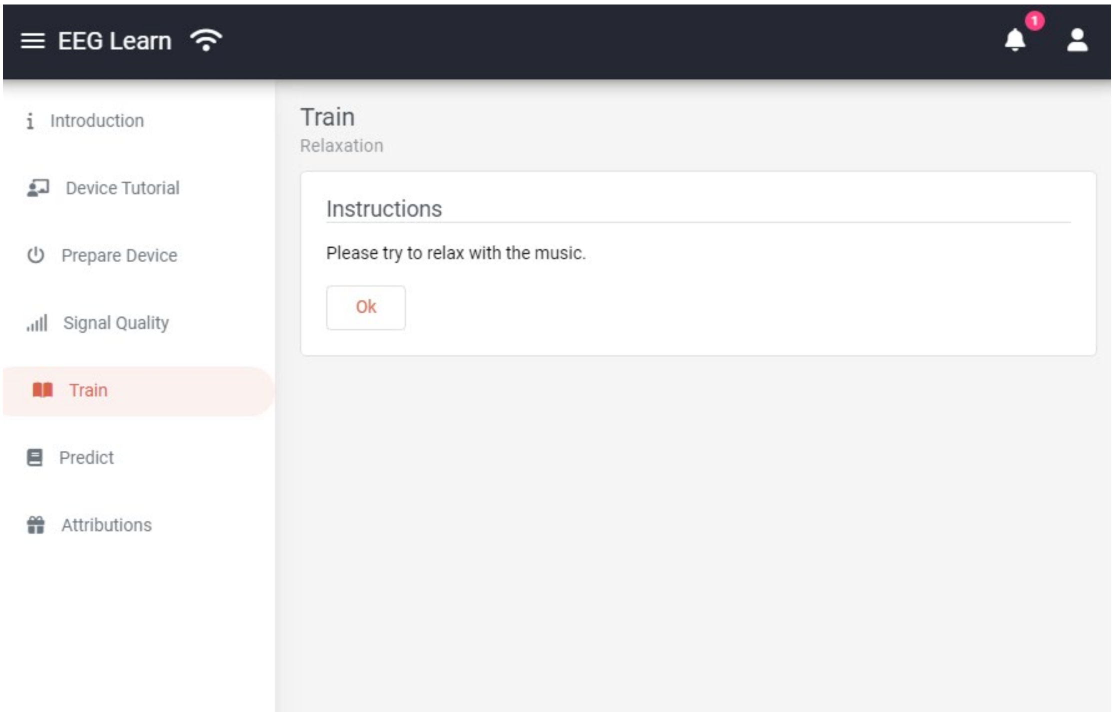

Figure 47: Train relax instructions.

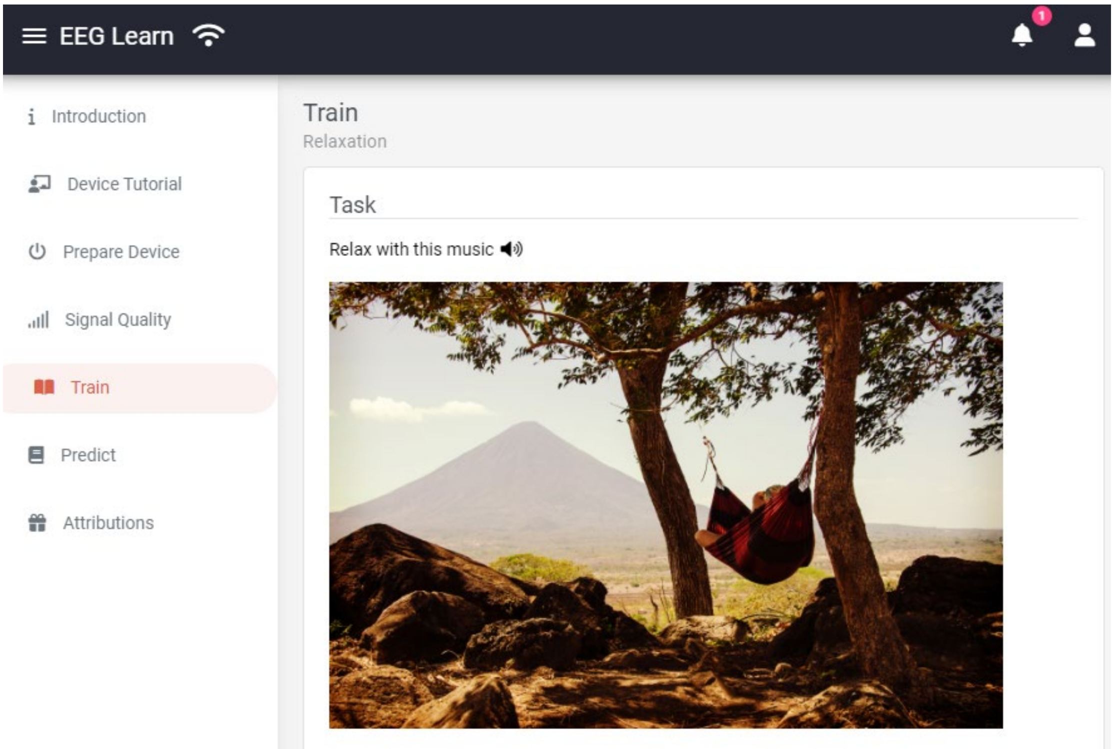

Figure 48: Train relax state.

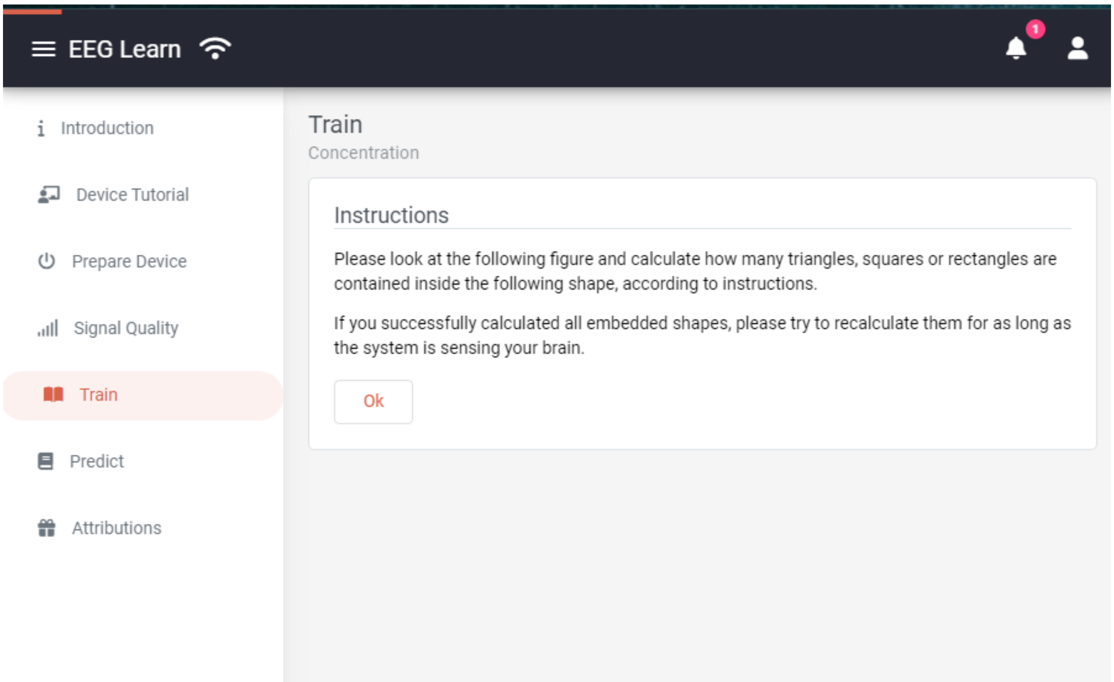

Figure 49: Train concentration instructions.

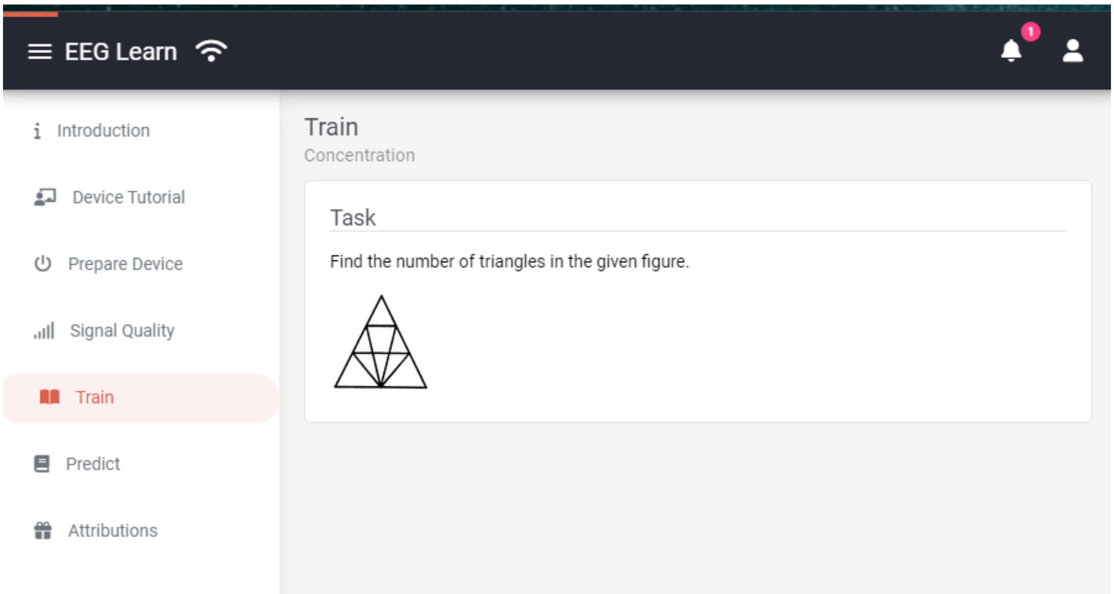

Figure 50: Train concentration.

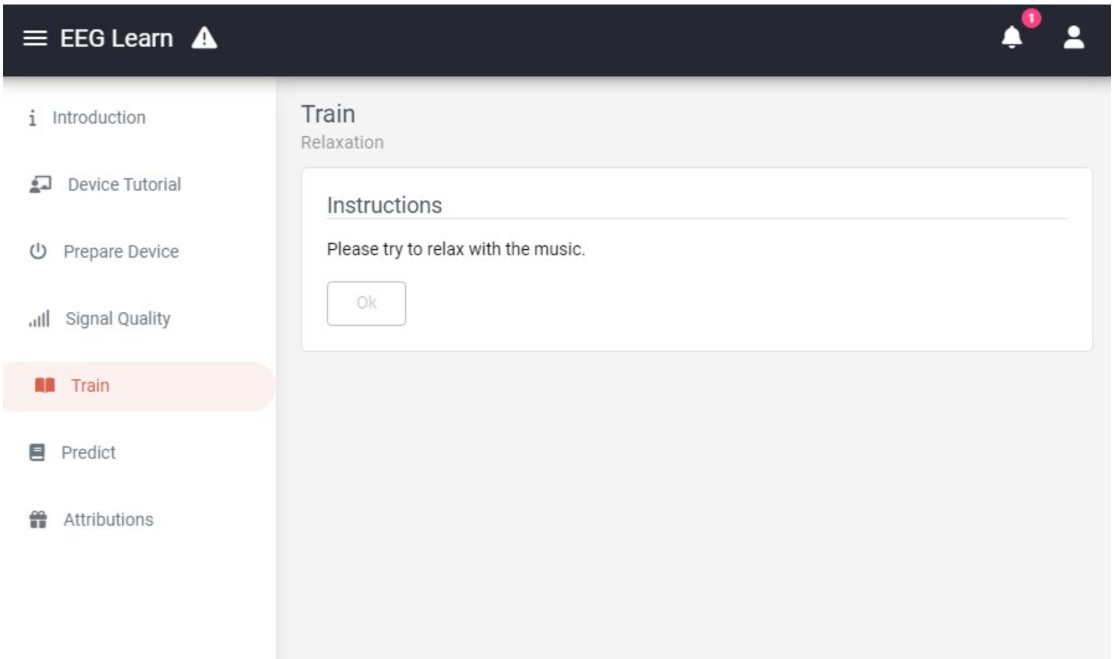

Figure 51: Connection not established (notice the icon next to *EEG Learn*). Also notice that train, predict and other features cannot start if there is no connectivity because buttons are disabled.

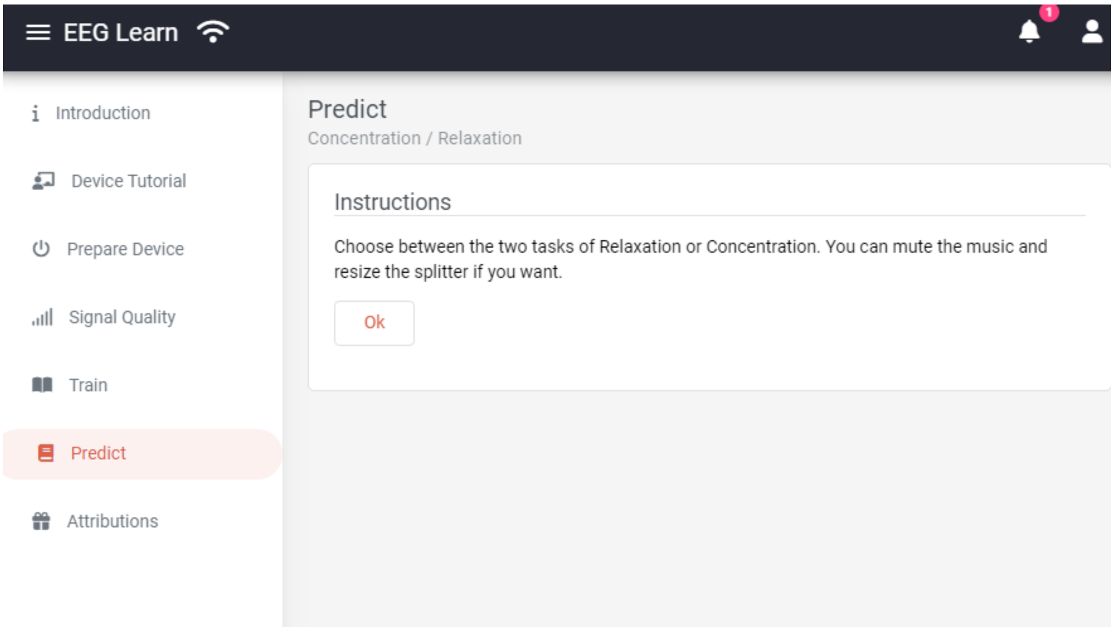

Figure 52: Predict instructions.

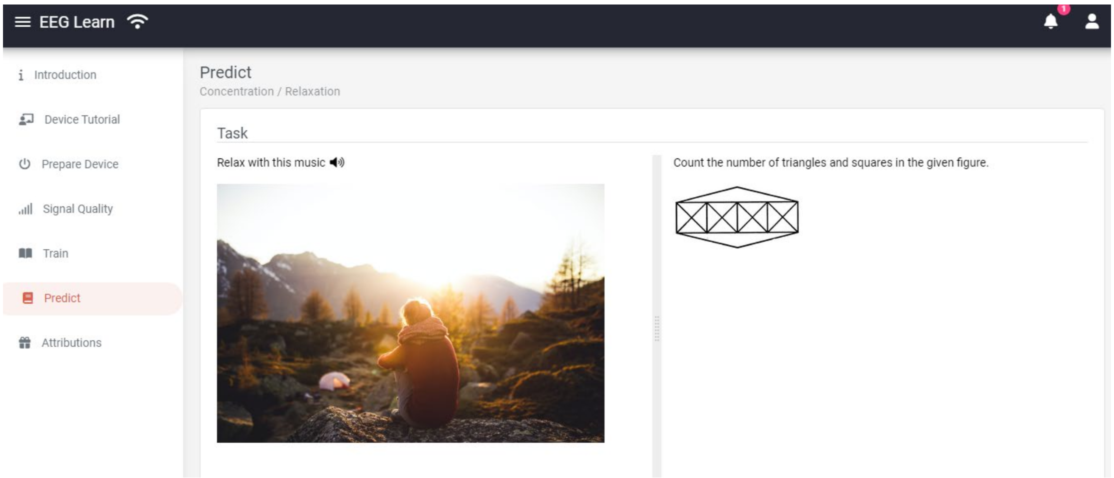

Figure 53: Predict. Choosing the task.

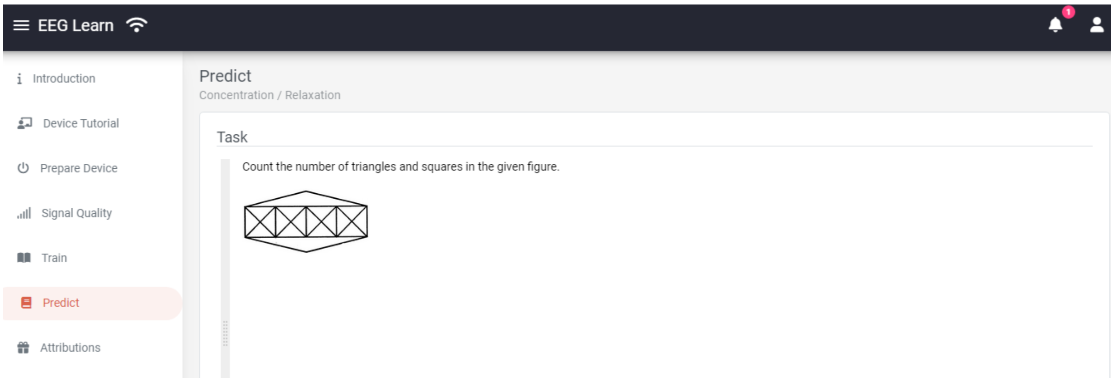

Figure 54: Predict. Moved the splitter to the left for concentrate state.

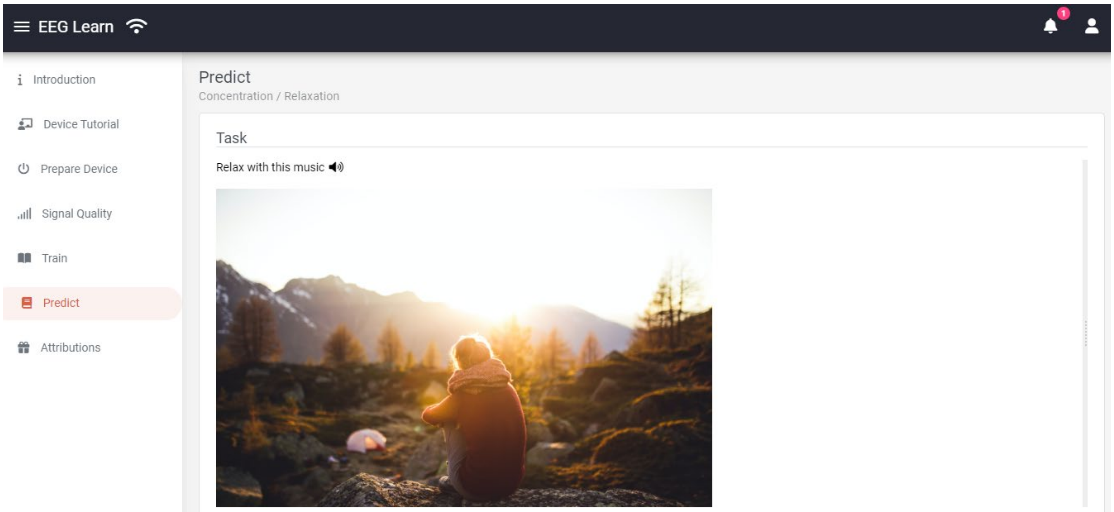

Figure 55: Predict. Moved the splitter to the right for relax state.

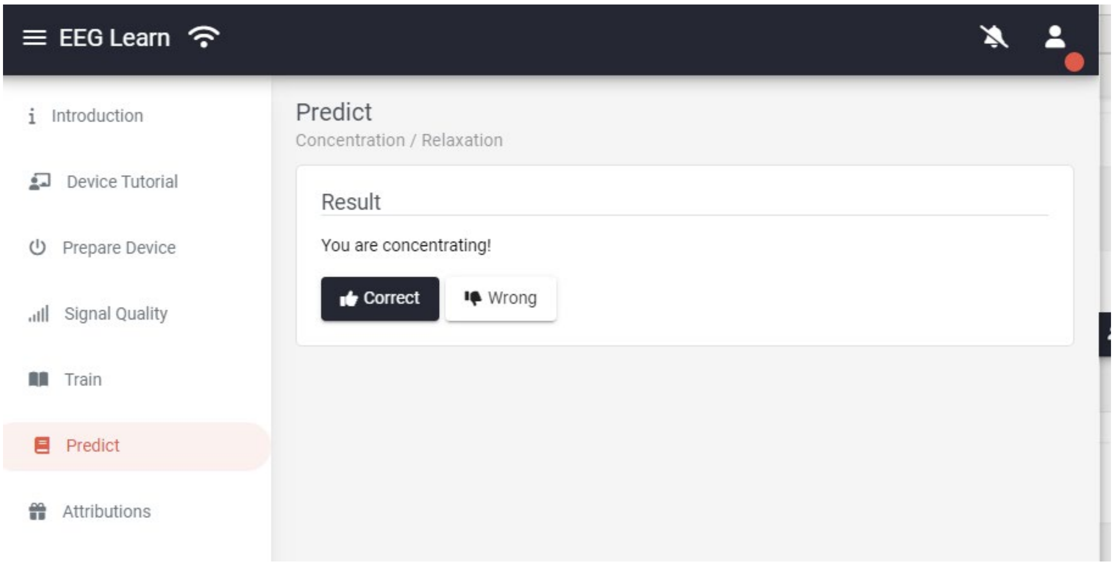

Figure 56: Predict result: Concentrate. Status changed to Busy, Notifications disabled.

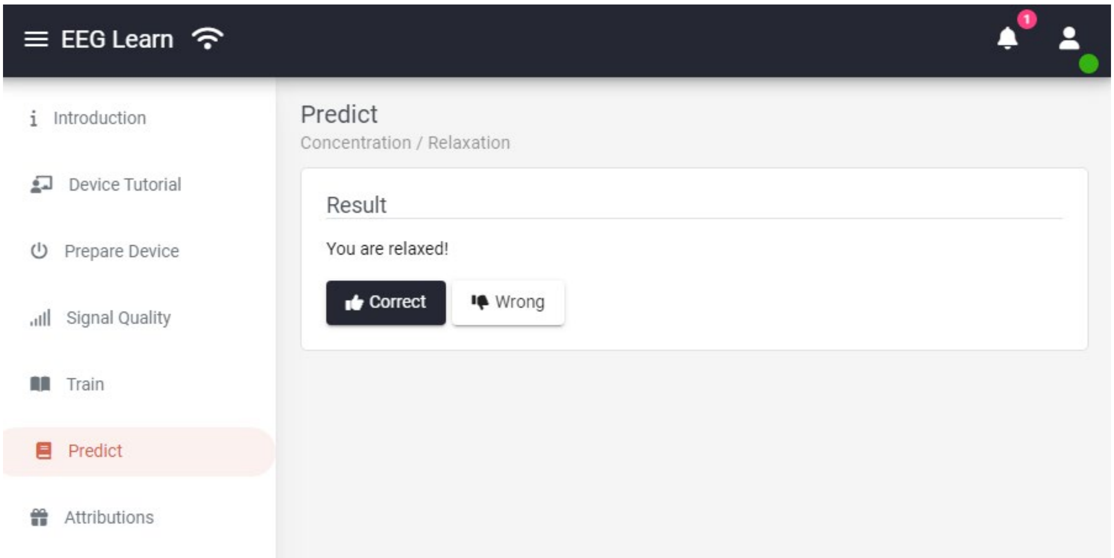

Figure 57: Predict result: Relax. Status changed to Available, Notifications enabled.

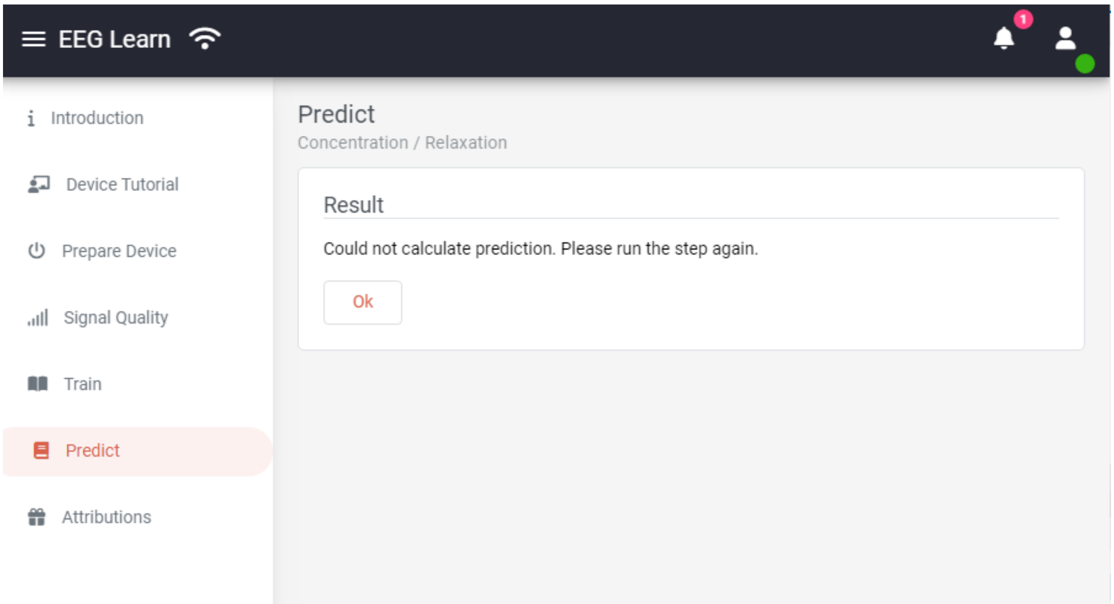

Figure 58: Prediction error that will repeat the current step.

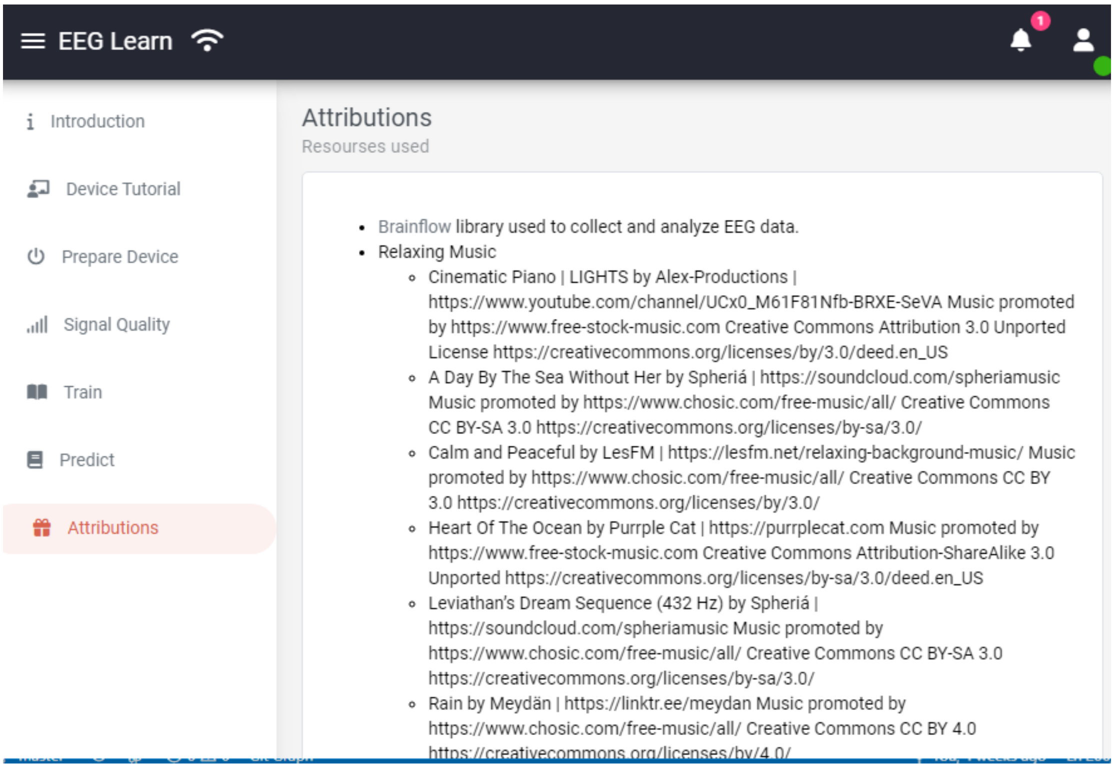

Figure 59: Attributions: Libraries, music tracks, photos.

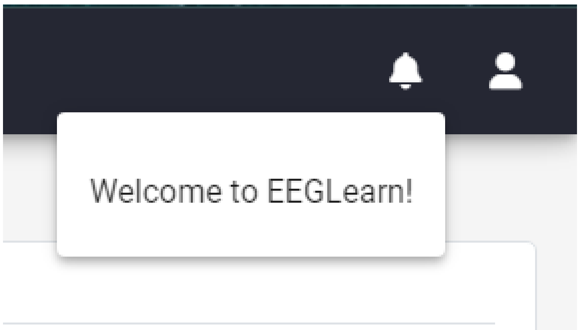

Figure 60: Notification area.

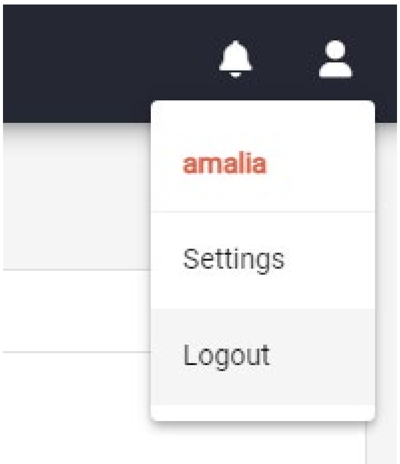

Figure 61: Settings and Logout options.

References

Abdulkader, S. N., Atia, A., & Mostafa, M.-S. M. (2015). Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, *16*(2), 213–230. https://doi.org/10.1016/j.eij.2015.06.002

Abujelala, M., Sharma, A., Abellanoza, C., & Makedon, F. (2016). Brain-EE: Brain enjoyment evaluation using commercial EEG headband. *ACM International Conference Proceeding Series*, *29-June-20*, 1–5. https://doi.org/10.1145/2910674.2910691

Aggarwal, S., & Chugh, N. (2022). Review of Machine Learning Techniques for EEG Based Brain Computer Interface. *Archives of Computational Methods in Engineering*, *29*(5), 3001–3020. https://doi.org/10.1007/s11831-021-09684-6

al Amrani, Y., Lazaar, M., & el Kadiri, K. E. (2018). Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis. *Procedia Computer Science*, *127*, 511–520. https://doi.org/10.1016/j.procs.2018.01.150

Al-Nafjan, A., & Aldayel, M. (2022). Predict Students' Attention in Online Learning Using EEG Data. *Sustainability*, *14*(11), 6553. https://doi.org/10.3390/su14116553

Badillo, S., Kam-thong, T., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Siebourg-polster, J., Steiert, B., & Zhang, J. D. (2020). *An Introduction to Machine Learning*. *107*(4), 871–885. https://doi.org/10.1002/cpt.1796

Baştanlar, Y., & Özuysal, M. (2014). *Introduction to Machine Learning* (M. Yousef & J. Allmer, Eds.; Vol. 1107, pp. 105–128). Humana Press. https://doi.org/10.1007/978-1-62703-748-8_7

Bellamy, S. (2021). *Can mental workload in EEG tasks be classified using machine learning algorithms?* (Issue January) [Northeastern University]. https://doi.org/10.17760/D20399921

Bera, T. K. (2021). A Review on The Medical Applications of Electroencephalography (EEG). *2021 Seventh International Conference on Bio Signals, Images, and Instrumentation (ICBSII)*, 1–6. https://doi.org/10.1109/ICBSII51839.2021.9445153

Bezold, M., & Minker, W. (2011). Adaptive Multimodal Interactive Systems. *Adaptive Multimodal Interactive Systems*. https://doi.org/10.1007/978-1-4419-9710-4

Bird, J. J., Manso, L. J., Ribeiro, E. P., Ekart, A., & Faria, D. R. (2018). A Study on Mental State Classification using EEG-based Brain-Machine Interface. *9th International Conference on Intelligent Systems 2018: Theory, Research and Innovation in Applications, IS 2018 - Proceedings*, *October*, 795–800. https://doi.org/10.1109/IS.2018.8710576

Brouwer, A. M., Zander, T. O., van Erp, J. B. F., Korteling, J. E., & Bronkhorst, A. W. (2015). Using neurophysiological signals that reflect cognitive or affective state: Six recommendations to avoid common pitfalls. *Frontiers in Neuroscience*, *9*(APR), 1–11. https://doi.org/10.3389/fnins.2015.00136

"Business Standard." (2019). *Facebook, Tesla want to read your mind: here's why you should be worried*. https://www.business-standard.com/article/technology/facebook-tesla-want-to-read-your-mind-here-s-why-you-should-be-worried-119082100165_1.html

Candra, H., Yuwono, M., Rifai Chai, Handojoseno, A., Elamvazuthi, I., Nguyen, H. T., & Su, S. (2015). Investigation of window size in classification of EEG-emotion signal with wavelet entropy and support vector machine. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, *2015-Novem*, 7250–7253. https://doi.org/10.1109/EMBC.2015.7320065

Casson, A. J. (2019). Wearable EEG and beyond. *Biomedical Engineering Letters*, *9*(1), 53–71. https://doi.org/10.1007/s13534-018-00093-6

Ceron, J., Baldiris, S., Quintero, J., Garcia, R. R., Saldarriaga, G. L. V., Graf, S., & Fuente Valentin, L. D. la. (2021). Self-Regulated Learning in Massive Online Open Courses: A State-of-the-Art Review. *IEEE Access*, *9*, 511–528. https://doi.org/10.1109/ACCESS.2020.3045913

Channon, S. (2004). Frontal lobe dysfunction and everyday problem-solving: Social and non-social contributions. *Acta Psychologica*, *115*(2–3), 235–254. https://doi.org/10.1016/j.actpsy.2003.12.008

Chaouachi, M., & Frasson, C. (2010). Exploring the relationship between learner EEG mental engagement and affect. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6095 LNCS*(PART 2), 291–293. https://doi.org/10.1007/978-3-642-13437-1_48

Chaouachi, M., & Frasson, C. (2012). Mental workload, engagement and emotions: An exploratory study for intelligent tutoring systems. *Lecture Notes in*

Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), *7315 LNCS*, 65–71. https://doi.org/10.1007/978-3-642-30950-2_9

Chaouachi, M., Jraidi, I., & Frasson, C. (2015). MENTOR: A Physiologically Controlled Tutoring System. In F. Ricci, K. Bontcheva, O. Conlan, & S. Lawless (Eds.), *User Modeling, Adaptation and Personalization. UMAP 2015. Lecture Notes in Computer Science* (Vol. 9146, pp. 56–67). Springer, Cham. https://doi.org/10.1007/978-3-319-20267-9

Coetzee, D., Fox, A., Hearst, M. A., & Hartmann, B. (2014). Chatrooms in MOOCs. *Proceedings of the First ACM Conference on Learning @ Scale Conference*, 127–136. https://doi.org/10.1145/2556325.2566242

di Flumeri, G., Aricò, P., Borghini, G., Sciaraffa, N., di Florio, A., & Babiloni, F. (2019). The Dry Revolution: Evaluation of Three Different EEG Dry Electrode Types in Terms of Signal Spectral Features, Mental States Classification and Usability. *Sensors*, *19*(6), 1365. https://doi.org/10.3390/s19061365

Díaz-Oreiro, I., López, G., Quesada, L., & Guerrero, L. (2019). Standardized Questionnaires for User Experience Evaluation: A Systematic Literature Review. *13th International Conference on Ubiquitous Computing and Ambient Intelligence UCAmI 2019*, *October 2018*, 14. https://doi.org/10.3390/proceedings2019031014

Douibi, K., le Bars, S., Lemontey, A., Nag, L., Balp, R., & Breda, G. (2021). Toward EEG-Based BCI Applications for Industry 4.0: Challenges and Possible Applications. *Frontiers in Human Neuroscience*, *15*(August), 1–8. https://doi.org/10.3389/fnhum.2021.705064

Ed-douibi, H., Cánovas Izquierdo, J. L., & Cabot, J. (2017). Example-Driven Web API Specification Discovery. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10376 LNCS* (pp. 267–284). https://doi.org/10.1007/978-3-319-61482-3_16

Edla, D. R., Mangalorekar, K., Dhavalikar, G., & Dodia, S. (2018). Classification of EEG data for human mental state analysis using Random Forest Classifier. *Procedia Computer Science*, *132*(Iccids), 1523–1532. https://doi.org/10.1016/j.procs.2018.05.116

Frasson, C., & Chalfoun, P. (2010). Managing learner's affective states in intelligent tutoring systems. *Studies in Computational Intelligence*, *308*, 339–358. https://doi.org/10.1007/978-3-642-14363-2_17

Gerjets, P., Walter, C., Rosenstiel, W., Bogdan, M., & Zander, T. O. (2014). Cognitive state monitoring and the design of adaptive instruction in digital environments: Lessons learned from cognitive workload assessment using a passive brain-computer interface approach. *Frontiers in Neuroscience*, *8*(DEC), 1–21. https://doi.org/10.3389/fnins.2014.00385

Germanakos, P., & Belk, M. (2016). *Human-Centred Web Adaptation and Personalization*. Springer International Publishing. https://doi.org/10.1007/978-3-319-28050-9

Hajare, R., & Kadam, S. (2021). Comparative study analysis of practical EEG sensors in medical diagnoses. *Global Transitions Proceedings*, *2*(2), 467–475. https://doi.org/10.1016/j.gltp.2021.08.009

Hammad, J., Hariadi, M., Hery Purnomo, M., Jabari, N., & Kurniawan, F. (2018). E-learning and Adaptive E-learning Review. *IJCSNS International Journal of Computer Science and Network Security*, *18*(2), 48.

Herrmann, C. S., Strüber, D., Helfrich, R. F., & Engel, A. K. (2016). EEG oscillations: From correlation to causality. *International Journal of Psychophysiology*, *103*, 12–21. https://doi.org/10.1016/j.ijpsycho.2015.02.003

Hosseini, M.-P., Hosseini, A., & Ahi, K. (2021). A Review on Machine Learning for EEG Signal Processing in Bioengineering. *IEEE Reviews in Biomedical Engineering*, *14*(c), 204–218. https://doi.org/10.1109/RBME.2020.2969915

Hu, P. C., & Kuo, P. C. (2017). Adaptive learning system for E-learning based on EEG brain signals. *2017 IEEE 6th Global Conference on Consumer Electronics, GCCE 2017*, *2017-Janua*(Gcce), 1–2. https://doi.org/10.1109/GCCE.2017.8229382

Jackson, A. F., & Bolger, D. J. (2014). The neurophysiological bases of EEG and EEG measurement: A review for the rest of us. *Psychophysiology*, *51*(11), 1061–1071. https://doi.org/10.1111/psyp.12283

Kakkos, I., Dimitrakopoulos, G. N., Sun, Y., Yuan, J., Matsopoulos, G. K., Bezerianos, A., & Sun, Y. (2021). EEG Fingerprints of Task-Independent Mental Workload Discrimination. *IEEE Journal of Biomedical and Health Informatics*, *25*(10), 3824–3833. https://doi.org/10.1109/JBHI.2021.3085131

Kamel, N. S., & Malik, A. S. (2014). *EEG/ERP Analysis* (K. Nidal & A. S. Malik, Eds.). CRC Press. https://doi.org/10.1201/b17605

Katmah, R., Al-Shargie, F., Tariq, U., Babiloni, F., Al-Mughairbi, F., & Al-Nashash, H. (2021). A Review on Mental Stress Assessment Methods Using EEG Signals. *Sensors*, *21*(15), 5043. https://doi.org/10.3390/s21155043

Kaur, M., Sakhare, S. R., Wanjale, K., & Akter, F. (2022). Early Stroke Prediction Methods for Prevention of Strokes. *Behavioural Neurology*, *2022*, 1–9. https://doi.org/10.1155/2022/7725597

Khedher, A. ben, Jraidi, I., & Frasson, C. (2019). Tracking Students' Mental Engagement Using EEG Signals during an Interaction with a Virtual Learning Environment. *Journal of Intelligent Learning Systems and Applications*, *11*(01), 1–14. https://doi.org/10.4236/jilsa.2019.111001

Kirschstein, T., & Köhling, R. (2009). What is the Source of the EEG? *Clinical EEG and Neuroscience*, *40*(3), 146–149. https://doi.org/10.1177/155005940904000305

Klašnja-Milićević, A., Vesin, B., Ivanoviä, M., Budimac, Z., & Jain, L. C. (2017). Introduction to e-learning systems. In *Intelligent Systems Reference Library* (Vol. 112, pp. 3–17). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-319-41163-7_1

Klem, G. H., Lüders, H. O., Jasper, H. H., & Elger, C. (1999). The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology. *Electroencephalography and Clinical Neurophysiology. Supplement*, *52*(2), 3–6. http://www.ncbi.nlm.nih.gov/pubmed/10590970

Kosmyna, N., & Maes, P. (2019). AttentivU: An EEG-Based Closed-Loop Biofeedback System for Real-Time Monitoring and Improvement of

Engagement for Personalized Learning. *Sensors*, *19*(23), 5200. https://doi.org/10.3390/s19235200

Krigolson, O. E., Hammerstrom, M. R., Abimbola, W., Trska, R., Wright, B. W., Hecker, K. G., & Binsted, G. (2021). Using Muse: Rapid Mobile Assessment of Brain Performance. *Frontiers in Neuroscience*, *15*(January), 1–11. https://doi.org/10.3389/fnins.2021.634147

Li, Z., Zhang, L., Zhang, F., Gu, R., Peng, W., & Hu, L. (2020). Demystifying signal processing techniques to extract resting-state EEG features for psychologists. *Brain Science Advances*, *6*(3), 189–209. https://doi.org/10.26599/BSA.2020.9050019

Light, G. A., Williams, L. E., Minow, F., Sprock, J., Rissling, A., Sharp, R., Swerdlow, N. R., & Braff, D. L. (2010). Electroencephalography (EEG) and event-related potentials (ERPs) with human participants. *Current Protocols in Neuroscience*, *SUPPL. 52*, 1–24. https://doi.org/10.1002/0471142301.ns0625s52

Lin, F. R., & Kao, C. M. (2018). Mental effort detection using EEG data in E-learning contexts. *Computers and Education*, *122*, 63–79. https://doi.org/10.1016/j.compedu.2018.03.020

Liu, X., & Ardakani, S. P. (2022). A machine learning enabled affective E-learning system model. *Education and Information Technologies*, *27*(7), 9913–9934. https://doi.org/10.1007/s10639-022-11010-x

Lotte, F. (2014). A Tutorial on EEG Signal-processing Techniques for Mental-state Recognition in Brain–Computer Interfaces. In *Guide to Brain-Computer Music Interfacing* (pp. 133–161). Springer London. https://doi.org/10.1007/978-1-4471-6584-2_7

Louis, E. K. S., Frey, L. C., Britton, J. W., Hopp, J. L., Korb, P. J., Koubeissi, M. Z., Lievens, W. E., & Pestana-Knight, E. (2016). *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants*.

Lu, X., & Hu, L. (2019). Electroencephalography, Evoked Potentials, and Event-Related Potentials. In L. Hu & Z. Zhang (Eds.), *EEG Signal Processing and Feature Extraction* (pp. 23–42). Springer Singapore. https://doi.org/10.1007/978-981-13-9113-2_3

Malik, A. S., & Amin, H. U. (2017). Chapter 1 - Designing an EEG Experiment. In A. S. Malik & H. U. Amin (Eds.), *Designing EEG Experiments for Studying the Brain* (pp. 1–30). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-811140-6.00001-1

Miklody, D., Uitterhoeve, W. M., van Heel, D., Klinkenberg, K., & Blankertz, B. (2017). Maritime cognitive workload assessment. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9961 LNCS*, 102–114. https://doi.org/10.1007/978-3-319-57753-1_9

Mühl, C., Jeunet, C., & Lotte, F. (2014). EEG-based workload estimation across affective contexts. *Frontiers in Neuroscience*, *8*(8 JUN), 1–15. https://doi.org/10.3389/fnins.2014.00114

Nagy, A. (2017). The Impact of E-Learning. In *E-Content* (Vol. 13, Issue 5, pp. 79–96). Springer-Verlag. https://doi.org/10.1007/3-540-26387-X_4

Nandi, A., Xhafa, F., Subirats, L., & Fort, S. (2021). Real-time emotion classification using eeg data stream in e-learning contexts. *Sensors*, *21*(5), 1–26. https://doi.org/10.3390/s21051589

OpenBCI. (n.d.-a). *All-in-one EEG electrode cap starter kit*. Retrieved August 10, 2022, from https://shop.openbci.com/products/openbci-eeg-electrocap-kit

OpenBCI. (n.d.-b). *Low-cost Biosensing Starter Kit*. Retrieved August 10, 2022, from https://shop.openbci.com/products/bundle2

Papatheocharous, E., Belk, M., Germanakos, P., & Samaras, G. (2014). Towards Implicit User Modeling Based on Artificial Intelligence, Cognitive Styles and Web Interaction Data. *International Journal on Artificial Intelligence Tools*, *23*(02), 1440009. https://doi.org/10.1142/S0218213014400090

Parfenov, A. (2022). *Brainflow*. https://brainflow.org/

Peng, W. (2019). EEG Preprocessing and Denoising. In *EEG Signal Processing and Feature Extraction* (pp. 71–87). Springer Singapore. https://doi.org/10.1007/978-981-13-9113-2_5

Pope, A. T., Bogart, E. H., & Bartolome, D. S. (1995). Biocybernetic system evaluates indices of operator engagement in automated task. *Biological Psychology*, *40*(1–2), 187–195. https://doi.org/10.1016/0301-0511(95)05116-3

Portillo-Lara, R., Tahirbegi, B., Chapman, C. A. R., Goding, J. A., & Green, R. A. (2021). Mind the gap: State-of-the-art technologies and applications for EEG-based brain–computer interfaces. *APL Bioengineering*, *5*(3), 031507. https://doi.org/10.1063/5.0047237

Qu, H., Zhang, M., & Pang, L. (2022). Mental Workload Classification Method Based on EEG Cross-Session Subspace Alignment. *Mathematics*, *10*(11), 1875. https://doi.org/10.3390/math10111875

Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*, 35–39. https://doi.org/10.1109/COMITCon.2019.8862451

Richer, R., Zhao, N., Amores, J., Eskofier, B. M., & Paradiso, J. A. (2018). Real-time Mental State Recognition using a Wearable EEG. *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *2018*, 5495–5498. https://doi.org/10.1109/EMBC.2018.8513653

Russell, D. M., Klemmer, S., Fox, A., Latulipe, C., Duneier, M., & Losh, E. (2013). Will massive online open courses (moocs) change education? *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*, 2395. https://doi.org/10.1145/2468356.2468783

Saeidi, M., Karwowski, W., Farahani, F. v., Fiok, K., Taiar, R., Hancock, P. A., & Al-Juaid, A. (2021). Neural Decoding of EEG Signals with Machine Learning: A Systematic Review. *Brain Sciences*, *11*(11), 1525. https://doi.org/10.3390/brainsci11111525

Sanei, S., & Chambers, J. A. (2007). *EEG Signal Processing*. John Wiley & Sons Ltd,. https://doi.org/10.1002/9780470511923

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 1–21. https://doi.org/10.1007/s42979-021-00592-x

Sawangjai, P., Hompoonsup, S., Leelaarporn, P., Kongwudhikunakorn, S., & Wilaiprasitporn, T. (2020). Consumer Grade EEG Measuring Sensors as Research Tools: A Review. *IEEE Sensors Journal*, *20*(8), 3996–4024. https://doi.org/10.1109/JSEN.2019.2962874

Schrepp, M. (2019). *User Experience Questionnaire Handbook Version 8*. *September 2015*, 1–15. https://doi.org/10.13140/RG.2.1.2815.0245

Sciaraffa, N., Aricò, P., Borghini, G., Flumeri, G. di, Florio, A. di, & Babiloni, F. (2019). On the Use of Machine Learning for EEG-Based Workload Assessment: Algorithms Comparison in a Realistic Task. *Communications in Computer and Information Science*, *1107*, 170–185. https://doi.org/10.1007/978-3-030-32423-0_11

Shaked, U. (2021). *musejs*. GitHub Repository. https://github.com/urish/muse-js

Shishkin, S. L. (2022). Active Brain-Computer Interfacing for Healthy Users. *Frontiers in Neuroscience*, *16*(April), 1–4. https://doi.org/10.3389/fnins.2022.859887

Shoorangiz, R., Weddell, S. J., & Jones, R. D. (2021). EEG-Based Machine Learning: Theory and Applications. In *Handbook of Neuroengineering* (pp. 1–39). https://doi.org/10.1007/978-981-15-2848-4_70-1

Slimen, I. ben, Boubchir, L., & Seddik, H. (2020). Epileptic seizure prediction based on EEG spikes detection of ictal-preictal states. *The Journal of Biomedical Research*, *34*(3), 162. https://doi.org/10.7555/JBR.34.20190097

Soufineyestani, M., Dowling, D., & Khan, A. (2020). Electroencephalography (EEG) technology applications and available devices. *Applied Sciences (Switzerland)*, *10*(21), 1–23. https://doi.org/10.3390/app10217453

Srinivasan, L., Scharnagl, J., & Schilling, K. (2013). Analysis of WebSockets as the New Age Protocol for Remote Robot Tele-operation. *IFAC Proceedings Volumes*, *46*(29), 83–88. https://doi.org/10.3182/20131111-3-KR-2043.00032

Tambe, N. R., & Khachane, A. (2017). Mood based E-learning using EEG. *Proceedings - 2nd International Conference on Computing, Communication, Control and Automation, ICCUBEA 2016*, 2–5. https://doi.org/10.1109/ICCUBEA.2016.7860018

Teplan, M. (2002). Fundamental of EEG Measurement. *Measurement Science Review*, *2*(2), 59–64.

Thompson, M. C. (2019). Critiquing the Concept of BCI Illiteracy. *Science and Engineering Ethics*, *25*(4), 1217–1233. https://doi.org/10.1007/s11948-018-0061-1

Tivadar, R. I., & Murray, M. M. (2019). A Primer on Electroencephalography and Event-Related Potentials for Organizational Neuroscience. *Organizational Research Methods*, *22*(1), 69–94. https://doi.org/10.1177/1094428118804657

Vidaurre, C., & Blankertz, B. (2010). Towards a cure for BCI illiteracy. *Brain Topography*, *23*(2), 194–198. https://doi.org/10.1007/s10548-009-0121-6

Walter, C., Rosenstiel, W., Bogdan, M., Gerjets, P., & Spüler, M. (2017). Online EEG-based workload adaptation of an arithmetic learning environment. *Frontiers in Human Neuroscience*, *11*(May), 1–10. https://doi.org/10.3389/fnhum.2017.00286

Wang, Z., Hope, R. M., Wang, Z., Ji, Q., & Gray, W. D. (2012). Cross-subject workload classification with a hierarchical Bayes model. *NeuroImage*, *59*(1), 64–69. https://doi.org/10.1016/j.neuroimage.2011.07.094

Xia, X., & Hu, L. (2019). EEG: Neural Basis and Measurement. In L. Hu & Z. Zhang (Eds.), *EEG Signal Processing and Feature Extraction* (pp. 7–21). Springer Singapore. https://doi.org/10.1007/978-981-13-9113-2_2

Xu, Y., & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, *2*(3), 249–262. https://doi.org/10.1007/s41664-018-0068-2

You, S. D. (2021). Classification of Relaxation and Concentration Mental States with EEG. *Information*, *12*(5), 187. https://doi.org/10.3390/info12050187

Zander, T. O., & Kothe, C. (2011). Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of Neural Engineering*, *8*(2), 025005. https://doi.org/10.1088/1741-2560/8/2/025005

Zander, T. O., Lehne, M., Ihme, K., Jatzev, S., Correia, J., Kothe, C., Picht, B., & Nijboer, F. (2011). A Dry EEG-System for Scientific Research and Brain–Computer Interfaces. *Frontiers in Neuroscience*, *5*, 1–10. https://doi.org/10.3389/fnins.2011.00053

Zhang, Z. (2019). Spectral and Time-Frequency Analysis. In *EEG Signal Processing and Feature Extraction* (pp. 89–116). Springer Singapore. https://doi.org/10.1007/978-981-13-9113-2_6