



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД



Методе мерења електричних
активности мозга у контексту
истраживања квантитативне
електроенцефалографије,
неурофидбека и неуропластичности

Докторска дисертација

Ментор:

Проф. др Платон Совиљ

Студент:

Никола Петровић

Нови Сад, 2024.

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА¹

Врста рада:	Докторска дисертација
Име и презиме аутора:	Никола Петровић
Ментор (титула, име, презиме, звање, институција)	др Платон Совиљ, редовни професор, Факултет техничких наука, Универзитет у Новом Саду
Наслов рада:	Методe мерења електричних активности мозга у контексту истраживања квантитативне електроенцефалографије, неурофидбека и неуропластичности
Језик публикације (писмо):	Српски (ћирилица)
Физички опис рада:	Унети број: Страница 210 Поглавља 8 Референци 131 Табела 2 Слика 133 Графикона 0 Прилога 4
Научна област:	Електротехничко и рачунарско инжењерство
Ужа научна област (научна дисциплина):	Електрична мерења, метрологија и биомедицина
Кључне речи / предметна одредница:	Методe мерења, мозак, електроенцефалографија, неурофидбек, неуропластичност
Резиме на језику рада:	<p>Електроенцефалографија (ЕЕГ) представља неинвазивну технику за мерење електричних активности мозга, њена примена у когнитивним неуронаукама даје могућност истраживања комплексних когнитивних феномена, као што су памћење, пажња, учење и неуропластичност. У складу са актуелним стањем у когнитивним неуронаукама, тема овог истраживања оријентисана је на доказивање хипотезе која је фокусирана на истраживање неуропластичности и претпоставке да се неуропластичност повећава с употребом неурофидбек система. Друга хипотеза је уверавање у поузданост система и његову способност да производи резултате који су у складу с већ постојећим конвенционалним системима у медицини и биомедицини.</p> <p>Употребом вишеканалног ЕЕГ уређаја и развијеног мозак - рачунар система, спроведен је низ експеримената где су испитаници помоћу менталних слика вршили управљање симулираним и физичким системима.</p> <p>Развијени систем се састоји од седам подсистема и њихова улога је да обезбеде мерење, анализу и обраду ЕЕГ сигнала, спровођење неурофидбек експеримената са симулираним и физичким излазима, психолошка тестирања и евалуацију, мапирање мозга на основу ЕЕГ сигнала као и класификацију и регресију когнитивних феномена. Систем</p>

¹ Аутор докторске дисертације потписао је и приложио следеће обрасце:

5б – Изјава о ауторству;

5в – Изјава о истоветности штампане и електронске верзије и о личним подацима;

5г – Изјава о коришћењу.

Ове изјаве се чувају на Факултету у штампаном и електронском облику и не корице се с тезом.

	<p>је дизајниран да обезбеди комуникацију између стручњака из области инжењерства и психологије.</p> <p>Основна идеја овог истраживања је развој иновативног неурофидбек система који користи виртуелне и физичке излазе за подстицање неуропластичности код испитаника. Циљ система је да се стимулише и обучи мозак да промени своје радне обрасце, чиме се побољшавају когнитивне способности и опште психичко стање.</p> <p>Неурофидбек систем је тестиран на двадесет испитаника и резултати мерења ЕЕГ сигнала заједно са опажањима из експеримента су нам дали у увид промене у когнитивним феноменима током експеримената.</p> <p>Евалуација је извршена помоћу алата за мултиспектрално мапирање мозга, психолошку евалуацију, класификацију и регресију.</p> <p>Добијени резултати су показали да употреба развијеног система код здравих испитаника успешно развија нове когнитивне функције које су задужене за контролу излазних вредности, ово истраживање даје темеље за развој система са више примена.</p>
Датум прихватања теме од стране надлежног већа:	27. 04. 2023.
Датум одбране: (Попуњава одговарајућа служба)	
Чланови комисије: (титула, име, презиме, звање, институција)	<p>Председник: др Драган Пејић, редовни професор, Факултет техничких наука, Нови Сад</p> <p>Члан: др Вања Ковић, редовни професор, Филозофски факултет, Београд</p> <p>Члан: др Немања Газивода, доцент, Факултет техничких наука, Нови Сад</p> <p>Члан: др Ђорђе Новаковић, доцент, Факултет техничких наука, Нови Сад</p> <p>Ментор: др Платон Совиљ, редовни професор, Факултет техничких наука, Нови Сад</p>
Напомена:	

KEY WORD DOCUMENTATION²

<i>Document type:</i>	Doctoral dissertation
<i>Author:</i>	Nikola Petrović
<i>Supervisor (title, first name, last name, position, institution)</i>	Dr. Platon Sovilj, Full Professor, Faculty of Technical Sciences, University of Novi Sad
<i>Thesis title:</i>	Methods of Measuring Brain Electrical Activity in the Context of Research on Quantitative Electroencephalography, Neurofeedback, and Neuroplasticity
<i>Language of text (script):</i>	Serbian language (Cyrillic)
<i>Physical description:</i>	Number of: Pages 205 Chapters 8 References 131 Tables 2 Illustrations 133 Graphs 0 Appendices 4
<i>Scientific field:</i>	Electrical and Computer Engineering
<i>Scientific subfield (scientific discipline):</i>	Electrical Measurements, Metrology, and Biomedicine
<i>Subject, Key words:</i>	Measurement methods, brain, electroencephalography, neurofeedback, neuroplasticity
<i>Abstract in English language:</i>	<p>Electroencephalography (EEG) is a non-invasive technique for measuring the electrical activity of the brain. Its application in cognitive neuroscience offers the opportunity to research complex cognitive phenomena such as memory, attention, learning, and neuroplasticity.</p> <p>In line with the current state of cognitive neuroscience, the focus of this research is to test a hypothesis centered on the study of neuroplasticity, particularly the theory that neuroplasticity increases with the use of neurofeedback systems. A secondary hypothesis concerns the system's reliability and its capacity to yield results consistent with those of existing conventional systems in medicine and biomedicine.</p> <p>Utilizing a multi-channel EEG device and a developed brain-computer interface, a series of experiments were conducted in which subjects controlled both simulated and physical systems using mental imagery.</p> <p>The developed system comprises seven subsystems, each designed to measure, analyze, and process EEG signals, conduct neurofeedback experiments with both simulated and physical outputs, and perform psychological testing and evaluation. Additionally, it supports brain mapping based on EEG signals and facilitates the classification and regression of cognitive phenomena. The system aims to enhance communication between professionals in engineering and psychology.</p> <p>The core objective of this research is to devise an innovative neurofeedback system that employs both virtual and physical outputs to promote neuroplasticity in subjects. The system's goal is to stimulate and train the brain</p>

² The author of doctoral dissertation has signed the following Statements:

5б – Statement on the authority,

5в – Statement that the printed and e-version of doctoral dissertation are identical and about personal data,

5г – Statement on copyright licenses.

The paper and e-versions of Statements are held at the faculty and are not included into the printed thesis.

	<p>to alter its operational patterns, thus enhancing cognitive abilities and overall mental health.</p> <p>The neurofeedback system was tested on twenty subjects, and the analysis of EEG signal measurements, coupled with observations from the experiments, provided insights into cognitive changes during the study. Evaluation utilized multispectral brain mapping tools, psychological assessments, and techniques for classification and regression.</p> <p>The findings indicate that the application of the developed system in healthy subjects successfully fosters new cognitive functions responsible for managing output values. This research lays the groundwork for developing systems with broader applications.</p>
<i>Accepted on Scientific Board on:</i>	27. 04. 2023.
Defended: (Filled by the faculty service)	
<i>Thesis Defend Board: (title, first name, last name, position, institution)</i>	<p>Chairman: Dragan Pejić PhD, Full Professor, Faculty of Technical Sciences, Novi Sad</p> <p>Member: Vanja Ković PhD, Full Professor, Faculty of Philosophy, Belgrade</p> <p>Member: Nemanja Gazivoda PhD, Assistant Professor, Faculty of Technical Sciences, Novi Sad</p> <p>Member: Đorđe Novaković PhD, Assistant Professor, Faculty of Technical Sciences, Novi Sad</p> <p>Mentor: Platon Sovilj PhD, Full Professor, Faculty of Technical Sciences, Novi Sad</p>
Note:	

Сажетак

Когнитивне неуронауке представљају мост између когнитивне психологије и биолошких неуронаука исписујући своју научну и клиничку сврху кроз изучавање можданих процеса који леже у основи когнитивних функција. Електроенцефалографија (ЕЕГ) је кључна у истраживањима у области когнитивних неуронаука омогућавајући прецизно временско праћење електричних активности мозга. Један од најзагонетнијих аспеката истраживања мозга јесте неуропластичност, способност мозга да мења своју структуру и функцију као одговор на искуства и обуку. Неурофидбек, техника која користи ЕЕГ сигнале за процену когнитивних стања и когнитивних активности, омогућава индивидуама да узимају активну улогу у регулацији своје мождане активности и на тај начин остваре значајно побољшање у поспешивању неуропластичности, подржавајући како клиничку интервенцију тако и когнитивно обогаћивање као и учење.

У контексту овог истраживања, развијен је савремен систем који комбинује напредне методе ЕЕГ-а и технике неурофидбека с циљем појачаног стимулације и мониторинга неуропластичних промена. Систем је дизајниран да обезбеди крајњем кориснику интензивну и директну повратну информацију о његовој можданој активности у виду симулација, виртуелне реалности или роботизованих система, омогућавајући тако дубље разумевање и контролу когнитивних процеса који су у току.

Испитиване су различите експерименталне фазе и ситуације, користећи се разнообразним демографским чиниоцима као што су пол, старост и предзнање учесника. Прва фаза, која је служила као увод и припрема за даља истраживања, омогућила је тестирање различитих методологија и алата. У наредним фазама, укључујући пилот-фазу, основну и проширену експерименталну фазу, детаљно је анализирано како испитаници реагују на различите стимулусе и технологије, водећи рачуна о њиховим индивидуалним разликама.

Свака фаза истраживања доносила је са собом различите изазове. Почевши од питања интеграције опреме и софтвера, преко проблема адаптације испитаника различитим стимулусима, све до тежњи да се оствари конзистентност у извођењу експеримената. Сваки од изазова је пажљиво разматран и адресиран, а решења су интегрисана у финални дизајн истраживања. Да би се постигла максимална робустност и квалитет, уведене су многобројне контролне и валидационе мере. Оне укључују систематско праћење неурофизиолошког стања учесника (ЕЕГ, психолошке процене, мапирање мозга), дубоке анализе података, упоредне студије и валидацију резултата с другим поузданим изворима.

Кључни налази овог истраживања указују на то да адаптација особа на неурофидбек систем резултира променама на централном нервном систему које се могу дефинисати као појачана неуропластичност. Ова открића представљају значајан корак напред у области когнитивних неуронаука и постављају темеље за интеграцију различитих научних дисциплина, укључујући неурологију, психологију, психијатрију и техничке науке.

Abstract

Cognitive neuroscience represents a bridge between cognitive psychology and biological neurosciences, delineating its scientific and clinical purpose through the study of brain processes that underlie cognitive functions. Electroencephalography (EEG) is pivotal in research in the realm of cognitive neurosciences, allowing for precise temporal monitoring of the brain's electrical activity. One of the most enigmatic aspects of brain research is neuroplasticity, the ability of the brain to change its structure and function in response to experiences and training. Neurofeedback, a technique that utilizes EEG signals to assess cognitive states and cognitive activities, allows individuals to take an active role in regulating their brain activity, thereby achieving significant enhancement in promoting neuroplasticity, supporting both clinical intervention and cognitive enrichment as well as learning.

In the context of this research, a modern system has been developed that combines advanced EEG methods and neurofeedback techniques with the aim of enhanced stimulation and monitoring of neuroplastic changes. The system is designed to provide the end user with intensive and direct feedback about his/her brain activity in the form of simulations, virtual reality, or robotic systems, thus allowing a deeper understanding and control of ongoing cognitive processes.

Different experimental phases and situations were examined, utilizing a variety of demographic factors such as gender, age, and prior knowledge of participants. The first phase, which served as an introduction and preparation for further research, allowed for testing of various methodologies and tools. In subsequent phases, including the pilot, basic, and extended experimental phases, it was meticulously analyzed how subjects respond to different stimuli and technologies, taking into account their individual differences.

Each research phase brought its own set of challenges. Starting from the issue of integrating equipment and software, through the problems of adapting subjects to various stimuli, all the way to striving for consistency in conducting experiments. Each challenge was carefully considered and addressed, with solutions integrated into the final research design. To achieve maximum robustness and quality, numerous control and validation measures were introduced. These include systematic tracking of the neurophysiological state of participants (EEG, psychological assessments, brain mapping), in-depth data analysis, comparative studies, and validation of results with other reliable sources.

The key findings of this research indicate that the adaptation of individuals to the neurofeedback system results in changes in the central nervous system that can be defined as enhanced neuroplasticity. These discoveries represent a significant step forward in the field of cognitive neurosciences and lay the foundation for the integration of various scientific disciplines, including neurology, psychology, psychiatry, and technical sciences.

Захвалница

Са дубоким поштовањем се захваљујем ментору, проф. др Платону Совиљу. Његово знање, стрпљење, преданост и неизмерна подршка у протеклих десет година не само да су ми отворили врата науке и истраживања, већ су били кључни у обликовању мене као особе спремне да се суочи са животним и научним изазовима. Његова способност да у сваком тренутку пружи подршку и мотивацију, играла је пресудну улогу у мом академском и личном развоју.

Посебну захвалност дугујем проф. др Светлани Боројевић, чија ми је стручна помоћ и пружени материјали из области психологије били од непроцењивог значаја. Њен рад и ентузијазам су ме инспирисали за даља истраживања и улазак у свет психологије. Такође бих се захвалио и проф. др Стевану Цветићанину на подршци, саветима у вези техничких и научно-истраживачких корака, као и на мотивацији која ми је помогла да превазиђем многе препреке.

Најдубљу захвалност дугујем својој породици. Моји родитељи, отац Јовица и мајка Јадранка, који су ме увек безусловно подржавали, верујући у мене чак и када сам ја сумњао. Њихова љубав, жртва и охрабрење су били мој најчвршћи ослонац. Мојој сестри Никол и брату Давиду, који су ми били подршка и уточиште, пружајући осмех и растерећење у најизазовнијим тренуцима.

Овај рад посвећујем својој породици и будућим генерацијама, с надом да ће моје истраживање допринети како њиховом разумевању, тако и инспирацији за даља истраживања.

Списак коришћених скраћеница

Скраћеница	Опис
ЕЕГ	Електроенцефалографија
Ост-20	Стандард за распоред ЕЕГ електрода
QEEG	Квантитативна електроенцефалографија
BCI	Brain-Computer Interface – Мозак-рачунар систем
АЛС	Амиотрофична латерална склероза
ADHD	Attention deficit hyperactivity disorder - Поремећај пажње и хиперактивност
фМРИ	Функциона магнетна резонанца
ПЕТ	Позитронска електротомографија
ТМС	Транскранијална магнетна стимулација
МЕГ	Магнетоенцефалографија
ERP	Event Related Potential - Потенцијал везан за догађај
tDCS	Транскранијална директна струјна стимулација
XR	Extended Reality - Продужена реалност
VR	Virtual Reality - Виртуелна реалност
AR	Augmented Reality - Побољшана реалност
MR	Mixed Reality - Измешана реалност
NFB	Neurofeedback - Неурофидбек
ВИ	Вештачка интелигенција
DNNs	Deep Neural Networks - Дубоке неуралне мреже
API	Application Programming Interface - Програмски интерфејс апликације
IDE	Integrated Development Environment - Интегрисано развојно окружење
I2C	Inter-Integrated Circuit - Интеринтегрисано коло
SPI	Serial Peripheral Interface - Серијски периферни интерфејс
ARM	Advanced RISC Machine
USB	Universal Serial Bus - Универзална серијска магистрала
PWM	Pulse-width modulation - Модулација ширине импулса
OS	Operative System - Оперативни систем
FPV	First Person View - Поглед из првог лица
ЦНС	Централни нервни систем
CSV	Comma Separated Values - Вредности одвојене зарезима
EDF	European Data Format - Европски формат података
DICOM	Digital Imaging and Communications in Medicine - Дигиталне слике и комуникације у медицини
ЛЕП-БЛ	Лабораторија за експерименталну психологију - Бања Лука
MNE	Minnesota NeuroElectroMagnetics
STFT	Short-Time Fourier Transform - Кратка Фуријеова трансформација
CNN	Convolutional Neural Network - Конволутивна неурална мрежа
WMQ	Working Memory Questionnaire - Упитник о радној меморији
ИД	Идентификација
json	JavaScript Object Notation - JavaScript нотација објеката
ОАС	Основне академске студије
ICA	Independent Component Analysis - Анализа независних компонената
ЕМГ	Електромиографија
ЕКГ	Електрокардиографија
F1	F1 оцена класификатора
MCC	Matthew's correlation coefficient - Метјуов коефицијент корелације
TP	Стопа стварних позитива
TN	Стопа стварних негатива
FP	Стопа лажних позитива
FN	Стопа лажних негатива

Садржај

1	Увод.....	1
1.1	Проблем истраживања	1
1.2	Електроенцефалографија.....	1
1.2.1	Значај ЕЕГ-а у когнитивним неуронаукама.....	3
1.2.2	Улога ЕЕГ-а у неурологији	3
1.2.3	ЕЕГ у психологији и психијатрији.....	3
1.3	Когнитивне неуронауке.....	4
1.4	Неуропластичност.....	6
1.4.1	Механизми неуропластичности	6
1.4.2	Фактори који утичу на неуропластичност.....	7
1.4.3	Улога неуропластичности у обучавању и терапији.....	7
1.4.4	Изазови и будући правци истраживања.....	7
1.4.5	Неуропластичност и машинско учење	8
1.4.6	Закључак.....	8
1.5	Когнитивни феномени	8
1.5.1	Памћење.....	8
1.5.2	Пажња.....	9
1.5.3	Размишљање.....	9
1.5.4	Учење	9
1.5.5	Менталне слике	9
1.5.6	Визуализација	9
1.5.7	Концентрација.....	10
1.5.8	Закључак.....	10
1.6	Мозак-рачунар системи (BCI)	11
1.6.1	Технологије за снимање мозга.....	11
1.6.2	Обрада и декодирање сигнала	11
1.6.3	Апликације BCI-ја.....	12
1.6.4	Изазови и будући развој	12
1.7	Мапирање мозга.....	13
1.8	Продужена реалност	13
1.9	Системи машинског учења	14
1.9.1	Системи за класификацију и регресију	15
1.9.2	Валидациони параметри	16
2	Актуелно стање у области.....	18
2.1	Преглед владајућих ставова и схватања.....	18

2.1.1	Неурофидбек системи.....	18
2.1.2	Развој и испитивање неуропластичности.....	19
2.1.3	Системи вештачке интелигенције и њихова примена над ЕЕГ сигналама.....	19
2.2	Образложење о потребама научног истраживања.....	20
2.3	Циљ истраживања и хипотеза.....	20
3	Методологија.....	21
3.1	Опрема.....	21
3.1.1	<i>Emotiv Insight</i> и <i>Emotiv Epos</i> : Алати за мерење електричне активности мозга.....	21
3.1.2	Arduino развојно окружење.....	34
3.1.3	Unity програмско окружење.....	37
3.1.4	Oculus Quest 2 систем.....	39
3.2	Опис експеримента.....	41
3.2.1	Припрема испитаника.....	42
3.2.2	Једнодимензионалне неурофидбек вежбе.....	42
3.2.3	Мултимедијалне дидактичке вежбе.....	43
3.2.4	Актуаторске неурофидбек вежбе.....	46
3.2.5	Тестирање и преглед испитаника.....	46
3.2.6	Место истраживања.....	47
3.2.7	Спроведени експерименти.....	48
3.3	Архитектура система.....	48
3.3.1	Систем за аквизицију сигнала.....	49
3.3.2	Систем за складиштење, анализу и обраду сигнала.....	50
3.3.3	Неурофидбек симулација.....	50
3.3.4	Неурофидбек актуатор.....	56
3.3.5	Мултиспектрално мапирање мозга.....	59
3.3.6	Систем за класификацију и регресију когнитивних стања и когнитивних активности.....	61
3.3.7	Психолошко тестирање и евалуација.....	64
4	Резултати.....	71
4.1	Неурофидбек.....	71
4.1.1	Запажања за обуку и вежбање одрживости менталне концентрације.....	71
4.1.2	Запажања за обуку и вежбање агилности менталне концентрације.....	71
4.1.3	Запажања за обуку и вежбање контроле менталне концентрације.....	71
4.1.4	Arduino платформа.....	71
4.1.5	Mentis Oculi.....	72
4.2	Мултиспектрално мапирање мозга.....	72
4.3	Систем за класификацију и регресију когнитивних стања и когнитивних активности..	75

4.4	Дигитална интерактивна психологија.....	79
5	Закључак.....	80
5.1	Даљи развој система.....	80
5.2	Примена развијеног система.....	81
5.2.1	Медицинска примена.....	81
5.2.2	Рехабилитациона примена.....	81
5.2.3	Едукативна примена.....	81
5.2.4	Асистивна примена.....	82
5.2.5	Научноистраживачка примена.....	82
6	Литература.....	83
7	Прилози.....	89
7.1	Додатни резултати.....	89
7.1.1	Мапирање мозга.....	89
7.2	Кодови.....	114
7.2.1	Неурофидбек симулација.....	114
7.2.2	Неурофидбек актуатор.....	143
7.2.3	Мултиспектрално мапирање мозга.....	147
7.2.4	Систем за класификацију и регресију.....	167
7.2.5	Психолошко тестирање и евалуација.....	183
7.3	Списак објављених научних радова.....	203
7.4	Биографија кандидата.....	204

Списак слика

Слика 1 Распоред ЕЕГ електрода 10-20 [4].....	1
Слика 2 Приказ основних таласа ЕЕГ сигнала [6].....	2
Слика 3 Приказ релација између развоја и учења као и њихов утицај на неуралну пластичност [29]6	
Слика 4 Пример мозак-рачунар система [70]	11
Слика 5 Графички приказ тока информација у оквиру продужене реалности [91]	14
Слика 6 Методологија неурофидбек система [100]	18
Слика 7 Emotiv Insight уређај [110]	22
Слика 8 Emotiv Ерос уређај [113]	23
Слика 9 Emotiv Insight електроде [110]	23
Слика 10 Emotiv Ерос електроде [112]	24
Слика 11 Emotiv Launcher [115]	25
Слика 12 Мерење и приказ ЕЕГ сигнала у реалном времену [116]	26
Слика 13 Приказ фреквенцијских компонената изабраног канала [116].....	26
Слика 14 Мерење и приказ когнитивних стања испитаника [116]	27
Слика 15 Мерење и приказ сигнала са жироскопа, акцелерометра и магнетометра [116]	27
Слика 16 Преглед сачуваних ЕЕГ сигнала током експеримената [116]	28
Слика 17 Emotiv BCI [117].....	29
Слика 18 Emotiv BCI и рад у реалном времену [117]	29
Слика 19 Алат за тренирање и праћење фацијалних експресија [117]	30
Слика 20 Emotiv OSC [117]	31
Слика 21 Emotiv BrainViz [118].....	31
Слика 22 Arduino UNO [120]	35
Слика 23 Unity програмско окружење	38
Слика 24 Oculus Quest 2 с контролерима [126]	39
Слика 25 Oculus Quest 2 контролери [126].....	40
Слика 26 Oculus link апликација [126]	40
Слика 27 Кориснички интерфејс контролисан помоћу препознавања покрета руку [126].....	41
Слика 28 Пример правилног положаја седења	42
Слика 29 Снимак симулације вожње аутомобила уз приказ мождане активности.....	44
Слика 30 Снимак симулатора акционих маневара за агилност менталне концентрације.....	45
Слика 31 Снимак симулатора летења за контролу менталне концентрације	45
Слика 32 Архитектура система	48
Слика 33 Архитектура неурофидбек симулатора	50
Слика 34 Конфигурација система са рачунаром.....	53
Слика 35 Конфигурација система са VR уређајем	53
Слика 36 Подешавања симулације.....	54
Слика 37 Симулација из трећег лица.....	54
Слика 38 Симулација из првог лица	55
Слика 39 Снимак симулације, фацијалних експресија и мапирања мозга	55
Слика 40 Архитектура неурофидбек актуатора	56
Слика 41 Блок шема актуаторског неурофидбек система	59
Слика 42 Софтверска архитектура алата за мапирање мозга	59

Слика 43 Кораца система за класификацију и регресију.....	62
Слика 44 Организација података	62
Слика 45 Методологија развијеног алата	65
Слика 46 Приказ исечка табеле правилно форматираног теста	65
Слика 47 Приказ учитавања прилагођеног теста	66
Слика 48 Тестирање	66
Слика 49 Панел за чување резултата теста	67
Слика 50 Хијерархија резултата једног теста	68
Слика 51 Учитавање резултата.....	68
Слика 52 Приказ остварених резултата.....	69
Слика 53 Приказ датих одговора	69
Слика 54 Исечак снимка мапирања мозга	73
Слика 55 Мапирање мозга током фазе одмора	74
Слика 56 Мапирање мозга током успешног рачунања.....	74
Слика 57 Мапирање мозга током релаксације	74
Слика 58 Мапирање мозга током лошег рачунања	75
Слика 59 Губици модела током 50 епоха обуке	76
Слика 60 Тачност модела током 50 епоха обучавања	77
Слика 61 Матрица конфузије за валидациони узорак скупа података (TP = 12337, TN = 444, FP = 78, FN = 49).	78
Слика 62 Мапирање мозга - прилог	90
Слика 63 Мапирање мозга - прилог	91
Слика 64 Мапирање мозга - прилог	91
Слика 65 Мапирање мозга - прилог	91
Слика 66 Мапирање мозга - прилог	92
Слика 67 Мапирање мозга - прилог	92
Слика 68 Мапирање мозга - прилог	92
Слика 69 Мапирање мозга - прилог	93
Слика 70 Мапирање мозга - прилог	93
Слика 71 Мапирање мозга - прилог	93
Слика 72 Мапирање мозга - прилог	94
Слика 73 Мапирање мозга - прилог	94
Слика 74 Мапирање мозга - прилог	94
Слика 75 Мапирање мозга - прилог	95
Слика 76 Мапирање мозга - прилог	95
Слика 77 Мапирање мозга - прилог	95
Слика 78 Мапирање мозга - прилог	96
Слика 79 Мапирање мозга - прилог	96
Слика 80 Мапирање мозга - прилог	96
Слика 81 Мапирање мозга - прилог	97
Слика 82 Мапирање мозга - прилог	97
Слика 83 Мапирање мозга - прилог	97
Слика 84 Мапирање мозга - прилог	98
Слика 85 Мапирање мозга - прилог	98

Слика 130 Мапирање мозга - прилог	113
Слика 131 Мапирање мозга - прилог	113
Слика 132 Мапирање мозга - прилог	114
Слика 133 Мапирање мозга – прилог.....	114

Списак табела

Табела 1 Структура сачуваних резултата једног теста	67
Табела 2 Списак слика мапирања с додатним подацима	90

1 Увод

1.1 Проблем истраживања

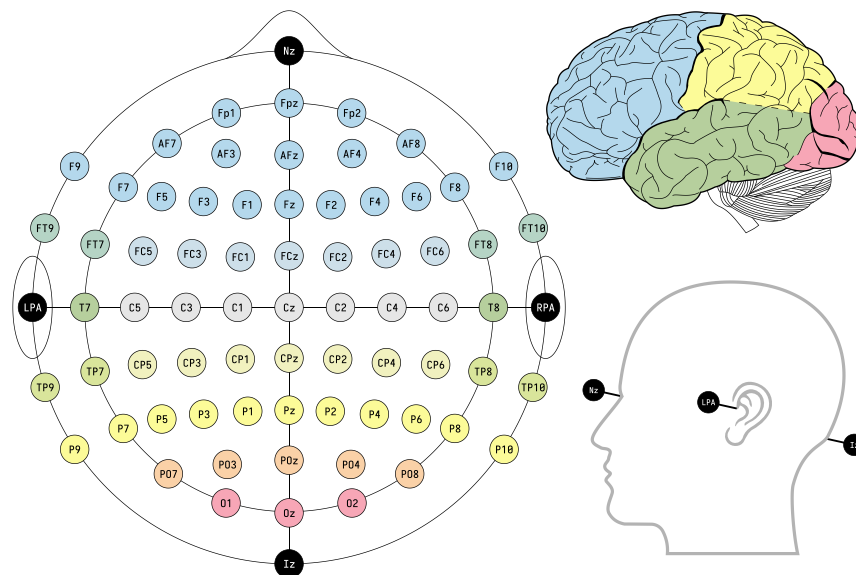
У фокусу ове дисертације се налазе методе мерења електричних активности мозга у контексту истраживања квантитативне електроенцефалографије, неурофидбека и неуропластичности. Овај рад има за циљ даљи развој наведених метода и одговарајућих система за обуку и тестирање испитаника, истовремено обезбеђујући мерење, обраду, анализу, класификацију и регресију сигнала.

Мотивација за ову тезу је побољшање постојећих метода мерења електричних активности мозга – са циљем унапређења неурофидбек система и алата за истраживање појаве неуропластичности. Целокупно истраживање је подељено у целине које обухватају развој, тестирање, рад с испитаницима, обраду података и доношење закључака.

Методе мерења које су предмет наведеног побољшања су електроенцефалографија и квантитативна електроенцефалографија. Предложена побољшања имају за даљи циљ да омогуће испитаницима унапређење когнитивних функција и стања. Овај рад се такође може посматрати као основа за даља истраживања у области когнитивних неуронаука, која су дефинисана у дискусији и закључку.

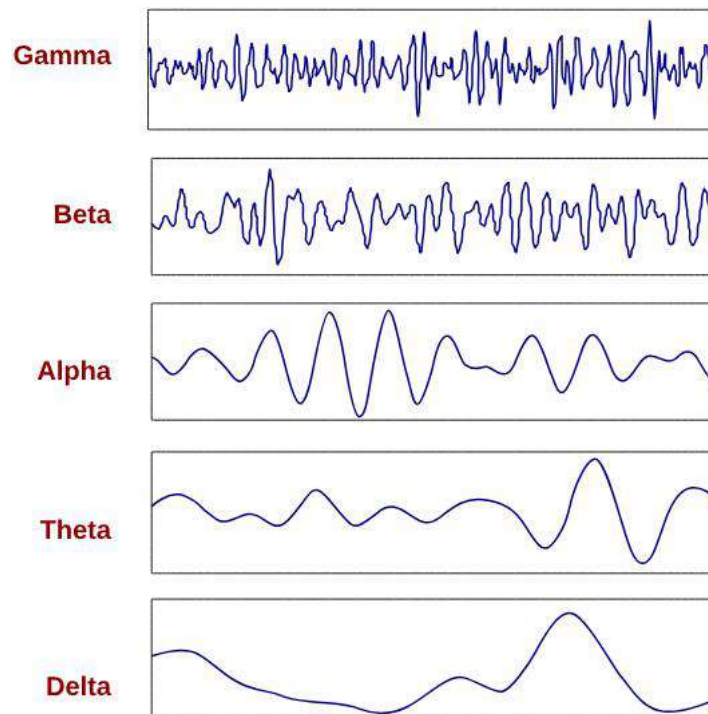
1.2 Електроенцефалографија

Електроенцефалографија (ЕЕГ) представља неинвазивну технику за мерење електричних активности мозга, која се широко користи у истраживањима, медицинској дијагностици и терапији [1]. ЕЕГ сигнал се састоји од варијација напона које се мере на површини главе и које су резултат сумираних активности пирамидалних неурона [2]. ЕЕГ сигнали се региструју коришћењем електрода које се стављају на површину главе испитаника, у складу са стандардним системом позиционирања електрода, као што је 10-20 систем [3].



Слика 1 Распоред ЕЕГ електрода 10-20 [4]

ЕЕГ сигнали се користе за детекцију и анализу шаблона електричних активности мозга, који се најчешће деле у четири основна фреквенцијска опсега: делта (од 0,5 Hz до 4 Hz), тета (од 4 Hz до 8 Hz), алфа (од 8 Hz до 13 Hz), бета (од 13 Hz до 30 Hz) и гама (од 30 Hz до 45 Hz) [5]. Сваки опсег се повезује с различитим когнитивним стањима и функцијама. На пример, алфа ритам се често повезује с релаксацијом и затвореним очима, док се бета ритам повезује с активним мишљењем и фокусирањем на задатак. На слици 2 су приказани примери ЕЕГ таласа различитих фреквенцијских опсега, илуструјући карактеристике сваког спектра.



Слика 2 Приказ основних таласа ЕЕГ сигнала [6]

Квантитативна електроенцефалографија (*QEEG*) је техника која користи математичке анализе за обраду и анализу ЕЕГ сигнала [7]. *QEEG* се може користити за идентификацију аномалија у електричној активности мозга, укључујући промене у амплитуди и фази сигнала, локализацију генератора сигнала и функционалне везе између различитих региона мозга [8]. *QEEG* може бити користан у дијагностици и мониторингу различитих неуролошких поремећаја, као што су епилепсија, шизофренија, деменција и аутизам [9]. ЕЕГ се користи у истраживању когнитивних процеса, као што су пажња, памћење, емоције, говор и вештине решавања проблема [10]. Такође, ЕЕГ се широко користи у изучавању мозга током спавања и у истраживањима која се баве утицајем спавања на когнитивне функције [11].

Употребом ЕЕГ мерне методе, могуће је снимити мождану активност у реалном времену, што омогућава истраживачима да студирају динамичне процесе који се дешавају у мозгу током различитих когнитивних и емоционалних стања. Ово може допринети бољем разумевању начина на који мозак функционише, као и развоју нових терапија и интервенција за лечење неуролошких и психијатријских поремећаја [12].

У скорије време ЕЕГ се користи у развоју мозак-рачунар система (BCI), који омогућавају директну комуникацију између мозга и спољног уређаја, као што су рачунари или роботи [13]. BCI системи су показали велик потенцијал у примени за рехабилитацију и асистивне технологије, поготово код особа с ограниченом моториком или комуникацијским способностима, као што су пацијенти с тетраплегијом, амиотрофичном латералном склерозом (АЛС) или „*locked-in*“ синдромом [14]. Без обзира на огроман потенцијал, примена BCI система још увек је у раној фази развоја, и постоји потреба за даљим истраживањима да би се побољшала њихова ефикасност и усвојивост.

Једна од области која се бави применом ЕЕГ-а представљају неурофидбек системи, они користе интерпретиране ЕЕГ сигнале да би особа могла да научи да контролише одређене аспекте своје мождане активности [15]. Неурофидбек тренинг се показао ефективним у лечењу неких психијатријских поремећаја, као што су атентивно-хиперактивни поремећај (*ADHD*) и анксиозност, као и у унапређењу когнитивних функција код здравих особа [16, 17].

У закључку, ЕЕГ је моћан алат који омогућава мерење и анализу мождане активности у реалном времену, са широким спектром примена у научним истраживањима, медицинској дијагностици и терапији, као и у развоју нових технологија попут BCI система и неурофидбек тренинга.

1.2.1 Значај ЕЕГ-а у когнитивним неуронаукама

Електроенцефалографија (ЕЕГ) је једна од најпопуларнијих техника мапирања мозга у когнитивним неуронаукама. Због своје високе временске резолуције, ЕЕГ омогућава истраживачима да прате динамичке процесе који се одвијају у мозгу током когнитивних задатака, као што су учење, памћење, пажња, и језик. Овај увид доприноси разумевању како се информације процесуирају и интегришу у мозгу [1].

1.2.2 Улога ЕЕГ-а у неурологији

У неурологији ЕЕГ је широко коришћен у дијагностици и лечењу различитих мозговних патологија, као што су епилепсија, спавање и постављање дијагноза когнитивних поремећаја [18]. ЕЕГ може помоћи у идентификацији фокуса епилептичких напада и надгледање протокола лечења, као и у разумевању утицаја нервних поремећаја на когнитивне функције.

1.2.3 ЕЕГ у психологији и психијатрији

ЕЕГ има важну улогу у психологији и психијатрији јер омогућава истраживање можданих механизма који леже у основи емоционалног, социјалног и когнитивног функционисања. У психијатрији, ЕЕГ може бити користан у дијагностици и лечењу поремећаја као што су шизофренија, биполарни поремећај, депресија и анксиозни поремећаји. ЕЕГ маркери, као што су асиметрије у активности мозга и специфични евоцирани потенцијали, могу помоћи у идентификацији биолошких основа ових поремећаја и мониторирању реакција на лечење [19].

У психологији ЕЕГ је користан алат за истраживање различитих аспеката когнитивног функционисања, укључујући памћење, пажњу, решавање проблема и емоционално процесуирање. ЕЕГ експерименти могу помоћи у разумевању како мозак интегрише и обрађује информације на различитим нивоима и како се ови процеси мењају у различитим ситуацијама и у складу с различитим стимулусима [20].

1.3 Когнитивне неуронауке

Когнитивне неуронауке представљају мултидисциплинарну научну област која се бави проучавањем можданих механизма који леже у основи когнитивних процеса као што су памћење, учење, пажња, језик, емоције, решавање проблема и одлучивање [18]. Циљ когнитивних неуронаука је да интегришу знање из различитих области, укључујући психологију, неуробиологију, компјутерске науке и филозофију, како би се добио бољи увид у начине на које мозак обрађује информације и ствара менталне репрезентације.

Технике које се користе у когнитивним неуронаукама варирају од класичних психолошких експеримената и компјутерских модела до савремених метода визуализације можданих активности као што су функционална магнетна резонанца (фМРИ), позитронска емисиона томографија (ПЕТ), транскранијална магнетна стимулација (ТМС), магнетоенцефалографија (МЕГ) и електроенцефалографија (ЕЕГ) [21].

ЕЕГ, као што је раније објашњено, омогућава праћење електричне активности мозга у реалном времену, што га чини погодном техником за истраживање временски променљивих когнитивних процеса. ЕЕГ се користи у когнитивним неуронаукама да би се истражила временска структура можданих активности које су повезане с различитим когнитивним функцијама, као и да би се идентификовали специфични обрасци мождане активности који су повезани с различитим когнитивним стањима [10]. Према томе, ЕЕГ се показао као користан алат у различитим когнитивним доменима, као што су визуелна и аудитивна пажња, радна меморија, спонтане мисли и медитација и декодирање менталних стања [22].

Од почетка когнитивних неуронаука ЕЕГ студије су континуирано доприносиле нашем разумевању функционалних односа између мозга и когнитивних процеса. На пример, „*event related potentials*“ (ERP) представљају временски локализоване промене у ЕЕГ сигналу које су повезане са специфичним когнитивним догађајима као што су визуелни подстицаји, аудитивни сигнали или моторни одговори [23]. Испитивањем ERP компонената истраживачи су успели да добију увид у временску динамику различитих когнитивних процеса, као и у њихове функционалне везе с различитим можданим регионима.

Један од кључних изазова у когнитивним неуронаукама је интеграција информација с различитих мерних уређаја. За то се често користе методе вишеповратне анализе, које омогућавају истовремено праћење функционалних и структурних мерења на различитим просторним и временским скалама [24]. Такође, развој нових алгоритама и метода за анализу ЕЕГ сигнала, као што су временско-фреквенцијске анализе, конективност и машинско учење, допринео је бољем разумевању сложених и динамичких можданих структура које су укључене у когнитивне функције [25].

Додатно, когнитивне неуронауке су имале важну улогу у развоју нових терапија и интервенција за различите психолошке и неуролошке поремећаје, као и у побољшању когнитивних перформанси код здравих особа. На пример, истраживања у области неуропластичности, како је описано у потпоглављу 1.4 Неуропластичност, показала су да је мозак способан да се адаптира и мења кроз време као одговор на околности и стимулацију, што је довело до развоја нових терапијских приступа, као што је неурофидбек [26].

У светлу свих ових напредака, когнитивне неуронауке су оствариле значајан успех у разумевању механизма који леже у основи когнитивних функција и унапређивању наших способности да предвидимо, дијагностикујемо и третирамо различите когнитивне поремећаје. Међутим, иако смо значајно проширили своје знање о мозгу и когнитивним процесима, много непознатих и изазова остаје.

У будућности, когнитивне неуронауке ће наставити да се развијају и да проширују свој домет у области мозга и когнитивних функција. Неки од потенцијалних праваца истраживања могли би да укључују:

1. Развој бољих модела можданих функција: Са све већим бројем доступних података и савремених алгоритама за анализу, истраживачи ће вероватно наставити да развијају детаљније и тачније моделе можданих функција који се користе за обраду информација.
2. Интеграција више техника мерења: Комбиновање података из различитих извора, као што су ЕЕГ, фМРИ, ТМС, и других техника, може да пружи додатне информације о томе како се мозак и когнитивни процеси мењају кроз време и у различитим контекстима.
3. Примена вештачке интелигенције и машинског учења: Вештачка интелигенција и машинско учење могу да помогну у анализи сложених података, идентификацији образаца активности и предвиђању когнитивних поремећаја, што може да допринесе развоју нових дијагностичких и терапијских приступа.
4. Унапређење терапијских метода: На основу дубљег разумевања механизма који стоје у основи когнитивних функција, истраживачи ће вероватно развити ефикасније терапије и интервенције за лечење когнитивних поремећаја и побољшање когнитивних способности здравих особа. Ово може укључивати примену техника неуромодулације, као што је неурофидбек или транскранијална директна струјна стимулација (tDCS), као и когнитивни тренинг и рехабилитацију.
5. Персонализовани приступи у когнитивним неуронаукама: Усвајањем персонализованих приступа у когнитивним неуронаукама, истраживачи ће можда успети да развију појединачне стратегије за унапређење когнитивних функција и лечење поремећаја на основу специфичних потреба и профила сваког појединца.
6. Етичка разматрања и социјални утицаји: Како когнитивне неуронауке напредују, етичка питања и социјални утицаји ових истраживања постаће све важнији. Потребно је размотрити како нова открића и технологије могу утицати на појединце и друштво као целину, укључујући питања приватности, дискриминације и приступа новим терапијама.

Когнитивне неуронауке представљају изузетно динамичну и обећавајућу научну област која ће наставити да расветљава тајне мозга и когнитивних функција. Даља истраживања и иновације у овој области допринеће развоју нових дијагностичких и терапијских приступа, као и побољшању квалитета живота за особе које се боре с когнитивним поремећајима или желе да оптимизују своје когнитивне способности. Овај напредак такође ће допринети развоју етичких и социјалних норми које ће подржати одговорну и одрживу примену нових технологија и знања у когнитивним неуронаукама.

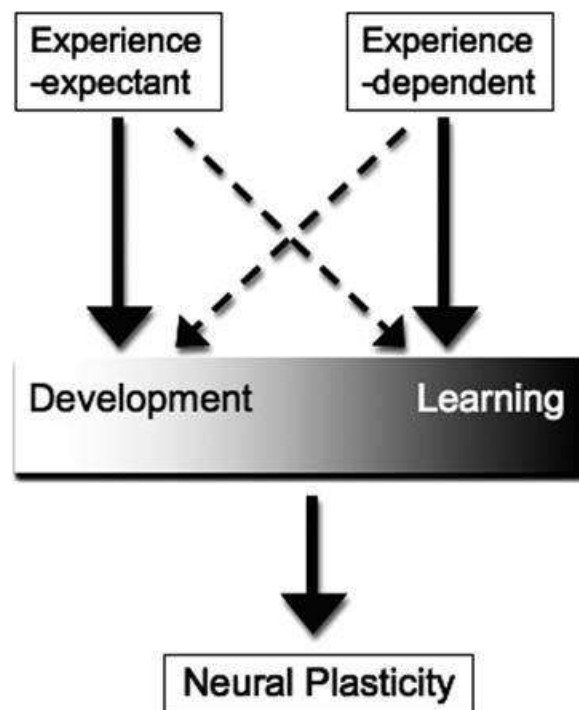
Осим тога, будућа истраживања могла би да истраже међусобну везу између когнитивних функција и других аспеката људске свести, као што су емоције, социјална кохезија, креативност и самосвест.

Такође, могло би да се истражи како се когнитивне функције мењају током животног доба и како се наши мозгови адаптирају на нове околности и изазове у свету који се брзо мења.

Кроз даља истраживања и сарадњу међу научницима из различитих дисциплина, когнитивне неуронауке наставиће да откривају нове увиде о мозгу и когнитивним функцијама, дајући нам веће могућности да разумемо и унапредимо људски потенцијал.

1.4 Неуропластичност

Неуропластичност је способност мозга да се мења и адаптира кроз цео живот као одговор на различите стимулусе, искуства и повреде. Термин се односи на процесе који воде до промена у структури и функцијама нервног система, укључујући промене у синапсама, неуралним путањама и укупној организацији мозга [27, 28].



Слика 3 Приказ релација између развоја и учења као и њихов утицај на неуралну пластичност [29]

1.4.1 Механизми неуропластичности

Три главна механизма који доприносе неуропластичности су:

1. Синаптичка пластичност: Овај процес укључује промене у јачини и ефикасности синапса, веза између неурона. Синаптичка пластичност може бити дуготрајна (што се дешава код дуготрајног потенцирања) или краткотрајна (што се дешава код депресије синапсе) [30, 31].
2. Структурна пластичност: Овај процес укључује промене у структури мозга, као што су раст и пренос неурона, стварање нових синапса, и измене у микроструктури мозга [32].
3. Функционална пластичност: Овај процес укључује промене у активности мозга и организацији можданих мрежа, које се могу створити као резултат искуства, обучавања или повреде мозга [33].

1.4.2 Фактори који утичу на неуропластичност

Неки од фактора који утичу на неуропластичност јесу:

1. Генетика: Генетски фактори могу утицати на способност мозга да се адаптира и мења [34].
2. Старење: Неуропластичност се смањује са старењем, али мозак још увек задржава одређени ниво пластичности који омогућава старијим особама да уче нове ствари и адаптирају се на нове околности [35].
3. Искуства и обучавање: Вежбање и обучавање могу побољшати неуропластичност и подстаћи раст нових синапси и неурона [36].
4. Повреде мозга: Повреде мозга могу довести до промена у функцији и структури мозга, али такође могу стимулисати процесе неуропластичности који могу помоћи у опоравку [37].

1.4.3 Улога неуропластичности у обучавању и терапији

Неуропластичност има велик потенцијал у области обучавања и терапије. На пример, истраживања показују да вежбе које побољшавају радну меморију могу подстаћи неуропластичност и довести до дуготрајних промена у когнитивним функцијама [38]. Такође, терапије које се заснивају на неуропластичности, као што су когнитивна рехабилитација и неурофидбек, могу помоћи особама с когнитивним поремећајима или повредама мозга да поврате или унапреде своје когнитивне способности [39].

1.4.4 Изазови и будући правци истраживања

Иако неуропластичност нуди многе могућности за обучавање и терапију, постоје и изазови. На пример, тренутно немамо довољно знања о оптималним методама стимулације неуропластичности у различитим контекстима и за различите индивидуе. Будућа истраживања морају да објасне како да се најбоље искористи неуропластичност у различитим ситуацијама и с различитим особама.

Неки од потенцијалних праваца будућих истраживања су:

1. Идентификација оптималних метода и техника за подстицање неуропластичности у различитим старосним групама, културним контекстима и специфичним стањима мозга [40].
2. Истраживање молекуларних и ћелијских механизма који стоје иза неуропластичности, како би се развило боље разумевање начина на који се мозак мења и адаптира [41].
3. Развој персонализованих приступа који се заснивају на мерењима неуропластичности код сваке особе, како би се осигурале најефикасније интервенције и третмани [42].
4. Истраживање улоге неуропластичности у развоју и одржавању когнитивних поремећаја, попут деменције или шизофреније, како би се развиле нове терапије и превентивне стратегије [43].
5. Примена нових технологија, као што су оптичке неуронауке и генетско инжењерство, за прецизно утицање на мождане структуре као и стимулацију неуропластичности [44].

Неуропластичност представља фундаментални аспект функционисања мозга и има велик потенцијал за примену у области обучавања и терапије. Међутим, много феномена још увек треба да се истражи о механизмима и факторима који утичу на неуропластичност, као и о оптималним

начинима за искоришћавање ових процеса у различитим контекстима. Будућа истраживања у овој области имају потенцијал да промене начин на који разумемо и лечимо когнитивне поремећаје, повреде мозга и старење.

1.4.5 Неуропластичност и машинско учење

Студије неуропластичности могу имати значајан утицај и на област машинског учења. Научници у овој области трагају за начинима да разумеју и реплицирају процесе који се дешавају у људском мозгу, укључујући промене у синапсама и неуралним путевима. Ово разумевање може допринети развоју вештачких неуралних мрежа које се могу мењати и адаптирати током времена, слично нашем мозгу [45].

Прилагођавајући знања о неуропластичности на машинско учење, научници могу успоставити мреже које се ефикасније адаптирају на нове задатке, уче из мање количине података и боље оптимизују своје перформансе током времена [46].

У будућности, спој изучавања неуропластичности и машинског учења може довести до значајног напретка у развоју вештачке интелигенције, нарочито у области адаптивног учења и роботике. Такође, може допринети развоју нових терапија и алата за унапређење и очување когнитивног здравља, као и примену вештачке интелигенције у медицини и неуронаукама [47].

1.4.6 Закључак

Неуропластичност је основни механизам који омогућава нашем мозгу да се адаптира на различите услове и изазове током читавог живота. У овом поглављу обрађена су основна својства, механизми и улога неуропластичности у обучавању, терапији и машинском учењу. Разумевање ових процеса може допринети развоју нових стратегија и интервенција за подршку когнитивног здравља и адаптације на старење, као и унапређењу техника машинског учења.

Будућа истраживања у овој области могу настојати да идентификују оптималне методе стимулације неуропластичности у различитим контекстима, развијају персонализоване терапије и истражују улогу неуропластичности у развоју и одржавању когнитивних поремећаја. Такође, истраживање симбиозе између неуропластичности и машинског учења може допринети значајном напретку у развоју вештачке интелигенције и примени вештачке интелигенције у медицини и неуронаукама.

1.5 Когнитивни феномени

Когнитивни феномени обухватају различите аспекте менталног процесуирања као што су памћење, пажња, размишљање, учење и вербално и визуелно разумевање. У овом поглављу су описани неки од главних когнитивних феномена, њихова улога у обради информација и њихове импликације на истраживање мозга и развој биомедицинских технологија.

1.5.1 Памћење

Памћење је кључна когнитивна функција која омогућава индивидуама да кодирају, чувају и преузимају информације [48]. Оно се обично дели на краткорочно (радно) и дугорочно памћење. Краткорочно памћење омогућава привремено чување информација, док дугорочно памћење служи за трајно чување информација [49]. Многа истраживања покушала су да идентификују неуралне

механизме који леже у основи различитих видова памћења, као и да разумеју како се ови процеси могу подстаћи или поправити у случају поремећаја памћења [50].

1.5.2 Пажња

Пажња је когнитивни процес који омогућава селективно фокусирање на одређене информације у окружењу, док се истовремено игноришу друге [51]. Она је од суштинског значаја за успешно обављање свакодневних задатака и учествовање у сложеним когнитивним активностима. Истраживања на пољу когнитивних неуронаука показала су да се пажња углавном обрађује у паријеталном и предњем цингулуму мозга, а истраживања на основу ЕЕГ-а помогла су у идентификацији осцилаторних механизма који су укључени у регулацију пажње [22, 52].

1.5.3 Размишљање

Размишљање представља сет менталних процеса који омогућавају индивидуама да формирају и манипулишу менталним репрезентацијама, да разматрају и решавају проблеме, да доносе одлуке и да планирају [53]. Размишљање је уско повезано с другим когнитивним функцијама, као што су радно памћење, пажња и инхибиција, и од суштинског је значаја за адаптивно понашање и менталну флексибилност [54]. Разни делови мозга, укључујући префронтални кортекс, учествују у процесу размишљања, а истраживања на основу ЕЕГ-а допринела су разумевању динамике функционисања мозга током ових когнитивних процеса [55].

1.5.4 Учење

Учење је когнитивни процес који омогућава индивидуама да стекну нова знања, вештине и способности кроз искуство [54]. Учење се обично ослања на пластичност мозга, што омогућава адаптивне промене у везама између неурона и мрежа које дефинишу когнитивне функције [56]. Различите врсте учења, као што су усвојено учење, имплицитно учење и експлицитно учење, повезане су с различитим мрежама у мозгу и могу се испитивати уз помоћ техника као што је ЕЕГ [57, 58]. Истраживање учења и механизма који стоје иза њега има важне импликације за развој интервенција и терапија које могу побољшати когнитивне способности у различитим контекстима, као што су образовање, рад и старење [59].

1.5.5 Менталне слике

Менталне слике представљају процес стварања менталних репрезентација које нису директно доступне сензорним системима [60]. Овај когнитивни феномен омогућава индивидуама да симулирају и предвиђају окружење и догађаје на основу претходног искуства, што је од суштинског значаја за планирање, решавање проблема и креативност [61]. Менталне слике се најчешће повезују с визуелним доменом, али могу се односити и на друге сензорне модалитете, као што су слух, додир и мирис [62]. Неурални механизми који подржавају менталне слике могу се истражити уз помоћ метода као што је ЕЕГ, који пружа увид у временске и просторне активности мозга у току ових когнитивних процеса [63].

1.5.6 Визуализација

Визуализација је способност да се створе, манипулишу и интерпретирају менталне слике које представљају визуелне информације. Она је повезана с различитим когнитивним функцијама, као што су памћење, пажња, учење и решавање проблема, и углавном се обрађује у окулипиталном и темпоралном кортексу мозга [64]. Истраживања на основу ЕЕГ-а и других техника визуализације

допринела су разумевању механизма који стоје иза визуализације, укључујући ролу осцилација и синхронизације у визуелној обради и интеграцији информација [65, 66].

1.5.7 Концентрација

Концентрација представља способност одржавања и усмеравања фокуса и менталних ресурса на одређени задатак или информацију. Она је уско повезана с пажњом и контролом когнитивних процеса, а истраживања су показала да се концентрација углавном обрађује у префронталном кортексу и другим областима кортекса које учествују у извршној функцији [67]. Технике као што је ЕЕГ могу да пруже увид у динамику можданих активности током задатака који захтевају концентрацију, као и да идентификују потенцијалне биомаркере пада концентрације у различитим патолошким стањима, као што су *ADHD* и деменција [68, 69].

1.5.7.1 Аспекти менталне концентрације

На основу креираног плана истраживања и постигнутих резултата, дефинисани су следећи аспекти менталне концентрације:

1. Одрживост менталне концентрације: Означава способност испитаника да одржи визуелну менталну слику најдужи могући временски период изнад одређеног нивоа интензитета.
2. Агилност менталне концентрације: Описује способност испитаника да прави промене у интензитету визуализације менталне слике у што краћем временском периоду.
3. Контрола менталне концентрације: Односи се на способност испитаника да одржи интензитет визуализације менталне слике у дефинисаним толерисаним границама у односу на дефинисани ниво интензитета.
4. Временска издржљивост у контроли концентрације: Означава временски период за који испитаник може вољно да управља системом пре него што се његове способности одржавања менталних слика знатно смање или уследи појава замора или бола.
5. Способност паралелног обављања задатака: Описује способност испитаника да истовремено обавља више вољних моторних и когнитивних функција, укључујући и управљање системом.
6. Отпорност на спољашње дистракције: Означава способност испитаника да игнорише спољашње шуме који би могли да утичу на његово обављање когнитивних функција.

1.5.8 Закључак

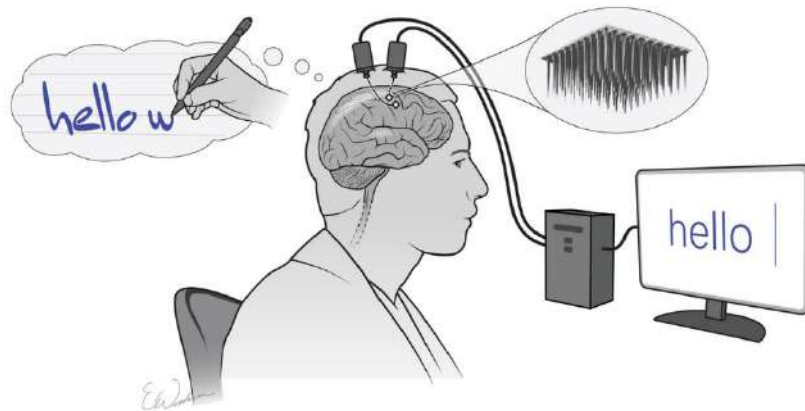
У овом поглављу обрађени су различити когнитивни феномени који су од суштинског значаја за свакодневно функционисање индивидуа. Разумевање ових процеса и механизма који их подржавају има важне импликације за истраживање мозга, развој биомедицинских технологија и примену ових сазнања у клиничким и образовним контекстима. Како настављамо да истражујемо ова поља истраживања, можемо очекивати да ће се наша разумевања ових феномена и њихова веза са структуром и функцијом мозга додатно развити. Ово ће отворити нове могућности за креирање иновативних терапија и интервенција које могу побољшати когнитивне способности код шире популације корисника.

Когнитивне неуронауке представљају мултидисциплинарни приступ истраживању мозга који се фокусира на когнитивне процесе, њихову молекуларну и целуларну основу, а такође и на функционалне и структурне аспекте мозга који стоје иза тих процеса.

Како когнитивне неуронауке настављају да напредују и развијају се, очекује се да ће разумевање мозга и когнитивних процеса стално расти и даље интегрисати нова сазнања из различитих дисциплина. Ова интеграција знања омогућиће нам да развијемо боље методе за идентификацију и третман когнитивних поремећаја, као и да побољшамо квалитет живота и функционалне способности људи широм света.

1.6 Мозак-рачунар системи (BCI)

Мозак-рачунар системи (BCI) представљају склоп технологија које омогућавају директну комуникацију између људског мозга и екстерног рачунарског система. Ови системи се користе у различите сврхе, укључујући истраживања, медицинске апликације и контролу рачунара или других уређаја користећи когнитивна стања или когнитивне активности корисника. На слици 4 је приказан упрошћен пример који илуструје начин рада BCI система.



Слика 4 Пример мозак-рачунар система [70]

1.6.1 Технологије за снимање мозга

BCI се ослањају на технологије за снимање мозга које се користе за мерење активности мозга. Неке од најчешћих метода за снимање мозга у BCI-у укључују електроенцефалографију (ЕЕГ), магнетоенцефалографију (МЕГ) и функционалну магнетну резонанцу (*fMRI*) [12]. Свака од ових метода има своје предности и мане у погледу резолуције, просторно-временских односа и инвазивности.

1.6.2 Обрада и декодирање сигнала

Добијени сигнали из мозга морају се обработити и декодирати да би се идентификовали одговарајући когнитивни процеси или намере корисника. Ово подразумева коришћење различитих метода обраде сигнала, статистичких анализа и машинског учења, као што су филтрирање, просторно-временска анализа, класификација и регресија [71, 72].

1.6.3 Апликације BCI-ја

BCI има широк спектар апликација, укључујући истраживања о когнитивним процесима, неурорехабилитацију особа с физичким ограничењима и контролу асистивних технологија, као и гејминг и виртуелну реалност [73, 74].

1.6.3.1 Медицинске апликације

У медицини BCI се користе за рехабилитацију особа које имају тешкоће с контролом мишића или комуникацијом, као што су особе с тетраплегијом, амиотрофичном латералном склерозом (АЛС) или после можданог удара [75, 76]. BCI се такође користи за контролу протеза, роботских егзоскелета и других асистивних уређаја који омогућавају покрет и самосталност особама с овим стањима [77, 78].

1.6.3.2 Истраживачке апликације

У научним истраживањима BCI се користи за проучавање когнитивних процеса, учења и памећења, емоција и социјалног разумевања [79, 80]. Ово омогућава истраживачима да разумеју како мозак кодира, обрађује и интерпретира информације, што може допринети развоју нових терапија и интервенција за различита когнитивна и неуролошка стања.

1.6.3.3 Гејминг и виртуелна реалност

BCI системи се такође користе у гејминг индустрији и виртуелној реалности, где корисници могу да контролишу игре и виртуелна окружења користећи своја когнитивна стања и когнитивне функције [81, 82]. Ово отвара нове могућности за интерактивност и имерсивност корисника, као и потенцијалне апликације у образовању, тренингу и терапији.

1.6.4 Изазови и будући развој

Иако BCI системи представљају велик потенцијал за различите апликације, они такође представљају неколико изазова. Један од главних изазова је слаб и шумовит сигнал, који се често добија из неинвазивних техника снимања мозга, као што је ЕЕГ [83]. Боље технике за ублажавање шума и екстракцију релевантних карактеристика сигнала могле би помоћи у побољшању тачности BCI-а.

Други изазов је индивидуална варијабилност у мозгу и когнитивним процесима [84]. Ово може утицати на то колико добро BCI може да се прилагоди између различитих особа и ситуација. Технике за адаптивно учење и пренос знања могу помоћи у превазилажењу овог изазова.

Коначно, етички изазови у вези с приватношћу, сигурношћу и злоупотребом BCI технологије морају бити узети у обзир [85]. Ово подразумева стварање правила, протокола и стандарда који штите права и интересе корисника.

У будућности BCI системи могу створити нове апликације и иновације у различитим областима, као што су медицина, образовање, гејминг и виртуелна реалност. Такође, побољшање у технологији снимања мозга, обради сигнала и алгоритмима машинског учења може допринети бољем разумевању и контроли когнитивних процеса и менталних стања.

1.7 Мапирање мозга

Мапирање мозга је процес који омогућава визуализацију активности различитих области мозга. ЕЕГ системи играју кључну улогу у овом процесу, дајући научницима и здравственим стручњацима увид у функционисање мозга на електричном нивоу.

Комбиновањем података с различитих канала ЕЕГ сигнала можемо формирати целокупну "мапу" активности мозга.

Добијене мапе се представљају као графички прикази, где различите боје или интензитети боја представљају различите нивое активности. На пример, боја може варирати од хладних (ниска активност) до топлих боја (висока активност), што омогућава брзо и једноставно читање активности мозга.

Мапирање мозга помоћу ЕЕГ система је изузетно вредно средство за психологе, психијатре и неурологе. То им омогућава да боље разумеју како различите области мозга комуницирају и раде заједно, што може помоћи у дијагнози и лечењу многих болести и стања.

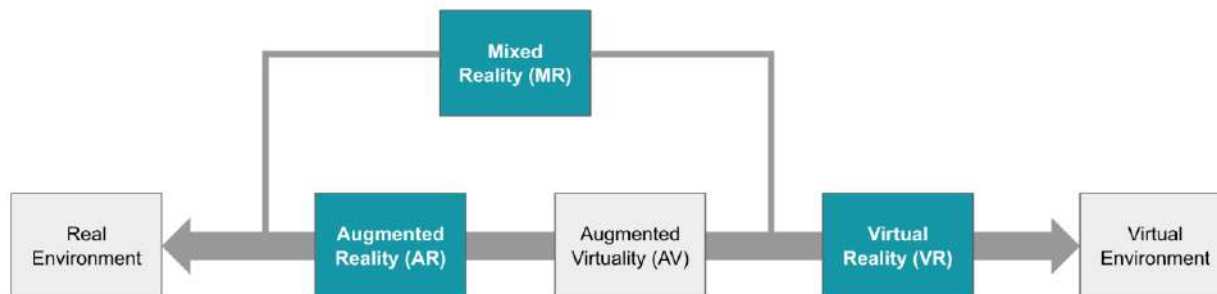
При примени ЕЕГ мапирања мозга могу се уочити промене у активности мозга које могу бити повезане с одређеним когнитивним стањима или болестима, као што су депресија, анксиозност, шизофренија, епилепсија, и многе друге. Ово може омогућити рану дијагнозу и интервенцију, што може значајно помоћи у процесу лечења.

1.8 Продужена реалност

Продужена реалност (*Extended Reality* - XR) је широк појам који обухвата све форме виртуелне и аугментоване реалности. Он обједињава виртуелну реалност (VR), аугментовану реалност (AR) и мешовиту реалност (MR), стварајући интерактивни, имерсивни дигитални свет [85]. На слици 5 су приказани сви системи који су обухваћени продуженом реалношћу, као и њихове међусобне релације од стварног окружења до виртуелног окружења.

Виртуелна реалност (VR) омогућава корисницима да потпуно уђу у вештачко, виртуелно окружење. Ово се често постиже помоћу VR система, као што је *Oculus Rift* или *HTC Vive*. Корисници могу интерактивно учествовати у овом окружењу које представља рад у потпуно симулираном окружењу [86]. Аугментована реалност (AR) је технологија која проширује нашу физичку реалност усађивањем виртуелних објеката у реални свет, најчешће преко камере мобилног уређаја или AR наочара [87]. Мешовита реалност (MR) комбинује елементе VR и AR омогућавајући корисницима да виде и интерактивно учествују с виртуелним објектима у реалном свету. Ово је постало могуће помоћу уређаја као што је *Microsoft HoloLens* [88].

Продужена реалност има многобројне примене. У области образовања она може обезбедити интерактивне и образовне сценарије који подржавају учење [89]. У здравству VR се користи за обуку хирурга и лечење посттрауматског стресног поремећаја [90]. AR се све више користи у маркетингу и рекламама омогућавајући корисницима да "пробају" производе пре него што их купе. У области развоја софтвера и игара XR пружа нове и узбудљиве могућности за креирање имерзивних и интерактивних искустава. Због тога продужена реалност представља знатан напредак у интеракцији између човека и компјутера, стварајући нов начин на који можемо да искусимо и интерактујемо с дигиталним светом.



Слика 5 Графички приказ тока информација у оквиру продужене реалности [91]

1.9 Системи машинског учења

Машинско учење је једна од подобласти вештачке интелигенције, која се фокусира на развој алгоритама и модела способних да уче и доносе одлуке на основу података [92]. У контексту когнитивних наука и неурофидбек система, машинско учење игра кључну улогу у анализи и интерпретацији комплексних електрофизиолошких података, као што су ЕЕГ сигнали [93]. Модел машинског учења је математичка структура која се користи за предвиђање или доношење одлука на основу улазних података. Скуп података који се користи за обуку и тестирање модела се обично дели на три дела: обука, валидација и тестирање. Функција губитка је мера која процењује колико добро модел предвиђа жељене излазе и циљ обуке је минимизовати ову функцију[92].

Постоје три главна облика обуке у машинском учењу:

- Надзирано учење (*Supervised Learning*): У овом облику обуке, модел учи из унапред означених података, где сваки улаз има одговарајући излаз. Циљ је научити однос између улазних и излазних података како би се правилно предвидели излази за нове улазе. Надзирано учење се користи код:
 - Класификација - задатак где је циљна вредност категоријска, што значи да модел предвиђа којој од неколико дискретних категорија припадају улазни подаци [92].
 - Регресија - задатак је да се на основу улазних података изврши предвиђање циљне вредности која се налази у опсегу [92].
- Ненадзирано учење (*Unsupervised Learning*): У овом облику обуке, модел учи из необележених података, односно без унапред дефинисаних излаза. Циљ је открити скривене обрасце или структуре у подацима [92]. Основни типови задатака у ненадзираном учењу су:
 - Кластеризација: Процес груписања сличних података на основу њихових обележја, тако да подаци унутар истог кластера буду што сличнији, а подаци из различитих кластера што различитији [93].
 - Смањење димензионалности: Техника која има за циљ поједностављење података и уклањање шума кроз пројектовање података у простор са мање димензија, уз задржавање што више важних информација [93].
- Подстицајно учење (*Reinforcement Learning*): У овом облику обуке, модел учи путем интеракције са окружењем, добијајући награде или казне за своје акције. Циљ је максимизовати кумулативну награду кроз оптималне одлуке. Појачано учење се често

користи у контекстима као што су учење игре шаха, управљање роботима или развој паметних система за доношење одлука [93].

Обука модела машинског учења почиње постављањем иницијалних параметара модела и дефинисањем функције губитка и оптимизационог алгорита. Током обуке, модел се итеративно ажурира на основу података из тренинг скупа. Основни циљ је минимизовати функцију губитка, која мери разлику између предвиђених и стварних вредности. Овај процес се извршава кроз више епоха, што представља један пролаз кроз цео тренинг скуп података [92].

Епоха представља један циклус проласка кроз цео тренинг скуп података. Током једне епохе, алгоритам ажурира тежинске коефицијенте модела на основу функције губитка израчунате за сваки пример у скупу података. Обука се извршава итеративно кроз више епоха, при чему се током сваке епохе тежине модела ажурирају на основу грешака направљених на тренинг подацима. Процес се наставља све док се не испуне одређени критеријуми за прекид обуке, као што су достизање максималног броја епоха, стабилизација функције губитка или постизање задовољавајућих перформанси на валидационом скупу података [93].

1.9.1 Системи за класификацију и регресију

Неуралне мреже су инспирисане биолошким нервним системом и представљају један од најефикаснијих и најпопуларнијих модела машинског учења. Основне јединице неуралне мреже су неурони, који су организовани у слојеве. Слојеви мреже се деле на улазни слој, скривени слојеви и излазни слој. Сваки неурон у једном слоју је повезан са неуронима у наредном слоју преко тежинских коефицијената. Током обуке, мрежа учи да прилагођава ове тежинске коефицијенте како би што боље предвиђала излазне вредности [93].

Конволутивне неуралне мреже (*CNN*) представљају посебан тип неуралних мрежа дизајниран за обраду структурираних података као што су слике и аудио сигнали. *CNN* мреже користе конволутивне слојеве који примењују филтере на улазне податке како би изоловали важне карактеристике. Ове мреже су посебно ефикасне за задатке као што су класификација слика, препознавање образаца и обрада природног језика [94].

Једнодимензионалне конволутивне неуралне мреже (*1D CNN*) су специјална врста *CNN*-а која се користи за обраду секвенцијалних података. Ове мреже врше једнодимензионалне конволуције дуж временске или секвенцијалне димензије омогућавајући моделовање локалних обележја унутар секвенци [94].

Линеарна регресија је један од основних модела машинског учења који се користи за предвиђање континуираних вредности. Циљ линеарне регресије је да пронађе линеарну везу између улазних карактеристика и циљне вредности. Овај модел претпоставља да је циљна вредност линеарна комбинација улазних карактеристика, уз додатак случајног шума [95].

Random Forest је метода у машинског учења која се користи за класификацију, регресију и друге задатке који укључују анализу података. Овај алгоритам комбинује више стабала одлучивања како би побољшао тачност предвиђања и смањио ризик од претренирања (*overfitting*). Основна идеја *Random Forest* алгоритма је да уместо једног стабла одлучивања користи више стабала, од којих свако узима у обзир различите узорке података и подскупове карактеристика. Током обуке *Random Forest* модела, свака појединачна стабла се граде коришћењем метода случајног узорковања

(*bootstrap sampling*), што значи да се за свако стабло користи различит подскуп података из оригиналног скупа. Додатно, при сваком раздвајању унутар стабла, алгоритам бира најбољи разделник из случајног подскопа карактеристика, чиме се обезбеђује разноликост међу стаблима [96].

1.9.2 Валидациони параметри

Код вршења класификације у машинском учењу, могуће је класификационе исходе поделити на четири основна случаја: тачно позитивни (*True Positive - TP*), тачно негативни (*True Negative - TN*), лажно позитивни (*False Positive - FP*) и лажно негативни (*False Negative - FN*). Ови исходи су кључни за разумевање перформанси модела и за израчунавање валидационих метрика, које су наведене у [97] и описане у изразима од (1) до (5). У процени перформанси модела класификације, примењује се више различитих валидационих параметара. Први и основни параметар је тачност, која представља проценат исправно класификованих случајева у односу на укупан број случајева. Тачност се израчунава као однос броја исправно класификованих и укупног броја случајева:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad .(1)$$

Прецизност је параметар који мери проценат исправно класификованих позитивних случајева у односу на укупан број случајева класификованих као позитивни. Прецизност је важна у ситуацијама где је битно смањити број лажно позитивних случајева:

$$Precision = \frac{TP}{TP + FP} \quad .(2)$$

Одзив (*Recall*), познат и као сензитивност, представља проценат исправно класификованих позитивних примера у односу на укупан број стварно позитивних примера. Одзив је од значаја у контекстима где је критично идентификовати што више позитивних примера:

$$Recall = \frac{TP}{TP + FN} \quad .(3)$$

F1 скор је метрика која комбинује прецизност (*Precision*) и одзив (*Recall*) у једну меру перформанси модела класификације. F1 скор је посебно користан у случајевима када је важно одржати равнотежу између прецизности и одзива, нарочито у ситуацијама са неуравнотеженим класама:

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad .(4)$$

Receiver Operating Characteristic (ROC) крива је графички приказ перформанси бинарних класификационих модела, који приказује однос између стопе стварних позитива и стопе лажних позитива за различите граничне вредности (*thresholds*). Различите граничне вредности одређују прагове на основу којих се класификују позитивни и негативни случајеви. Граничне вредности представљају праг који одређује на којој тачки вероватноће модел означава улазни податак као позитиван или негативан. Промена граничне вредности утиче на број истински позитивних и лажно

позитивних предвиђања. Виша гранична вредност може смањити стопу лажно позитивних (FPR), али и стопу стварних позитива (TPR), и обрнуто [97].

Матјузов корелациони коефицијент (MCC) је још један важан параметар који узима у обзир тачност класификације и односе између TP , TN , FP и FN . MCC вредност се креће од -1 до 1, где 1 означава савршену класификацију, 0 случајну класификацију, а -1 инвертовану класификацију:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad .(5)$$

Код процене перформанси модела регресије, примењују се различити валидациони параметри који се рачунају на основу стварне вредности излаза (y_i), предвиђених вредности (\hat{y}_i) и просечне вредности излазне променљиве (\bar{y}), а изрази од (6) до (10) су описани у [95]. Први и основни параметар је R^2 , који мери колико добро модел објашњава варијабилност излазних података у односу на просечну вредност. R^2 вредност близу 1 указује на то да модел добро објашњава податке, док вредност близу 0 указује на то да модел не објашњава варијабилност у подацима:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (6)$$

Средња квадратна грешка (MSE) представља просечну квадратну грешку између стварних и предвиђених вредности. Квадратни корен средње квадратне грешке ($RMSE$) је мерило грешке у истим јединицама као и излазне вредности, пружајући интуитивнију меру грешке:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (7)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (8)$$

Средња апсолутна грешка (MAE) представља просечну апсолутну грешку између стварних и предвиђених вредности, пружајући директну меру просечне грешке модела. Средња апсолутна процентуална грешка ($MAPE$) мери просечну апсолутну процентуалну грешку, пружајући меру грешке у процентима:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

$$MAPE = 100 \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (10)$$

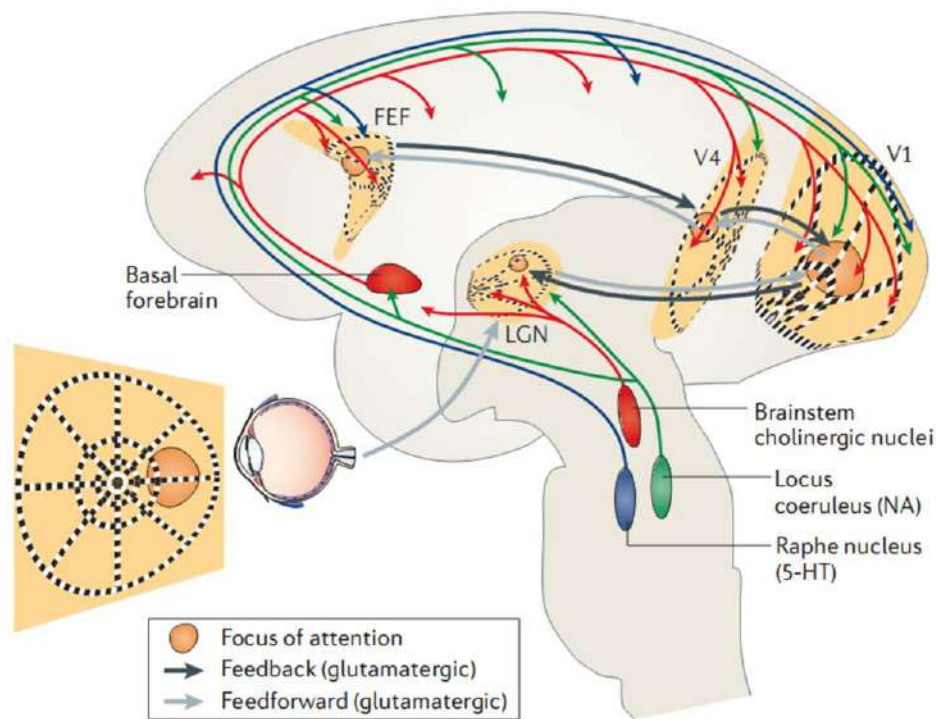
2 Актуелно стање у области

2.1 Преглед владајућих ставова и схватања

2.1.1 Неурофидбек системи

Неурофидбек (NFB) системи, такође познати као EEG биофидбек системи, представљају методу која омогућује особама да контролишу своје когнитивне и неуролошке процесе путем директног повратног информацијског тока о активности мозга [98]. Ова технологија је стекла популарност у последњих неколико деценија, нарочито у области психотерапије, тренинга памћења и концентрације, као и у лечењу различитих психолошких поремећаја. Слика 6 приказује основну методологију неурофидбек система.

Од својих раних примена у 70-им годинама 20. века, NFB системи су значајно напредовали [99]. Актуелни радови попут оних које су написали Ros и сарадници (2014) указују на то да неурофидбек може резултирати пластичним променама у нервним путевима [100].



Слика 6 Методологија неурофидбек система [100]

Терапије које користе неурофидбек показале су обећавајуће резултате у лечењу различитих стања, укључујући инсомнију [101], па чак и појединих поремећаја пажње као што је *ADHD*.

Научна заједница је препознала потенцијалне предности и примену неурофидбека не само као терапију већ и као средство за истраживање когнитивних функција. Sitaram и сарадници (2017) у свом прегледном раду објаснили су научне основе и клиничке примене неурофидбек система, истичући њихов потенцијал у развоју можданих функција [102].

Осим као терапијски алат, неурофидбек системи постају све популарнији у области побољшања когнитивних вештина код здравих особа. Ово укључује области попут побољшања концентрације, креативности и опуштања [99].

Неурофидбек системи представљају важну и иновативну област истраживања која обећава у погледу побољшања когнитивних функција, лечења различитих психолошких и нервних поремећаја, као и пружања увида у рад и функционисање мозга [103].

2.1.2 Развој и испитивање неуропластичности

Неуропластичност, позната и као пластичност мозга, означава способност мозга да се мења током читавог живота као одговор на нова искуства, учење и различите стимулусе. Ова способност мозга да реорганизује своје структурне и функционалне везе основа је за опоравак после повреда и за усвајање нових вештина [28].

У последњих неколико деценија истраживања у области неуропластичности постала су све прецизнија због примене савремених техника визуализације неуралних активности. Радови попут оних које је објавио *Draganski* и колеге (2004) демонстрирали су директне промене у структури мозга као резултат учења нове вештине, као што је жонглирање [36].

Занимљива област истраживања у контексту неуропластичности је и улога сна. *Diekelmann* и сарадници (2014) у свом раду указују на то да сан може активирати и поткрепити неуропластичне процесе, омогућавајући консолидацију новонасталих меморијских трагова [104].

Такође, истраживања показују да активности као што су медитација и неурофидбек могу индуцирати неуропластичне промене. *Tang* и колеге (2015) објавили су да медитација не само да може да подстакне неуропластичност већ и да омогућује бољу саморегулацију и когнитивне способности [105].

Неуропластичност игра кључну улогу у разумевању нашег потенцијала за учење и опоравак, а савремена истраживања непрестано откривају нове аспекте и могућности које ова способност мозга пружа.

2.1.3 Системи вештачке интелигенције и њихова примена над ЕЕГ сигналина

Системи вештачке интелигенције (ВИ) играју изузетно важну улогу у области анализе и обраде електроенцефалограмских (ЕЕГ) сигнала. ВИ може ефикасно препознавати обрасце и извршавати сложене задатке анализе који су тежи за човека.

Најновије истраживање у овој области фокусира се на примену дубоких неуронских мрежа (*Deep Neural Networks – DNNs*) у анализи ЕЕГ сигнала. *DNNs* су постали основни алат за класификацију и регресију ЕЕГ сигнала. У раду који су објавили *Schirrmeyer* и сарадници (2017) користећи конволуционе дубоке неуронске мреже, постигли су високу прецизност при класификацији различитих когнитивних задатака из ЕЕГ сигнала [106].

Поред класификације, ВИ се такође користе за регресију ЕЕГ сигнала, што омогућава прецизно прогнозирање когнитивних стања и одговарајућих активности. Велике дубоке неуралне мреже у радовима као што је рад *Tabar* и сарадници (2016) постигле су изузетне резултате у прогнозирању ЕЕГ сигнала за моторне задатке [107].

Примена ВИ над ЕЕГ сигналама је велика и обухвата широку палету области, укључујући и медицину, где се користе за дијагностику епилепсије, аутизма и других неуролошких поремећаја. Vandecasteele и колеге (2017) су развили систем који користи ВИ за детекцију епилептичних анфалса из ЕЕГ сигнала с високом тачношћу [108].

Ови примери истраживања и примене сведоче о значају система вештачке интелигенције у анализи и примени ЕЕГ сигнала, као и о брзом напретку ове области у научном и клиничком контексту.

2.2 Образложење о потребама научног истраживања

Методе мерења електричних активности мозга су разноврсне и, између осталог, имају своју примену у контексту истраживања квантитативне електроенцефалографије, неурофидбека и неуропластичности. Постоји простор за даљи развој наведених метода и одговарајућих система за обуку и тестирање испитаника, истовремено обезбеђујући мерење, обраду, анализу, класификацију и регресију сигнала.

Претходно наведена истраживања имају истакнут значај у области развоја неурофидбек система, неуропластичности, система вештачке интелигенције као и њихове примене. Основна идеја за даљи развој наведених метода је да то омогући унапређење неурофидбек *Brain-Computer Interface (BCI)* система, који се издвајају тако да комбинују виртуелне и физичке излазе како би стимулисао и обучио мозак испитаника.

Потенцијални допринос овог истраживања је развој платформе за обуку и коришћење флексибилног и адаптивног BCI система. Оваква платформа би укључивала интеграцију *Emotiv* ЕЕГ уређаја, *Python* програмирање, *Arduino* развојно окружење и виртуелну реалност.

2.3 Циљ истраживања и хипотеза

Циљ овог истраживања је унапређење метода за мерење (укључујући анализу и интерпретацију) можданих електричних сигнала добијених помоћу електроенцефалографских система, а у контексту истраживања квантитативне електроенцефалографије, неурофидбека и неуропластичности. Постизање овог циља може да допринесе даљем развоју и унапређењу биомедицинских технологија у области когнитивних неуронаука, као и разумевању сложених процеса који се одвијају у мозгу током различитих когнитивних активности и стања.

Очекивани резултат (хипотеза) овог истраживања је да се развије неурофидбек *BCI* систем који користи виртуелне и физичке излазе за подстицање неуропластичности код испитаника. Циљ система је да се стимулише и обучи мозак да промени своје радне обрасце, чиме се побољшавају когнитивне способности и опште психичко стање. Као додатни аспект овог истраживања развијена је симулација за неурофидбек користећи програмски језик *C#*, *Unity* развојно окружење и *Emotiv Cortex* пакет.

3 Методологија

Поглавље обухвата детаљан преглед коришћене опреме, осмишљених експеримената, архитектуре система, места истраживања и спроведених експеримената. У истраживању су примењени савремени хибридни ЕЕГ уређаји и лиценцирани психолошки тестови, како би се прецизно анализирале когнитивне активности испитаника. Посебна пажња је посвећена развоју неурофидбек симулационог система, који омогућава синхронизовано праћење когнитивних параметара и интеракција унутар симулације.

Важно је напоменути да су експерименти над испитаницима, описани у овом поглављу, спроведени током истраживања за мастер рад под називом "Развој мерно-информационог система за истраживање когнитивних појава заснованог на мерењу ЕЕГ сигнала" из 2018. године, и нису део овог докторског истраживања. Иако нису део овог истраживања, добијени резултати и закључци се користе као основа за даљи развој система и додатна проширења.

3.1 Опрема

3.1.1 *Emotiv Insight* и *Emotiv Epos*: Алати за мерење електричне активности мозга

Emotiv Insight и *Emotiv Epos* представљају уређаје у области ЕЕГ мерења, које је произвео Emotiv Systems. Они су намењени за мерење и анализу мождане активности и као такви налазе широку примену у различитим областима: истраживању, образовању, развоју видео-игара и многим другим.

3.1.1.1 *Emotiv Insight*

Emotiv Insight, приказан на слици 7, је бежични ЕЕГ уређај који пружа петоканално мерење ЕЕГ сигнала. Сензори су позиционирани у складу с међународним 10-20 системом на позицијама AF3, AF4, T7, T8 и Pz. Уређај има два референтна сензора на левој мастоидној кости, а материјал сензора је хидрофилна полусува полимерна гума. Фреквенцијски опсег ЕЕГ сигнала је од 0,5 Hz до 43 Hz с дигиталним notch филтером на 50 Hz и 60 Hz.

Emotiv Insight користи *Bluetooth Low Energy* за бежичну комуникацију, а антена подржава *Bluetooth 5*. Семпловање се врши 128 пута у секунди по каналу, с резолуцијом од 16 бита, при чему је $LSB=0,1275 \mu V$. Уређај обезбеђује динамички опсег од $8.400 \mu V(pp)$, а све вредности су АС спрегнуте. Батерија капацитета 480 mAh омогућује рад до 20 сати с једним пуњењем.

У склопу уређаја налази се и систем, чип ICM-20948 с фреквенцијом семпловања од 32 Hz и резолуцијом од 16 бита, за мерење кретања, који се састоји из троосног акцелерометра, жироскопа и магнетометра. Мерни опсег акцелерометра је $\pm 8g$, жироскопа је $\pm 2000 dps$, који су конвертовани у четири нормализована кватерниона, магнетометар има мерни опсег $\pm 12 gauss$ -а [109, 110].

Emotiv Insight се користи за детекцију различитих менталних команди и фацијалних експресија, као што су трептање, намигивање, изненађење, мрштење, осмех и стисак зуба [111].



Слика 7 Emotiv Insight уређај [110]

3.1.1.2 Emotiv Epos

Emotiv Epos, приказан на слици 8, је уређај који пружа 14 канала мерења и који обезбеђује детаљан приказ мождане активности. Сензори су позиционирани у складу с међународним 10-20 системом на позицијама AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 и AF4. ЕЕГ електроде су направљене од хидрофилних карбонских сунђера, натопљених у физиолошком раствору. Фреквенцијски опсег ЕЕГ сигнала је од 0,16 Hz до 43 Hz с дигиталним notch филтером на 50 Hz и 60 Hz.

Уређај пружа могућност комуникације путем USB пријемника или користећи *Bluetooth Low Energy* комуникацију. Семпловање унутар уређаја се врши на фреквенцији од 2.048 Hz које се затим спушта на 128 Hz или 256 Hz, у зависности од избора корисника уз резолуцију од 16 бита, при чему је $LSB=0,1275 \mu V$. Уређај обезбеђује динамички опсег од $8.400 \mu V(pp)$ и све вредности су AC спрегнуте.

Као и код *Emotiv Insight* уређаја, *Emotiv Epos* у себи има исти систем за мерење помераја, ротације и густине магнетног флукса помоћу истог чипа. Батерија уређаја има капацитет 595 mAh и омогућава рад до 12 сати користећи USB пријемник или до шест сати користећи *Bluetooth Low Energy*. Тежина уређаја је 170 грама и димензије су 9 cm x 15 cm x 15 cm [112, 113].

Овај уређај се такође користи за детекцију различитих менталних команди и фацијалних експресија, али поред оних које има и *Insight*, додатно укључује и детекцију смеха и полуосмеха с обе стране [114].



Слика 8 Emotiv EPOC уређај [113]

3.1.1.3 Електроде уређаја Emotiv Insight и Emotiv EPOC

Електроде које се користе у *Emotiv Insight* и *Emotiv EPOC* уређајима су развијене да би обезбедиле квалитетан и стабилан пренос ЕЕГ сигнала. Слика 9 и слика 10 приказују електроде *Emotiv Insight* и *Emotiv EPOC* уређаја, респективно.



Слика 9 Emotiv Insight електроде [110]



Слика 10 *Emotiv Epos* електроде [112]

3.1.1.3.1 Материјал

У уређају *Emotiv Insight* електроде су израђене од хидрофилног полусувог полимера. Ова врста полимера може апсорбовати воду без стварања мокре површине, што је идеално за пренос биоелектричних сигнала. С друге стране, уређај *Emotiv Epos* користи јастучиће натопљене физиолошким раствором за своје сензоре. Они се натапају у сланој води пре употребе, што им омогућава ефикасан пренос ЕЕГ сигнала.

3.1.1.3.2 Дизајн

Оба уређаја су дизајнирана да буду лагана и удобна за ношење, чиме се смањује притисак на главу корисника. Ово је нарочито важно за дуготрајне експерименте и истраживање у когнитивним неуронаукама, где је комфор од суштинске важности. Сензори су позиционирани у складу с међународним 10-20 стандардом.

3.1.1.3.3 Функционалност

Електроде оба *Emotiv* уређаја су дизајниране да раде без потребе за додатним проводницима или абразивним препаратима, који су често потребни у традиционалним ЕЕГ системима за постизање доброг контакта с кожом. Ово је важно за истраживање у области *Brain-Computer Interface* (BCI) система и когнитивних неуронаука, пошто омогућава бржу и лакшу припрему, а такође смањује ризик од иритације коже.

3.1.1.3.4 Предности

Предности *Emotiv Insight* и *Emotiv Epos* уређаја у односу на конвенционалне ЕЕГ системе обухватају низ аспеката:

1. **Брза и лака припрема:** Једна од великих предности ових уређаја је употреба хидрофилних полусувих полимера или јастучића натопљених физиолошким раствором, који отклањају потребу за додатним проводницима или абразивним препаратима. Ово обезбеђује брзу и laku припрему уређаја, која је посебно корисна у истраживањима и применама где је важна ефикасност и корисничко искуство.
2. **Комфор:** Дизајн ових електрода омогућава удобно ношење уређаја током релативно дужег периода у односу на конвенционалне електроде и својим обликом обезбеђује стабилнију везу са кожом. Лаган и флексибилан дизајн доприноси удобности током дужег времена, што је кључно за релативно дуже експерименте и истраживања у области BCI и когнитивних неуронаука.
3. **Повећана прецизност и стабилност сигнала:** Коришћење специјално израђених материјала омогућава ефикасан пренос ЕЕГ сигнала и стабилан контакт с кожом, што доприноси

побољшаној прецизности и стабилности сигнала у односу на конвенционалне металне електроде.

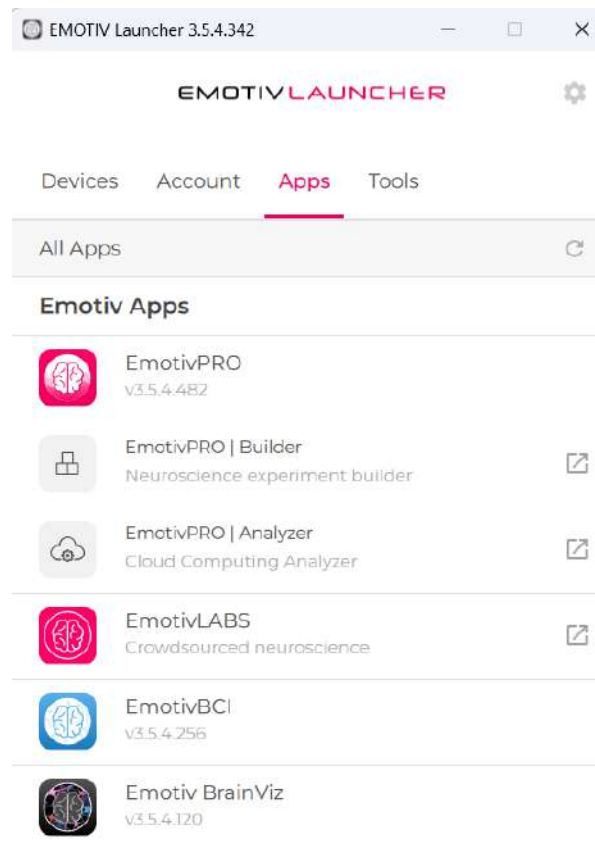
Наведене карактеристике чине их подобним за примену у области *BCI* система и експериментима у когнитивним неуронаукама, где је важно постићи високу прецизност, ефикасност и минималну дистракцију корисника. У будућности се могу очекивати даља унапређења у дизајну и материјалима *Emotiv* електрода. Ово би могло водити ка још већој прецизности и стабилности сигнала, као и ка додатној удобности и једноставности за кориснике.

3.1.1.4 *Emotiv* програмски пакети

Компанија *Emotiv* нуди разне софтверске алате који омогућавају коришћење, анализу и интеграцију података с *Emotiv* ЕЕГ уређајима. Овде ће бити описани неки од њихових главних софтверских алата.

3.1.1.4.1 *Emotiv* Launcher

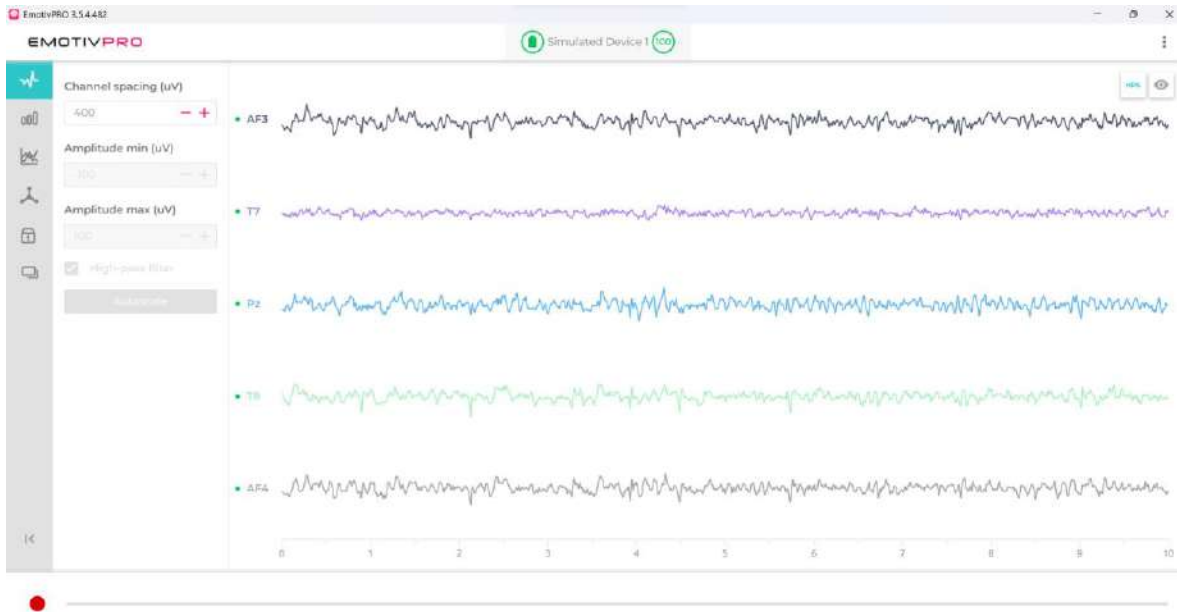
Emotiv Launcher је централни софтвер који омогућава приступ свим осталим алатима компаније *Emotiv*. Он омогућава увоз и извоз података, повезивање више уређаја и креирање виртуелних уређаја. Путем овог алата корисници могу да управљају својим уређајима и да учитавају и прегледају податке с *Emotiv* уређаја. Основни кориснички интерфејс *Emotiv Launcher* апликације је приказан на слици 11.



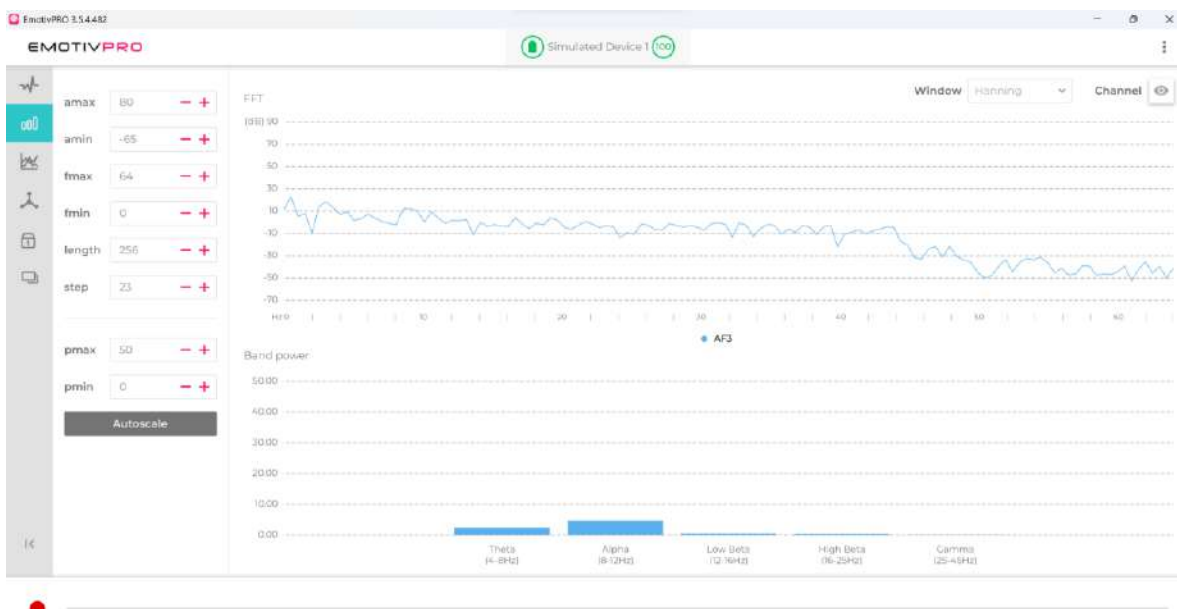
Слика 11 *Emotiv Launcher* [115]

3.1.1.4.2 EmotivPRO

EmotivPRO је софтвер за праћење и регистрацију података с *Emotiv* уређаја у реалном времену. Он омогућава корисницима да прате своја когнитивна стања и когнитивне активности уживо, што је приказано на сликама 12 и 14. *EmotivPRO* такође нуди алате за анализу и визуелизацију података као што су спектрална анализа која је приказана на слици 13 и сигнале жироскопа, акцелерометра и магнетометра који су приказани на слици 15. Овај софтвер такође пружа могућност и прегледа, слања и експортовања постојећих снимака, што је приказано на слици 16.



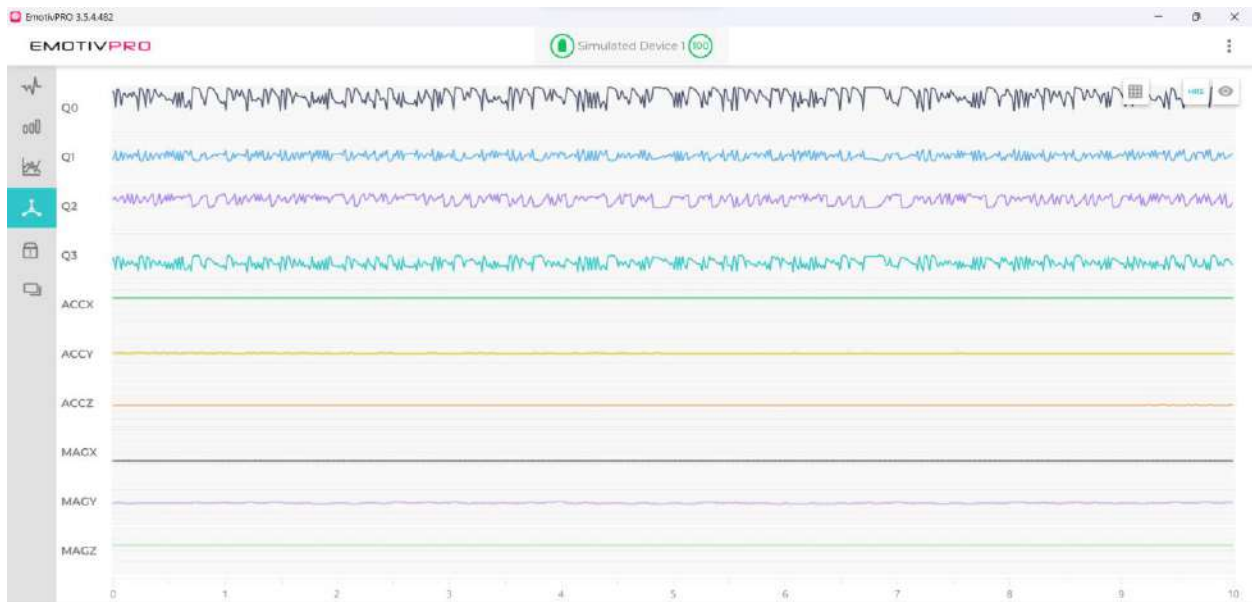
Слика 12 Мерење и приказ ЕЕГ сигнала у реалном времену [116]



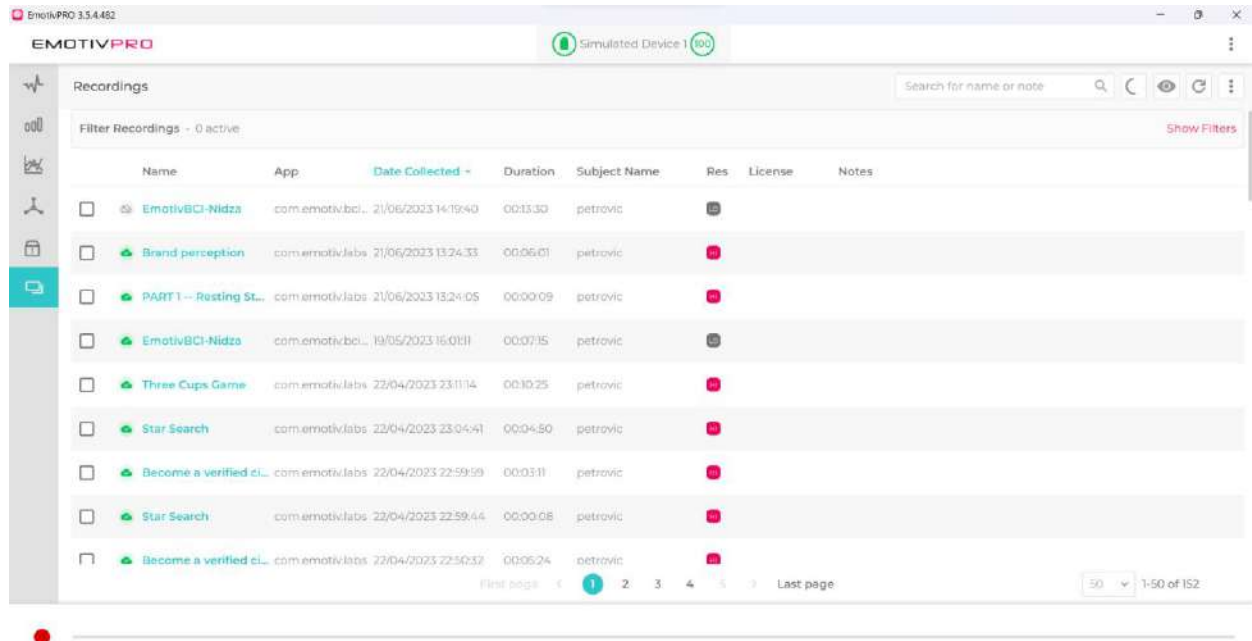
Слика 13 Приказ фреквенцијских компонента изабраног канала [116]



Слика 14 Мерење и приказ когнитивних стања испитаника [116]



Слика 15 Мерење и приказ сигнала са жироскопа, акцелерометра и магнетометра [116]



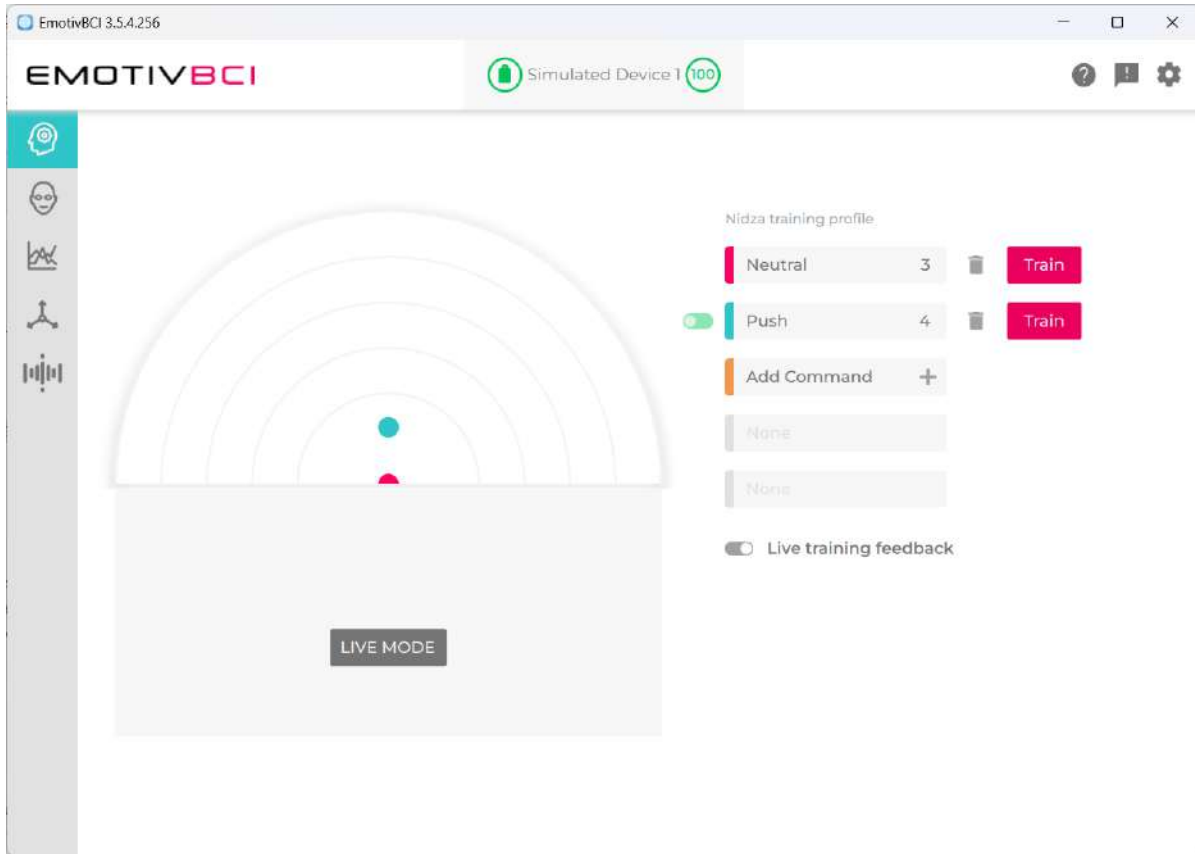
Слика 16 Преглед сачуваних EEG сигнала током експеримената [116]

3.1.1.4.3 EmotivLABS

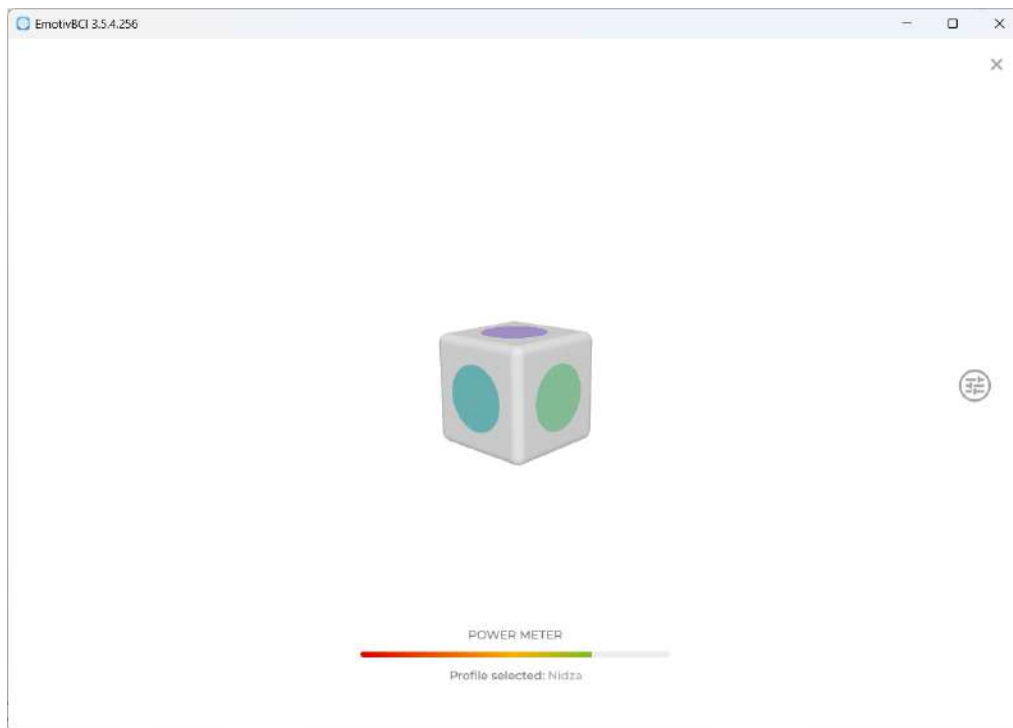
EmotivLABS је алат за дизајнирање и тестирање експеримената и прототипа у области когнитивних истраживања и неуронаука. Он омогућава истраживачима да развијају нове апликације и тестирају идеје за коришћење *Emotiv* уређаја у различитим контекстима [116].

3.1.1.4.4 EmotivBCI

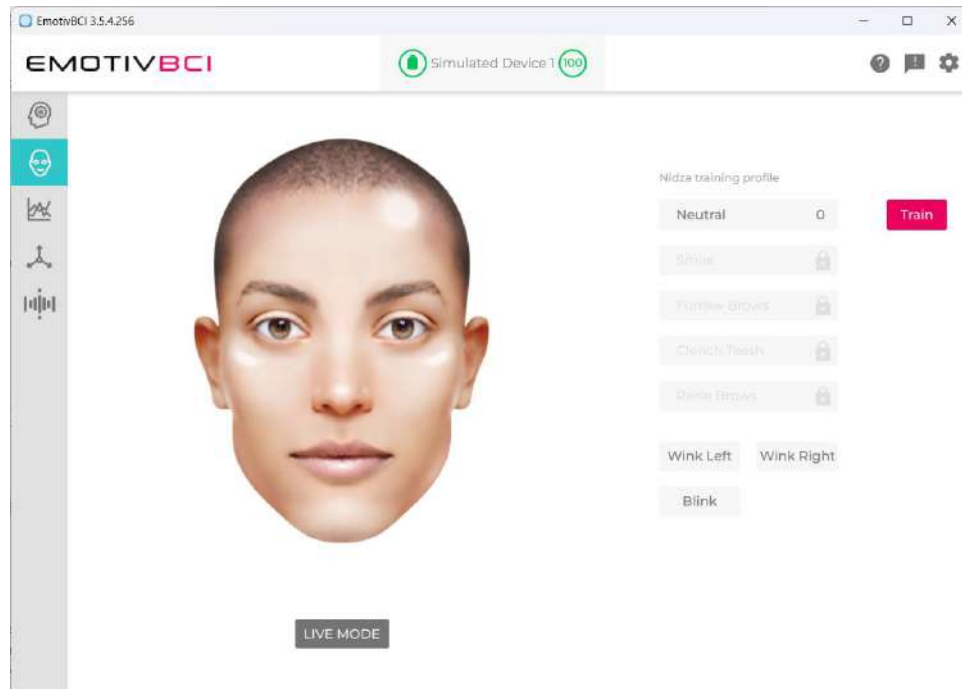
EmotivBCI је софтвер за развој и примену BCI система. Он омогућава корисницима да тренирају и тестирају BCI моделе, као и да развијају и прилагођавају апликације које користе ове моделе. *EmotivBCI* нуди графички кориснички интерфејс за дефинисање и управљање когнитивним командама, као и интеграцију с другим софтверским и хардверским компонентама [117]. Основни мени, приказан на слици 17, пружа могућност одабира когнитивних активности које ће испитаник вежбати. Током увежбавања, процењена когнитивна активност се испитанку приказује у реалном времену, чиме се остварује неурофидбек петља, што је приказано на слици 18. Поред тога, могуће је вршити и препознавање и тренирање фацијалних експресија, што је приказано на слици 19.



Слика 17 Emotiv BCI [117]



Слика 18 Emotiv BCI и рад у реалном времену [117]



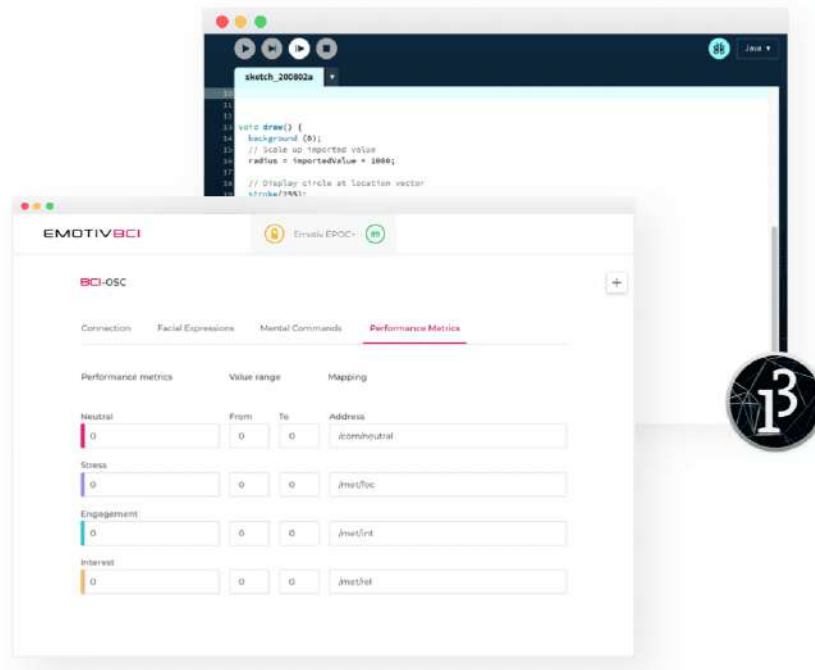
Слика 19 Алат за тренирање и праћење фаџијалних експресија [117]

3.1.1.4.5 Emotiv OSC

Emotiv OSC је софтверски алат који омогућава слање података о мозгу у реалном времену из *Emotiv BCI* апликације до било ког другог уређаја или апликације кроз мрежу користећи *Open Sound Control* (OSC) протокол. Овај алат је дизајниран да обезбеди брз, лак и ефикасан начин за слање и примање података, омогућавајући непрекидну комуникацију између различитих платформи и апликација.

Emotiv OSC се једноставно конфигурише и обезбеђује брз пренос података без потребе за сложеним кодирањем. Он подржава различите врсте података, укључујући ЕЕГ податке (електроенцефалограм), податке о перформансама (нпр. интерес, фокус), менталне команде (нпр. гурање, вучење), и фаџијалне експресије (нпр. трептање, померање обрва, стискање вилице и померање усана).

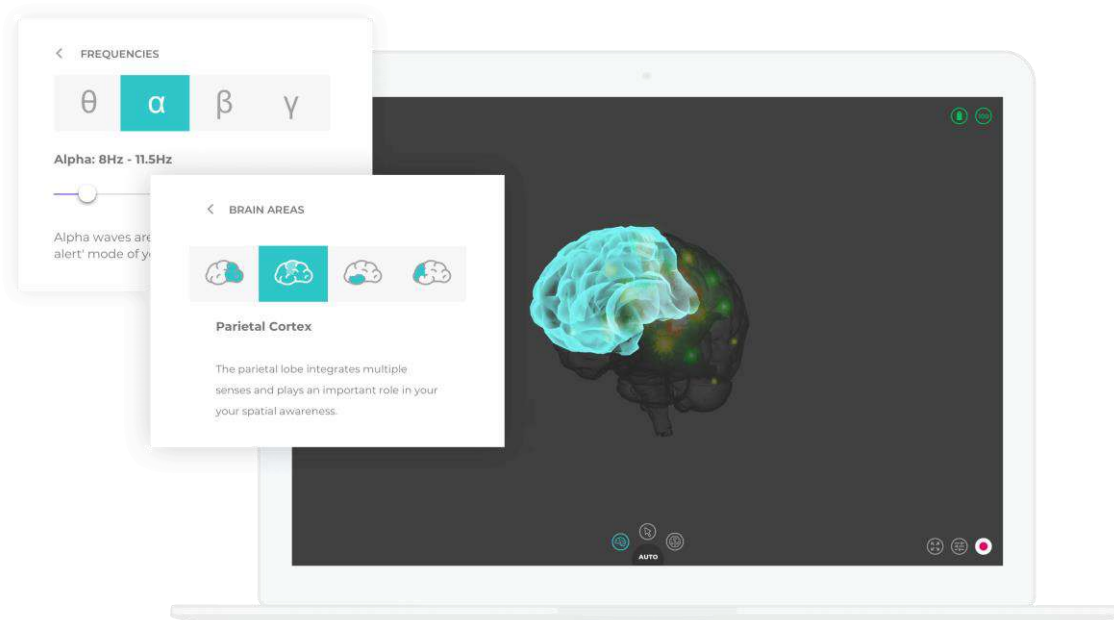
Овај алат је користан у ситуацијама где је потребно пренети и анализирати податке о мозгу у реалном времену, што може бити корисно у различитим контекстима, укључујући истраживања, образовање, развој апликација и игара, и многе друге области. Слика 20 приказује основни панел *Emotiv OSC*-а.



Слика 20 Emotiv OSC [117]

3.1.1.4.6 Emotiv BrainViz

Emotiv BrainViz је алат за визуализацију тродимензионалне мождане активности који омогућава корисницима да прегледају, анализирају и интерпретирају податке с *Emotiv* уређаја. Он нуди разне опције за визуализацију података, укључујући топографске мапе, временске серије и спектралне графиконе. Такође, омогућава корисницима да увозе и извозе податке и да сачувају визуализације за касније коришћење [118]. Слика 21 приказује основни панел алата *Emotiv BrainViz*.



Слика 21 Emotiv BrainViz [118]

3.1.1.4.7 Emotiv Cortex

Emotiv Cortex представља апликациони програмски интерфејс (API) који је компанија *Emotiv* развила у циљу омогућавања програмерима да интегришу и користе *Emotiv* уређаје и сервисе у својим апликацијама. Овај API је специјално дизајниран да олакша рад с подацима из *Emotiv* уређаја и да допринесе развоју иновативних апликација заснованих на когнитивним истраживањима и неуронаукама [119].

Cortex API подржава различите програмске језике, укључујући *Python*, *Java*, *C++*, *JavaScript* и друге, што програмерима омогућава да развијају апликације које могу да комуницирају с *Emotiv* уређајима на различитим платформама. *Cortex API* има функционалности које омогућавају програмерима да:

1. Управљају *Emotiv* уређајима, укључујући повезивање, одјављивање и контролу статуса уређаја.
2. Обрађују и анализирају податке о мозгу, укључујући спектралну анализу, топографске мапе и израчунавање метрика когнитивних функција и когнитивних стања.
3. Интегришу апликације с *Emotiv* сервисима и ресурсима, као што су база за чување података, кориснички профили и аутентикација.
4. Имплементирају *Brain-Computer Interface (BCI)* системе и апликације које користе когнитивне команде и когнитивна стања за контролу електронских уређаја, како би корисници могли да управљају разним уређајима и системима користећи своје мисли и концентрацију. Користе функције за приступ сензорима у уређајима као што су акцелерометар, жироскоп и магнетометар, који могу да прате покрете и оријентацију корисника.
5. Развијају алгоритме и апликације за праћење и анализу пажње, релаксације, стреса и других менталних стања, у циљу подршке истраживању у области психологије, неурологије као и когнитивних неуронаука.
6. Интегришу *Emotiv* технологију с виртуелном и проширеном реалношћу, у циљу стварања реалистичних и интензивних корисничких искустава која се прилагођавају менталним стањима и реакцијама корисника.

Emotiv Cortex API пружа програмерима и истраживачима могућност да развијају персонализоване апликације и системе који могу да искористе капацитете *Emotiv* уређаја и технологије. Коришћење *Cortex API-ја* омогућава развојним инжењерима да створе иновативна решења у разним индустријама, попут образовања, здравства, развоја видео-игара, роботике и многих других области.

Додатно, *Emotiv* нуди и опширну документацију и подршку за коришћење *Cortex API-ја*, укључујући упутства, примере кода, туторијале и форуме за размењивање идеја и решавање проблема у раду с овим апликационим програмским интерфејсом. Ово омогућава програмерима да брзо и ефикасно развијају своје апликације, стекну дубље разумевање могућности и ограничења *Emotiv* технологије и искористе све предности *Cortex API-ја* у својим пројектима.

Поред тога, *Emotiv* организује и радионице, обуке и вебинаре који омогућавају програмерима да уче од стручњака из области неуронаука и инжењеринга, да размењују идеје с колегама и да истражују нове могућности и примене *Emotiv Cortex API-ја*. Овакав вид образовања и сарадње омогућава

развој иновативних решења и промовише коришћење *Emotiv* технологије у различитим индустријама и доменима.

Emotiv Cortex API пружа могућности програмерима и истраживачима да учествују у развоју технологије и науке, стварајући апликације које могу да унапреде начин на који живимо, учимо, радимо и комуницирамо [119].

3.1.1.5 Предности у односу на конвенционалне ЕЕГ уређаје

Emotiv уређаји пружају неколико предности у односу на конвенционалне ЕЕГ уређаје:

1. Преносивост и лакоћа коришћења: *Emotiv* уређаји су компактни и лагани, што омогућава лако ношење и коришћење у различитим окружењима, укључујући дом, канцеларију или лабораторију.
2. Удобност и бежична комуникација: Захваљујући полусувим електродама и бежичној *Bluetooth* вези, *Emotiv* уређаји су удобнији за кориснике и омогућавају глатко и брзо повезивање с компатибилним уређајима.
3. Брзо и једноставно постављање: Постављање *Emotiv* уређаја траје неколико минута, док конвенционални ЕЕГ уређаји често захтевају дуже време и више додатне опреме за постављање. Уређај не захтева причвршћивање за остатак тела.
4. Прилагодљивост и персонализација: *Emotiv* уређаји имају могућност да се прилагоде различитим истраживачким потребама, као што су стимулација у различитим доменима (визуелним, аудитивним, тактилним) или праћење специфичних когнитивних процеса.
5. Приступачност и ниска цена: *Emotiv* уређаји су знатно јефтинији од већине конвенционалних ЕЕГ уређаја, што их чини доступним већем броју истраживача и применама у различитим областима.

Укупно гледајући, *Emotiv Insight* и *Epos* нуде многе предности у односу на конвенционалне ЕЕГ уређаје, што их чини погодним за широк спектар истраживачких и клиничких примена у области когнитивних неуронаука, неурофидбека, медицинске дијагностике и терапије, као и за развој иновативних технологија које укључују управљање уређајима путем мозга, виртуелну реалност и образовање [114].

3.1.1.6 Интеграција с другим технологијама и платформама

Emotiv уређаји могу се лако интегрисати с различитим технологијама и платформама, што их чини савршеним за истраживаче и инжењере који желе да развијају иновативне апликације. Неки од могућих начина интеграције укључују:

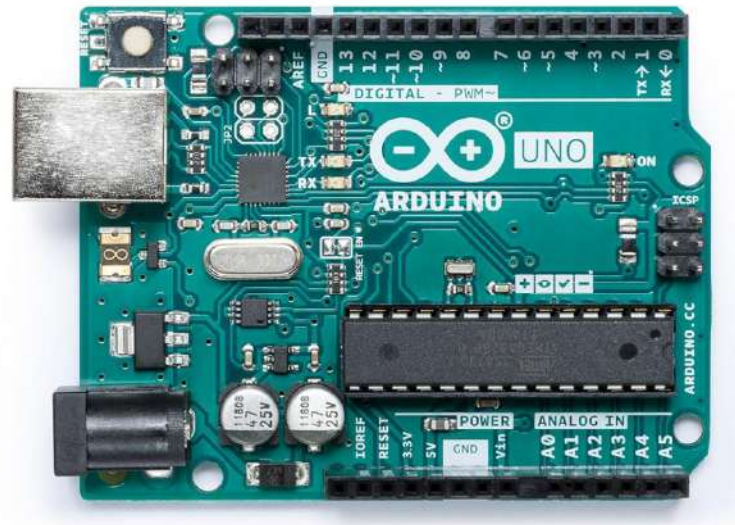
1. Комбиновање ЕЕГ података с подацима других мерних уређаја, као што су акцелерометри, жироскопи, сензори температуре или ЕКГ уређаји, да би се добиле шире информације о стању корисника.
2. Упаривање с роботским системима или беспилотним летелицама, где ЕЕГ сигнали могу бити коришћени за контролу покрета или радњи робота/дрона, као и за анализу корисничковог стања током рада с тим уређајима.

3. Развој апликација за мобилне уређаје и паметне телефоне, које могу користити ЕЕГ податке за мониторинг здравља, стреса, спавања или когнитивних способности корисника у реалном времену.
4. Коришћење ЕЕГ података у образовним апликацијама и тренинг програмима, где се ЕЕГ сигнали могу користити за праћење напретка учења, пажње, као и за прилагођавање материјала и метода обучавања на основу индивидуалних потреба ученика.
5. Примена у неурофидбек терапији, где ЕЕГ сигнали могу бити коришћени за обучавање корисника да контролишу своја когнитивна стања, побољшају концентрацију, управљају стресом и побољшају своје когнитивне способности.
6. Интеграција с биометријским системима за аутентификацију и безбедност, где се ЕЕГ сигнали могу користити као један од параметара за проверу идентитета корисника и заштиту приватности података.
7. Истраживање у области когнитивних неуронаука, где се *Emotiv* уређаји могу користити за студије о мозгу, когнитивним процесима, емоцијама, моторичким функцијама и другим аспектима људске психофизиологије.
8. Уметничке инсталације и перформанси, где се *Emotiv* уређаји могу користити за превођење ЕЕГ сигнала у аудио или визуелне израде, стварајући јединствено уметничко дело које одражава унутрашње стање уметника или гледаоца.
9. Развој апликација виртуелне реалности (VR) и проширене реалности (AR), где се *Emotiv* уређаји могу користити за интеграцију когнитивних и емоционалних података корисника у интерактивним окружењима. Ово може довести до стварања богатијих и реалистичнијих виртуелних искустава која се адаптирају на основу уникатних потреба и реакција корисника.
10. Биоинформатички алгоритми и статистичке анализе, где се *Emotiv* уређаји могу користити за добијање велике количине ЕЕГ података из различитих извора. Ови подаци могу затим бити анализирани и обрађени ради идентификације образаца, предвиђања и утврђивања веза између различитих когнитивних и емоционалних стања.
11. Спортска наука и перформансе, где се *Emotiv* уређаји могу користити за праћење менталног стања и фокуса спортиста, као и за оптимизацију тренинга и стратегија такмичења на основу индивидуалних когнитивних профила.

3.1.2 Arduino развојно окружење

Arduino је отворена хардверска и софтверска платформа за развој пројеката који укључују контролу, аутоматизацију и мониторинг у реалном времену. Основан на лако доступној микроконтролерској платформи, *Arduino* омогућава корисницима да развијају прототипе и решења без потребе за посебним знањем о електроници или комплексном програмирању. *Arduino* микроконтролери имају велики број улаза и излаза који се могу користити за повезивање различитих сензора, актуатора и других компоненти. Улази могу бити аналогни или дигитални, што омогућава прикључак сензора који могу мерити различите физичке величине, као што су температура, светлост, звук, притисак и друге. Излази се такође могу конфигурисати као аналогни или дигитални, што омогућава управљање актуаторима као што су мотори, серво механизми, светлеће диоде и

друге компоненте. Програмски језик *Arduino* је базиран на средствима *Wiring* (стандардна програмска библиотека за програмирање микроконтролера) и *C/C++* језику, који омогућава лако писање и читање кода, а такође и брзо и ефикасно извршавање програма на микроконтролеру. *Arduino* интегрисано развојно окружење (IDE) је једноставно и интуитивно, што олакшава развој, тестирање и примену различитих програма [120].



Слика 22 Arduino UNO [120]

У оквиру овог истраживања је коришћена *Arduino UNO* развојна плоча, приказана на слици 22, за развој неурофидбек система с физичким излазом за управљање мини-дронем на основу когнитивних стања испитаника. *Python* апликација врши аквизицију ЕЕГ сигнала, процењених когнитивних стања и активности помоћу *Emotiv Cortex API*-а, након чега се процењене когнитивне активности конвертују у одговарајуће вредности које се затим шаљу *Arduino UNO* развојној плочи. Плоча врши конверзију добијених вредности у одговарајуће *PWM* сигнале који се затим конвертују у аналогни напон на управљачу дрона, који на основу тога контролише покрете самог дрона. Детаљан опис методологије неурофидбек система са актуаторским излазом је описан у поглављу 4.4.

Arduino платформа нуди бројне предности за овакав тип пројекта:

1. Лако програмирање и приступачност: *Arduino* је једноставан за коришћење и програмирање, што омогућава брз развој и тестирање система.
2. Флексибилност и могућност прилагођавања: *Arduino* омогућава повезивање различитих сензора и актуатора, што га чини погодним за различите апликације и експерименте.
3. Ниска цена: *Arduino* платформа је доступна по ниској цени, што чини овакав систем доступним за широк спектар корисника и истраживача.
4. Подршка за комуникацију: *Arduino* подржава различите комуникационе протоколе, као што су *USB*, *I2C*, *SPI* и друге, што омогућава лаку интеграцију с другим системима и уређајима.

Коришћењем *Arduino* платформе у овом истраживању систем показује велик потенцијал у примени ЕЕГ технологије и *BCI* система у различитим областима, укључујући образовање, забаву, здравство

и рехабилитацију. Овакви системи могу имати широк спектар примена, као што су едукација, терапија, рехабилитација, когнитивна стимулација и

3.1.2.1 Архитектура *Arduino* система

Arduino системи су засновани на микроконтролерима, који су мали рачунари на једном интегрисаном колу. Микроконтролери се могу програмирати да извршавају различите задатке и да контролишу друге електронске компоненте. *Arduino* микроконтролери су најчешће базирани на *Atmel AVR* серији, али постоје и верзије засноване на *ARM* и другим архитектурама.

Arduino плоче су дизајниране тако да буду једноставне за коришћење и да омогућавају laku везу с другим компонентама. Оне обично имају више аналогних и дигиталних улазно/излазних пинова, који омогућавају повезивање сензора, актуатора и других електронских компонента. Такође, *Arduino* плоче имају интегрисане стабилизаторе напајања, комуникационе интерфејсе и програмске библиотеке које олакшавају програмирање и комуникацију с другим уређајима и компонентама [121].

3.1.2.2 Конективност, улази и излази

Arduino системи имају улазе и излазе који омогућавају повезивање са различитим компонентама. Улази и излази се могу поделити на:

- **Дигитални улази и излази (*Digital I/O*):** Ови пинови могу бити конфигурисани као улази или излази и користе се за читање или слање дигиталних сигнала (вредности 0 или 1).
- **Аналогни улази (*Analog Inputs*):** Пинови који могу читати аналогне сигнале и конвертовати их у дигиталне вредности у распону од 0 до 1023. Они се често користе за читање сензорских података.
- **PWM излази (*PWM Outputs*):** Дигитални пинови који подржавају *Pulse Width Modulation*, омогућавајући симулацију аналогних сигнала коришћењем дигиталних излаза.

Arduino системи користе више протокола за комуникацију и повезивање са спољашњим уређајима:

- **USB:** Користи се за повезивање плоче са рачунаром ради програмирања и серијске комуникације. Овај интерфејс омогућава пренос података између плоче и рачунара у реалном времену.
- **I2C (*Inter-Integrated Circuit*):** Комуникациони протокол који омогућава повезивање са различитим сензорима и уређајима користећи само два пина (*SDA* и *SCL*). Овај протокол подржава комуникацију са више уређаја на истој магистралаи.
- **SPI (*Serial Peripheral Interface*):** Брзи серијски комуникациони протокол који омогућава повезивање са уређајима као што су *SD* картице и екрани. Користи четири пина (*MISO*, *MOSI*, *SCK* и *SS*) за комуникацију.

3.1.2.3 Интеграција са рачунарима и другим компонентама

Arduino системи су веома прилагодљиви и могу се лако интегрисати у разне пројекте, укључујући комуникацију с рачунарима и другим компонентама. У случају примене у оквиру овог пројекта, *Arduino* може бити повезан с рачунаром преко *USB* интерфејса и користити серијску комуникацију за примање команди из *Python* програма. На основу тих команди, *Arduino* може да контролише мини-дрон путем аналогних или дигиталних излаза и актуатора.

Комуникација између *Arduino* система и рачунара може се успоставити помоћу различитих програмских библиотека, као што су *PySerial* и *Firmata*. Ове библиотеке пружају једноставан начин за програмирање и управљање комуникацијом између рачунара и *Arduino* плоче, омогућавајући тако лакшу интеграцију и контролу *Arduino* система из спољашњих апликација [122].

3.1.2.4 Примери примене *Arduino* у когнитивним истраживањима и *BCI* системима

Arduino системи су већ коришћени у разним истраживањима и апликацијама у области когнитивних наука и *BCI* система. Неки од примера су:

1. Неурофидбек тренинг системи - *Arduino* се користи за развој система који прате когнитивне активности корисника и дају им фидбек у виду звука, светла или других стимулуса. Ово може помоћи корисницима да развију вештине самоконтроле својих когнитивних процеса и стања.
2. Бежични *BCI* системи – *Arduino* системи се могу користити за изградњу бежичних комуникационих модула који омогућавају пренос података између ЕЕГ уређаја и рачунара или других компонената. То омогућава корисницима већу слободу кретања и удобност током истраживања или терапије.
3. Реализација виртуелне и проширене реалности – *Arduino* може бити коришћен за контролу виртуелних реалности или аугментационих система на основу когнитивних активности корисника. Ово може створити искуства која комбинују когнитивне активности и виртуелна окружења.
4. Игре и забавни садржаји – *Arduino* системи могу се користити у развоју интерактивних игара и забавних садржаја који се контролишу користећи когнитивне активности корисника. Ово може створити нове и уникатне играчке искуства која комбинују технологију и умањену стварност.
5. Асистивне технологије и роботика – *Arduino* системи могу бити коришћени за развој асистивних технологија које омогућавају особама с ограниченим функционалним способностима да контролишу протетичке уређаје, роботе или друге системе помоћу својих когнитивних активности. Ово може значајно повећати квалитет живота и независност особа с инвалидитетом.

3.1.3 *Unity* програмско окружење

Unity је вишеплатформско програмско окружење које омогућава развој апликација и игара за различите платформе, укључујући рачунаре, телефоне и *Oculus VR* системе. Основан на програмском језику *C#*, *Unity* нуди графички интуитивне алате који олакшавају креирање и манипулацију 3D објеката и окружења. *Unity* се широко користи у различитим индустријама, као што су забава, образовање, здравство, аутомобилска индустрија и архитектура, за развој виртуелних и проширених реалности апликација, симулација и тренинга [123].

У оквиру овог истраживања *Unity* програмско окружење користи се за развој симулације која интегрише неурофидбек *BCI* систем. Програмски језик *C#* користи се за креирање скрипти које контролишу интеракцију између корисника и симулације, док се пакет *Emotiv Cortex* користи за комуникацију са *Emotiv* ЕЕГ уређајима. Комбинацијом *Unity-ja* и програмског језика *C#* овде се развија флексибилна и адаптивна платформа за неурофидбек *BCI* систем, која може да се користи у различитим областима и применама [124].

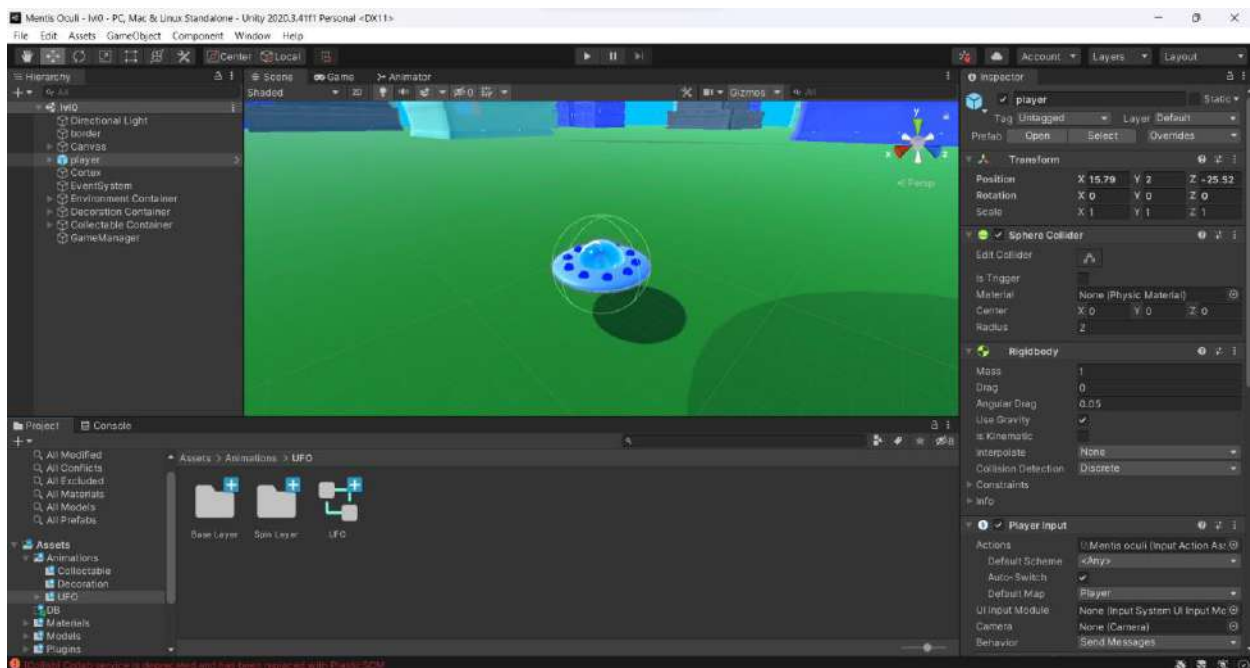
3.1.3.1 Развој апликација у Unity-ју

За сваку од претходно наведених платформи, *Unity* нуди различите опције и алатке које олакшавају процес развоја и оптимизације апликација.

Развој апликација за рачунаре у *Unity-ју* великим делом фокусира се на стварање интуитивних и ефикасних интерфејса за кориснике. Ово подразумева коришћење графичких елемената, анимација и звукова да би се створило функционално корисничко искуство.

Кад је у питању развој апликација за мобилне уређаје, *Unity* нуди оптимизацију перформанси и ефикасност батерије. Ово омогућава креирање апликација које раде ефикасно на различитим телефонима и таблетима.

Unity такође подржава развој апликација за *Oculus VR* системе, који се користе у различитим областима, као што су образовање, тренинг, медицина и забава. У оквиру овог истраживања развијена је симулација уз помоћ *Unity-ја* која омогућава кориснику да се потпуно уживи у виртуелни свет и контролише окружење помоћу својих когнитивних активности [125]. На слици 23 је приказано *Unity* развојно окружење током креирања неурофидбек система са симулационим излазом, који је описан у поглављу 4.3.



Слика 23 Unity програмско окружење

3.1.3.2 Коришћење програмског језика C# у Unity-ју

C# је објектно-оријентисани програмски језик који се широко користи у развоју апликација и игара у *Unity-ју*. Он омогућава писање јасних и ефикасних скрипти које контролишу померање објеката, интеракцију с корисником и друге аспекте апликације. Уз помоћ програмског језика *C#*, развојни инжењери могу да креирају комплексне и динамичке системе у оквиру *Unity* апликација.

3.1.3.3 Примена Unity-ја и програмског језика C# у развоју симулације за неурофидбек

У оквиру овог пројекта програмски језик C# и Unity програмско окружење коришћени су за развој симулације за неурофидбек. Циљ ове симулације је да омогући корисницима да контролишу виртуелне објекте и окружење користећи своје когнитивне активности, док им се у исто време пружа реално и задовољавајуће искуство.

Симулација је развијена тако да користи податке добијене из *Emotiv Cortex API-ја*, који се затим прослеђују кроз Unity сцену као параметри за контролу објекта и окружења. Ово омогућава корисницима да створе јединствену везу између својих мисли и активности у виртуелном свету.

За развој симулације коришћен је низ C# скрипти које обрађују податке с *Emotiv Cortex API-ја*, контролишу објекте у сцени и реагују на корисничке активности. Ове скрипте су интегрисане у Unity окружење, што омогућава стабилно и глатко функционисање симулације.

Коришћењем Unity програмског окружења и програмског језика C#, ова симулација представља иновативни и флексибилни алат који може да се користи у различитим истраживачким областима, као што су образовање, рехабилитација и тренинг.

3.1.4 Oculus Quest 2 систем

3.1.4.1 Оперативни систем и хардвер

Oculus Quest 2, приказан на слици 24, је систем виртуелне реалности (VR) који је развила компанија *Oculus*, партнер Фејсбука. Он функционише на основу оперативног система *Android* и користи сопствени *Oculus OS*. *Oculus Quest 2* представља значајан напредак у односу на предходника, с бољим процесором, већом резолуцијом екрана и смањеном тежином.



Слика 24 Oculus Quest 2 с контролерима [126]

3.1.4.2 Пратећа опрема

Осим самог VR система, *Oculus Quest 2* се испоручује с два контролера, који прате рад корисника у окружењу виртуелне стварности. Контролери, приказани на слици 25, су дизајнирани да буду

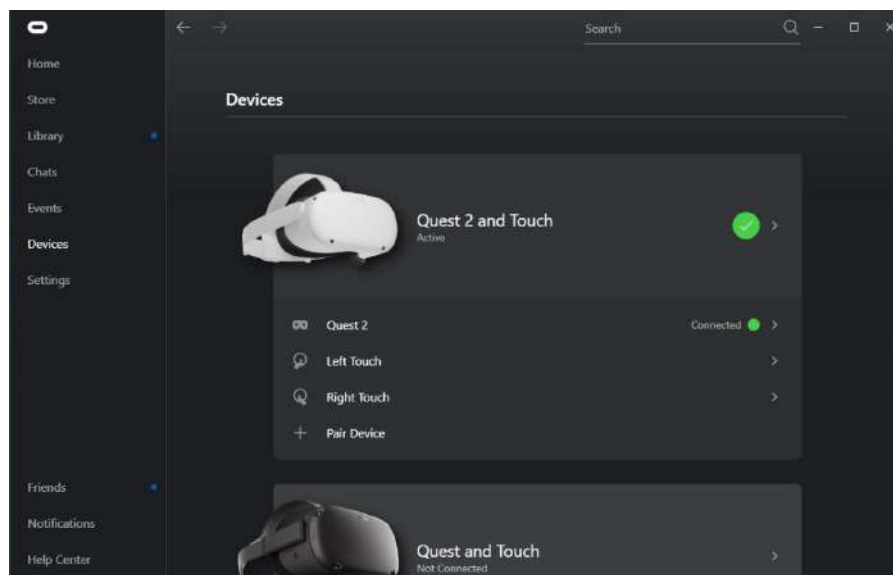
ергономски и да омогућавају прецизно управљање. *Oculus Quest 2* такође садржи четири уграђене камере које служе за праћење позиције у окружењу као и усклађивање активности из стварног и симулираног окружења.



Слика 25 *Oculus Quest 2* контролери [126]

3.1.4.3 Повезивање с телефоном и рачунаром

Oculus Quest 2 може се повезати с телефоном користећи мобилну апликацију *Oculus*, која омогућава корисницима да управљају својим уређајем, преузимају апликације и игре и прате активности других корисника. Такође, *Oculus Quest 2* може се повезати с рачунаром помоћу *Oculus Link* кабла, који омогућава приступ VR апликацијама и играма које су доступне на рачунару. Апликација за мобилне телефоне и рачунаре која врши повезивање је приказана на слици 26.



Слика 26 *Oculus link* апликација [126]

3.1.4.4 Могућности, значај и предности

Oculus Quest 2 нуди широк спектар могућности за кориснике, укључујући игре, образовне апликације и социјалне платформе. Виртуелна стварност пружа потпуно симулирано искуство које

може помоћи корисницима да се сасвим уживе у светове које стварају разне апликације и игре, што пружа значајне предности у области образовања, обуке и рехабилитације.

У контексту овог пројекта *Oculus Quest 2* се користити за стварање VR симулације која ради у склопу неурофидбек ВЦИ система. Испитаници могу користити своје когнитивне активности да управљају виртуелним објектима или окружењем, што доприноси стимулацији неуропластичности и унапређењу когнитивних способности.

Предности *Oculus Quest 2* укључују његову преносивост, једноставност употребе и доступност широком спектру апликација. Овај уређај представља ефикасан алат за развој и испитивање идеја у области когнитивних наука, укључујући и овај пројекат. Слика 27 приказује кориснички интерфејс контролисан помоћу препознавања покрета руку, што демонстрира могућности *Oculus Quest 2* уређаја за интеракцију са виртуелним окружењем на интуитиван и природан начин.

Укратко, *Oculus Quest 2* пружа велики потенцијал за интеграцију у ВЦИ системе и неурофидбек експерименте, као и за стварање иновативних виртуелних окружења која могу допринети стимулацији неуропластичности и унапређењу когнитивних способности [125, 127, 128].



Слика 27 Кориснички интерфејс контролисан помоћу препознавања покрета руку [126]

3.2 Опис експеримента

Експерименти су осмишљени с циљем да омогуће испитаницима управљање различитим системима путем визуелних менталних слика које су повезане с основним геометријским кретањем. Ови експерименти су дизајнирани да процене шест кључних аспеката менталне концентрације: одрживост, агилност, контрола, временска издржљивост, способност паралелног обављања задатака и отпорност на спољашње дистракције.

Приликом осмишљавања овог експеримента, разматрано је како ће свака фаза допринети процени и развоју менталне концентрације испитаника. Намера је била да се добије увид у стање и потенцијал за развој код испитаника у вези са поменутиим аспектима, а резултати би омогућили дефинисање индивидуалних планова за унапређење вештина.

3.2.1 Припрема испитаника

Припрема испитаника за спровођење експеримента обухвата следеће кораке:

На самом почетку неопходно је упознати испитаника с опремом. Овај процес подразумева кратак опис сврхе уређаја, при чему је важно напоменути да је уређај у потпуности неинвазиван и безопасан.

Након тога прелази се на давање инструкција које се односе на физички положај испитаника током калибрације. Од испитаника се тражи да се потпуно физички опусти и заузме правилан положај као што је приказано на слици 28, што подразумева да су му леђа наслоњена на наслон столице, дланови постављени на горњу страну бутина, стопала сасвим ослоњена на под у међусобном паралелном односу и на размаку ширине кукова. Додатно, истиче се важност избегавања фацијалних експресија уз нормално дисање и трептање. Испитаник се такође упозорава да током калибрације мора остати у мирном стању.



Слика 28 Пример правилног положаја седења

Усмеравање фокуса на когнитивно стање испитаника подразумева вођење кратког опуштајућег разговора. Циљ је да се испитаник потпуно психофизички опусти, што ће му омогућити да успешно изврши предстојеће когнитивне вежбе (3.2.2–3.2.5). Време потребно за опуштање је индивидуално и захтева субјективну процену особе која надгледа извршење експеримента. Испитаник треба да буде упознат с концептима менталних слика, њиховог активног одржавања као и измена.

Након припреме испитаника потребно је у што краћем временском року прећи на следећи корак (3.2.2–3.2.5). Током трајања експеримента особа која надгледа експеримент је у обавези да прати реакције испитаника и уколико оне знатно одступају од задатих упутстава, неопходно је подсетити испитаника на њих.

3.2.2 Једнодимензионалне неурофидбек вежбе

У следећој фази испитанику се представља тродимензионални геометријски облик, који може бити у стању мировања или кретања, при чему брзина зависи од интензитета визуализације кретања. Током ове фазе испитаник развија две визуелне менталне слике и учи да вољно контролише њихову визуализацију.

Калибрација, као кључни корак, подразумева аквизицију и обраду ЕЕГ сигнала испитаника помоћу програмског пакета у оквиру *Emotiv Cortex-a*. Помоћу добијених ЕЕГ сигнала током калибрације

спроводи се обука система машинског учења који процењује да ли је испитаник замишљао померај и с којим интензитетом. Процес калибрације дели се на два дела:

1. Калибрација визуализације мировања тродимензионалног геометријског облика: Од испитаника се тражи да се фокусира на представљено стање геометријског облика.
2. Калибрација визуализације одмака тродимензионалног геометријског облика од тачке гледишта испитаника: Од испитаника се захтева да изврши визуализацију како се тродимензионални геометријски облик убрзано удаљава од њега. Поред тога, кад геометријски облик дође до зида, испитаник мора да настави да замишља даље кретање.

Важно је напоменути да је потребно вршити ове две калибрације наизменично све док се не постигне задовољавајући ниво квалитета управљања. По завршетку калибрације испитаник је оспособљен да по команди помера геометријски облик и враћа га у стање мировања.

Процес учења се може посматрати из два угла, техничког и биолошког:

- Технички: Систем машинског учења се обучава помоћу ЕЕГ сигнала испитаника током калибрационог мерења. Врши се краћа и једноставнија обука с малим скупом података који се допуњује у свакој калибрационој сесији. По завршетку обуке, над самим скупом за обуку ради се валидација и добијање основних резултата који имају за циљ давање оцене система машинског учења као и повезаности *BCI* система.
- Биолошки: Испитаник покушава да посредно генерише ЕЕГ таласе (визуализацијом референтне вредности) који би пружили могућност обављања задатка и на тај начин долази до појаве неуропластичности, која има улогу да оспособи мозак за обављање те радње.

Кад се стручно лице које надгледа експеримент увери да је испитаник у стању да вољно помера геометријски облик и враћа га у стање мировања, од испитаника се захтева да у ограниченом времену што више пута узастопно изврши померај до граничног зида и врати у стање мировања. Стручно лице, обављајући ову проверу, добија увид о тренутном стању испитаникових вештина. У зависности од квалитета управљања, стручно лице одлучује да ли ће се прећи на следећи корак.

3.2.3 Мултимедијалне дидактичке вежбе

Мултимедијалне дидактичке вежбе представљају комплексан сет вежби специјално осмишљених за усавршавање својствености менталне концентрације. С намером да подрже процес учења и стимулишу неуропластичност код испитаника, ове вежбе укључују употребу висококвалитетних мултимедијалних симулатора. У својој актуелној верзији ови симулатори пружају аудио и визуелне стимулусе, јачајући тиме вишесензорни приступ.

Први у низу ових симулатора намењен је увежбавању одрживости менталне концентрације. Овај симулатор користи игру *Need for Speed: Payback* у научно-истраживачке сврхе и представља интерактивну игру вожње аутомобила, где испитаник контролише брзину аутомобила на основу интензитета своје визуелне менталне слике једнодимензионалног кретања. Кад је интензитет менталне слике изнад одређеног прага, аутомобил убрзава. Команде за скретање и кочење контролишу се ручно. Аутомобил којим испитаник управља приказан је из перспективе трећег лица, у оквиру реалистичне симулације градског саобраћаја. Снимак саме симулације се чува са снимком тродимензионалног мапирања мозга, што је приказано на слици 29.

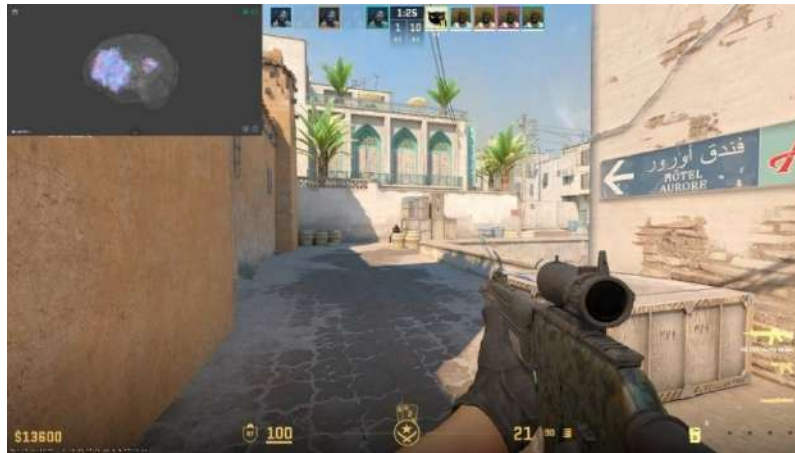
Ова вежба има за циљ да помогне усавршавање више аспекта менталне концентрације: отпорности, временске издржљивости, способности паралелног обављања задатака и отпорности на спољашње дистракције. На тај начин она пружа уникатну платформу за вежбање и развој вештина неопходних за одржавање сталне менталне концентрације на дужи временски период.



Слика 29 Снимак симулације вожње аутомобила уз приказ мождане активности

Следећи симулатор представља инструмент за увежбавање агилности менталне концентрације, који је приказан заједно са мапирањем мозга на слици 30. Осмишљен је као симулатор акционих маневара који учесника поставља у окружење првог лица, с реалистичном симулацијом окружења и интеракцијом с другим виртуелним лицима. Команда за пуцање везана је за интензитет визуализације менталне слике. Кад је овај интензитет изнад задатог прага, изабрано оружје започиње с пуцањем. Команде за кретање и померање нишана контролишу се ручно.

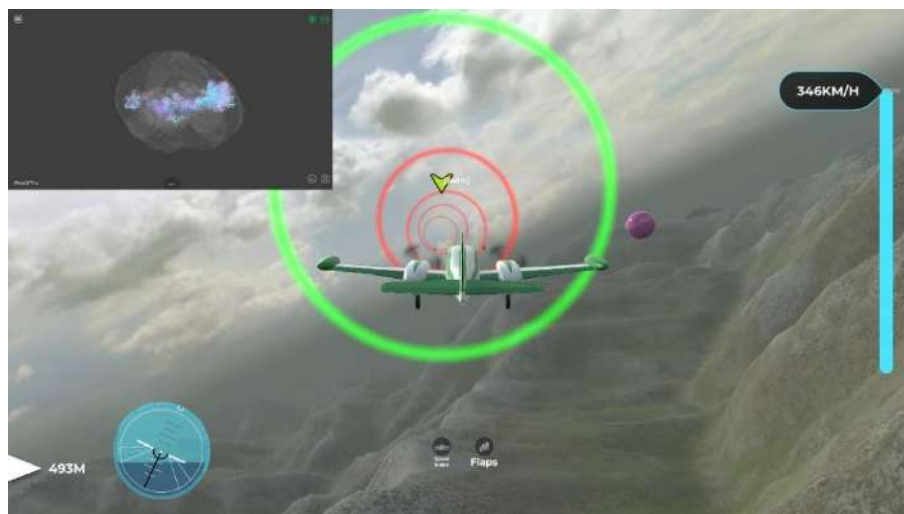
Овај симулатор је развијен с намером да подстакне развој способности за брзе промене менталне концентрације. Уз тестирање и вежбање брзине промене, он такође помаже у усавршавању вештина попут временске издржљивости менталне концентрације, способности за паралелно обављање различитих задатака и отпорности на спољашње дистракције. Кроз ову интерактивну и динамичну симулацију, учесници могу побољшати своју способност да брзо и ефикасно промене ниво своје менталне концентрације.



Слика 30 Снимак симулатора акционих маневара за агилност менталне концентрације

Следећи симулатор је дизајниран да подржи усавршавање контроле менталне концентрације. То је симулатор летења који приказује сценарије из перспективе трећег лица, с основним вежбама летења, као што су узлетање, слетање и одржавање жељене висине лета. Команда за контролу висине авиона везана је за интензитет визуализације менталне слике испитаника. Ако је интензитет превише низак, авион ће почети да понире, ако је превише висок, авион ће почети да се успиње. Ако је интензитет у дефинисаним границама, авион ће летети равно. Ручно се управља командама за скретање авиона.

Симулатор лета је развијен с циљем да подстакне учеснике у развијању вештина контроле менталне концентрације. Поред баланса, симулатор такође помаже у развоју вештина попут временске издржљивости менталне концентрације, способности за паралелно обављање различитих задатака и отпорности на спољашње дистракције. Кроз динамично и интерактивно окружење летења, учесници могу побољшати своју способност да вољно и прецизно контролишу своју менталну концентрацију.



Слика 31 Снимак симулатора летења за контролу менталне концентрације

3.2.4 Актуаторске неурофидбек вежбе

Актуаторске неурофидбек вежбе представљају последњи део експеримента у ком испитаник учествује. Овај корак обухвата најзахтевније активности у оквиру целокупног експеримента из угла техничког решења као и из угла психофизичких способности испитаника. Кад испитаници дођу до неопходних перформанси у оквиру мултимедијалних дидактичких вежби, које су описане у претходном поглављу, могуће је прећи на вежбе које ће им дозволити да увежбане когнитивне алгоритме користе у стварности, а не у симулацији.

Претходно наведена симулација, у којој испитаници управљају тродимензионалним летећим објектом, служи као припрема за управљање беспилотном летелицом – тркачким дроном. Дрон има шест степени слободе, што обезбеђује дванаест могућих команди испитаницима. Додатна безбедносна мера овде је дупли систем команди, који омогућава надзорнику да интервенише у случају потребе.

Задаци које испитаници добијају подељени су по тежинским категоријама, при чему надзорно лице одлучује на основу перформанси испитаника у којој категорији ће испитаник радити.

Прва категорија подразумева једнодимензионално померање уз контролу положаја и брзине. У овој категорији испитаник користи три менталне команде: стање мировања, кретање у позитивном смеру и кретање у негативном смеру.

Друга категорија подразумева дводимензионално кретање до задате тачке и по задатој путањи. У овој категорији испитаник користи пет менталних команди: стање мировања, кретање у позитивном и негативном смеру по две осе.

Трећа категорија представља тродимензионално кретање уз пролазак препрека, задатих тачака и путања уз контролу брзине и положаја. У овој категорији испитаник користи седам менталних команди: стање мировања, кретање у позитивном и негативном смеру по три осе.

Четврта категорија представља проширење претходне и поред транслације додаје могућност ротације дрона. У овој категорији испитаник може когнитивним командама користити све команде које дрон има. Од испитаника се тражи да пролази путање препреке и стазе које су на нивоу захтевности обуке за физичким управљањем дроном. Ова категорија је знатно тежа од претходне због тога што се правац дрона мења у складу с ротацијом, па је неопходно да се испитаник концентрише на разлику између апсолутног и релативног положаја као и команди. У овој категорији испитаник користи девет менталних команди: стање мировања, кретање у позитивном и негативном смеру по три осе, ротацију око вертикалне осе у позитивном и негативном смеру.

Пета категорија представља финални облик који обухвата високе перформансе лета као и контролу из првог лица (*FPV-drone*). Менталне команде које испитаник користи остају на истом нивоу као у претходној категорији, једина разлика је што испитаник мора бити у стању све да их користи из првог и трећег лица.

3.2.5 Тестирање и преглед испитаника

Циљ тестирања било је истраживање утицаја развијеног неурофидбек система на неуропластицитет особа које су учествовале у истраживању. С развојем модерних технологија и метода, нагласак је стављен на примену неурофидбека као инструмента који може потенцијално стимулисати и убрзати процесе неуропластицитета. Коришћењем савремене опреме као што су ЕЕГ и мапирање

мозга, у комбинацији с различитим психолошким тестовима, намера је била да се добије дубоки увид у то како индивидуализоване сесије неурофидбека могу утицати на структурне и функционалне промене у мозгу, и на који начин те промене корелирају с когнитивним способностима учесника.

У почетној фази фокус је био на униформној групи студената узраста од 22 до 24 године, сви с истим усмерењем и на истој години основних академских студија. Ова група од 20 студената послужила је као основа за пилот-истраживање, где је сваки учесник одабран на основу својих академских обавеза. Ова тестирања су спроведена у оквиру истраживања мастер рада под називом "Развој мерно-информационог система за истраживање когнитивних појава заснованог на мерењу ЕЕГ сигнала".

Процедура тестирања у истраживању методично је разрађена и представља интегрисани приступ оцени ефеката неурофидбека на неуропластицитет. Процес тестирања подразумева неколико кључних компонената. Прво, функционални тестови су спроведени за проверу способности испитаника да ефикасно извршавају задатке. Ови тестови су спроведени на почетку и на крају сваке сесије. Затим, општи психолошки тестови су коришћени за процену когнитивних способности као што су пажња, визуализација, концентрација и памћење. Ови тестови су спроведени на крају сесије, у пратњи стручних коментара надзорног лица. На крају, коришћена су неурофизиолошка мерења кроз ЕЕГ и мапирање мозга како би се добио увид у стање централног нервног система током целог експеримента.

Две групе су формиране ради додатне верификације налаза – експериментална група, која је користила развијени неурофидбек систем, и контролна група, која је извршавала исте задатке без употребе неурофидбека. Ова двострука стратегија омогућила је праћење разлике у начину и брзини учења између две групе.

Алати и ресурси који су тренутно коришћени у истраживању обухватају савремене хибридне ЕЕГ уређаје као што су *Emotiv Epos* и *Emotiv Insight*. Подаци добијени помоћу ових уређаја се анализирају помоћу софтвера *Emotiv PRO* и *EDFBrowser*, што омогућава стручњацима да прегледају и обраде ЕЕГ сигнале. С обзиром на важност тачности и интегритета података, у будућности ће се такође користити и клинички ЕЕГ уређаји како би се омогућила додатна валидација резултата од стране стручњака. Поред тога, у истраживању се користе лиценцирани психолошки тестови за процену и праћење психофизичког стања учесника. Интерпретацију и процену воде су стручњаци из релевантних области, укључујући психологе, психијатре и лекаре-специјалисте.

3.2.6 Место истраживања

Истраживања су спровођена у следћим лабораторијама:

- Лабораторија за биомедицинско инжењерство и инструментацију, Факултет техничких наука, Универзитет у Новом Саду;
- Лабораторија за мерно-информационе системе, Факултет техничких наука, Универзитет у Новом Саду;
- Лабораторија за метрологију, Факултет техничких наука, Универзитет у Новом Саду;
- Лабораторија за експерименталну психологију, Филозофски факултет, Универзитет у Бањој Луци.

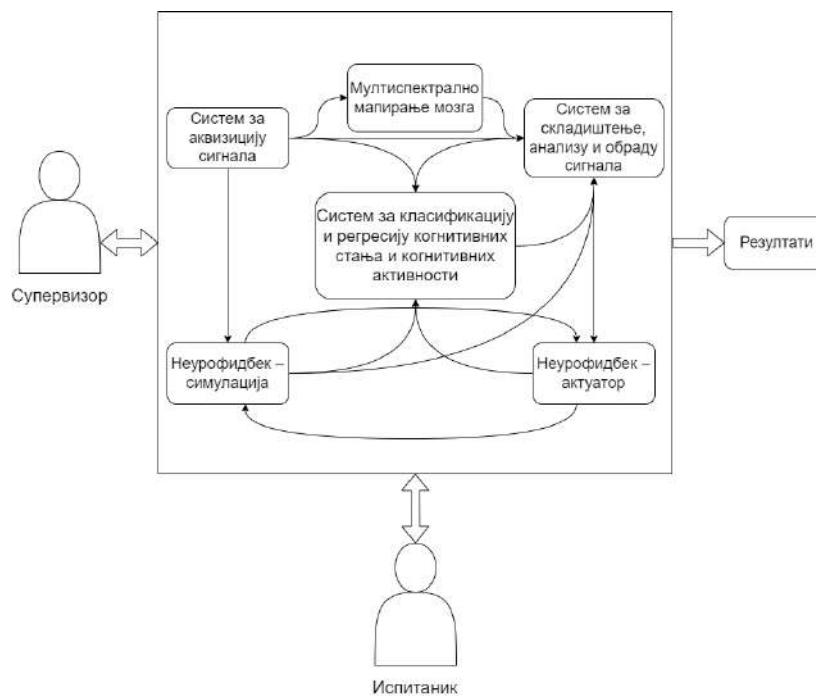
3.2.7 Спроведени експерименти

Експерименти описани у овом поглављу су осмишљени и спроведени током истраживања за мастер рад под називом "Развој мерно-информационог система за истраживање когнитивних појава заснованог на мерењу ЕЕГ сигнала" из 2018. године. Ови експерименти су били усмерени на испитивање потенцијала визуелних менталних слика у управљању различитим системима, као и на процену кључних аспеката менталне концентрације, укључујући одрживост, агилност, контролу, временску издржљивост, способност паралелног обављања задатака и отпорност на спољашње дистракције.

Иако су експерименти над испитаницима били ограничени на период мастер студија, резултати и закључци добијени током тих истраживања били су од суштинског значаја за развој и постављање темеља за креирање платформе *Mentis Oculi*. Ова платформа представља интегрисани систем који може обављати све активности које су биле предмет експеримената, али на свеобухватнији и ефикаснији начин. *Mentis Oculi* платформа је осмишљена да омогући синхронизовано управљање и праћење когнитивних параметара, интегришући све аспекте које су раније били истраживани у појединачним експериментима.

3.3 Архитектура система

Узимајући у обзир специфичне захтеве који су наметнути методологијом експеримента, архитектура система осмишљена је тако да испуни све неопходне функције. Систем је развијен с циљем да обезбеди све тражене функционалности, као и једноставан и разумљив приказ података, ток експеримента и резултата, који ће бити доступни свим члановима мултидисциплинарног тима. Овај тим укључује психологе, психијатре, стручњаке за анализу података, софтверске инжењере, као и биомедицинске инжењере. У наредним поглављима сваки аспект архитектуре система биће детаљно размотрен, а у наставку се налази слика 32 с графичким приказом целокупне архитектуре.



Слика 32 Архитектура система

3.3.1 Систем за аквизицију сигнала

Систем за аквизицију сигнала развијен је као две компоненте, користећи програмске језике *Python* и *C#*. Обе компоненте имају различите примене и функционалности у оквиру пројекта, задовољавајући разноврсне захтеве и потребе. На самом почетку потребно је прикупити иницијалне податке који ће бити сачувани заједно с подацима добијеним током експеримента. Иницијални подаци укључују основне информације о испитанику, као и специфичне податке везане за сам експеримент.

Прва компонента, која користи *Python*, омогућава аквизицију сигнала у реалном времену, као и асинхрону аквизицију. Овај приступ је користан за прикупљање података без повратне спреге, као што су праћење биофизичких параметара, тестирање и калибрација, као и за неурофидбек системе с физичким излазом. У оквиру ове компоненте, имплементација система обезбеђује један улаз и два излаза: један за актуаторе с физичким излазом и други за чување података. Учитавање улазних података обавља се помоћу *Emotiv Cortex* програмског пакета за програмски језик *Python*. Ако је потребно проследити податке излазу за актуаторе, на основу одговарајућих сигнала израчунавају се актуаторске вредности и преносе до актуаторског система путем *USB Serial Port* везе. У случају самог читавања података, систем почиње с прикупљањем и складиштењем података у локалним променљивима, да би их на крају експеримента сачувао заједно с иницијалним подацима. Подаци се могу сачувати у *json*, *edf* и *CSV* форматима.

С друге стране, друга компонента система, која користи *C#* програмски језик, фокусирана је на аквизицију сигнала у реалном времену за симулационе неурофидбек системе, као и за психолошке тестове. Ова варијанта омогућава брзу и ефикасну обраду и пренос података, што је кључно за успешну реализацију ових апликација.

Ова компонента има за циљ пружање неопходних података за апликације и дизајнирана је да има један улаз и један излаз. На улазу, читавање података се врши помоћу *Emotiv* пакета за *C#* програмски језик, док се пренос података обавља комуникацијом између модула.

Због једноставности имплементације, прва компонента се користи у случајевима кад је потребно извршити технички једноставније задатке, као што су асинхрона мерења или прослеђивање вредности. Друга компонента се користи кад је потребно употребити комплексније рачунарске симулације. У складу с тим, развијен је и остатак целокупног система, који се заснива на излазима система за аквизицију сигнала.

Тако развијен и структуриран, систем за аквизицију сигнала обавља све неопходне функције у оквиру пројекта, усаглашавајући се с потребама и захтевима различитих апликација и база података. Управо због тога, систем представља веома важан елемент пројекта, који омогућава ефикасну и прецизну обраду и пренос података, адаптирану конкретним потребама и захтевима разноврсних примена. У обе варијанте система за аквизицију сигнала, *Python* и *C#*, приоритет је поузданост и прецизност у прикупљању и обради података. Дизајн и архитектура система омогућавају laku интеграцију с различитим апликацијама и базама података, као и синхронизацију с другим компонентама пројекта. Овај систем такође обезбеђује оптималну перформансу и стабилност како би истраживачки тим могао да се фокусира на анализу и интерпретацију резултата.

Додатно, систем за аквизицију сигнала обезбеђује подршку за разноврсне сензоре и уређаје, укључујући електроенцефалографске (EEG) сензоре, електромиографске (EMG) сензоре и друге

биомедицинске сензоре. Ово омогућава већу флексибилност у избору опреме и метода за прикупљање података, што може допринети квалитету и обиму истраживања.

Систем за аквизицију сигнала представља кључну компоненту пројекта, базирану на савременим технологијама и иновативним решењима. Овај систем омогућава истраживачком тиму ефикасно и прецизно прикупљање и анализу података, уз истовремену примену интегрисаних неурофидбек система и других психолошких тестова.

3.3.2 Систем за складиштење, анализу и обраду сигнала

На основу самог имена овог система, закључује се да је његова улога подељена на две основне целине, прву представља складиштење, док другу представља модул за анализу и обраду.

Модул за складиштење података увезује и комуницира са свим системима и већином модула у оквиру целокупног система. Његов кључни задатак је да изврши упис, измене и ажурирања свих података који су добијени из свих обављених експеримената.

Предност мерења ЕЕГ сигнала са *Emotiv* системом доводи до тога да можемо да синхронизовано преузимамо и организујемо податке с више уређаја и с више рачунара за одговарајуће системе, при чему пермутације између испитаника, уређаја, експеримената и руководиоца не ремете процесе модула.

3.3.3 Неурофидбек симулација

Овај систем представља један од кључних алата с којима се срећу корисници у свом раду и обуци. Као један од првих корака употребе целокупног система, симулациони неурофидбек има више кључних улога од којих зависи остатак прогреса експеримената као и напретка испитаника: упознавање, припрема, обука, усавршавање и тестирање. Архитектура система и сви његови делови су приказани на слици 33.



Слика 33 Архитектура неурофидбек симулатора

3.3.3.1 Мерење ЕЕГ сигнала

Врши се употребом система за аквизицију сигнала описаног у потпоглављу 4.1 и његов задатак је да обезбеди ЕЕГ одбирке у што краћем року.

3.3.3.2 Обрада ЕЕГ сигнала

У овом кораку се прикупља скуп ЕЕГ одбирака и врши се њихово програмско филтрирање након чега се прослеђују даље.

3.3.3.3 *Анализа ЕЕГ сигнала*

Над обрађеним ЕЕГ подацима врши се анализа помоћу следећих корака:

1. **Примена *Short Time Fourier Transform (STFT)***: Примена STFT на ЕЕГ сигнал омогућава анализу фреквенцијске структуре сигнала у различитим временским интервалима. STFT примењује Фуријеову трансформацију на мале временске сегменте сигнала, чиме се добија фреквенцијски спектар сигнала у различитим временским тачкама.
2. **Добијање средњих вредности за фреквенцијске опсеге**: Након примене STFT, добијени сигнал се додатно анализира. За релевантне фреквенцијске опсеге, као што су делта, тета, алфа, бета и гама, израчунавају се средње вредности амплитуда у тим опсегима.
3. **Организација сигнала**: Добијене вредности из претходног корака слажу се у одговарајуће нивозе, након чега су спремни за следеће кораке.

3.3.3.4 *Процена когнитивне активности и когнитивних стања*

Након добијања свих неопходних података, приступа се процени когнитивних активности и когнитивних стања помоћу свих претходно наведених сигнала.

3.3.3.5 *Симулациони приказ*

Симулациони приказ обухвата све што испитаник види током експеримента. Симулације могу бити дводимензионалне и тродимензионалне, док се тродимензионалне додатно деле на виртуелну реалност и конвенционалне мониторе.

У свакој од наведених симулација параметри се прилагођавају испитанику на основу његових података, биофизичких параметара, као и на основу његових перформанси у току саме симулације.

Параметри који се мењају у току симулације могу се поделити на пасивне и активне, у зависности од тога како испитаник утиче на њих. Пасивни параметри представљају оне вредности које се мењају без директног утицаја испитаника на њих и од испитаника се не тражи да свесно утиче на њих, већ се само упућује на који начин се мењају. Активни параметри могу бити когнитивни и физички, они представљају вредности помоћу којих испитаник контролише симулацију.

3.3.3.6 *Mentis Oculi сусџем*

Систем *Mentis Oculi* развијен је као одговор на потребу за покретањем неурофидбек симулација које омогућавају синхронизовано праћење когнитивних параметара, као и свих акција унутар симулације. Симулација у оквиру система развијена је коришћењем *Unity Engine-a* и *Emotiv API-ja*, што омогућава мерење, прикупљање, обраду и анализу ЕЕГ података. Когнитивне активности које систем *Mentis Oculi* користи за управљање се могу описати као менталне слике једнодимензионалних вектора у тродимензионалном простору, оне укључују транслације и ротације у свим правцима. Овај систем омогућава унапређење различитих когнитивних феномена кроз вежбање, праћење и анализу когнитивних активности. У прилогу 9.2.1 се налази код за *Unity* симулацију, а алгоритам 1 описује псеудокод саме платформе.

Алгоритам 1: Псеудокод система *Mentis Oculi*

Иницијализација система

Учитавање основних подешавања и приказ GUI-а за избор испитаника.

Регистрација и избор испитаника

Уколико је у питању нови испитаник, унос података и додела ID-а.

Уколико је у питању постојећи испитаник, учитавање претходно унетих података.

Избор уређаја за симулацију

Препознавање доступних уређаја

Одабир симулационог уређаја: "Рачунар" или "VR".

Иницијализација система у складу са изабраним уређајем.

Подешавање параметара симулације

Одабир трајања експеримента, величине мапе и броја објеката за сакупљање.

Потврда одабраних параметара.

Подешавање контроле објекта

Постављање свих праваца кретања у једно од четири могућа стања: "Искључено", "Ручно", "Ментално" или "Комбиновано".

Покретање симулације

Извршење симулације са дефинисаним параметрима.

Мерење EEG сигнала, процена когнитивних стања, мапирање мозга, праћење параметара симулације и снимање фацијалних експресија.

Завршетак симулације и чување података

По завршетку симулације, чување свих података у формату: ID_ispitanika_oblik_testa_datum_vreme.

Могућност извоза података у изабраном формату (csv, json, edf, mkv).

Могућност покретања нове симулације са истим или новим испитаником, са истим или новим условима и истим или другим уређајем.

Затварање система

Затварање GUI-а и излазак из система.

Систем обухвата неколико основних функционалности, почевши од регистрације испитаника, калибрације неурофидбека и одабира опција за праћење когнитивних феномена. У почетној фази, испитаник уноси своје податке или бира свој идентификациони број у случају да је већ регистрован у систему. Након тога, систем омогућава избор уређаја за симулацију, где се препознају доступни уређаји као што су "Рачунар" или "VR". На слици 34 приказана је конфигурација система за коришћење са рачунаром и VR уређајем на слици 35.

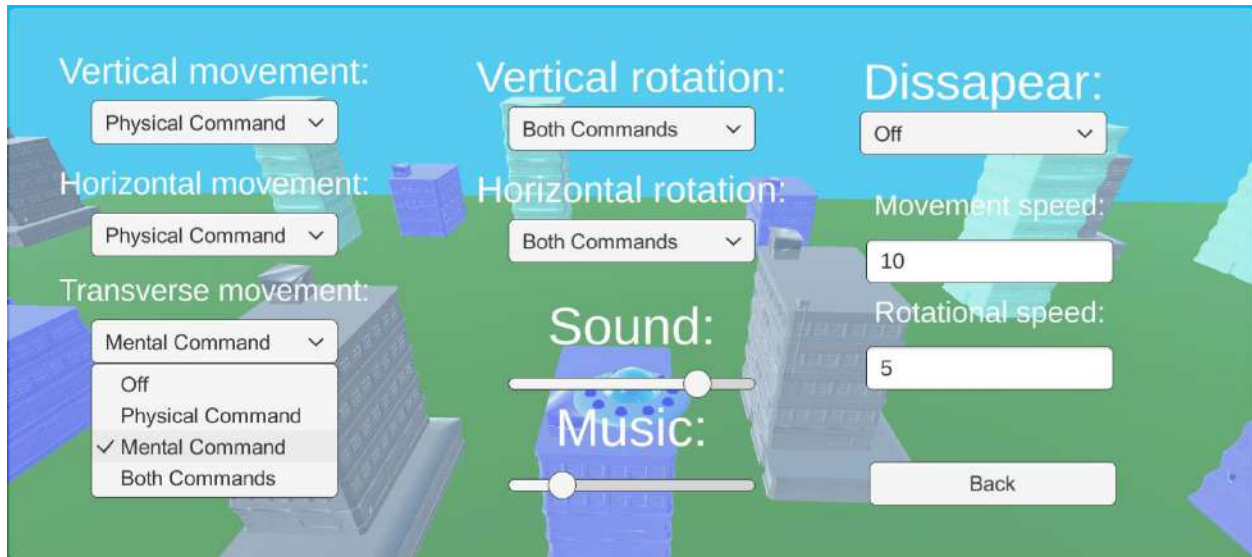


Слика 34 Конфигурација сисџема са рачунаром



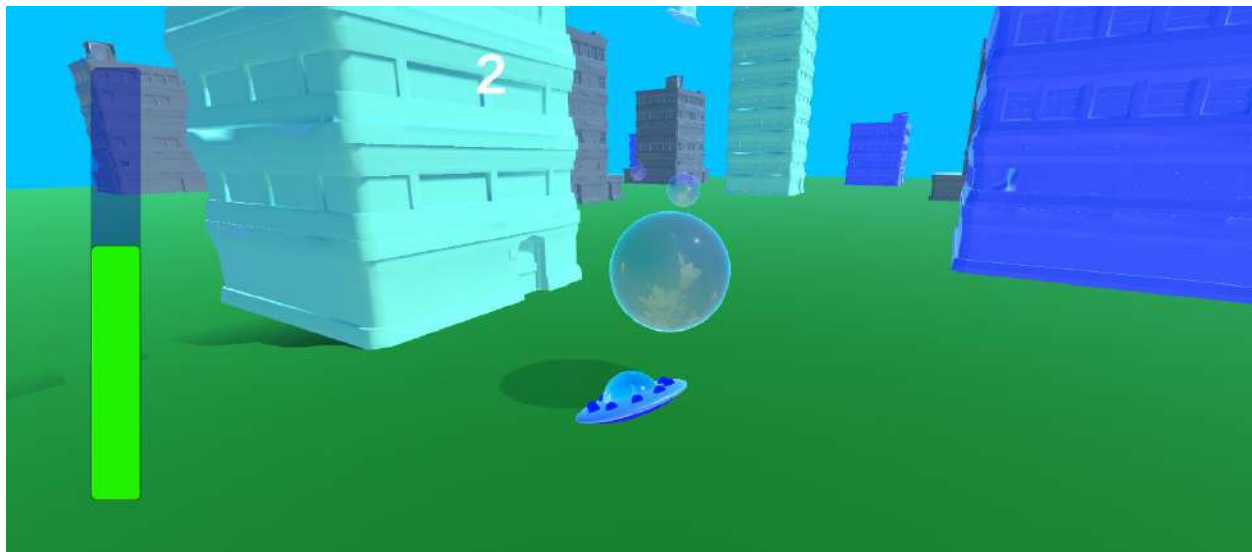
Слика 35 Конфигурација сисџема са VR уређајем

Након избора уређаја, корисник приступа подешавању параметара симулације, где се бирају трајање експеримента, величина мапе и број објеката за сакупљање. Поред тога, корисник има могућност да подеси начин контроле објекта у симулацији, где сваки правац кретања може бити постављен на једно од четири могућа стања: "Искључено", "Ручно", "Ментално" или "Комбиновано", што је приказано на слици 36.



Слика 36 Подешавања симулације

Када су сви параметри подешени, покреће се симулација која омогућава истовремено мерење EEG сигнала, процену когнитивних стања и праћење параметара симулације. Симулација обухвата контролу објекта у тродимензионалном простору, где се објекат може контролисати из перспективе трећег лица или у оквиру виртуелне стварности, као што је приказано на сликама 37 и 38. Објекти у симулацији, укључујући контролисани објекат у стању мировања, објекте који се сакупљају и декоративне елементе као што су облаци, не поседују стварно кретање, већ имају анимације које имитирају покрет. Ове анимације имају за циљ да помогну испитанику у формирању реалније менталне слике тродимензионалних објеката.



Слика 37 Симулација из трећег лица



Слика 38 Симулација из првог лица

По завршетку симулације, сви прикупљени подаци се чувају у формату који укључује *ID* испитаника, облик теста, датум и време извршења, овај процес омогућава детаљну анализу података. Сви подаци се могу извести у различитим форматима као што су *csv*, *json*, *edf* или *tkv*. Слика 39 приказује исечак снимка експеримента, где се може видети симулација, фацијална експресија испитаника и мапирање мозга.



Слика 39 Снимак симулације, фацијалних експресија и мапирања мозга

3.3.4 Неурофидбек актуатор

Комбиновањем *Arduino* платформе и програмског језика *Python*, систем омогућава испитанику да управља дроном користећи своје когнитивне активности. Овај систем представља иновативан и интерактиван начин за испитивање можданих сигнала и примену ЕЕГ технологије у области когнитивних наука и *BCI* система. Архитектура система је приказана на слици 40.



Слика 40 Архитектура неурофидбек актуатора

3.3.4.1 Мерење ЕЕГ сигнала

Као што је описано у поглављу 4.1 Систем за аквизицију сигнала, мерење се врши помоћу једноставнијег *Python* модула који ради мање захтевну обраду података уз њихово чување у бази података за одговарајући експеримент и испитаника.

3.3.4.2 Обрада ЕЕГ сигнала

Узимајући у обзир да је систем морао бити оптимизован за рад у реалном времену, извршена обрада података обухвата само операције које имају за циљ добијање неопходних података за рад и функционалност неурофидбек система.

3.3.4.3 Анализа ЕЕГ сигнала

Анализа ЕЕГ сигнала обавља се ретроактивно тек по завршетку експеримента и складишти се с остатком података добијених у току експеримента.

3.3.4.4 Процена когнитивне активности

На основу обрађеног ЕЕГ сигнала, систем машинског учења врши процену когнитивне активности и њено прослеђивање.

3.3.4.5 Хардверско повезивање

У циљу прослеђивања процењене когнитивне активности до актуатора, *Python* апликација се повезује с *Arduino* развојном плочом путем USB серијске конекције. Прослеђена порука састоји се из два дела – први говори о томе која когнитивна активност је у питању, а други о њеном интензитету.

3.3.4.6 Актуаторско управљање

Arduino развојна плоча врши претварање добијеног сигнала са серијског порта у одговарајући напон. У зависности од физичке изведбе актуаторског управљања, зависи облик излазног сигнала из *Arduino UNO* развојне плоче. Тренутна имплементација система реализована је као пилот-

експеримент и има за циљ тестирање управљања микродроном, блок шема је приказана на слици 41. Даљински контролер микродрона је преправљен тако да су напони на потенциометрима за контролу дрона замењени *PWM* сигнаlima *Arduino* плоче који су конвертовани у аналогни сигнал помоћу нископропусног филтра.

Кораци од 3.3.4.1 до 3.3.4.4 су обухваћени у следећем алгоритму:

Алгоритам 2: Псеудокод имплементације мерења, обраде, анализе и процене когнитивних активности

Конфигурација серијског порта

Подеси серијски порт (*serial_port*) на */dev/ttyS6*.

Подеси брзину преноса (*baud_rate*) на 9600.

Отвори серијски порт користећи конфигурацију.

Конфигурација Emotiv API

Постави *client_id* и *client_secret* на одговарајуће вредности (празне за овај пример).

Постави URL за Cortex API WebSocket на *wss://localhost:6868*.

Иницијализација варијабли за покрет

Иницијализуј променљиве за покрет:

```
movement = {"LX": 0, "LY": 0, "RX": 0, "RY": 0}
```

Функција за мапирање снаге на опсег -50 до 50

```
def map_power_to_range(power):
```

Конвертуј *power* у опсег од -50 до 50 и врати вредност.

Функција за слање података о покрету преко серијског порта

```
def send_movement():
```

Конструиши поруку у формату *LX_NNN_LY_NNN_RX_NNN_RY_NNN*.

Шаљи поруку преко серијског порта.

Обрада WebSocket порука

```
def on_message(ws, message):
```

Конвертуј поруку у JSON формат.

Ако порука садржи менталне команде:

Извлачи команду и снагу команде.

Мапирај снагу команде у опсег од -50 до 50.

Ако је команда "push", постави *RY* на мапирану вредност.

Ако је команда "pull", постави *RY* на негативну мапирану вредност.

Ако је команда "rotate_left", постави *LX* на негативну мапирану вредност.

Ако је команда "rotate_right", постави *LX* на мапирану вредност.

Ако је команда "lift", постави *LY* на мапирану вредност.

Ако је команда "drop", постави *LY* на негативну мапирану вредност.

Ако је команда "left", постави *RX* на негативну мапирану вредност.

Ако је команда "right", постави *RX* на мапирану вредност.

Пошаљи ажуриране податке о покрету преко серијског порта.

Функције за обраду WebSocket грешака и затварања

```
def on_error(ws, error):
```

Испиши поруку о грешци.

```
def on_close(ws, close_status_code, close_msg):
```

Испиши поруку о затварању везе.

Функција за отварање WebSocket везе

```
def on_open(ws):
    Исписи поруку о отварању везе.
    Пошаљи захтев за ауторизацију са client_id и client_secret.
    Пошаљи захтев за претплату на менталне команде и когнитивне податке.
    Иницијализација:
        Постави LY и RY на -50 и пошаљи податке о покрету.
        Чекај 1 секунду.
        Постави LY и RY на 50 и пошаљи податке о покрету.
        Чекај 1 секунду.
        Постави LY и RY на 0 и пошаљи податке о покрету.
```

Покретање WebSocket апликације

Омогући WebSocket трагове.
Креирај и покрени WebSocket апликацију са горе наведеним функцијама.
Затвори серијски порт на крају извршавања.

Имплементација алгоритма се налази у прилогу 8.2.1. Алгоритам 3 описује претварање команди за дрон у одговарајуће вредности *PWM* напона, чија је имплементација дата у прилогу 8.2.2 и блок шема приказана на слици 35.

Алгоритам 2: Псеудокод имплементације претварања команди за дрон у одговарајуће напоне

Иницијализација Servo објекта и PWM вредности

Дефиниши Servo објекте за леви и десни хоризонтални и вертикални помак.
Постави минималне, максималне и централне PWM вредности.

Подешавање и иницијализација

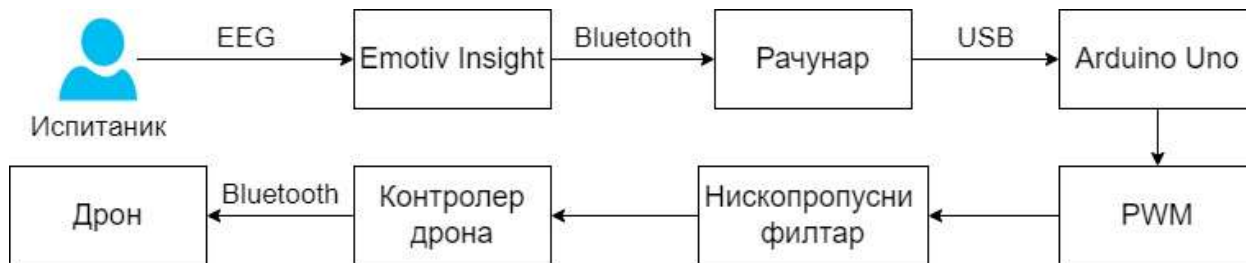
```
void setup():
    Постави брзину серијске комуникације на 9600.
    Додели пинове за Servo објекте.
    Постави почетне PWM вредности на централне.
```

Главни програмски ток

```
void loop():
    Ако постоје доступни серијски подаци:
        Прочитај серијску поруку до новог реда.
        Извлачи вредности за LX, LY, RX и RY.
        Мапирај и постави PWM вредности за сваки од Servo објекта.
```

Функција за извлачење вредности из серијске поруке

```
int extractValue(String data, String key):
    Извлачи вредност за дати кључ из серијске поруке и враћа ту вредност.
```



Слика 41 Блок шема актиуајорској неурофидбек сисџема

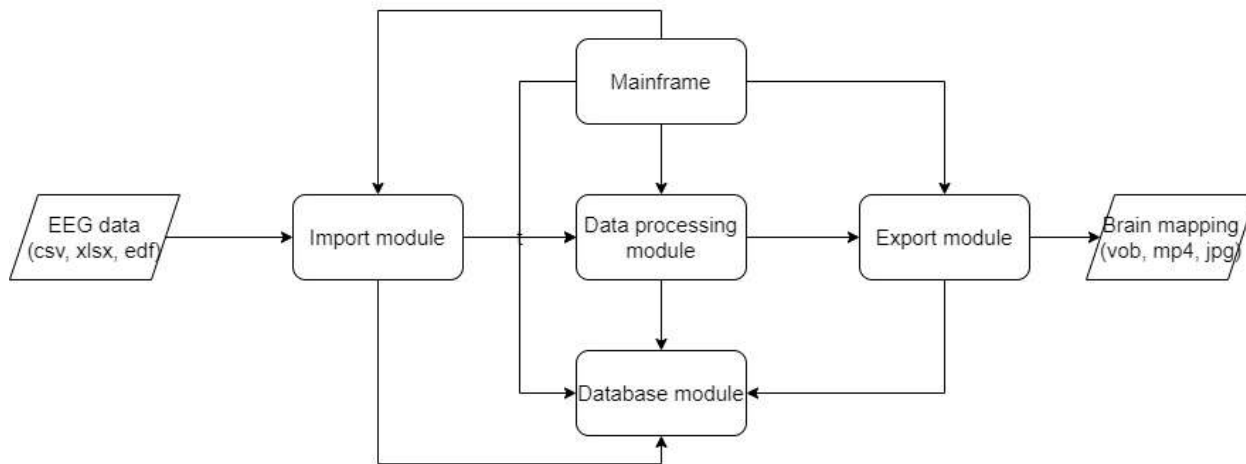
3.3.5 Мултиспектрално мапирање мозга

У сврху детаљнијег истраживања когнитивних појава, развијен је свеобухватан алат за читавање, обраду, анализу, графички приказ, чување и извоз ЕЕГ података. Захваљујући овом алату, истраживачи имају олакшану и прилагођену анализу ЕЕГ података, што им омогућава идентификацију и квантификацију образаца неуралне активности повезане с одређеним когнитивним процесима.

Слика 42 илуструје детаљну архитектуру развијеног алата, укључујући различите модуле. Сваки модул је дизајниран да обавља одређени скуп задатака, и подаци се преносе између модула да би се обезбедио гладак и ефикасан рад.

Алат је развијен користећи програмски језик *Python*, који пружа флексибилну платформу за научно рачунање и анализу података. Употребљено је неколико библиотека отвореног кода за имплементацију различитих функционалности алата, укључујући *scikit-learn*, *pandas*, *numpy*, *pickle*, *CSV*, *matplotlib*, *MNE* и *fooof*.

У целини, развијени алат пружа истраживачима погодан и свеобухватан интерфејс за анализу ЕЕГ података, омогућавајући им да ефикасније истраже и тумаче сложене појаве неуралних активности повезаних с когнитивним процесима. Алат има потенцијал да олакша нова открића у области когнитивне неуронауке и да прошири наше разумевање људског мозга.



Слика 42 Софтверска архитектура алата за мапирање мозга

Систем представљен у овом поглављу служи за анализу и валидирање добијених резултата неурофидбек система. Његовом употребом могуће је детектовати когнитивне активности које нису видљиве конвенционалним методама мапирања мозга. Због своје флексибилности и компатибилности, систем је применљив у експериментима из области когнитивних неуронаука, неурологије, психологије и биомедицинског инжењерства.

3.3.5.1 Централни модул

Централни модул је главна процесна јединица развијеног алата. Његова примарна улога је да издаје команде другим модулима и да извештава о напретку, статусу и резултатима корисника. Модул ради на основу постављених параметара, које обезбеђује корисник или систем. Команде се шаљу одговарајућим модулима за обраду података, анализу, графички приказ, чување и извоз. Централни модул такође комуницира с базом података ради добављања и ажурирања сачуваних информација. У тренутној имплементацији, са централним модулом може се бити у директној интеракцији, али се у будућности планира модул за кориснички интерфејс који ће вршити интеракцију са централним модулом и чинити га приступачнијим за корисника.

3.3.5.2 Модул за увоз података

Систем је дизајниран да буде аутоматизован, омогућавајући да се након примања команде од централног модула покрене процес читавања датотека за обраду података. Током процеса читавања, систем аутоматски препознаје број и назив канала из ЕЕГ података, као и остале релевантне параметре, као што су учесталост узорковања, ознаке, анотације и метаподатке, који су важни за даљу анализу ЕЕГ података.

Једном кад су ЕЕГ подаци успешно учитани, систем потом шаље податке модулу за обраду података на анализу. Модул за обраду података одговоран је за широк спектар задатака, укључујући филтрирање сигнала, екстракцију карактеристика, уклањање артефаката и статистичку анализу, између осталог. Обрађени подаци затим се шаљу бази података, која је одговорна за чување свих релевантних информација и резултата анализе.

Аутоматским препознавањем релевантних параметара и без прекида слањем података између модула, систем своди ризик од грешака на минимум и обезбеђује тачност резултата анализе.

3.3.5.3 Модул за обраду података

Модул за обраду података кључна је компонента развијеног алата, одговорна за филтрирање и анализу ЕЕГ података. Модул прима улазне податке из увозног модула и на основу команди примљених из централног модула, примењује одговарајуће филтере за препроцесуирање података. Филтери могу бити прилагођени захтевима корисника, као што су нископропусни, високопропусни или тракастопропусни филтери, да би се елиминисао нежељени шум или артефакти из сигнала.

Након филтрирања модул за обраду података врши анализу фреквенција сигнала и генерише дводимензионални или тродимензионални графички приказ активности мозга. Графички приказ показује јачину сигнала у различитим фреквенцијским опсезима, представљеним различитим бојама, у складу са 10-20 стандардом. Током обраде добијени резултати се континуирано приказују на екрану, омогућавајући кориснику да прати напредак анализе.

По завршетку, добијени резултати мапирања мозга чувају се у бази података за даљу анализу и поређење с другим резултатима. Модул је високоприлагодљив и пружа могућност избора обраде и приказа свих шест ЕЕГ спектара, укључујући делта, тета, алфа, бета, гама и му, чиме се обезбеђује целовит преглед изучаваних когнитивних појава.

3.3.5.4 База података

Модул базе података има кључну улогу у развијеном алату, јер служи као централизована јединица за чување свих података обрађених системом. Сачувани подаци су доступни и могу се додати у остале модуле према потреби. Да би се обезбедиле оптималне перформансе и компатибилност, база података је имплементирана помоћу две стандардне *Python* библиотеке - *CSV* и *pickle* - омогућавајући чување података и у формату вредности раздвојених зарезима (*CSV*) и у изворном *Python* формату серијализације (*pickle*). Поред тога, подацима сачуваним у бази података може се приступити и офлајн и онлајн преко *cloud* складишта.

Како би се омогућило ефикасно добављање и управљање подацима, архитектура базе података је дизајнирана да архивира и ажурира податке према ИД испитаника, ИД експеримента, ИД теста и ИД датотеке. Осим тога, процес ажурирања се аутоматски покреће након уноса сваког записа, обезбеђујући да најновији подаци увек буду доступни другим модулима. На крају, модул базе података је интегрисан и беспрекорно комуницира са свим другим модулима приказаним на слици 1, омогућавајући глатак и непрекидан ток података кроз читав систем.

3.3.5.5 Модул за извоз података

Модул за извоз представља последњи корак у процесу обраде података. Његова главна функција је извлачење података из базе података и смештање на жељену локацију на хард-диску. Тренутно, систем подржава само локално складиштење и извоз, али будуће верзије алата подржаваће и *cloud* складишна решења. То ће омогућити истраживачима да приступе својим подацима из било ког дела света и да сарађују с колегама на даљину.

Модул за извоз активира централни модул, који шаље команду за извлачење података за одређеног испитаника или експеримент. Једном кад су подаци извучени, модул их сачува у датотеку у формату који је одредио корисник. Подржани формати укључују *CSV*, *Excel*, *MATLAB*, *vob*, *tr4* и *jpg* датотеке.

Модул за извоз осигурава да су извезени подаци добро организовани и лако разумљиви. Узима у обзир ИД испитаника, ИД експеримента, ИД теста и ИД датотеке како би се обезбедило да се извезени подаци лако могу повезати с извором. Ова функција је посебно корисна за истраживаче који морају да прате велике количине података из више експеримената и испитаника.

3.3.6 Систем за класификацију и регресију когнитивних стања и когнитивних активности

Овај сегмент система је посвећен накнадној обради података добијених кроз аквизицију сигнала. У овом контексту његова примарна функција је вршење независне процене когнитивних стања испитаника употребом програмског језика *Python*. Модул је дизајниран да може примити било које стандардизоване ЕЕГ податке, без обзира на то да ли долазе од *Emotiv*-а или неког другог ЕЕГ система.

Подаци се читавају из *EDF* фајлова који садрже ЕЕГ снимке испитаника. Уз помоћ модула *mne* учитавају се и обрађују подаци. За сваки ЕЕГ сигнал примењује се *Short Time Fourier Transform (STFT)*

да би се добили спектрограми свих канала. Добијени спектрограми се даље анализирају кроз различите фреквенцијске опсеге: делта (од 0,5 до 4 Hz), тета (од 4 до 8 Hz), алфа (од 8 до 12 Hz), бета (од 12 до 30 Hz) и гама (од 30 до 45 Hz). На основу ових фреквенцијских опсега израчунава се средња вредност спектралне снаге унутар сваког опсега, што представља кључна обележја за улаз развијеног система. Сакупљени и обрађени подаци нормализују се користећи *StandardScaler* из „*sklearn.preprocessing*“ пакета. Нормализовани подаци се даље користе за тренирање конволутивне неуралне мреже (*CNN*) користећи *TensorFlow* и *Keras* библиотеке.

Модел је саграђен као секвенца слојева, укључујући два конволутивна слоја, два слоја за максимално агрегирање и три пуно повезана слоја. За тренирање модела користи се адам оптимизатор, а функција губитка је бинарна унакрсна ентропија. Слика 43 приказује све дефинисане кораке које систем за класификацију и регресију обавља.

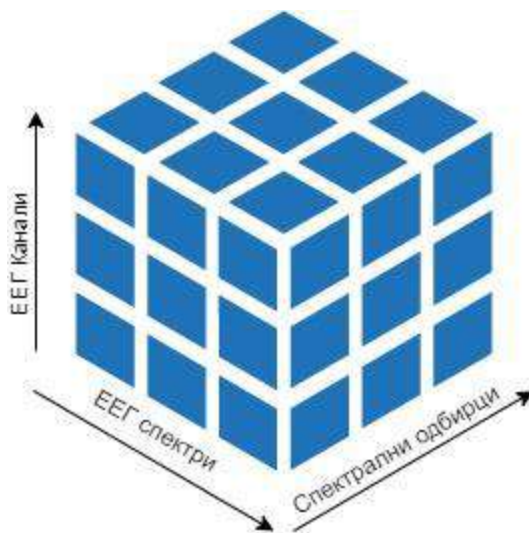


Слика 43 Кораки система за класификацију и регресију

3.3.6.1 Систем за класификацију

Овај сегмент система врши класификацију когнитивних стања испитаника на основу података добијених из EEG сигнала. За развој апликације коришћен је програмски језик Python, уз примену различитих библиотека и модула за обраду сигнала и машинско учење.

Подаци се очитавају из EDF фајлова који садрже EEG снимке испитаника. За учитавање и обраду података коришћен је модул mne, који омогућава ефикасно руковање EEG подацима. Сваки EEG сигнал подвргнут је *Short Time Fourier Transform (STFT)* како би се добили спектрограми свих канала. Спектрограми су затим обрађени у фреквенцијским опсезима који су наведени на почетку поглавља 3.3.6, а организација података је приказана на слици 44 [129].



Слика 44 Организација података

За сваки фреквенцијски опсег израчуната је средња вредност спектралне снаге, што представља кључна обележја која се користе као улазни подаци за класификациони модел. Добијени подаци су нормализовани коришћењем `StandardScaler` из `sklearn.preprocessing` пакета, како би се обезбедила конзистентност и побољшала прецизност модела.

За класификацију когнитивних стања коришћена је конволутивна неурална мрежа (CNN) развијена уз помоћ `TensorFlow` и `Keras` библиотеке. Мрежа је саграђена као секвенца слојева, која укључује два конволутивна слоја, два слоја за максимално агреирање и три пуно повезана слоја. Адам оптимизатор је коришћен за тренирање модела, док је бинарна унакрсна ентропија коришћена као функција губитка. Модел је трениран на подацима који су подељени на тренинг и тест сетове, а током тренирања коришћен је и механизам раног заустављања како би се избегло претренирање.

Након тренирања, модел је тестиран и евалуиран коришћењем више метрика као што су тачност, прецизност, осетљивост (*recall*), *F1* скор и *ROC* крива. Резултати су представљени кроз графике и матрицу конфузије, како би се добио јасан увид у перформансе модела.

Развијени систем пружа напредне методе за анализу ЕЕГ сигнала и њихову примену у когнитивним неуронаукама. Модел показује високу прецизност и поузданост, што омогућава његову примену и унапређење у различитим експерименталним условима.

3.3.6.2 Систем за регресију

Систем за регресију когнитивних стања представља наставак и унапређење система за класификацију, омогућавајући рад са *CSV* фајловима и процену когнитивних способности. Систем садржи четири методе регресије когнитивних стања: конволутивну неуралну мрежу (*CNN* регресију), линеарну регресију, *random forest* и *feedforward* мрежу.

Подаци се читавају из *CSV* фајлова који садрже ЕЕГ сигнале и когнитивне параметре испитаника. Прво се врши попуњавање недостајућих когнитивних параметара применом напредних и назадних испуна. Након тога, подаци се на исти начин обрађују као и код система за класификацију али уз синхронизацију са одговарајућим когнитивним параметрима.

CNN модел је развијен користећи *TensorFlow* и *Keras* библиотеке. Мрежа укључује три конволутивна слоја, три слоја за максимално агреирање, *flatten* слој и три густо повезана слоја. Модел је трениран коришћењем Адам оптимизатора и мерења *MSE* као функције губитка. Резултати тренинга су визуализовани и евалуирани кроз различите метрике као што су *RMSE* и R^2 .

Линеарни модел је креиран и трениран помоћу *scikit-learn* библиотеке. Подаци су трансформисани како би одговарали захтевима линеарне регресије, а резултати су евалуирани кроз метрике као што су *MSE*, *RMSE* и R^2 .

Random Forest модел је развијен и трениран користећи *scikit-learn* библиотеку. Модел укључује више дрвета одлуке како би се побољшала тачност и стабилност резултата. Евалуација је извршена кроз мерење *MSE*, *RMSE* и R^2 метрика.

Feedforward мрежа је развијена употребом *TensorFlow* и *Keras* библиотеке. Мрежа укључује неколико густо повезаних слојева и примењује дропаут технику за избегавање претренирања. Резултати су евалуирани кроз метрике *MSE*, *RMSE* и R^2 .

3.3.7 Психолошко тестирање и евалуација

У оквиру психолошког тестирања и евалуације посебна пажња се посвећује Упитнику радне меморије (*WMQ*) и његовом значају у проценама различитих когнитивних функција испитаника. *WMQ* обухвата три основна домена која су од суштинског значаја за процену когнитивних капацитета.

Први домен обухвата мерење способности краткорочног складиштења, што се односи на одржавање информација у краткорочној меморији у кратком временском периоду. Други домен, који се односи на пажњу, укључује способност усмеравања менталних активности на одређене садржаје, као и менталну пажњу и отпорност на дистракцију. Ово подразумева и обраду двоструких задатака. Трећи домен, егzekутивна контрола, фокусира се на извршне аспекте радне меморије, као што су доношење одлука и планирање. Извршне или егzekутивне функције представљају скуп когнитивних вештина неопходних за контролу и саморегулацију понашања.

Као допуну *WMQ* тесту размотрена је и могућност коришћења резултата Струповог задатка као контролне мере за мерење инхибиције и селективне пажње. Овакав приступ омогућава униформност учесника у истраживању и пружа додатни увид у резултате који су производ манипулације током експеримента и анализе података. Од суштинског је значаја истраживање утицаја употребе ових тестова на процену различитих аспеката когнитивних функција испитаника у психолошким испитивањима.

Струпов задатак је когнитивни тест који се користи у психолошким испитивањима за мерење инхибиције и селективне пажње испитаника. Његова основна функција је да процени способност испитаника да суздржи своје уобичајене реакције и да селективно усмери пажњу на конкретан стимулус, игноришући дистракције.

Током рада на Струповом задатку, испитанику се приказују боје које су назначене речима, а његов задатак је да име боје која је назначена одређеном речи прочита, а не да назове саму реч. Ово изазива колизију између боје и семантике речи, што испитује способност испитаника да суздржи уобичајену реакцију и да се фокусира само на боју.

Резултати овог задатка омогућавају увид у ниво инхибиције и пажње испитаника, као и њихову способност да контролишу и регулишу своје когнитивне процесе. Ово је од суштинског значаја у психолошким испитивањима, посебно у истраживањима која се баве познавањем и менталним процесима испитаника.

У циљу практичне реализације експеримента и валидације когнитивних стања испитаника, развијен је посебан алат под називом дигитална интерактивна психологија. Развијени алат представља иновативно решење које омогућује интеграцију психолошких тестова у хибридне системе који се користе у области психологије, психијатрије, неурологије, биомедицинског инжењерства и когнитивних неуронаука. Применом овог алата у другим системима могуће је извршити психолошку евалуацију без промене окружења испитаника и самим тим се квалитет експеримента одржава на жељеном нивоу без промене окружења или ометања испитаника. Осим интеграције с постојећим експериментима који се одвијају на рачунару, телефону, таблети или уређајима за виртуелну и измењену реалност, овај алат такође пружа могућност аутоматизованог читавања психолошких тестова, креирање прикладних упитника, тестирање испитаника, прикупљање биофизичких података, складиштење, обраду података као и представљање резултата експертима.

Имајући у виду да је алат планиран за коришћење у оквиру међународних пројеката, тренутна имплементација омогућује рад на енглеском, хинди, руском и српском језику.

Ради једноставније визуализације како целокупан алат функционише, на слици Слика 45 приказана је методологија алата, а наредних седам потпоглавља описује сваку целину посебно.



Слика 45 Методологија развијеног алата

3.3.7.1 Прилагођавање теста

Конвенционални психолошки тестови најчешће се користе у виду *PDF* докумената или упитника којима се приступа помоћу претраживача. Ради њихове припреме за учитавање у развијеном алату, неопходно је одговарајућа питања, одговоре, домене и секторе претворити у табелу у *CSV* документу, као што је приказано на слици Слика 46. Прва колона представља текст питања, друга понуђене одговоре за то питање, трећа и четврта колона представљају домен и сектор ком питање припада.

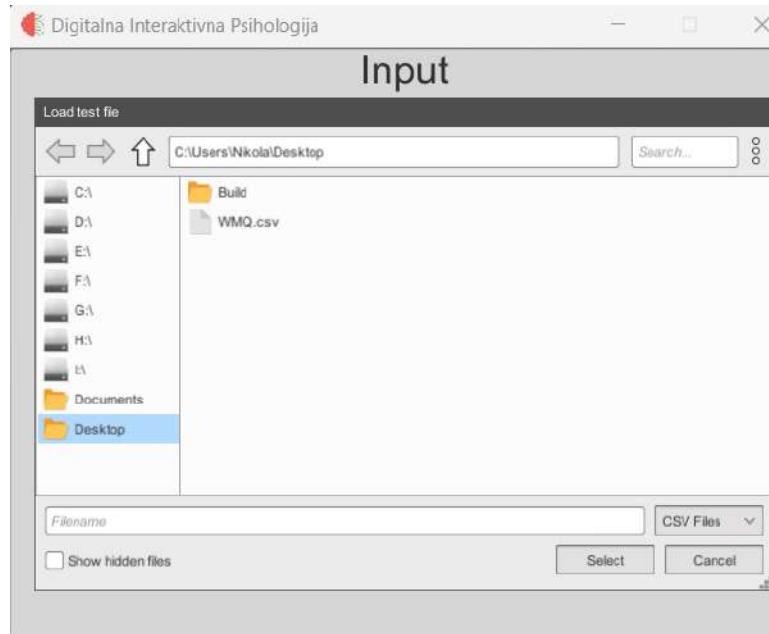
На овај начин могуће је извршити употребу било ког психолошког теста који се налази у текстуалном формату, узимајући у обзир да постоје одговарајуће дозволе за коришћење теста.

	A	B	C	D
1	Questions	Answers:	Domain	Sector
2	1- Do you	not at all;	Attention	1
3	2- Do you	not at all;	Executive	1
4	3- Do you	not at all;	Storage	1
5	4- Do you	not at all;	Attention	1
6	5- Do you	not at all;	Storage	1
7	6- When y	not at all;	Executive	1
8	7- Do you	not at all;	Storage	1
9	8- When y	not at all;	Attention	1
10	9- Do you	not at all;	Executive	1
11	10- Do nee	not at all;	Attention	1
12	11- Do you	not at all;	Storage	1

Слика 46 Приказ исечка табеле правилно форматираног теста

3.3.7.2 Учитавање теста

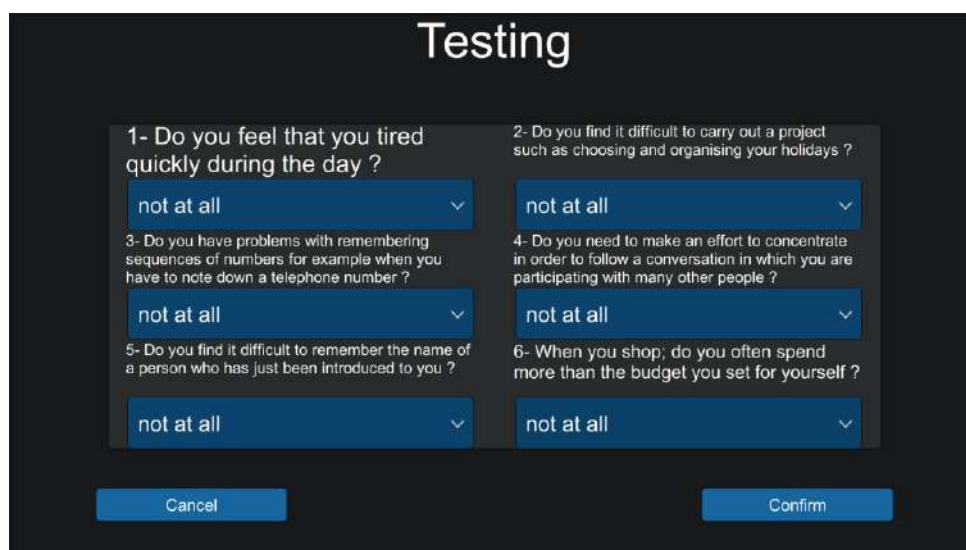
У циљу учитавања теста који је прилагођен за алат, неопходно је у алату одабрати сачувани тест, као што је приказано на слици 47. На тај начин алат аутоматски пролази кроз целокупан CSV документ и креира класе питања с одговарајућим доменима и секторима.



Слика 47 Приказ учитавања прилагођеног теста

3.3.7.3 Тестирање

Након учитавања прилагођеног теста и аутоматизованог креирања одговарајућих класа питања, стручно лице одређује колико ће тестова бити приказано по једној страни. Осим тога, неопходно је унети основне податке о тесту и испитанику који ће се сачувати заједно с тестом. На слици 48 је приказан део тестирања са задатим питањима и понуђеним одговорима.



Слика 48 Тестирање

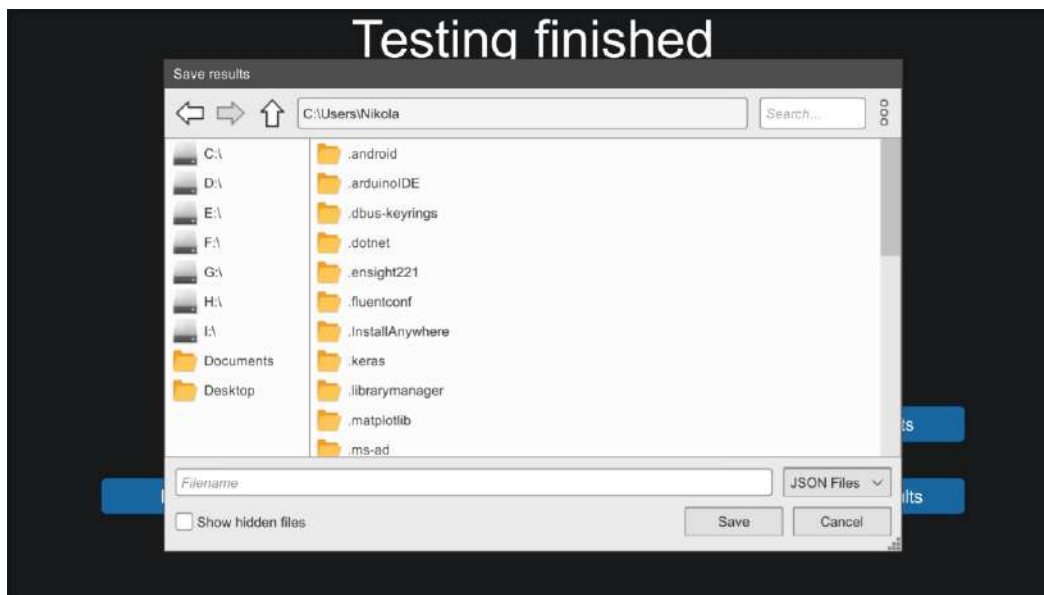
3.3.7.4 Складиштење података и анализа

По завршетку тестирања креира се вишеслојна класа која у себи садржи поља и класе које су описане у табели 1.

Назив објекта	Опис
<i>Title</i>	Назив теста
<i>TestID</i>	ИД теста
<i>SubjectID</i>	ИД испитаника
<i>Score</i>	Укупан резултат
<i>ElapsedTime</i>	Трајање теста у секундама
<i>Date</i>	Датум и време тестирања
<i>UserResponses</i>	Речник свих испитаникових одговора где сваки елемент садржи: текст питања, текст одабраног одговора, домен и сектор
<i>QuestionDictionary</i>	Речник свих питања у тесту при чему сваки елемент садржи: текст питања, понуђене одговоре, домен и сектор
<i>Sectors</i>	Резултати за све секторе и њихове одговарајуће домене

Табела 1 Структура сачуваних резултата једног теста

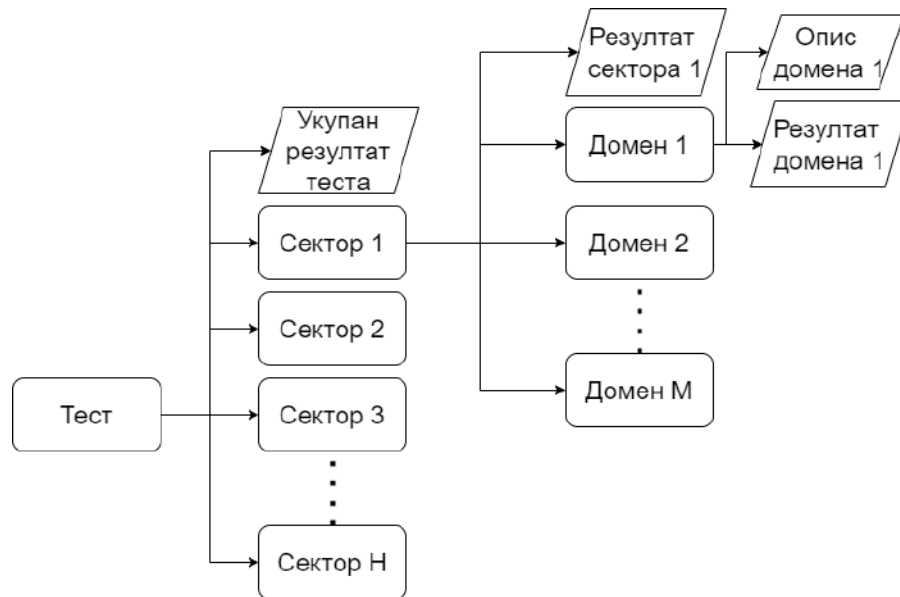
Кориснику алата се на екрану нуди опција да сачува тест, врати се на главни мени или да пређе на приказ резултата. У случају одабира опције за чување теста, отвара се претраживач који нуди испитанику да одабере локацију и назив теста чиме ће он бити сачуван, као што је приказано на слици 49.



Слика 49 Панел за чување резултата теста

Описана структура сачуваног теста из табеле пружа могућност да се изврши стандардизација тестирања испитаника, понављање експеримената, праћење стања испитаника кроз више експеримената, поређење више испитаника као и рад с групама.

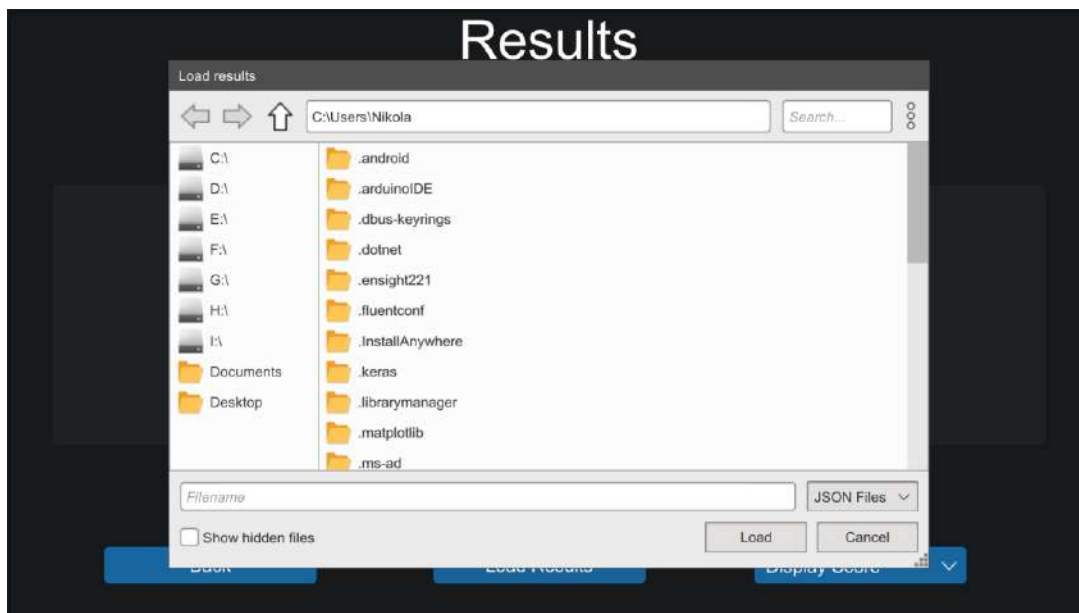
Обрада резултата подразумева тумачење свих одговора које је испитаник дао и њихово групно интерпретирање у складу с дефиницијом теста. На основу свих питања, одговора, домена и сектора креира се структурирана класа која прво рачуна резултат сваког појединачног домена, затим сваког сектора и на крају целокупног теста. Слика 50 приказује општи случај структуре резултата једног теста на основу доступних питања, домена и сектора.



Слика 50 Хијерархија резултата једног теста

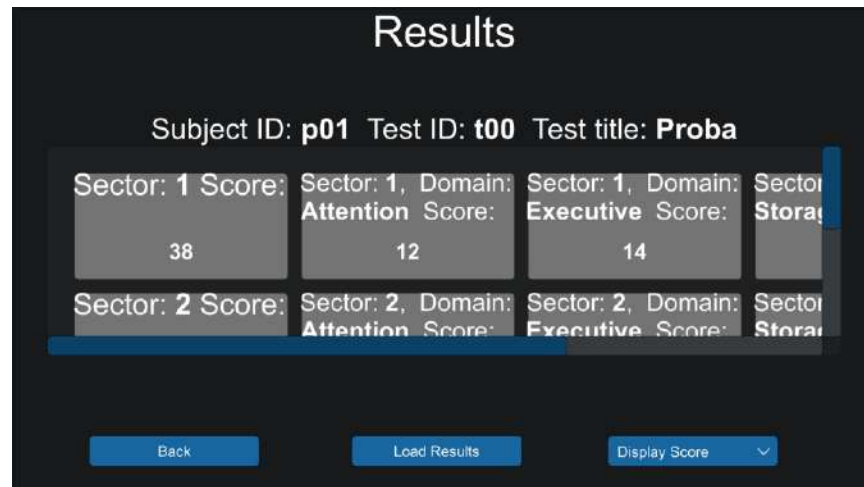
3.3.7.5 Приказ резултата

У оквиру панела за приказ резултата могу се учитати сачувани резултати и на тај начин извршити увид у дате одговоре као и остварене резултате. Панел за учитавање резултата је приказан на слици 51.

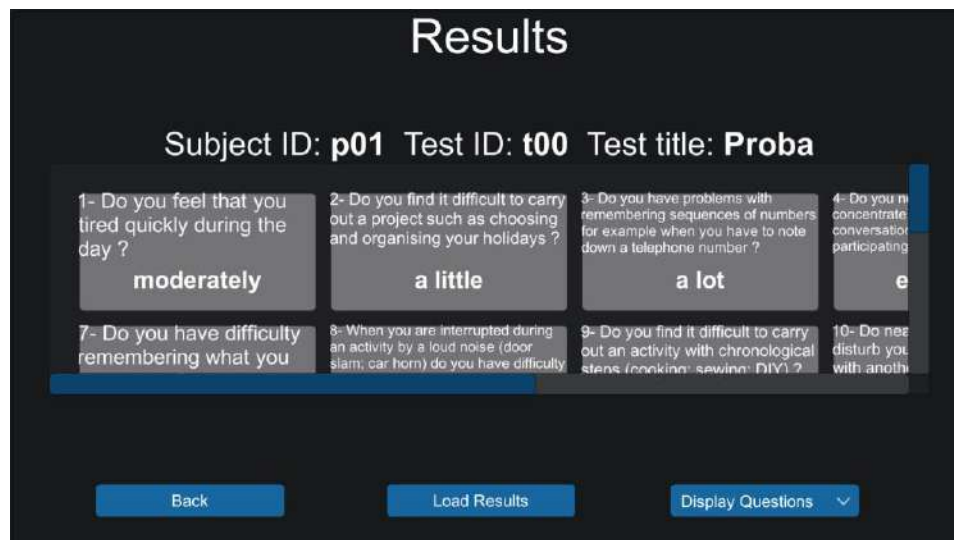


Слика 51 Учитавање резултата

Након учитавања кориснику се одмах пружа могућност да прегледа постигнуте резултате или дате одговоре, при чему се ИД испитаника, ИД теста и назив теста приказују у врху табеле. Постигнути резултати приказани су тако да сваки ред представља резултате једног сектора, при чему је први елемент укупан резултат тог сектора. Последњи елемент у целој табели је укупан постигнути резултат. Сlike 52 и 53 приказују резултате и дате одговоре у једном тесту, респективно, што је могуће мењати помоћу опција из падајућег менија.



Слика 52 Приказ остварених резултата



Слика 53 Приказ датих одговора

3.3.7.6 База података

Као што је већ приказано и описано, тренутно складиштење података одвија се локално, на уређају који покреће тест. Чување текстуалних тестова тренутно се врши у CSV формату, док се сви остали подаци чувају у json формату. Разлог је чињеница што json формат дозвољава структурирани запис и самим тим је могуће безбедно сачувати веће количине података и извршити њихову организацију. Осим тога, json формат дозвољава употребу бинарне енкрипције, чиме осигуравамо да ће само

овлашћени корисници имати приступ подацима и самим тим постижемо задовољавајућу безбедност.

3.3.7.7 *Имплементација*

Имплементација система извршена је у програмском језику *C#* као део *Unity* пројекта, уз одговарајуће библиотеке. Дата имплементација има за циљ да обезбеди свеобухватно решење за тестирање унутар целокупног пројекта. Обухвата функционалности за читање питања, чување одговора испитаника, рачунање резултата и представљање резултата у различитим формама. Користећи *Unity* компоненте корисничког интерфејса, екстерне библиотеке за руковање датотекама и *json* серијализацију, програм даје структурирану и модуларну базу кода за једноставно проширење и интеграцију. Организација података у класе побољшава јасноћу и оптимизовану употребу кода.

4 Резултати

На основу описаних активности и методологије, у циљу правилног разумевања рада система, у наставку следе резултати експеримената и анализе система.

4.1 Неурофидбек

Истраживање које је описано у овом раду спроведено је током мастер студија 2018. године на Факултету техничких наука. Лабораторијски услови и опрема који су коришћени били су у складу са тадашњим техничким могућностима и захтевима истраживања. Истраживање се спроводило у лабораторији на Факултету техничких наука, опремљеној свом потребном опремом и лабораторијским условима. Учествовало је 20 испитаника, студената IV године ОАС Биомедицинског инжењерства, с просечном старашћу од 23 године. Трајање истраживања било је три недеље, током које су испитаници пролазили кроз три експеримента, с временским размаком од седам дана између експеримената. Сваки експеримент је трајао 20 минута и састојао се од припреме испитаника (пет минута), вежбе замишљања простог геометријског кретања (пет минута), мултимедијалне дидактичке вежбе (пет минута) и тестирања испитаника уз кратке консултације (пет минута). Коришћена опрема обухватала је *Emotiv Eroc+* ЕЕГ уређај, лаптоп и неопходне софтверске пакете.

4.1.1 Запажања за обуку и вежбање одрживости менталне концентрације

Неутренирани испитаници нису у могућности одржати максималну менталну концентрацију дуже од једног минута у континуитету. На пажњу, визуализацију и менталну концентрацију испитаника позитивно утиче занимљивост објекта, посебно ако се објекат креће. Испитаници се лакше концентришу на мање детаље у оквиру геометријског тела, а негативно утичу шум из околине, умор и више независних покретних објеката у близини фокуса. Код неутренираних испитаника, поновљивост максималне менталне концентрације могућа је у првих 20 минута, уз задржавање од максимално пар минута.

4.1.2 Запажања за обуку и вежбање агилности менталне концентрације

Након пет минута вежбања одређени број испитаника постиже вољну импулсну концентрацију преко 50% максималне менталне концентрације с трајањем краћим од 0,1 s. Ова вештина умара испитанике више од прве вежбе. Постоји варијабилност у успеху испитаника између прве и друге вежбе.

4.1.3 Запажања за обуку и вежбање контроле менталне концентрације

Током првог вежбања ове вештине максимално време задржавања жељеног нивоа менталне концентрације је пет секунди. Испитаници лакше подижу концентрацију с нижег на средњи ниво у односу на спуштање с вишег на средњи ниво. Стечене вештине током претходних вежби примењују се у циљу постизања резултата у овој вежби.

4.1.4 Arduino платформа

У поглављу 4.4 детаљно је описан процес развоја прототипа система за управљање дроном коришћењем Arduino платформе. Овај прототип је развијен у циљу интеграције ЕЕГ сигнала са системом за управљање дроном, омогућавајући корисницима да путем менталних слика и когнитивних команди управљају летелицом. Комплетан код за овај систем дат је у прилогу 10.2.2. Иако је развој прототипа успешно завршен, систем није коришћен у практичним експериментима због потребе за додатном организацијом и добијањем неопходних дозвола.

Тренутна имплементација представља основу за даља истраживања у области актуаторских система у когнитивним неуронаукама и планирано је да се развија у сарадњи са колегама из сродних области које би унапредиле овај систем. Ови кораци и резултати представљају значајан допринос у развоју система за управљање дроном путем когнитивних команди и постављају основу за будућа истраживања и практичну примену у различитим областима.

4.1.5 Mentis Oculi

Систем "Mentis Oculi" је показао задовољавајућу ефикасност у симулацијама како на рачунару, тако и у окружењу виртуелне реалности. Током експерименталних сесија, систем је омогућио прецизно мерење и анализу ЕЕГ сигнала, као и адекватну визуелизацију и контролу когнитивних стања и функција испитаника.

У поређењу са претходно наведеним неурофидбек системима, *Mentis Oculi* се издваја својом интуитивношћу и једноставношћу употребе. Интегрисани интерфејс и алати олакшавају корисницима контролу, што омогућава бољу интеракцију и разумевање когнитивних процеса.

Систем се показао као стабилан и поуздан у различитим условима, укључујући лабораторијску и нелабораторијску примену. Овај интегрисани приступ нуди могућности за даљи развој и прилагођавање специфичним потребама у областима неурофидбека и когнитивних истраживања.

4.2 Мултиспектрално мапирање мозга

Како би се тестирао систем и потврдила хипотеза, извршен је експеримент коришћењем доступних података. Експеримент је укључивао приказивање слика мапирања мозга које одговарају тренутку когнитивне активности током извршавања аритметичке операције испитаника. ЕЕГ подаци који су снимљени према методи студије били су обрађени помоћу развијеног система како би се добиле слике мапирања мозга.

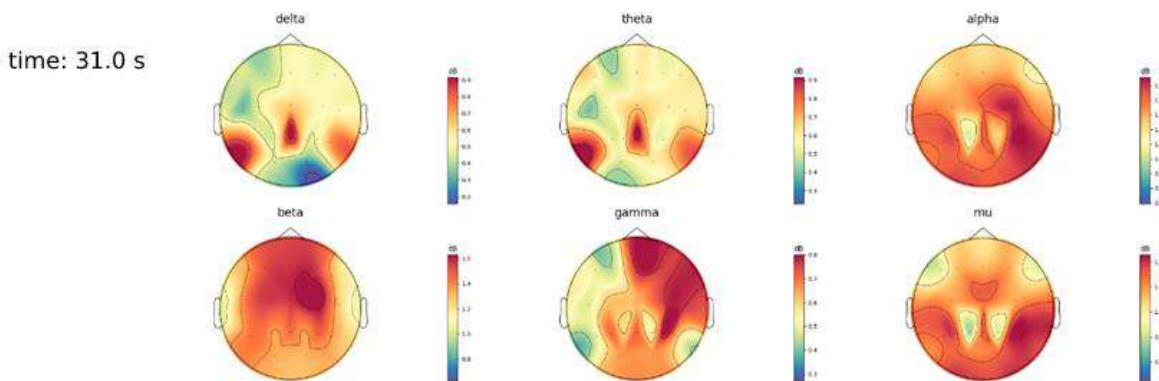
Добијене слике мапирања мозга упоређене су с оригиналним ЕЕГ подацима које су интерпретирале групе психолога експерата. Циљ овог упоређивања је потврђивање тачности и поузданости развијеног система у приказивању слика мапирања мозга које одговарају когнитивној активности током аритметичке операције.

Резултати експеримента указују да је развијени систем способан да генерише тачне слике мапирања мозга које приказују тренутак когнитивне активности током аритметичких операција. Конкретно, слике мапирања мозга показују јасан пораст активности у фронталном и паријеталном кортексу током аритметичког задатка. Ово потврђује почетну хипотезу да је систем способан да прикаже слике мапирања мозга које тачно одговарају тренутку когнитивне активности током аритметичке операције [130]. Ова открића су у складу с претходним истраживањима која су идентификовала поменуте кортексе као укључене у математичку обраду и радну меморију. Поред обезбеђивања тачних слика мапирања мозга, развијени систем такође може да генерише поуздане ЕЕГ податке. Сви снимци били су лишени артефаката коришћењем анализе независних компонената (ICA), која ефикасно уклања било какве артефакте као што су покрети очију, ЕМГ и ЕКГ сигнали. Развијени алат пружа велик потенцијал за когнитивне неуронауке, обезбеђујући истраживачима боље разумевање механизма укључених у математичку обраду и радну меморију. Међутим, мали број испитаника који су прошли кроз експеримент чини ове закључке негенерализованим. Стога су даља истраживања с већим бројем испитаника неопходна како би се потпуно утврдила тачност и поузданост развијеног система. Резултати откривају значајне промене

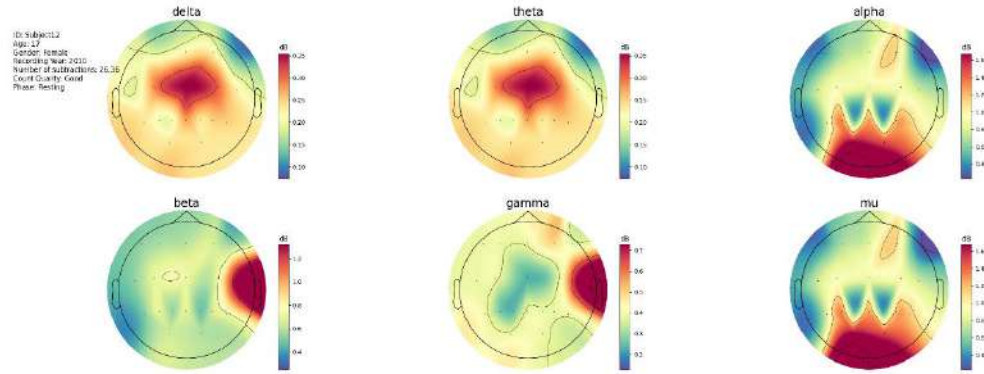
у неуралној активности у одговору на неурофидбек тренинг, посебно у регионима мозга повезаним с пажњом и извршном функцијом. Ова открића сугеришу да неурофидбек може имати значајан утицај на резултате мапирања мозга и истичу важност разматрања ефеката неурофидбека при интерпретацији података мапирања мозга. Слика 54 приказује тренутак добијен током експеримента у коме је ЕЕГ сигнал анализиран и обрађен развијеним алатом како би се генерисале слике мапирања мозга које представљају шест спектра, укључујући делта, тета, алфа, бета, гама и ми. Ово обезбеђује стручњацима комплетну графичку репрезентацију мождане активности испитаника у одређеном тренутку, при чему је у овом конкретном случају разлика у спектрима била највише уочљива у 31. секунди експеримента. Визуализацијом мождане активности, стручњаци могу анализирати и интерпретирати когнитивну активност испитаника и боље разумети подложне когнитивне процесе повезане са задатком изведеним током експеримента. Могућност прегледа мождане активности у више спектра такође обезбеђује додатне увиде у различита когнитивна стања испитаника, што је корисно у идентификацији и лечењу когнитивних поремећаја или унапређењу когнитивних перформанси помоћу неурофидбека.

Обрадом целокупне базе свих доступних испитаника из експеримента [130], алат је аутоматски генерисао слике мапирања мозга за све доступне сигнале у складу с њиховом организацијом коју је описао аутор експеримента. Сlike 55 и 56 приказују пример обраде сигнала једног испитаника током фазе релаксације и фазе рачунања коју је испитаник обавио успешно. У циљу валидације система слике 57 и 58 приказују резултате мапирања мозга испитаника током релаксације и рачунања, који није успешно обављао аритметичке операције. Поређењем слика 34 и 35 можемо јасно видети разлику у можданој активности током релаксације и рачунања код испитаника с израженом концентрацијом.

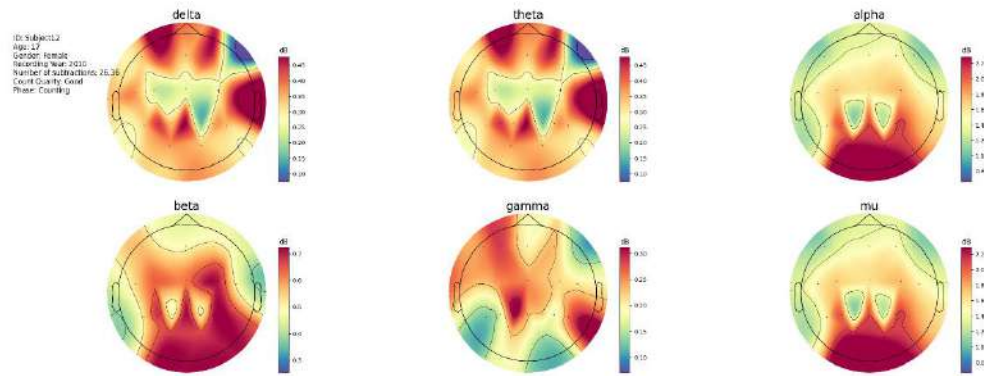
Добијени резултати са тренутно доступним подацима су објављени у часопису *Technology and Healthcare*, под насловом "*Cognitive Phenomena Measurement with Time Window-Based Multispectral Brain Mapping*". Овај рад детаљно разрађује иновативну методологију за мерење когнитивних феномена помоћу временски заснованог мултиспектралног мапирања мозга, показујући значајна унапређења у тачности и поузданости мерења у поређењу са традиционалним методама.[131]



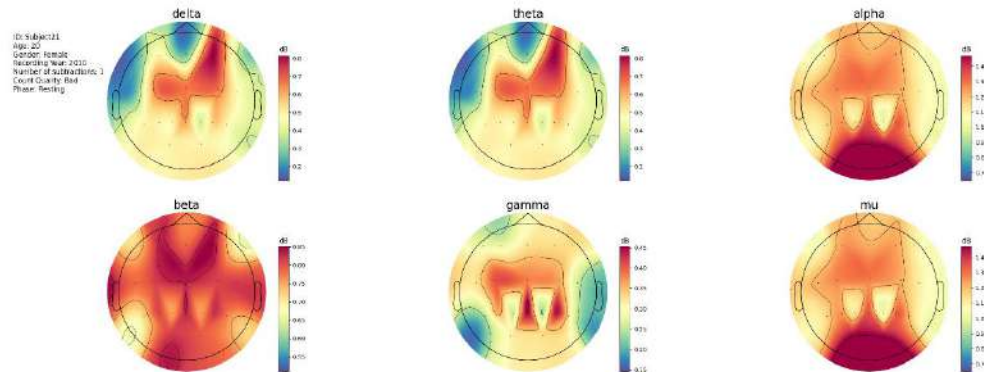
Слика 54 Исечак снимка мапирања мозга



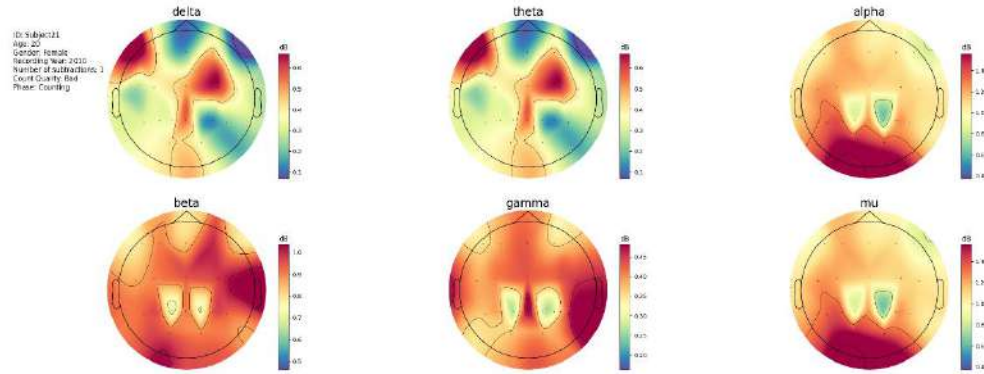
Слика 55 Мапирање мозга током фазе одмора



Слика 56 Мапирање мозга током успешног рачунања



Слика 57 Мапирање мозга током релаксације



Слика 58 Мапирање мозга током логичког рачунања

Укупно гледано, експеримент је успешно потврдио тачност и поузданост развијеног система у приказивању слика мапирања мозга. Овај нови алат, развијен посебно за истраживање когнитивних феномена, пружа уникатну прилику научној заједници да дубље истражи когнитивне процесе. Резултати демонстрирају потенцијалну корисност система у истраживању когнитивне неуронауке и отварају могућности за даља истраживања у овој области. Његове могућности за обраду података, анализу, графички приказ, чување и извоз чине га вредним алатом у истраживању когнитивних феномена, а користиће се као један од модула у више текућих истраживачких пројеката.

Како алат наставља с даљим развојем, очекује се да постане самостална апликација која се може извршавати на локалним машинама и *cloud* серверима. Ова проширења ће омогућити алату да ефикасније обавља задатке с већом обрадом података и даће му способност за складиштење и приступ великим количинама података. Ово ће омогућити дубље разумевање везе између неурофидбека и мапирања мозга, што ће допринети научним истраживањима у оптимизацији употребе неурофидбека у клиничким и истраживачким активностима.

4.3 Систем за класификацију и регресију когнитивних стања и когнитивних активности

У циљу тестирања развијеног система, коришћени ЕЕГ подаци састојали су се од сегмената дужине 0,5 секунди из 23 канала, што је додатно проширило базу података за обуку модела. Коришћење кратких сегмената података има предност у могућности класификације когнитивних стања у реалном времену, што је кључно за многе примене ЕЕГ података, као што су системи интерфејса мозак-рачунар и системи неурофидбека.

Истраживање је показало корисност неколико техника предобраде и анализе података. На пример, користили смо краткотрајну Фуријеву трансформацију (*STFT*) како бисмо претворили временске серијске податке ЕЕГ-а у фреквенцијски домен, омогућавајући нам да издвојимо особине спектралне густине снаге за сваки фреквенцијски опсег (делта, тета, алфа, бета и гама). Ове особине су затим коришћене као улаз у *CNN* модел. Овај приступ нам је омогућио да захватимо спектралне карактеристике ЕЕГ података, које су познате по својим везама с различитим когнитивним стањима.

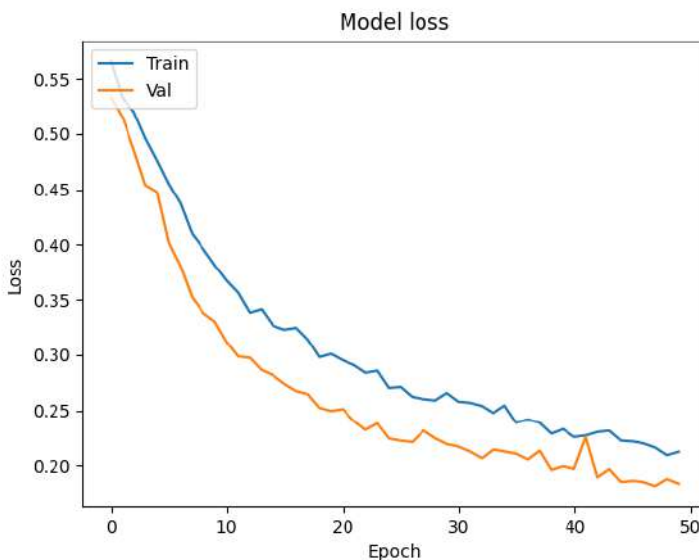
Резултати истраживања демонстрирају ефикасност модела конволутивне неуралне мреже (*CNN*) у класификацији когнитивних стања на основу електроенцефалограма (ЕЕГ) података. Модел је обучаван и валидиран на ЕЕГ подацима сакупљеним за време задатка менталне аритметике, добро

утврђеног когнитивног задатка који има неколико когнитивних процеса, укључујући радну меморију, пажњу и основне аритметичке процесе.

Процес обучавања модела конволутивне неуралне мреже (CNN) био је кључни аспект истраживања. Модел је обучаван у току 50 епоха, при чему су подаци за обуку подељени у тренинг сет (80%) и валидациони сет (20%). Ова подела је омогућила праћење перформанси модела на непознатим подацима који нису били доступни током процеса обучавања, пружајући индикацију способности модела да функционише с новим подацима.

Процес обучавања је вођен бинарном крос-ентропијом, одговарајућим избором за наш задатак бинарне класификације. Адам оптимизатор је коришћен за минимизацију ове функције губитка. Адам, алгоритам за оптимизацију на основу градијената првог реда, широко се користи у моделима дубоког учења због своје ефикасности и ниског захтева за меморијом.

Током процеса обучавања, забележено је константно опадање средње квадратне грешке (Слика 59), што указује на то да је модел ефикасно учио да класификује когнитивне активности на основу ЕЕГ података. Губитак обучавања је опао са 0,565 у првој епохи на 0,212 у 50. епохи. Слично томе, средње квадратна грешка је код валидације опала са 0,533 на 0,183 у истом периоду. Ови трендови сугеришу да модел није преобучен на подацима за обуку, што се види из истовременог смањења губитка на валидационом сету. Даље су примењене различите технике за оптимизацију перформанси модела. Ове технике укључују коришћење слојева искључења како бисмо спречили преобучавање, слојеве максималног узимања како бисмо смањили димензионалност података, као и позив за рани прекид тренинга кад би валидациони губитак престао да опада. Ове технике допринеле су робустним перформансама модела.

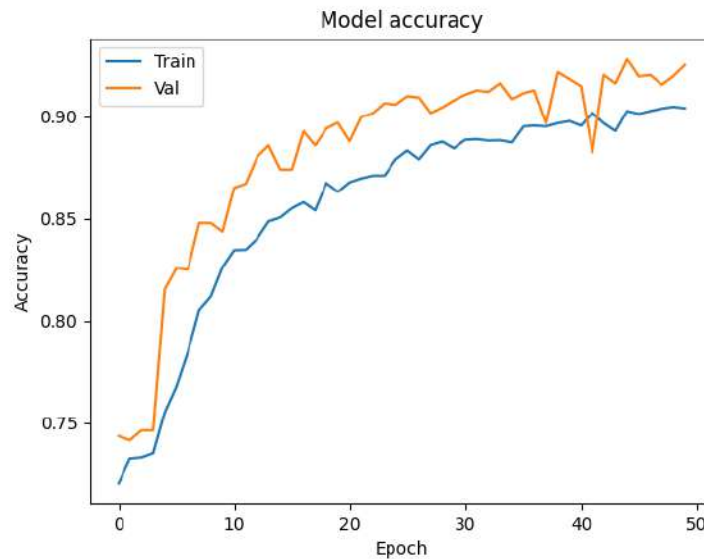


Слика 59 Губици модела током 50 епоха обуке

Да би се спречило преобучавање коришћени су слојеви искључења. Искључење је техника регуларизације која случајно поставља одређени део улазних јединица на 0 при сваком ажурирању током тренинга, што помаже у спречавању преобучавања осигуравајући да модел не зависи

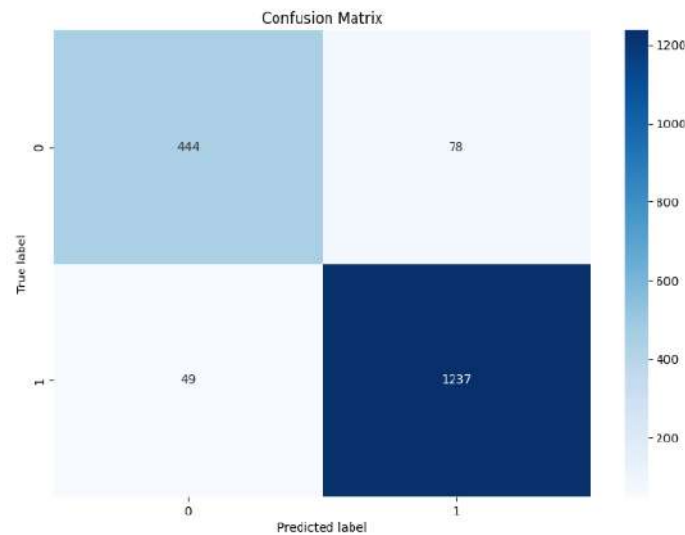
превише од било које појединачне особине. У нашем моделу стопа искључења од 0,25 коришћена је након конволуционих слојева и слојева максималног узимања, а стопа од 0,5 коришћена је пре финалног слоја.

Поред праћења губитка током тренинга, такође је праћена тачност модела на тренинг и валидационом сету. Модел је постигао крајњу тренинг тачност од 89,6% и валидациону тачност од 92,1% (Слика 60). Ове високе стопе тачности, заједно с ниским вредностима губитка, указују на то да се модел добро снашао и на тренинг сету и на валидационом сету.



Слика 60 Тачност модела током 50 епоха обучавања

За детаљнију евалуацију перформанси модела креирана је матрица конфузије на основу предвиђања модела на валидационом сету (Слика 61). Матрица конфузије пружа обухватан преглед перформанси модела, показујући број тачних позитива, тачних негатива, лажних позитива и лажних негатива. Ове информације могу се користити за израчунавање различитих метрика перформанси, као што су F1 скор и Матјев координатни коефицијент (MCC), пружајући нијансирано разумевање перформанси модела у односу на саму тачност.



Слика 61 Матрица конфузије за валидациони узорак скупа података ($TP = 12337$, $TN = 444$, $FP = 78$, $FN = 49$).

Сензитивност мери пропорцију стварних позитива који су тачно идентификовани, и она у овом случају износи 96,2%. Специфичност мери пропорцију стварних негатива који су тачно идентификовани, а у случају овог модела она износи 85%. Тачност је однос правилно предвиђених опажања у односу на укупна опажања, а за одговарајући модел износи 93%. Како би се потврдила посматрана тачност, коришћени су F1 скор, који је вагани просек прецизности и повратка, што је корисно за неравномерну расподелу класа, и Матјев координатни коефицијент, који је мера квалитета бинарних класификација, што је предност у односу на F1 скор као што узима у обзир истинске и лажне позитиве и негативе и сматра се балансираном мером која се може користити чак и ако су класе веома различитих величина. F1 скор и Матјев координатни коефицијент за одговарајући модел износе 96,2%, односно 82,7%.

Развој и тестирање модела за класификацију и регресију когнитивних стања и активности изведени су у сарадњи са стручњацима из области психологије и машинског учења. Током развоја система, стручњаци су учествовали у дефинисању параметара, избору алгоритама и подешавању модела. Њихови увиди и искуства су коришћени за оптимизацију система. Планирано је да се увиди и искуства стручњака даље искористе у будућим научним радовима, где ће њихове процене и препоруке бити детаљно документоване.

Један од значајних исхода овог истраживања је оптимизација система за класификацију когнитивних стања у реалном времену. Ово отвара нове могућности за примену овог система у системима интерфејса мозак-рачунар (BCI) с активношћу у реалном времену и системима за неурофидбек. Могућност класификације когнитивних стања у реалном времену може знатно побољшати функционалност и ефикасност ових система, чинећи ово истраживање вредним доприносом у области.

Развијени модел показао је робустност и значајан потенцијал за примену у области когнитивних наука. Модел, заснован на конволуционим неуронским мрежама (CNNs), обучен је за класификацију когнитивних стања током аритметичких задатака коришћењем ЕЕГ података. ЕЕГ

подаци, сакупљени са 23 канала с трајањем од 0,5 секунди, трансформисани су у фреквенцијске опсеге, именовано делта, тета, алфа, бета и гама, који су познати по вези с различитим когнитивним процесима.

Модел је обучаван током 50 епоха, показујући конвергенцију ка импресивној валидираној тачности од 94% и ниском валидираном губитку од 15%. Процес обучавања био је пажљиво надгледан и прилагођен коришћењем техника раних заустављања и смањења стопе учења како би се спречило преобучавање и осигурала општеприхваћеност модела на новим подацима. Перформансе модела су додатно потврдили стручњаци из области психологије и машинског учења, потврдивши његову исправност и прилагодљивост за даљи развој и унапређење.

Метрике перформанси модела – укључујући сензитивност, специфичност, тачност, F1 скор и Матјев координатни коефицијент (*MCC*) – израчунате су на основу матрице конфузије. Модел је постигао високу сензитивност од 0,962, специфичност од 0,850, тачност од 0,932, F1 скор од 0,962 и *MCC* од 0,827. Ове метрике указују на то да је модел веома ефикасан у тачном класификовању, с балансираним перформансама чак и у присуству дисбаланса класа.

Успех овог модела отвара могућности за његову примену у системима за интерфејс мозак-рачунар (*BCI*) и системима за неурофидбек у реалном времену. Он пружа стабилне основе за развој будућих модела који имају за циљ разумевање и интерпретацију когнитивних феномена у психологији, неурологији и когнитивним неуронаукама.

Овај систем је показао изводљивост и ефикасност коришћења модела дубоког учења, специфично *CNNs*, за класификацију когнитивних стања на основу ЕЕГ података. Високе перформансе модела, уз његов потенцијал за примену у реалном времену, чине га вредним алатом за напредно истраживање у области когнитивних наука. Будући рад може бити усмерен на усавршавање модела с већим скуповима података, истраживање других архитектура неуронских мрежа и примену модела на друге когнитивне задатке и услове.

4.4 Дигитална интерактивна психологија

Дигитална интерактивна психологија представља интеграцију савремених технологија са традиционалним психолошким методама, омогућавајући дубљи увид у когнитивне процесе и интеракције субјеката са дигиталним окружењима. У оквиру овог система, испитаницима је омогућено да учествују у интерактивним психолошким експериментима који користе виртуелну реалност (*VR*) и друге дигиталне платформе као инструменте за мерење и анализу њиховог когнитивног стања. Кроз примену дигиталних симулација, учесници могу бити изложени различитим сценаријима који захтевају концентрацију, решавање проблема и контролу емоција. Овај приступ не само да омогућава прецизније мерење реакција и когнитивних процеса у реалном времену, већ и ствара услове за динамичну адаптацију експерименталних задатака у складу са индивидуалним способностима и реакцијама испитаника.

Развијени систем за дигиталну интерактивну психологију је интегрисан са постојећом инфраструктуром за мерење и анализу ЕЕГ сигнала, омогућавајући детаљно праћење неурофизиолошких реакција током различитих задатака. Ова комбинација технологија пружа свеобухватан увид у сложене когнитивне и емоционалне процесе, што је кључно за развој напредних терапијских и образовних метода.

5 Закључак

У оквиру представљене дисертације истражена је потенцијална способност новоформираних система, базираних на савременим технологијама као што су ЕЕГ, неурофидбек, виртуелна реалност и роботика, да подрже и унапреде неуропластичност. Усвојена методологија укључивала је анализу ЕЕГ података, психолошке и неуролошке евалуације биомедицинских сигнала, као и примену система машинског учења у циљу регресије и класификације.

5.1 Даљи развој система

Циљеви за даљи развој система су:

- **Развој и побољшање опреме за аквизицију ЕЕГ сигнала:**
Оптимална аквизиција ЕЕГ сигнала представља темељ успешне обраде и анализе. У току је истраживање напредних технологија које би омогућиле прецизније, стабилније и брже снимање сигнала, чиме би се допринело квалитетнијем и усмеренијем анализирању информација.
- **Развој опреме и алгоритама за обраду ЕЕГ сигнала:**
Ово подразумева интеграцију најновијих компонената и програмских решења ради појачавања капацитета за обраду сигнала. Подаци се трансформишу тако да буду интерпретабилни и корисни за даљу анализу и примену.
- **Развој алгоритама за процену менталних стања и когнитивних функција испитаника:**
Кључни изазов је разумевање сложених неурофизиолошких процеса који се одигравају у мозгу. На основу ЕЕГ података разрадићемо алгоритме који прецизно идентификују и класификују различита ментална стања и когнитивне функције у реалном времену.
- **Унапређење алгоритама за процену менталних стања и когнитивних функција испитаника:**
Константан рад на побољшању алгоритама има за циљ појачање прецизности и ефикасности. Ово подразумева периодично тестирање, валидацију и адаптацију алгоритама у светлу нових научних сазнања и технолошких достигнућа.
- **Унапређење постојећих система обуке испитаника за употребу мерно-информационог система у зависности од примене:**
Како би се осигурала максимална ефикасност у коришћењу система, постојећи програми обуке ће се дорађивати. Циљ је да се корисници осете удобно, безбедно и сигурно током рада са системом, а уједно и да те програме примене на правилан и продуктиван начин.
- **Развој нових система обуке испитаника за употребу мерно-информационог система у зависности од примене:**
Поред унапређења постојећих програма, фокус ће бити и на креирању потпуно нових образовних модула који би адаптирали кориснике на најновије функције и аспекте система, а такође и обезбедили разумевање и примену система у различитим контекстима.

5.2 Примена развијеног система

У овом потпоглављу наведене су смернице за истраживање могућих примена развијеног система.

5.2.1 Медицинска примена

Медицинска примена овог система не подразумева само традиционални преглед централног нервног система већ усмерава фокус на истраживање функционалних целина мозга, омогућавајући стручњацима да сагледају рад и координацију различитих региона. Поред тога што систем нуди брзе резултате, он такође омогућава прецизно лоцирање аномалија или потенцијалних проблема. У идеалном клиничком окружењу овакав систем може бити коришћен као прва линија дијагностике централног нервног система, при чему би се конвенционални медицински алати и методе користили за дубљу и специфичну анализу. Ефикасност овакве методологије резултира краћим временом проведеним на прегледима, оптималнијом употребом медицинске опреме и стручњака, као и укупним смањењем трошкова дијагностике.

5.2.2 Рехабилитациона примена

Рехабилитациона примена развијеног система подразумева употребу за персонализовану рехабилитацију и неурорехабилитацију испитаника која се обавља интеграцијом с постојећим системима, где је крајњи циљ максимална стимулација неуропластичности. Рехабилитациони системи подразумевају стимулацију неопходних физичких функција у складу с процењеним когнитивним активностима у реалном времену које одговарају задатој функцији, при чему се у виртуелној реалности приказује жељена радња. Примена оваквог система може се користити за рехабилитацију следећих феномена: померање екстремитета, локомоторни рефлекс, фацијална експресија, регулација рада срца, регулација респираторног рефлекса и слично.

Неурорехабилитациона примена овог система подразумева персонализовани приступ лечењу и терапији когнитивних стања и поремећаја помоћу активних или пасивних вежби. Активне вежбе подразумевају свесно обављање задатака помоћу извођења задатих когнитивних функција и операција, уз приказ квалитета у реалном времену. Неки од примера оваквих вежби су: замишљање кретања различитих геометријских тела, замишљање промене интензитета и фреквенције звука и управљање физичким системима. Пасивне вежбе подразумевају обављање задатака на које когнитивна стања и когнитивне активности испитаника не утичу директно, али могу олакшати или отежати њихово обављање. Примери оваквих система су: обављање рачунских операција са сликом при чему број елемената у позадини слике зависи од концентрације испитаника, тражење разлика између две слике при чему њихова осветљеност зависи од пажње испитаника, боравак у виртуелној просторији при чему растојање зидова зависи од стреса испитаника, вежбање јавног говора при чему шум публике зависи од релаксације испитаника и управљање аутомобилом при чему максимална брзина аутомобила зависи од концентрације испитаника. Системи у овој области могу се користити за неурорехабилитацију следећих феномена: поремећај пажње, анксиозност, панични поремећај, стресни поремећај, фобије, поремећај спавања, трауматски поремећај као и поремећај прилагођавања.

5.2.3 Едукативна примена

Едукативна примена развијеног система наслања се на подстакнуту неуропластичност испитаника током употребе система, имајући у виду да подупире усвајање нових и унапређење постојећих способности. Ово је посебно значајно за здраве особе којима је потребна обука или додатно усавршавање.

Систем пружа персонализовани приступ образовању и обуци, прилагођавајући се когнитивним активностима и стањима сваког појединца. На овај начин, учење и обука постају далеко ефикаснији јер се садржај и методе усвајања информација прилагођавају специфичним потребама и способностима испитаника.

При унапређењу постојећих способности систем се фокусира на оптимизацију жељене активности. Праћењем релевантних параметара, активност се прилагођава на основу когнитивних стања корисника, чиме се постиже оптималан резултат у извођењу те активности.

Оваква едукативна примена система има широк спектар могућих примена: од спортских активности и физиологије спорта, преко школске едукације на свим нивоима, па до обуке меких вештина и наменске индустрије. Тако, на пример, спортиста може користити систем за оптимизацију своје тренинг рутине, док ученик или студент може користити исти такав систем за унапређење својих ученичких вештина и постизање бољих академских резултата.

5.2.4 Асистивна примена

Асистивна примена развијеног система подразумева да се, на основу процењених когнитивних активности, испитаницима пружа могућност управљања симулационим или физичким системима. Способност адаптације система омогућава испитаницима широку палету могућих система којима могу управљати. Предност оваквог система је у томе што прецизност, брзина и слобода коју испитаници остварују не захтевају физичке покрете, што је значајно како за особе с инвалидитетом тако и за здраве особе јер се избегава замор мишића или мишићних неуросинапси. Самим тим, док су физичке контроле ограничене на физичке способности испитаника, когнитивне функције су неограничене.

Неки од примера употебе овог система укључују управљање: аутомобилом, дроном, роботизованим системима, специјализованим возилима, гејминг системима и наменским уређајима.

5.2.5 Научноистраживачка примена

Научноистраживачка примена развијеног система представља витални корак у разумевању и истраживању когнитивних стања, активности, рефлекса и различитих феномена који се јављају у људском мозгу. Коришћење овог система као основе за научна истраживања омогућава дубље увиде у функционисање мозга, како на изолованим нивоима тако и у ширим когнитивним контекстима. Оваква примена представља каталитичку тачку за развој нових метода и техника у многим научним дисциплинама.

Важност система препозната је у широком спектру области, што ће резултирати формирањем мултидисциплинарних тимова који ће објединити стручњаке из различитих научних и истраживачких домена. Ово укључује физичаре, хемичаре, инжењере из области електротехнике, експерте у области рачунарских наука, медицинске стручњаке, психологе, психијатре и многе друге, али се не ограничава само на њих. Овакво обједињавање ресурса и експертског знања има за циљ не само разумевање већ и развијање нових приступа и решења за комплексне изазове који се постављају пред човечанство у области когнитивних наука.

6 Литература

- [1] E. Niedermeyer and F. L. da Silva, *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.
- [2] P. L. Nunez and R. Srinivasan, *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2006.
- [3] H. H. Jasper, "Ten-twenty electrode system of the international federation," *Electroencephalogr Clin Neurophysiol*, vol. 10, pp. 371-375, 1958.
- [4] L. R. Krol, "EEG electrode positions in the 10-10 system using modified combinatorial nomenclature, along with the fiducials and associated lobes of the brain.," ed, 2020.
- [5] G. Buzsaki, *Rhythms of the Brain*. Oxford university press, 2006.
- [6] J.-P. Barral and A. Croibier, "Manual therapy for the cranial nerves," (*No Title*), 2009.
- [7] D. R. Simkin, R. W. Thatcher, and J. Lubar, "Quantitative EEG and neurofeedback in children and adolescents: Anxiety disorders, depressive disorders, comorbid addiction and attention-deficit/hyperactivity disorder, and brain injury," *Child and Adolescent Psychiatric Clinics*, vol. 23, no. 3, pp. 427-464, 2014.
- [8] R. W. Thatcher, *Handbook of quantitative electroencephalography and EEG biofeedback*. Anipublishing Company, 2016.
- [9] S. Sanei and J. A. Chambers, *EEG signal processing*. John Wiley & Sons, 2013.
- [10] M. X. Cohen, *Analyzing neural time series data: theory and practice*. MIT press, 2014.
- [11] T. Åkerstedt and P. M. Nilsson, "Sleep as restitution: an introduction," *Journal of internal medicine*, vol. 254, no. 1, pp. 6-12, 2003.
- [12] S. Makeig, K. Gramann, T.-P. Jung, T. J. Sejnowski, and H. Poizner, "Linking brain, mind and behavior," *International Journal of psychophysiology*, vol. 73, no. 2, pp. 95-100, 2009.
- [13] J. R. Wolpaw and E. W. Wolpaw, "Brain-computer interfaces: something new under the sun," *Brain-computer interfaces: principles and practice*, vol. 14, 2012.
- [14] N. Birbaumer and L. G. Cohen, "Brain-computer interfaces: communication and restoration of movement in paralysis," *The Journal of physiology*, vol. 579, no. 3, pp. 621-636, 2007.
- [15] N. Lofthouse, L. E. Arnold, S. Hersch, E. Hurt, and R. DeBeus, "A review of neurofeedback treatment for pediatric ADHD," *Journal of attention disorders*, vol. 16, no. 5, pp. 351-372, 2012.
- [16] B. Zoefel, R. J. Huster, and C. S. Herrmann, "Neurofeedback training of the upper alpha frequency band in EEG improves cognitive performance," *Neuroimage*, vol. 54, no. 2, pp. 1427-1431, 2011.
- [17] H. Gevensleben *et al.*, "Neurofeedback training in children with ADHD: 6-month follow-up of a randomised controlled trial," *European child & adolescent psychiatry*, vol. 19, pp. 715-724, 2010.
- [18] R. B. Ivry and G. R. Mangun, *Cognitive Neuroscience: The biology of the mind*. Norton, 2014.
- [19] Z. Li, L. Zhang, F. Zhang, R. Gu, W. Peng, and L. Hu, "Demystifying signal processing techniques to extract resting-state EEG features for psychologists," *Brain Science Advances*, vol. 6, no. 3, pp. 189-209, 2020.
- [20] J. R. Hughes and E. R. John, "Conventional and quantitative electroencephalography in psychiatry," *The Journal of neuropsychiatry and clinical neurosciences*, vol. 11, no. 2, pp. 190-208, 1999.
- [21] P. S. Churchland and T. J. Sejnowski, *The computational brain*. MIT press, 1992.
- [22] W. Klimesch, "Alpha-band oscillations, attention, and controlled access to stored information," *Trends in cognitive sciences*, vol. 16, no. 12, pp. 606-617, 2012.
- [23] S. J. Luck, *An introduction to the event-related potential technique*. MIT press, 2014.
- [24] S. Makeig, A. Bell, T.-P. Jung, and T. J. Sejnowski, "Independent component analysis of electroencephalographic data," *Advances in neural information processing systems*, vol. 8, 1995.

- [25] A. M. Bastos and J.-M. Schoffelen, "A tutorial review of functional connectivity analysis methods and their interpretational pitfalls," *Frontiers in systems neuroscience*, vol. 9, p. 175, 2016.
- [26] R. T. Thibault, A. MacPherson, M. Lifshitz, R. R. Roth, and A. Raz, "Neurofeedback with fMRI: A critical systematic review," *Neuroimage*, vol. 172, pp. 786-807, 2018.
- [27] P. R. Huttenlocher and A. S. Dabholkar, "Regional differences in synaptogenesis in human cerebral cortex," *Journal of comparative Neurology*, vol. 387, no. 2, pp. 167-178, 1997.
- [28] A. Pascual-Leone, A. Amedi, F. Fregni, and L. B. Merabet, "The plastic human brain cortex," *Annu. Rev. Neurosci.*, vol. 28, pp. 377-401, 2005.
- [29] A. Galván, "Neural plasticity of development and learning," *Human brain mapping*, vol. 31, no. 6, pp. 879-890, 2010.
- [30] M. F. Bear and W. C. Abraham, "Long-term depression in hippocampus," *Annual review of neuroscience*, vol. 19, no. 1, pp. 437-462, 1996.
- [31] T. V. Bliss and G. L. Collingridge, "A synaptic model of memory: long-term potentiation in the hippocampus," *Nature*, vol. 361, no. 6407, pp. 31-39, 1993.
- [32] R. J. Zatorre, R. D. Fields, and H. Johansen-Berg, "Plasticity in gray and white: neuroimaging changes in brain structure during learning," *Nature neuroscience*, vol. 15, no. 4, pp. 528-536, 2012.
- [33] A. Pascual-Leone *et al.*, "Characterizing brain cortical plasticity and network dynamics across the age-span in health and disease with TMS-EEG and TMS-fMRI," *Brain topography*, vol. 24, pp. 302-315, 2011.
- [34] A. May, "Experience-dependent structural plasticity in the adult human brain," *Trends in cognitive sciences*, vol. 15, no. 10, pp. 475-482, 2011.
- [35] P. Greenwood, "Functional plasticity in cognitive aging: review and hypothesis," *Neuropsychology*, vol. 21, no. 6, p. 657, 2007.
- [36] B. Draganski *et al.*, "Temporal and spatial dynamics of brain structure changes during extensive learning," *Journal of Neuroscience*, vol. 26, no. 23, pp. 6314-6317, 2006.
- [37] R. J. Nudo, "Recovery after brain injury: mechanisms and principles," *Frontiers in human neuroscience*, vol. 7, p. 887, 2013.
- [38] S. M. Jaeggi, M. Buschkuhl, J. Jonides, and W. J. Perrig, "Improving fluid intelligence with training on working memory," *Proceedings of the National Academy of Sciences*, vol. 105, no. 19, pp. 6829-6833, 2008.
- [39] H. W. Mahncke, A. Bronstone, and M. M. Merzenich, "Brain plasticity and functional losses in the aged: scientific bases for a novel intervention," *Progress in brain research*, vol. 157, pp. 81-109, 2006.
- [40] A. Sale, N. Berardi, and L. Maffei, "Environment and brain plasticity: towards an endogenous pharmacotherapy," *Physiological reviews*, vol. 94, no. 1, pp. 189-234, 2014.
- [41] R. D. Fields, "White matter in learning, cognition and psychiatric disorders," *Trends in neurosciences*, vol. 31, no. 7, pp. 361-370, 2008.
- [42] M. Grosse-Wentrup, D. Mattia, and K. Oweiss, "Using brain-computer interfaces to induce neural plasticity and restore function," *Journal of neural engineering*, vol. 8, no. 2, p. 025004, 2011.
- [43] A. Holtmaat and K. Svoboda, "Experience-dependent structural synaptic plasticity in the mammalian brain," *Nature Reviews Neuroscience*, vol. 10, no. 9, pp. 647-658, 2009.
- [44] K. Deisseroth, "Optogenetics: 10 years of microbial opsins in neuroscience," *Nature neuroscience*, vol. 18, no. 9, pp. 1213-1225, 2015.
- [45] A. H. Marblestone, G. Wayne, and K. P. Kording, "Toward an integration of deep learning and neuroscience," *Frontiers in computational neuroscience*, vol. 10, p. 94, 2016.

- [46] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245-258, 2017.
- [47] K. Doya, "Reinforcement learning: Computational theory and biological mechanisms," *HFSP journal*, vol. 1, no. 1, p. 30, 2007.
- [48] A. Baddeley, "Working memory: looking back and looking forward," *Nature reviews neuroscience*, vol. 4, no. 10, pp. 829-839, 2003.
- [49] L. R. Squire, "Memory and brain systems: 1969–2009," *Journal of Neuroscience*, vol. 29, no. 41, pp. 12711-12716, 2009.
- [50] I. A. Clark and E. A. Maguire, "Remembering preservation in hippocampal amnesia," *Annual review of psychology*, vol. 67, pp. 51-82, 2016.
- [51] M. Carrasco, "Visual attention: The past 25 years," *Vision research*, vol. 51, no. 13, pp. 1484-1525, 2011.
- [52] S. K. Ungerleider and L. G. "Mechanisms of visual attention in the human cortex," *Annual review of neuroscience*, vol. 23, no. 1, pp. 315-341, 2000.
- [53] R. J. Sternberg, *Wisdom, intelligence, and creativity synthesized*. Cambridge University Press, 2003.
- [54] A. Diamond, "Executive functions," *Annual review of psychology*, vol. 64, pp. 135-168, 2013.
- [55] T.-P. Jung *et al.*, "Removing electroencephalographic artifacts by blind source separation," *Psychophysiology*, vol. 37, no. 2, pp. 163-178, 2000.
- [56] D. Gopher, T. Inui, J. L. McClelland, and A. Koriat, *Attention and performance XVI: information integration in perception and communication*. MIT Press, 1996.
- [57] N. Censor and D. Sagi, "Global resistance to local perceptual adaptation in texture discrimination," *Vision Research*, vol. 49, no. 21, pp. 2550-2556, 2009.
- [58] O. Hauk, M. H. Davis, F. Kherif, and F. Pulvermüller, "Imagery or meaning? Evidence for a semantic origin of category-specific brain activity in metabolic imaging," *European Journal of Neuroscience*, vol. 27, no. 7, pp. 1856-1866, 2008.
- [59] C. Lustig, P. Shah, R. Seidler, and P. A. Reuter-Lorenz, "Aging, training, and the brain: a review and future directions," *Neuropsychology review*, vol. 19, pp. 504-522, 2009.
- [60] S. M. Kosslyn, G. Ganis, and W. L. Thompson, "Neural foundations of imagery," *Nature reviews neuroscience*, vol. 2, no. 9, pp. 635-642, 2001.
- [61] J. Pearson and S. M. Kosslyn, "The heterogeneity of mental representation: Ending the imagery debate," *Proceedings of the national academy of sciences*, vol. 112, no. 33, pp. 10089-10092, 2015.
- [62] R. J. Zatorre and A. R. Halpern, "Mental concerts: musical imagery and auditory cortex," *Neuron*, vol. 47, no. 1, pp. 9-12, 2005.
- [63] S. M. Kosslyn and W. L. Thompson, "When is early visual cortex activated during visual mental imagery?," *Psychological bulletin*, vol. 129, no. 5, p. 723, 2003.
- [64] A. M. Albers, P. Kok, I. Toni, H. C. Dijkerman, and F. P. De Lange, "Shared representations for working memory and mental imagery in early visual cortex," *Current Biology*, vol. 23, no. 15, pp. 1427-1431, 2013.
- [65] O. Jensen and A. Mazaheri, "Shaping functional architecture by oscillatory alpha activity: gating by inhibition," *Frontiers in human neuroscience*, vol. 4, p. 186, 2010.
- [66] J. F. Hipp, A. K. Engel, and M. Siegel, "Oscillatory synchronization in large-scale cortical networks predicts perception," *Neuron*, vol. 69, no. 2, pp. 387-396, 2011.
- [67] S. Cortese *et al.*, "Toward systems neuroscience of ADHD: a meta-analysis of 55 fMRI studies," *American Journal of Psychiatry*, vol. 169, no. 10, pp. 1038-1055, 2012.

- [68] P. Arpaia *et al.*, "A systematic review on feature extraction in electroencephalography-based diagnostics and therapy in attention deficit hyperactivity disorder," *Sensors*, vol. 22, no. 13, p. 4934, 2022.
- [69] S. Epelbaum and F. Cacciamani, "Clinical Assessment of Brain Disorders," *Machine Learning for Brain Disorders*, p. 233, 2023.
- [70] E. L. The Scientist, Inspiring Innovation. "Brain-Computer Interface User Types 90 Characters Per Minute with Mind." <https://www.the-scientist.com/news-opinion/brain-computer-interface-user-types-90-characters-per-minute-with-mind-68762> (accessed).
- [71] C. Vidaurre, C. Sannelli, K.-R. Müller, and B. Blankertz, "Machine-learning-based coadaptive calibration for brain-computer interfaces," *Neural computation*, vol. 23, no. 3, pp. 791-816, 2011.
- [72] M. C. F Lotte, A Lécuyer, F Lamarche and B Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *Journal of Neural Engineering*, vol. 4(2), p. R1, 2007.
- [73] A. AlMofleh, M. Alseddiqi, O. Najam, L. Albaloooshi, A. Alheddi, and A. Alshaimi, "Brain Computer Interfaces: The Future of Communication Between the Brain and the External World," *Science, Engineering and Technology*, vol. 3, no. 2, 2023.
- [74] B.-H. Cho *et al.*, "Neurofeedback training with virtual reality for inattention and impulsiveness," *Cyberpsychology & Behavior*, vol. 7, no. 5, pp. 519-526, 2004.
- [75] R. Mane, K. K. Ang, and C. Guan, "Brain-Computer interface for stroke rehabilitation," *Handbook of Neuroengineering*, pp. 1285-1315, 2023.
- [76] S. R. Soekadar, M. Witkowski, N. Birbaumer, and L. G. Cohen, "Enhancing Hebbian learning to control brain oscillatory activity," *Cerebral cortex*, vol. 25, no. 9, pp. 2409-2415, 2015.
- [77] F. Pichiorri *et al.*, "Brain-computer interface boosts motor imagery practice during stroke recovery," *Annals of neurology*, vol. 77, no. 5, pp. 851-865, 2015.
- [78] M. A. Lebedev and M. A. Nicolelis, "Brain-machine interfaces: past, present and future," *TRENDS in Neurosciences*, vol. 29, no. 9, pp. 536-546, 2006.
- [79] G. Buzsáki and B. O. Watson, "Brain rhythms and neural syntax: implications for efficient coding of cognitive content and neuropsychiatric disease," *Dialogues in clinical neuroscience*, 2022.
- [80] K. N. Ochsner and J. J. Gross, "The cognitive control of emotion," *Trends in cognitive sciences*, vol. 9, no. 5, pp. 242-249, 2005.
- [81] J. Á. Ariza and J. M. Pearce, "Low-Cost Assistive Technologies for Disabled People Using Open-Source Hardware and Software: A Systematic Literature Review," *IEEE Access*, 2022.
- [82] C. Guger, B. Z. Allison, and A. Gunduz, *Brain-computer interface research: a state-of-the-art summary 10*. Springer, 2021.
- [83] R. Fazel-Rezai, "Brain-Computer Interface Systems: Recent Progress and Future Prospects," 2013.
- [84] C. E. Herff, "Speech Processes for Brain-Computer Interfaces," Universität Bremen, 2016.
- [85] M. N. Abbott and S. L. Peck, "Emerging ethical issues related to the use of brain-computer interfaces for patients with total locked-in syndrome," *Neuroethics*, vol. 10, pp. 235-242, 2017.
- [86] M. Slater and M. V. Sanchez-Vives, "Enhancing our lives with immersive virtual reality," *Frontiers in Robotics and AI*, vol. 3, p. 74, 2016.
- [87] R. T. Azuma, "A survey of augmented reality," *Presence: teleoperators & virtual environments*, vol. 6, no. 4, pp. 355-385, 1997.
- [88] G. Kipper and J. Rampolla, *Augmented reality: An emerging technologies guide to AR*. Elsevier, 2012.
- [89] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining "gamification"," in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, 2011, pp. 9-15.

- [90] A. Rizzo and S. T. Koenig, "Is clinical virtual reality ready for primetime?," *Neuropsychology*, vol. 31, no. 8, p. 877, 2017.
- [91] S. Doolani *et al.*, "A review of extended reality (xr) technologies for manufacturing training," *Technologies*, vol. 8, no. 4, p. 77, 2020.
- [92] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning* (no. 4). Springer, 2006.
- [93] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [94] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.
- [95] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.
- [96] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5-32, 2001.
- [97] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.
- [98] T. Ros *et al.*, "Mind over chatter: plastic up-regulation of the fMRI salience network directly after EEG neurofeedback," *Neuroimage*, vol. 65, pp. 324-335, 2013.
- [99] J. H. Gruzelier, "EEG-neurofeedback for optimising performance. I: A review of cognitive and affective outcome in healthy participants," *Neuroscience & Biobehavioral Reviews*, vol. 44, pp. 124-141, 2014.
- [100] T. Ros, B. J. Baars, R. A. Lanius, and P. Vuilleumier, "Tuning pathological brain oscillations with neurofeedback: a systems neuroscience framework," *Frontiers in human neuroscience*, vol. 8, p. 1008, 2014.
- [101] M. Schabus *et al.*, "Enhancing sleep quality and memory in insomnia using instrumental sensorimotor rhythm conditioning," *Biological psychology*, vol. 95, pp. 126-134, 2014.
- [102] R. Sitaram *et al.*, "Closed-loop brain training: the science of neurofeedback," *Nature Reviews Neuroscience*, vol. 18, no. 2, pp. 86-100, 2017.
- [103] R. T. Thibault, M. Lifshitz, and A. Raz, "The self-regulating brain and neurofeedback: Experimental science and clinical promise," *cortex*, vol. 74, pp. 247-261, 2016.
- [104] S. Diekelmann and J. Born, "The memory function of sleep," *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 114-126, 2010.
- [105] Y.-Y. Tang, B. K. Hölzel, and M. I. Posner, "The neuroscience of mindfulness meditation," *Nature reviews neuroscience*, vol. 16, no. 4, pp. 213-225, 2015.
- [106] R. T. Schirmer *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human brain mapping*, vol. 38, no. 11, pp. 5391-5420, 2017.
- [107] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *Journal of neural engineering*, vol. 14, no. 1, p. 016003, 2016.
- [108] K. Vandecasteele *et al.*, "Automated epileptic seizure detection based on wearable ECG and PPG in a hospital environment," *Sensors*, vol. 17, no. 10, p. 2338, 2017.
- [109] Emotiv. "Emotiv Insight Technical Specifications." <https://emotiv.gitbook.io/insight-manual/introduction/technical-specifications> (accessed).
- [110] Emotiv. "Emotiv Insight User manual." <https://emotiv.gitbook.io/insight-manual/> (accessed).
- [111] H. T. Atay, İ. Çilesiz, and K. K. Kurt, "EEG Controlled Semi-Autonomous Mobile Vehicle Design and Implementation," in *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2019: IEEE, pp. 1-5.
- [112] Emotiv. "Emotiv Epos+ User Manual." <https://emotiv.gitbook.io/epoc-user-manual/> (accessed).
- [113] Emotiv. "Emotiv Epos+ Technical Specifications." <https://emotiv.gitbook.io/epoc-user-manual/introduction-1/technical-specifications> (accessed).

- [114] R. Maskeliunas, R. Damasevicius, I. Martisius, and M. Vasiljevas, "Consumer-grade EEG devices: are they usable for control tasks?," *PeerJ*, vol. 4, p. e1746, 2016.
- [115] Emotiv. "Emotiv Launcher User Manual." <https://emotiv.gitbook.io/emotiv-launcher/> (accessed).
- [116] Emotiv. "EmotivPRO User Manual." <https://emotiv.gitbook.io/emotivpro-v3/> (accessed).
- [117] Emotiv. "EmotivBCI User Manual." <https://emotiv.gitbook.io/emotivbci/> (accessed).
- [118] Emotiv. "Emotiv BrainViz User Manual." <https://emotiv.gitbook.io/emotiv-brainviz/> (accessed).
- [119] Emotiv. "Emotiv App User Manual." <https://emotiv.gitbook.io/cortex-manual/> (accessed).
- [120] S. A. Arduino, "Arduino," *Arduino LLC*, vol. 372, 2015.
- [121] M. Banzi and M. Shiloh, *Getting started with Arduino*. Maker Media, Inc., 2022.
- [122] J. M. Hughes, *Arduino: a technical reference: a handbook for technicians, engineers, and makers*. "O'Reilly Media, Inc.", 2016.
- [123] U. Technologies. "Unity Engine." <https://unity.com/> (accessed).
- [124] A. Šmíd, "Comparison of unity and unreal engine," *Czech Technical University in Prague*, pp. 41-61, 2017.
- [125] O. Comber, R. Motschnig, H. Mayer, and D. Haselberger, "Engaging students in computer science education through game development with unity," in *2019 IEEE Global Engineering Education Conference (Educon)*, 2019: IEEE, pp. 199-205.
- [126] M. Platforms. "Meta Quest 2." <https://www.meta.com/quest/> (accessed).
- [127] J. Brookes, M. Warburton, M. Alghadier, M. Mon-Williams, and F. Mushtaq, "Studying human behavior with virtual reality: The Unity Experiment Framework," *Behavior research methods*, vol. 52, pp. 455-463, 2020.
- [128] J. Jerald, P. Giokaris, D. Woodall, A. Hartholt, A. Chandak, and S. Kuntz, "Developing virtual reality applications with Unity," in *2014 IEEE Virtual Reality (VR)*, 2014: IEEE, pp. 1-3.
- [129] A. Gramfort *et al.*, "MEG and EEG data analysis with MNE-Python," *Frontiers in neuroscience*, p. 267, 2013.
- [130] I. Zyma *et al.*, "Electroencephalograms during mental arithmetic task performance," *Data*, vol. 4, no. 1, p. 14, 2019.
- [131] N. Petrović, S. Mandić, S. Borojević, N. Gazivoda, and P. Sovilj, "Cognitive phenomena measurement with time window-based multispectral brain mapping," *Technology and Health Care*, no. Preprint, pp. 1-10, 2024.

7 Прилози

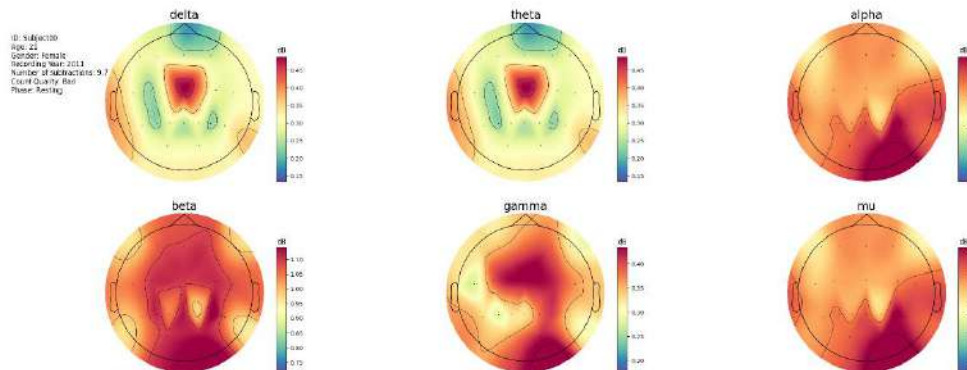
7.1 Додатни резултати

7.1.1 Мапирање мозга

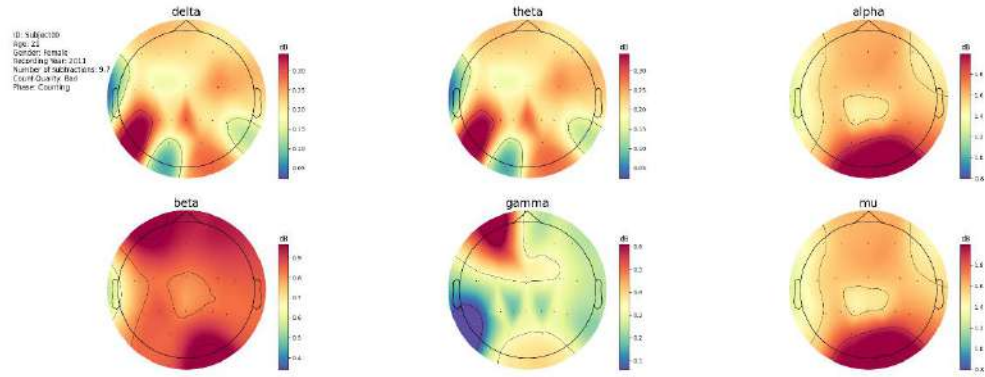
Назив слике	ИД испитаника	Старост	Пол	Година снимања	Број операција	Квалитет рачунања	Активност
Слика62	Subject00	21	Женски	2011.	9.7	Лош	Релаксација
Слика63	Subject00	21	Женски	2011.	9.7	Лош	Рачунање
Слика64	Subject01	18	Женски	2011.	29.35	Добар	Релаксација
Слика65	Subject01	18	Женски	2011.	29.35	Добар	Рачунање
Слика66	Subject02	19	Женски	2012.	12.88	Добар	Релаксација
Слика67	Subject02	19	Женски	2012.	12.88	Добар	Рачунање
Слика68	Subject03	17	Женски	2010.	31	Добар	Релаксација
Слика69	Subject03	17	Женски	2010.	31	Добар	Рачунање
Слика70	Subject04	17	Женски	2010.	8.6	Лош	Релаксација
Слика71	Subject04	17	Женски	2010.	8.6	Лош	Рачунање
Слика72	Subject05	16	Женски	2010.	20.71	Добар	Релаксација
Слика73	Subject05	16	Женски	2010.	20.71	Добар	Рачунање
Слика74	Subject06	18	Мушки	2011.	4.35	Лош	Релаксација
Слика75	Subject06	18	Мушки	2011.	4.35	Лош	Рачунање
Слика76	Subject07	18	Женски	2012.	13.38	Добар	Релаксација
Слика77	Subject07	18	Женски	2012.	13.38	Добар	Рачунање
Слика78	Subject08	26	Мушки	2011.	18.24	Добар	Релаксација
Слика79	Subject08	26	Мушки	2011.	18.24	Добар	Рачунање
Слика80	Subject09	16	Женски	2010.	7	Лош	Релаксација
Слика81	Subject09	16	Женски	2010.	7	Лош	Рачунање
Слика82	Subject10	17	Женски	2010.	1	Лош	Релаксација
Слика83	Subject10	17	Женски	2010.	1	Лош	Рачунање
Слика84	Subject11	18	Женски	2010.	26	Добар	Релаксација
Слика85	Subject11	18	Женски	2010.	26	Добар	Рачунање
Слика86	Subject12	17	Женски	2010.	26.36	Добар	Релаксација
Слика87	Subject12	17	Женски	2010.	26.36	Добар	Рачунање
Слика88	Subject13	24	Мушки	2012.	34	Добар	Релаксација
Слика89	Subject13	24	Мушки	2012.	34	Добар	Рачунање
Слика90	Subject14	17	Женски	2010.	9	Лош	Релаксација
Слика91	Subject14	17	Женски	2010.	9	Лош	Рачунање
Слика92	Subject15	17	Женски	2012.	22.18	Добар	Релаксација
Слика93	Subject15	17	Женски	2012.	22.18	Добар	Рачунање
Слика94	Subject16	17	Женски	2010.	11.59	Добар	Релаксација
Слика95	Subject16	17	Женски	2010.	11.59	Добар	Рачунање
Слика96	Subject17	17	Женски	2010.	28.7	Добар	Релаксација
Слика97	Subject17	17	Женски	2010.	28.7	Добар	Рачунање
Слика98	Subject18	17	Женски	2010.	20	Добар	Релаксација
Слика99	Subject18	17	Женски	2010.	20	Добар	Рачунање
Слика100	Subject19	22	Мушки	2010.	7.06	Лош	Релаксација
Слика101	Subject19	22	Мушки	2010.	7.06	Лош	Рачунање

Слика102	Subject20	17	Женски	2010.	15.41	Добар	Релаксација
Слика103	Subject20	17	Женски	2010.	15.41	Добар	Рачунање
Слика104	Subject21	20	Женски	2010.	1	Лош	Релаксација
Слика105	Subject21	20	Женски	2010.	1	Лош	Рачунање
Слика106	Subject22	19	Женски	2011.	4.47	Лош	Релаксација
Слика107	Subject22	19	Женски	2011.	4.47	Лош	Рачунање
Слика108	Subject23	16	Женски	2010.	27.47	Добар	Релаксација
Слика109	Subject23	16	Женски	2010.	27.47	Добар	Рачунање
Слика110	Subject24	17	Мушки	2012.	14.76	Добар	Релаксација
Слика111	Subject24	17	Мушки	2012.	14.76	Добар	Рачунање
Слика112	Subject25	17	Мушки	2012.	30.53	Добар	Релаксација
Слика113	Subject25	17	Мушки	2012.	30.53	Добар	Рачунање
Слика114	Subject26	17	Женски	2010.	13.59	Добар	Релаксација
Слика115	Subject26	17	Женски	2010.	13.59	Добар	Рачунање
Слика116	Subject27	19	Женски	2010.	34.59	Добар	Релаксација
Слика117	Subject27	19	Женски	2010.	34.59	Добар	Рачунање
Слика118	Subject28	19	Женски	2010.	27	Добар	Релаксација
Слика119	Subject28	19	Женски	2010.	27	Добар	Рачунање
Слика120	Subject29	19	Мушки	2011.	16.59	Добар	Релаксација
Слика121	Subject29	19	Мушки	2011.	16.59	Добар	Рачунање
Слика122	Subject30	17	Мушки	2011.	10	Лош	Релаксација
Слика123	Subject30	17	Мушки	2011.	10	Лош	Рачунање
Слика124	Subject31	19	Женски	2012.	19.88	Добар	Релаксација
Слика125	Subject31	19	Женски	2012.	19.88	Добар	Рачунање
Слика126	Subject32	20	Женски	2011.	13	Добар	Релаксација
Слика127	Subject32	20	Женски	2011.	13	Добар	Рачунање
Слика128	Subject33	17	Мушки	2010.	21.47	Добар	Релаксација
Слика129	Subject33	17	Мушки	2010.	21.47	Добар	Рачунање
Слика130	Subject34	18	Женски	2010.	31	Добар	Релаксација
Слика131	Subject34	18	Женски	2010.	31	Добар	Рачунање
Слика132	Subject35	17	Женски	2010.	12.18	Добар	Релаксација
Слика133	Subject35	17	Женски	2010.	12.18	Добар	Рачунање

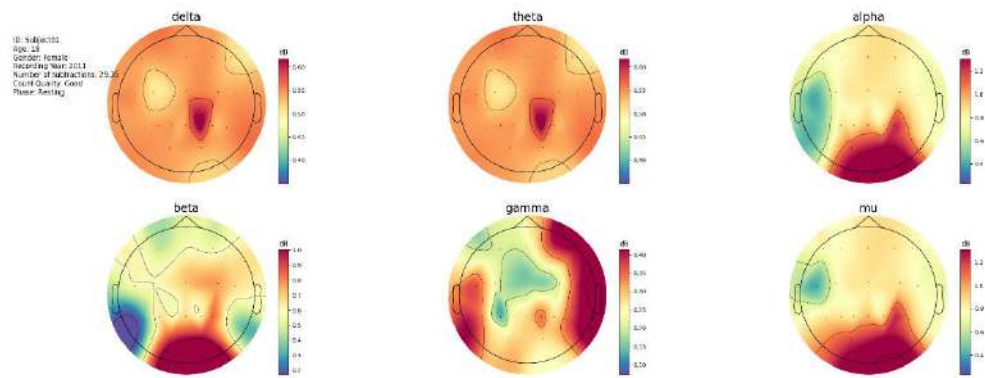
Табела 2 Списак слика мапирања с додатним подацима



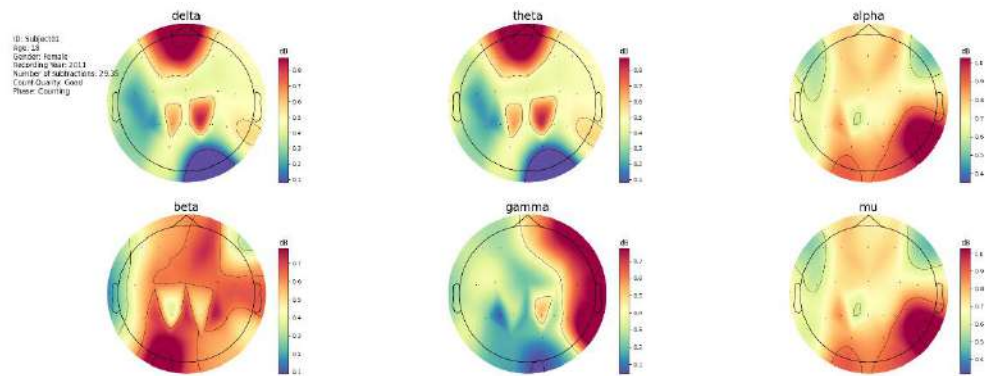
Слика 62 Мапирање мозга - прилог



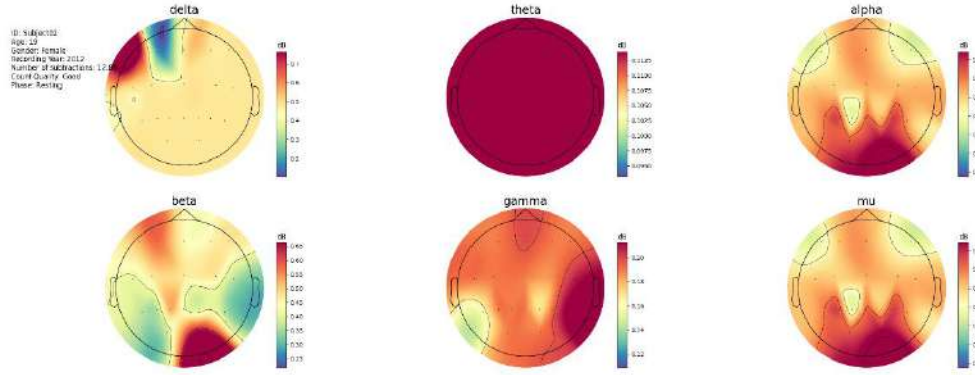
Слика 63 Мапирање мозга - прилог



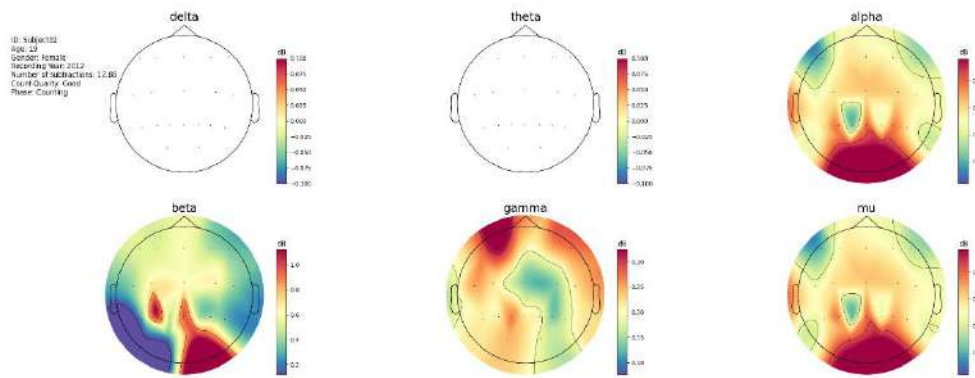
Слика 64 Мапирање мозга - прилог



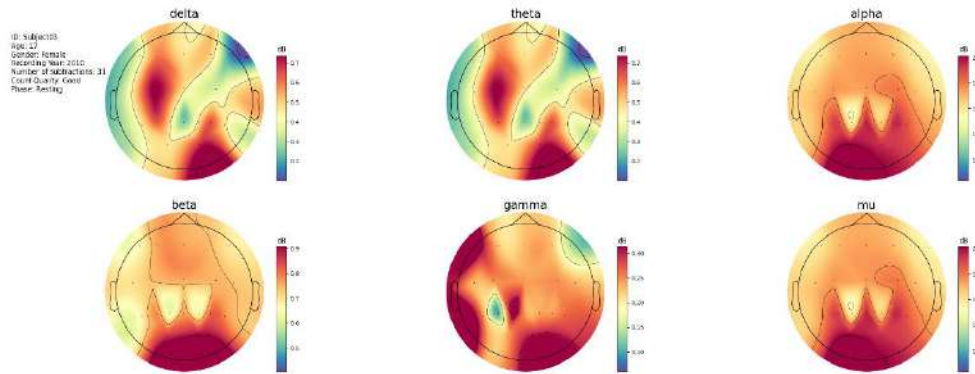
Слика 65 Мапирање мозга - прилог



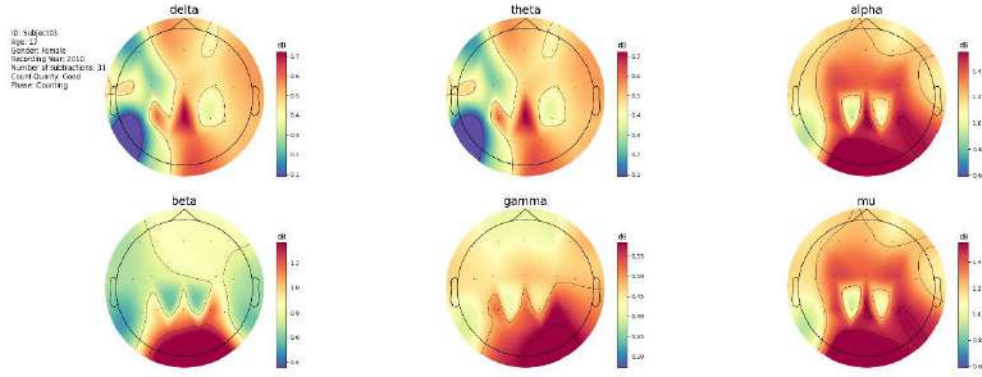
Слика 66 Мапирање мозга - прилог



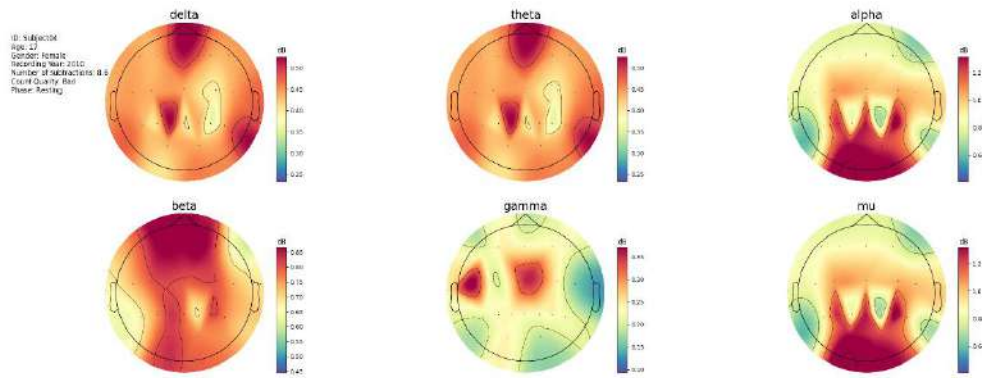
Слика 67 Мапирање мозга - прилог



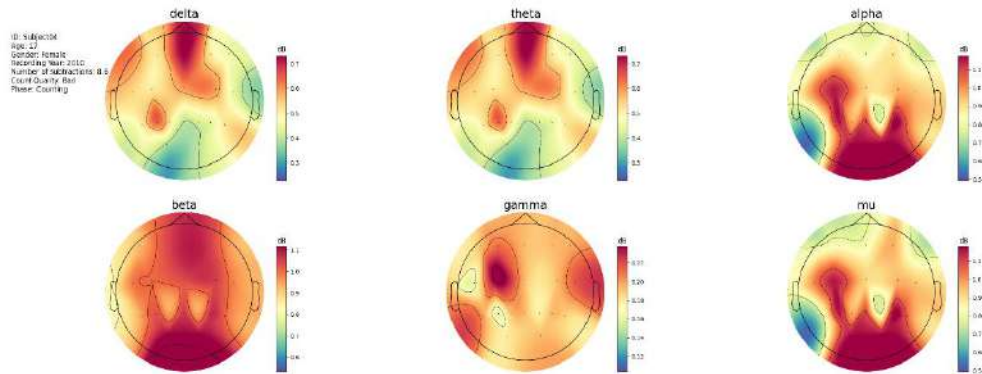
Слика 68 Мапирање мозга - прилог



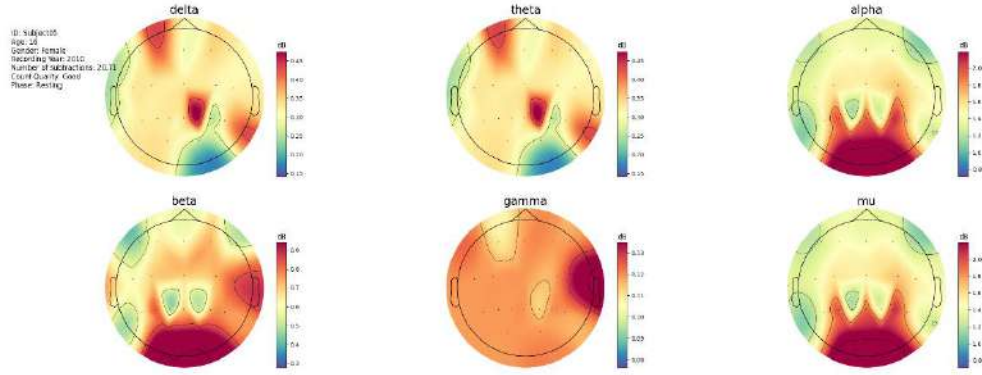
Слика 69 Мапирање мозга - прилог



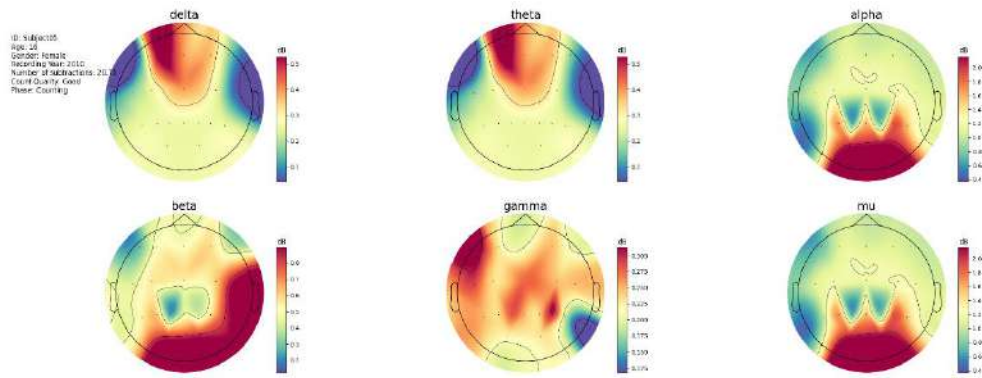
Слика 70 Мапирање мозга - прилог



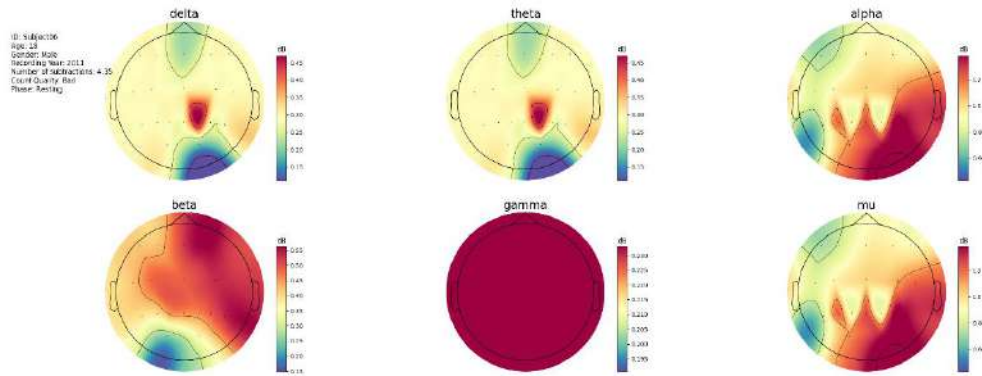
Слика 71 Мапирање мозга - прилог



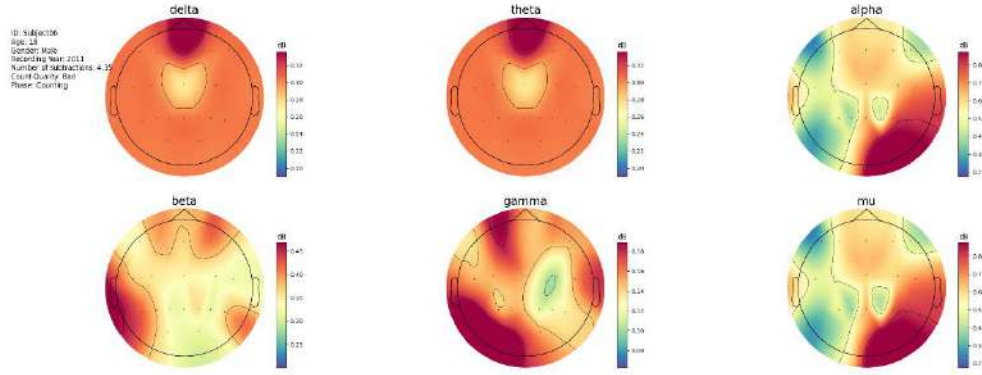
Слика 72 Мапирање мозга - прилог



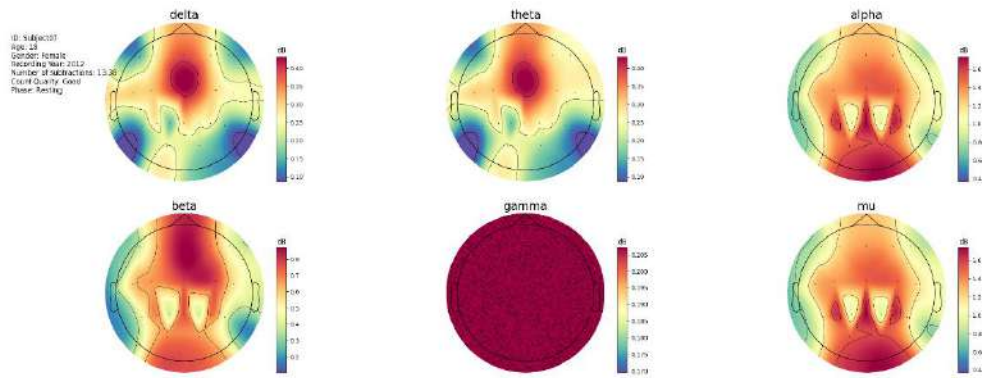
Слика 73 Мапирање мозга - прилог



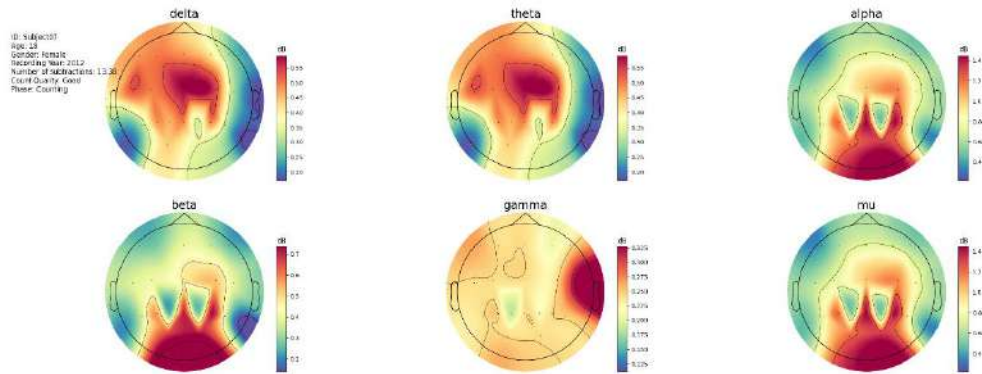
Слика 74 Мапирање мозга - прилог



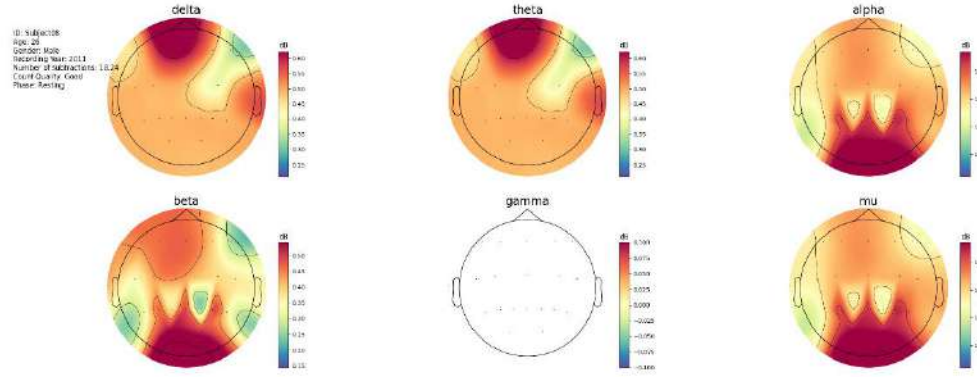
Слика 75 Мапирање мозга - прилог



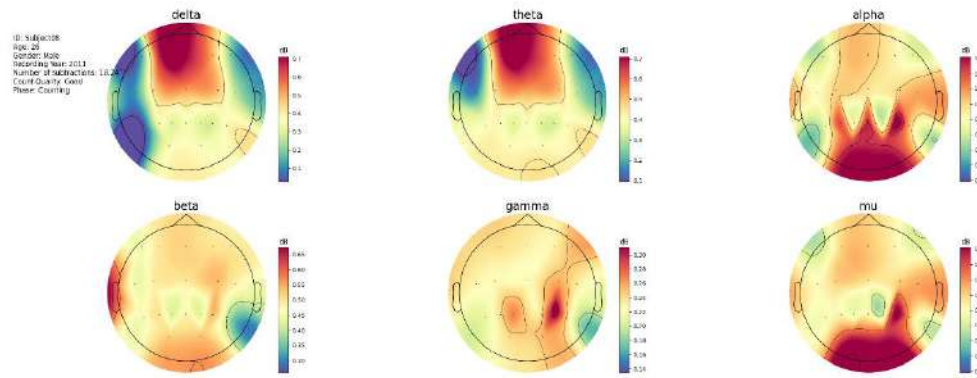
Слика 76 Мапирање мозга - прилог



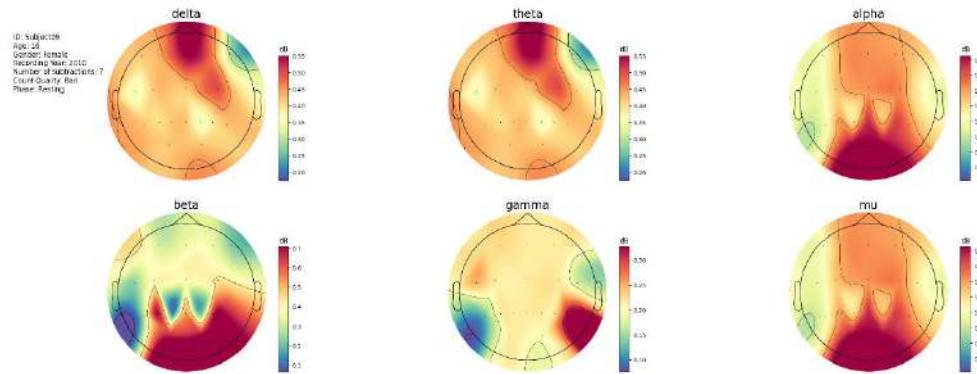
Слика 77 Мапирање мозга - прилог



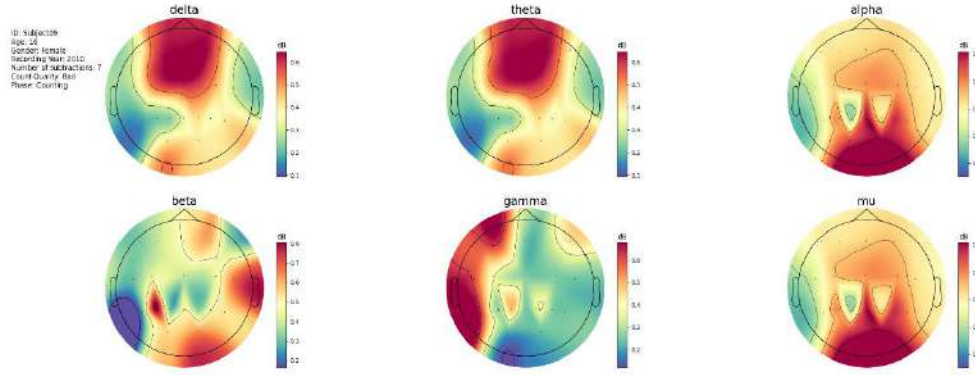
Слика 78 Мапирање мозга - прилог



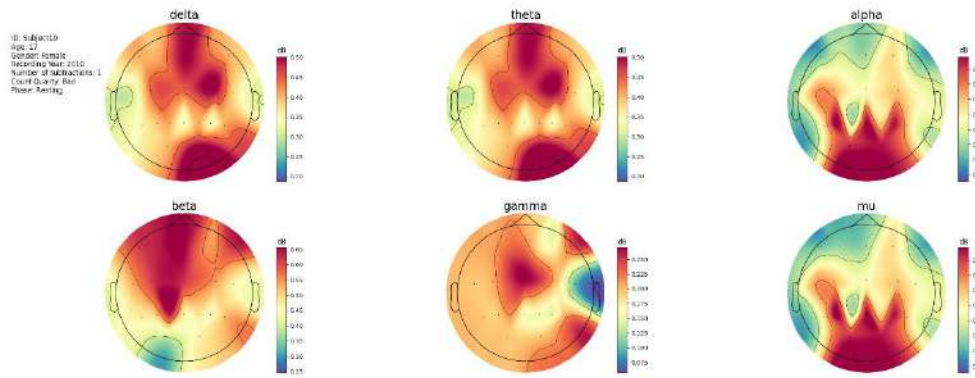
Слика 79 Мапирање мозга - прилог



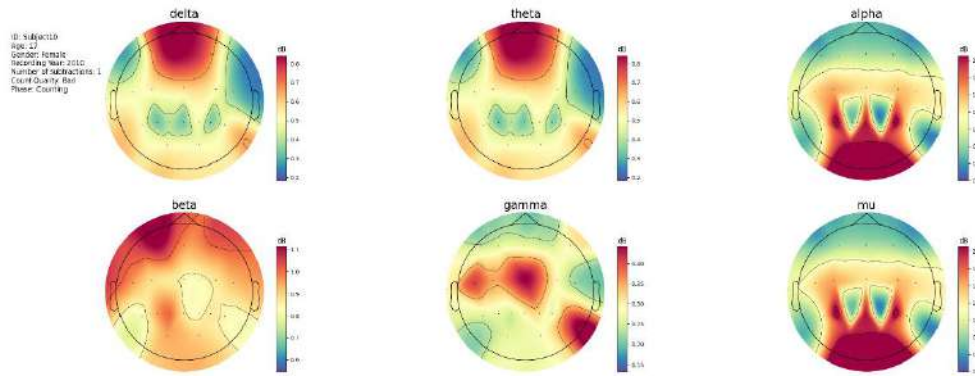
Слика 80 Мапирање мозга - прилог



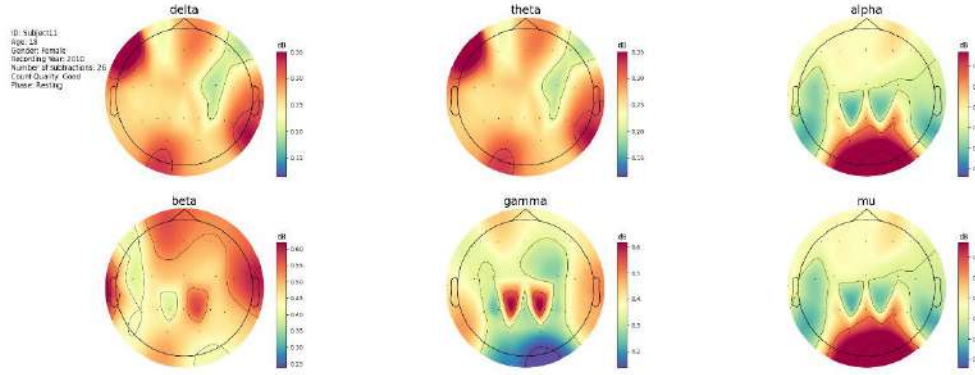
Слика 81 Мапирање мозга - прилог



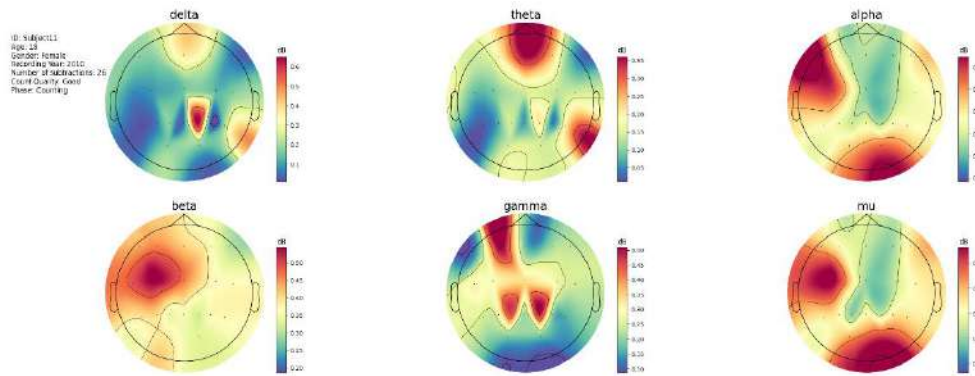
Слика 82 Мапирање мозга - прилог



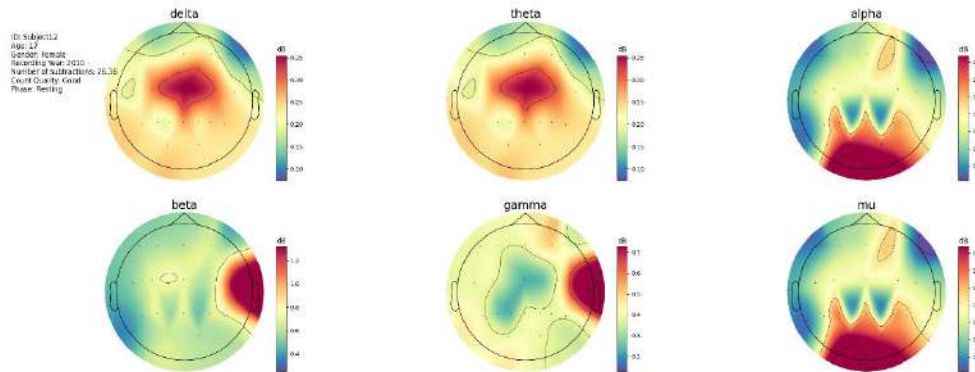
Слика 83 Мапирање мозга - прилог



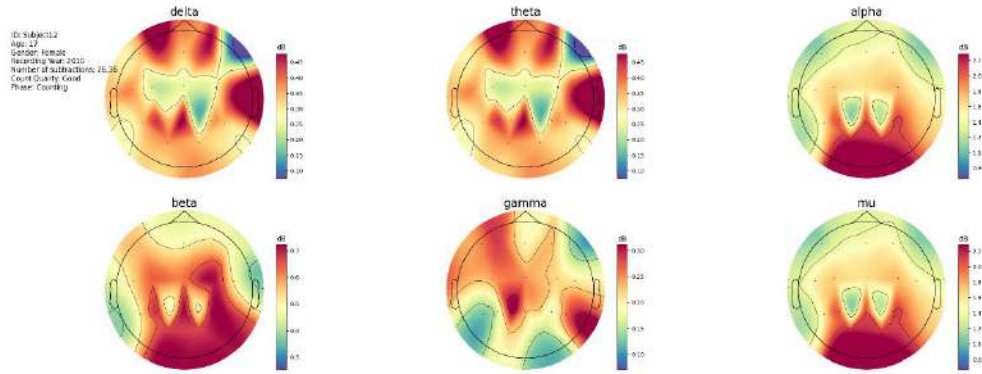
Слика 84 Мапирање мозга - прилог



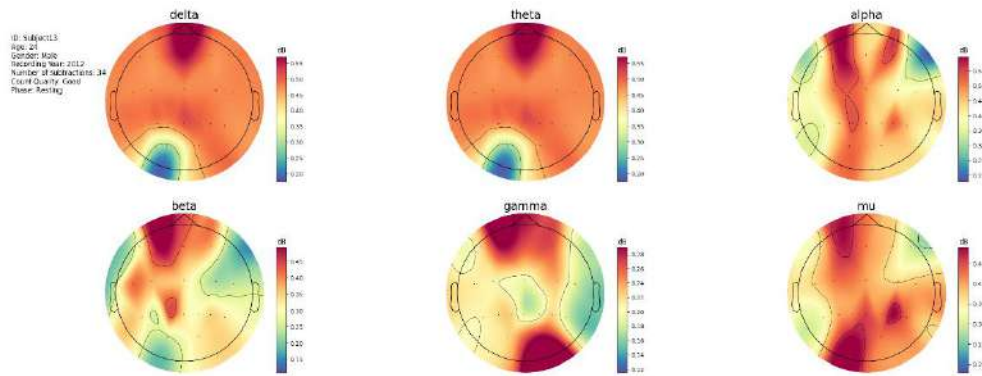
Слика 85 Мапирање мозга - прилог



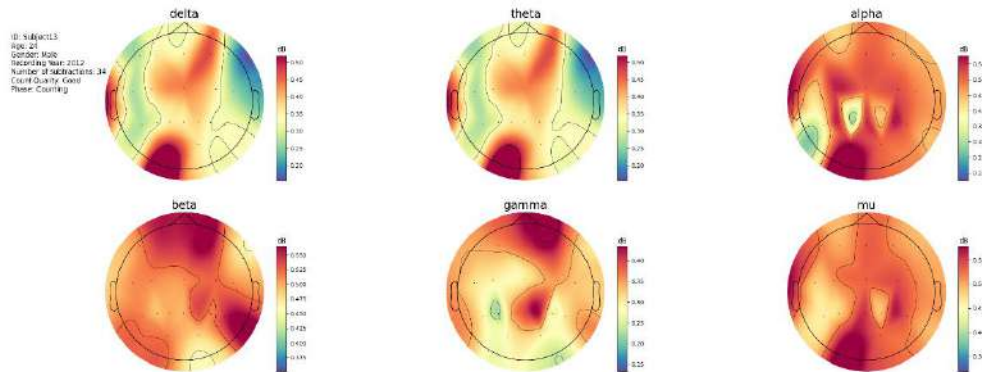
Слика 86 Мапирање мозга - прилог



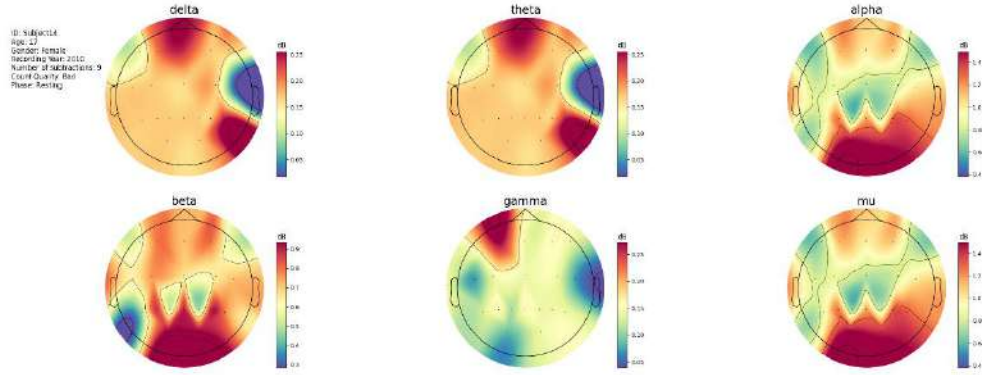
Слика 87 Мапирање мозга - прилог



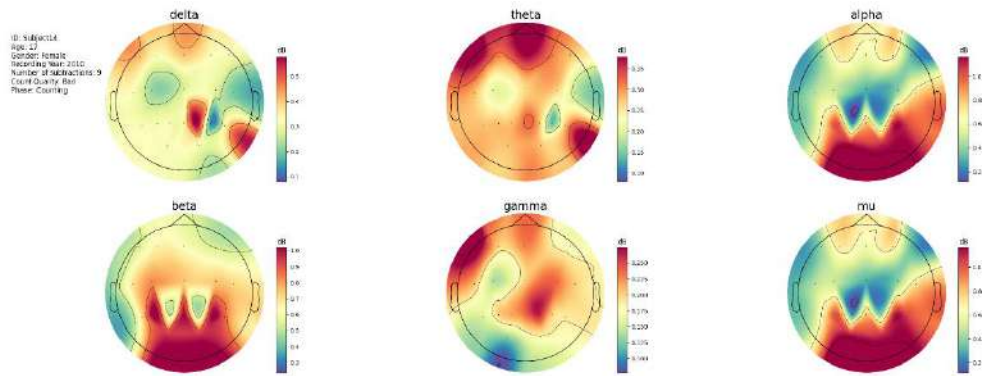
Слика 88 Мапирање мозга - прилог



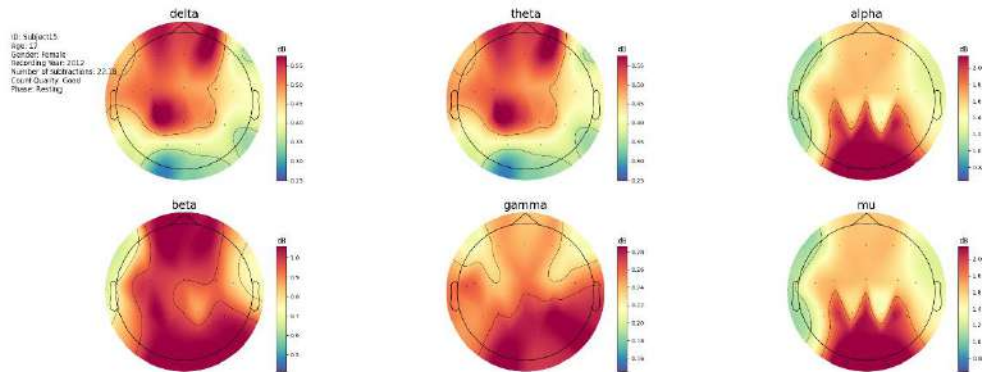
Слика 89 Мапирање мозга - прилог



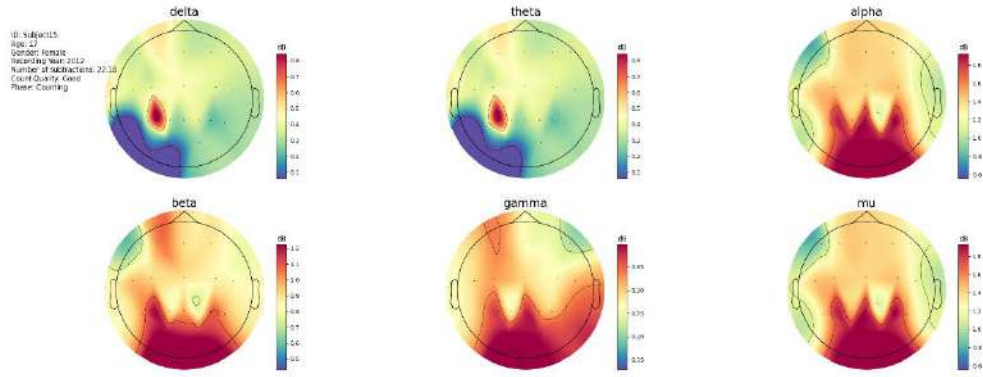
Слика 90 Мапирање мозга - прилог



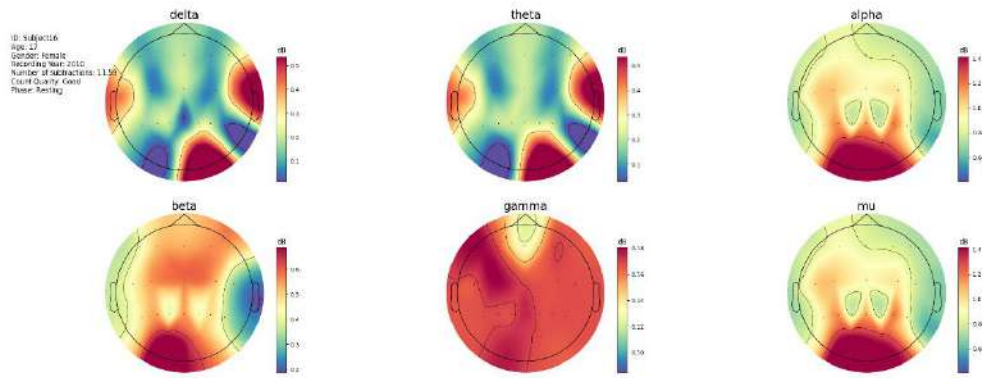
Слика 91 Мапирање мозга - прилог



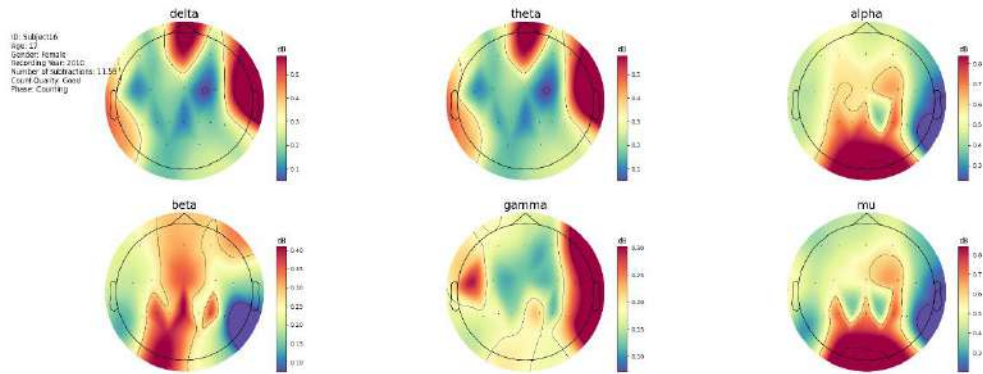
Слика 92 Мапирање мозга - прилог



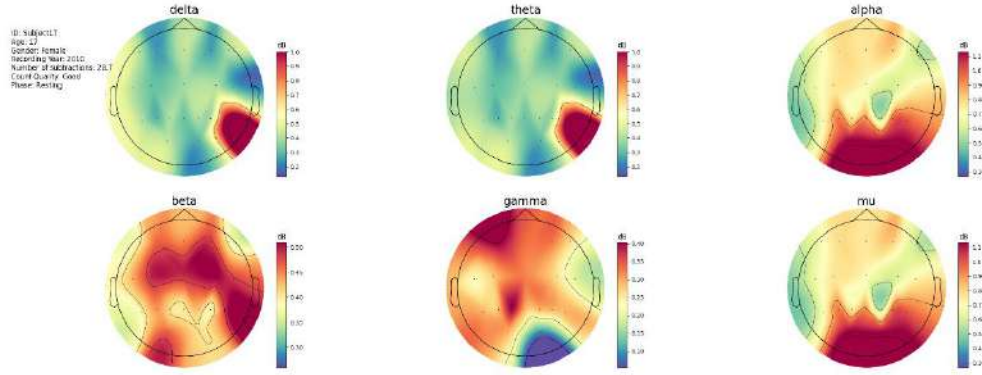
Слика 93 Мапирање мозга - прилог



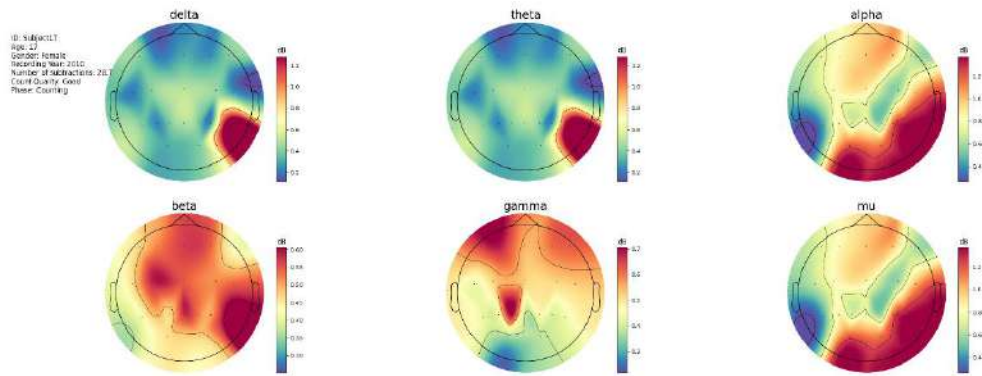
Слика 94 Мапирање мозга - прилог



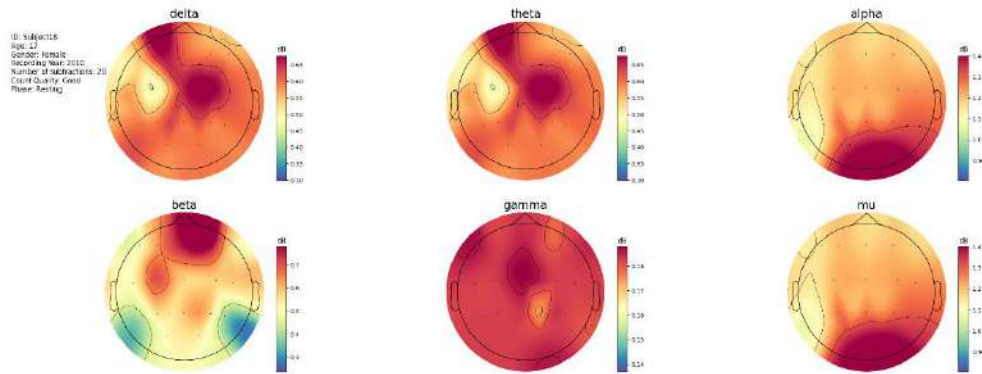
Слика 95 Мапирање мозга - прилог



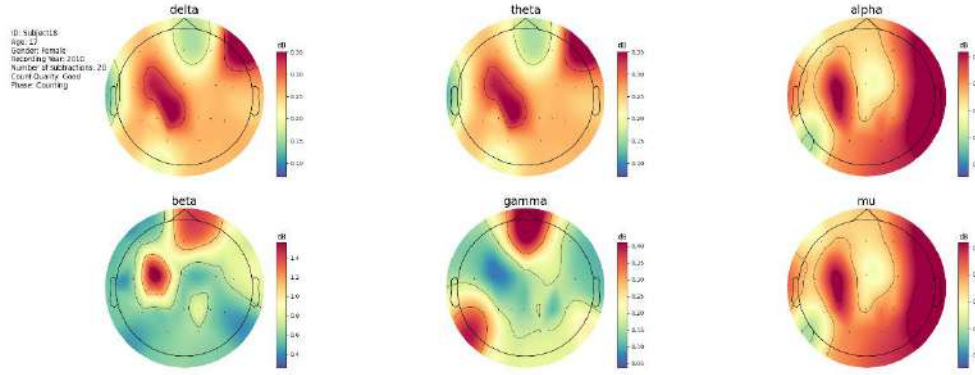
Слика 96 Мапирање мозга - прилог



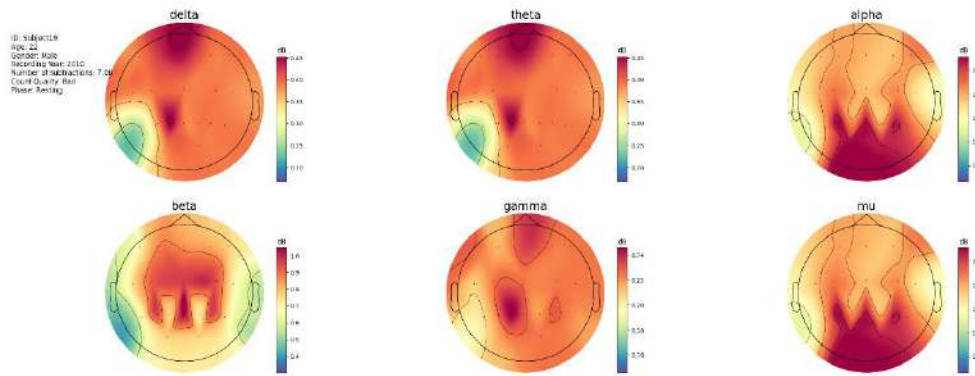
Слика 97 Мапирање мозга - прилог



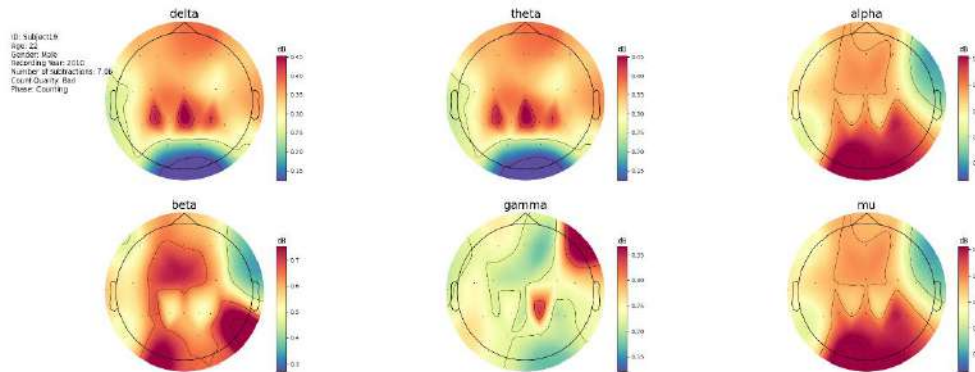
Слика 98 Мапирање мозга - прилог



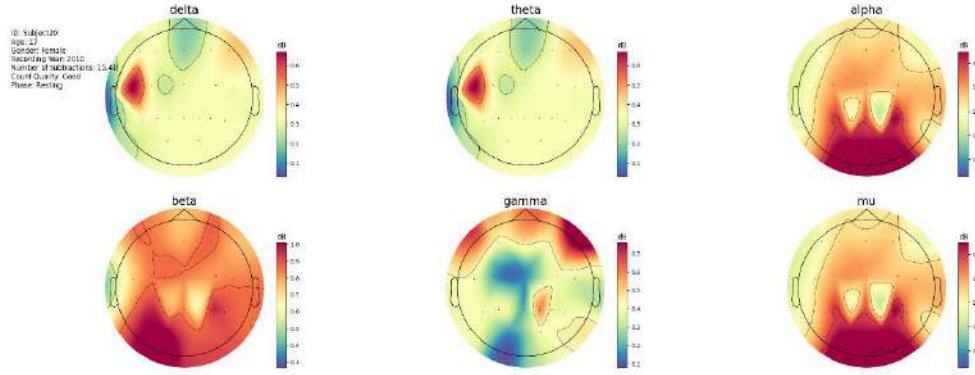
Слика 99 Мапирање мозга - прилог



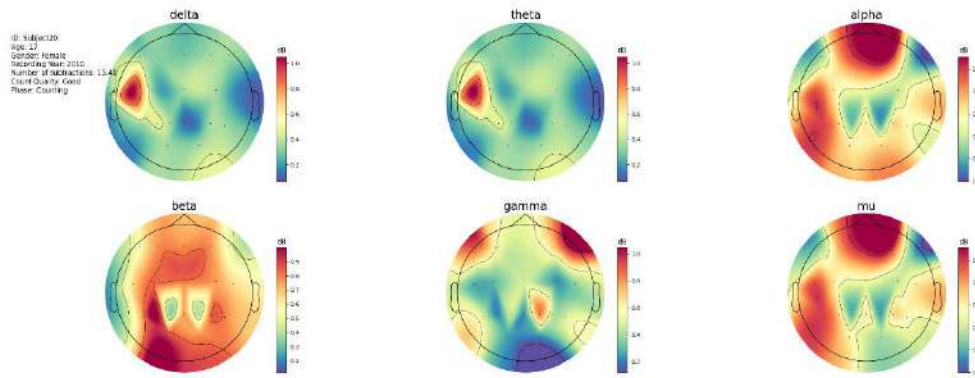
Слика 100 Мапирање мозга - прилог



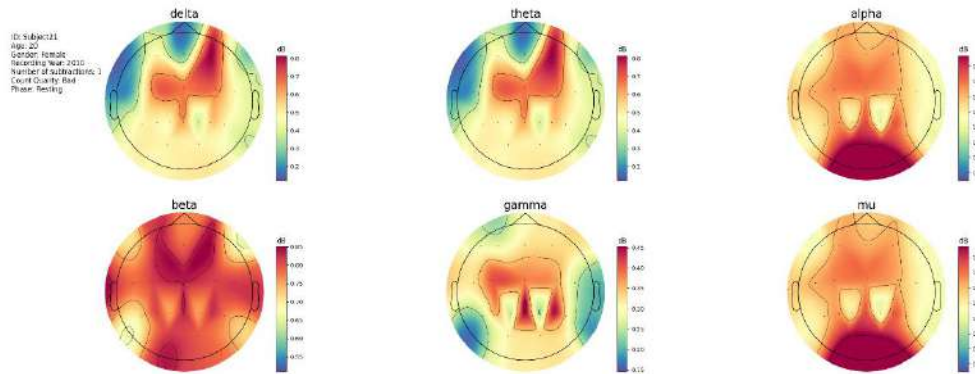
Слика 101 Мапирање мозга - прилог



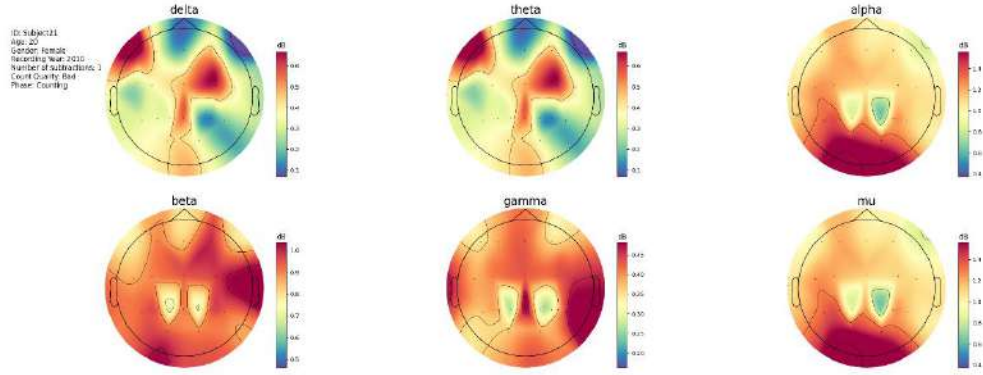
Слика 102 Мапирање мозга - прилог



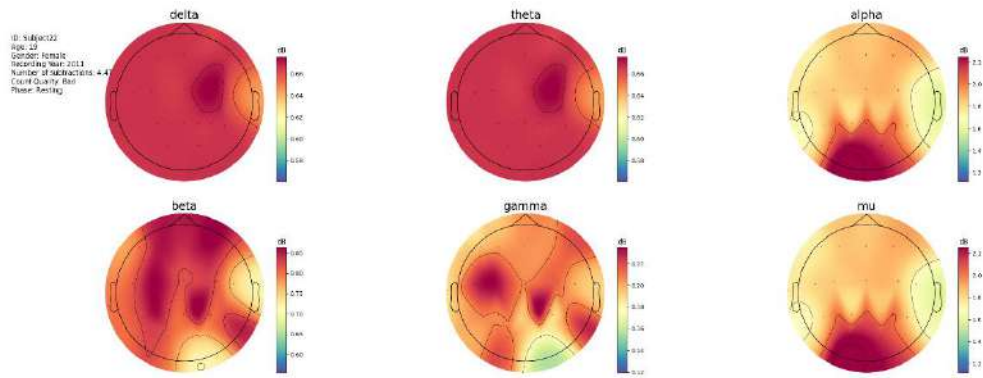
Слика 103 Мапирање мозга - прилог



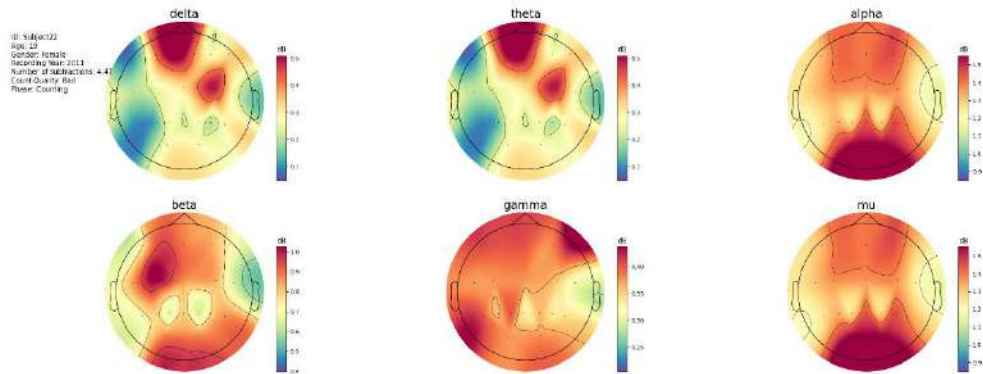
Слика 104 Мапирање мозга - прилог



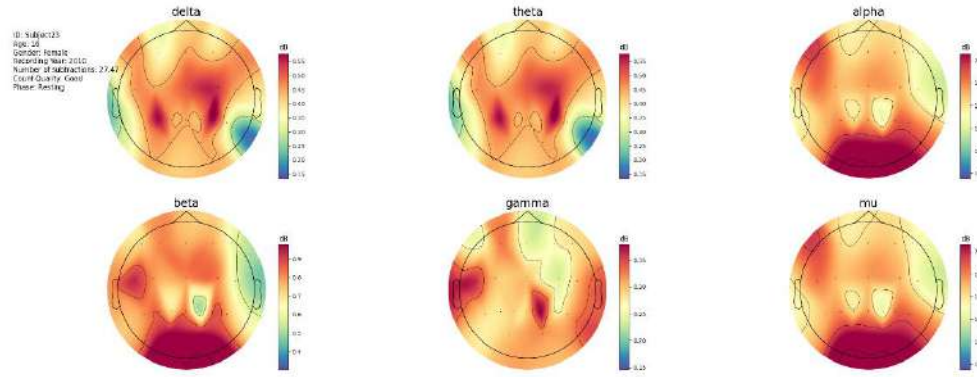
Слика 105 Мапирање мозга - прилог



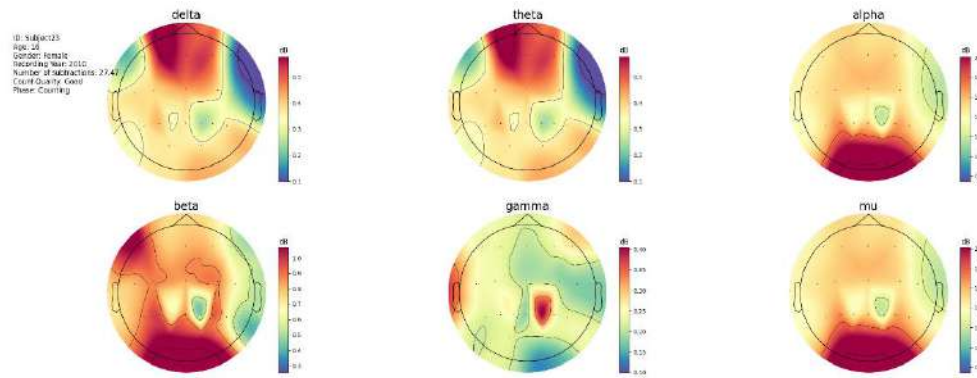
Слика 106 Мапирање мозга - прилог



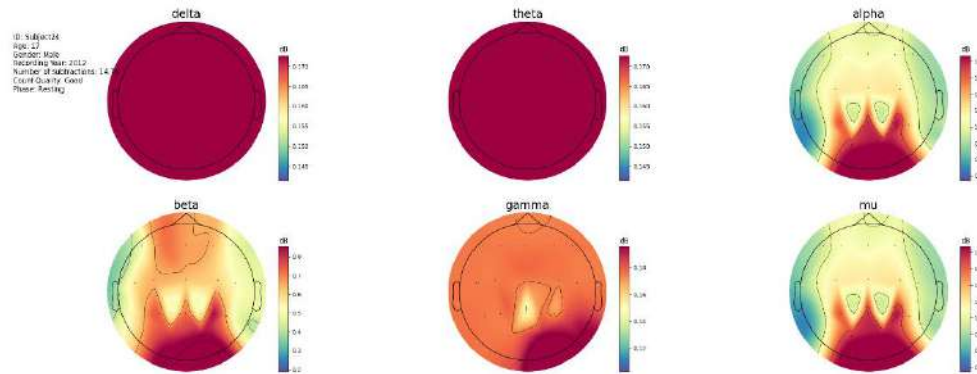
Слика 107 Мапирање мозга - прилог



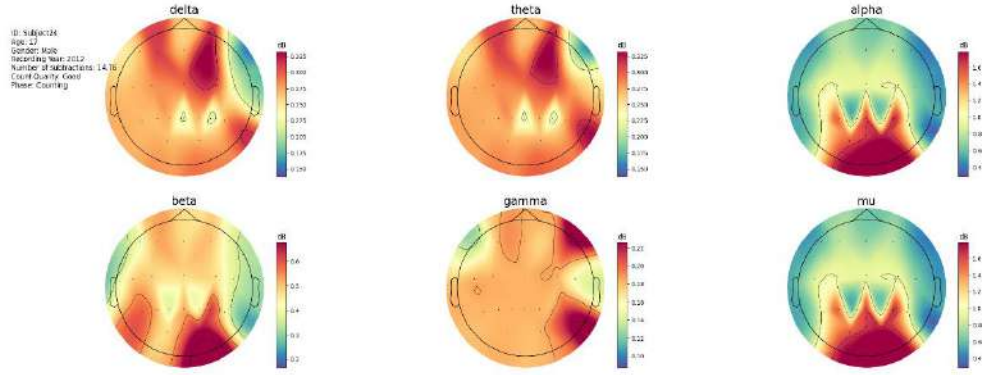
Слика 108 Мапирање мозга - прилог



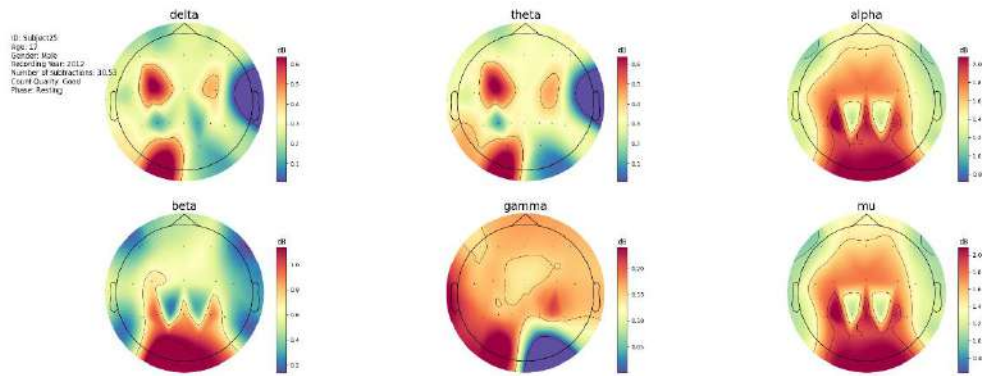
Слика 109 Мапирање мозга - прилог



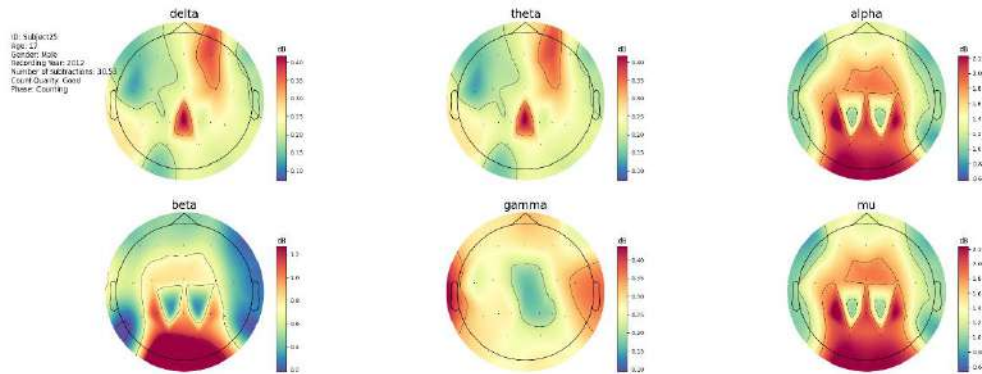
Слика 110 Мапирање мозга - прилог



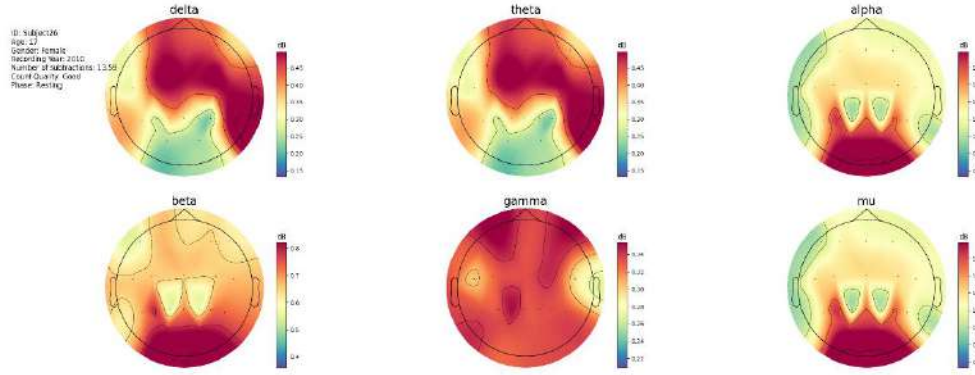
Слика 111 Мапирање мозга - прилог



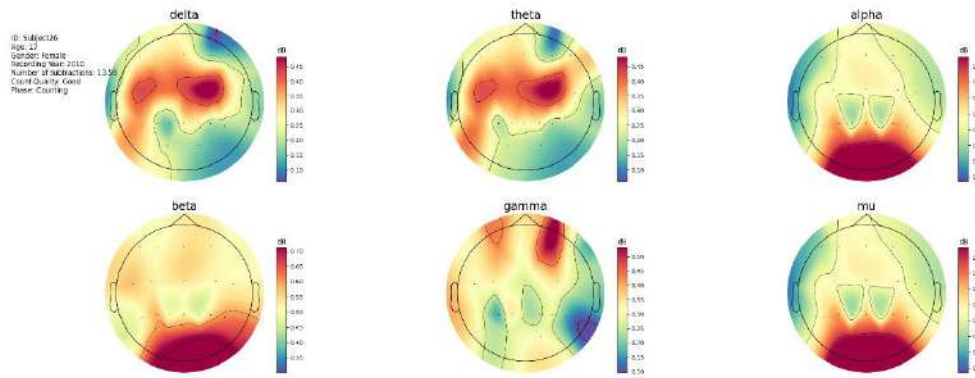
Слика 112 Мапирање мозга - прилог



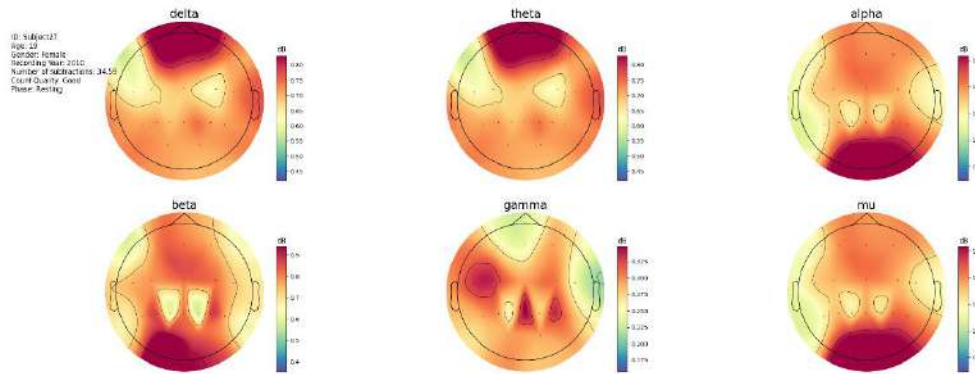
Слика 113 Мапирање мозга - прилог



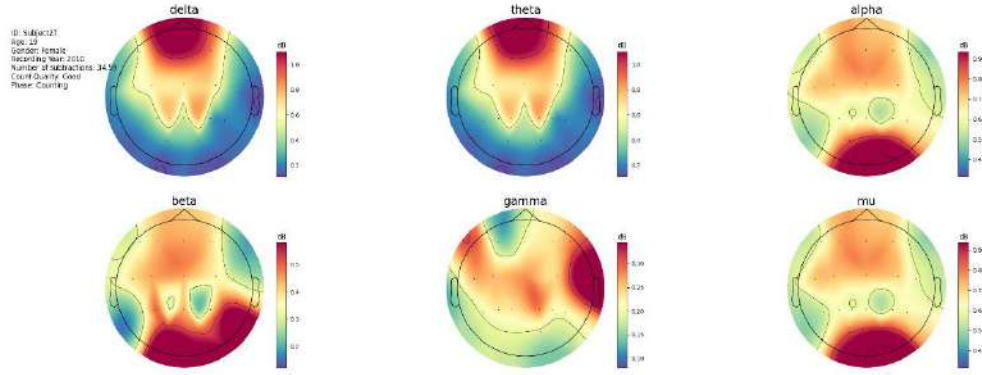
Слика 114 Мапирање мозга - прилог



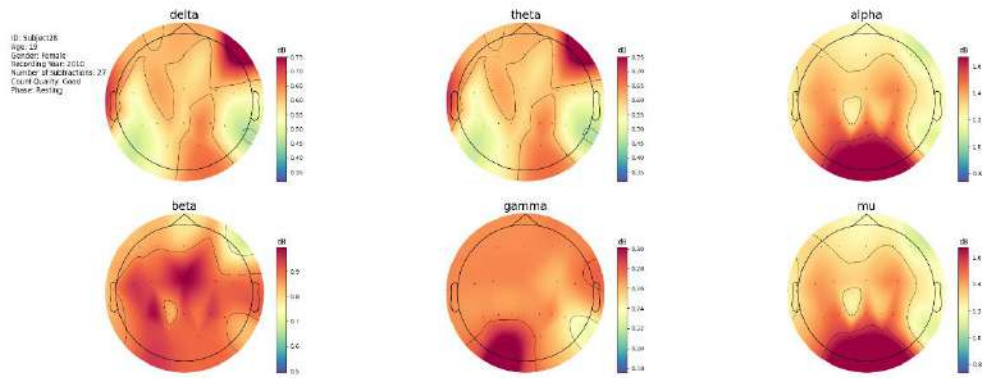
Слика 115 Мапирање мозга - прилог



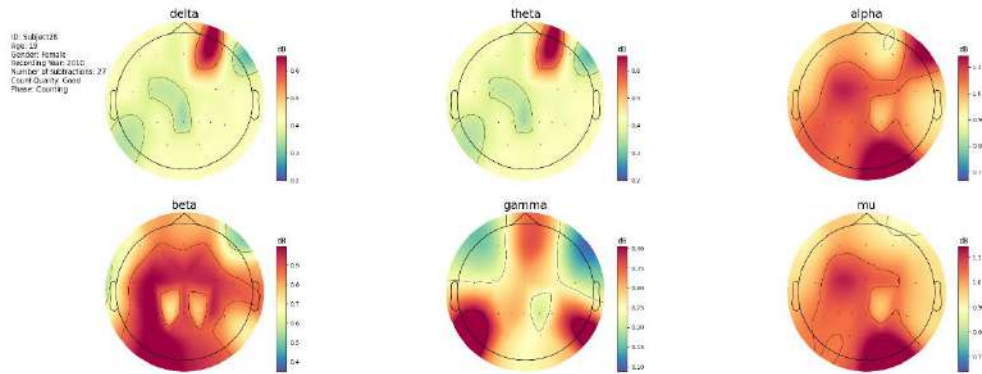
Слика 116 Мапирање мозга - прилог



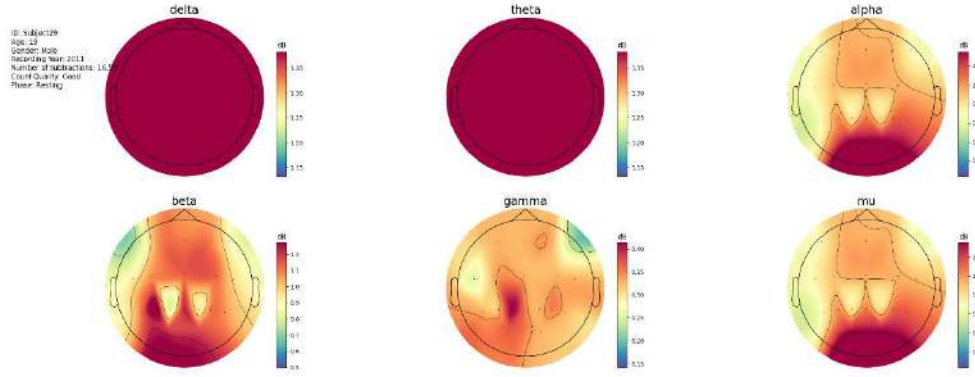
Слика 117 Мапирање мозга - прилог



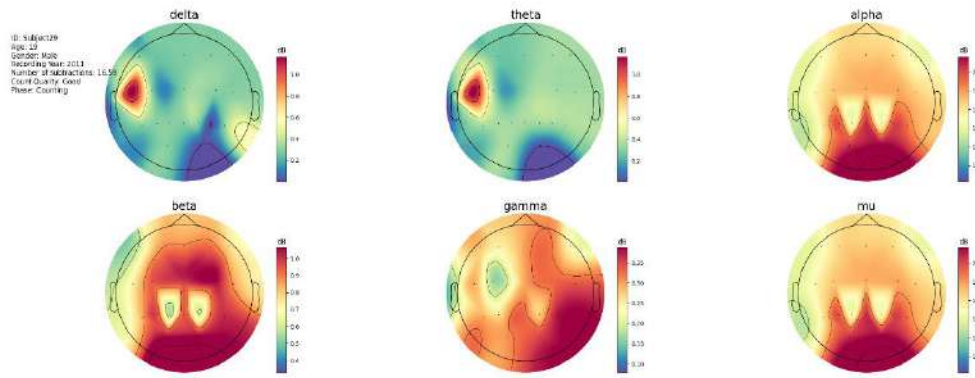
Слика 118 Мапирање мозга - прилог



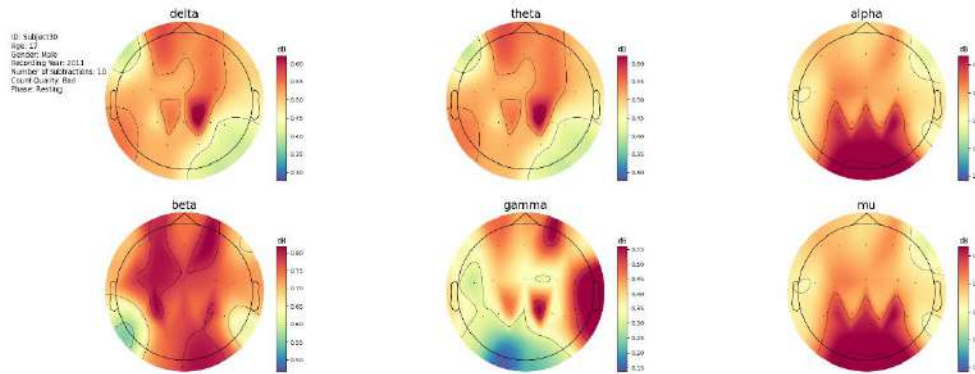
Слика 119 Мапирање мозга - прилог



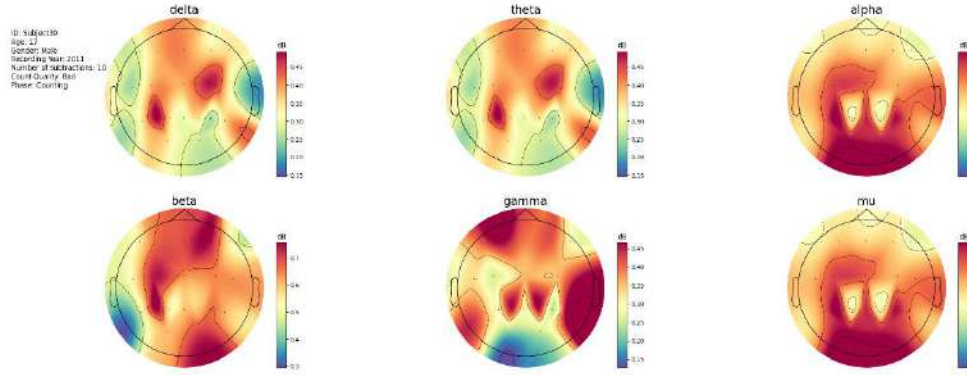
Слика 120 Мапирање мозга - прилог



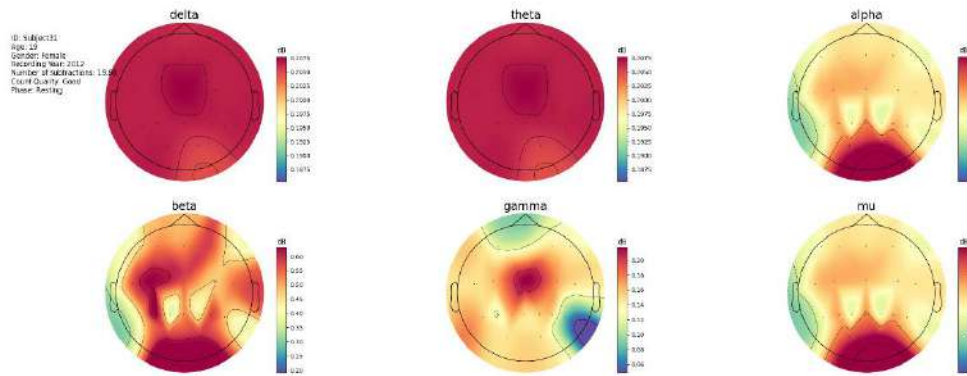
Слика 121 Мапирање мозга - прилог



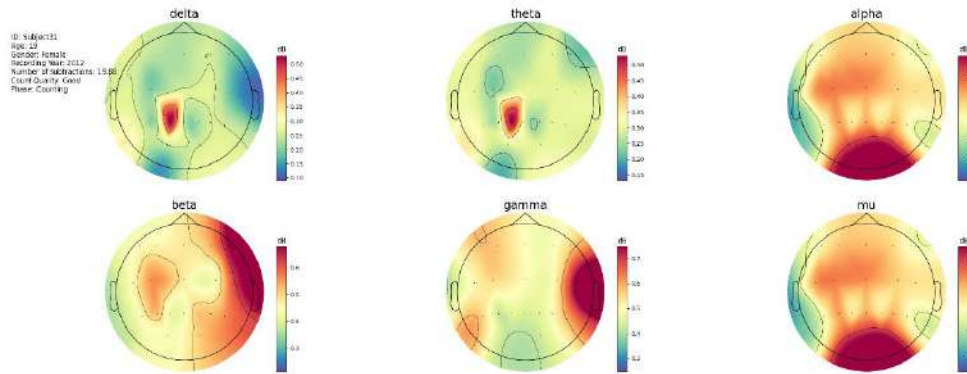
Слика 122 Мапирање мозга - прилог



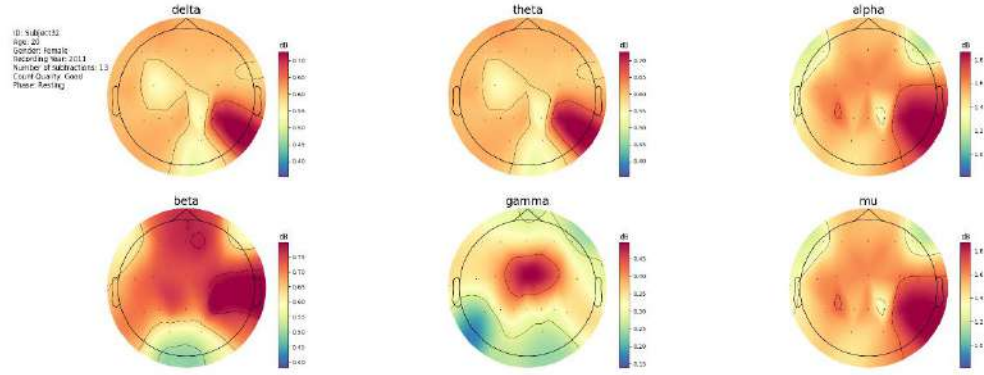
Слика 123 Мапирање мозга - прилог



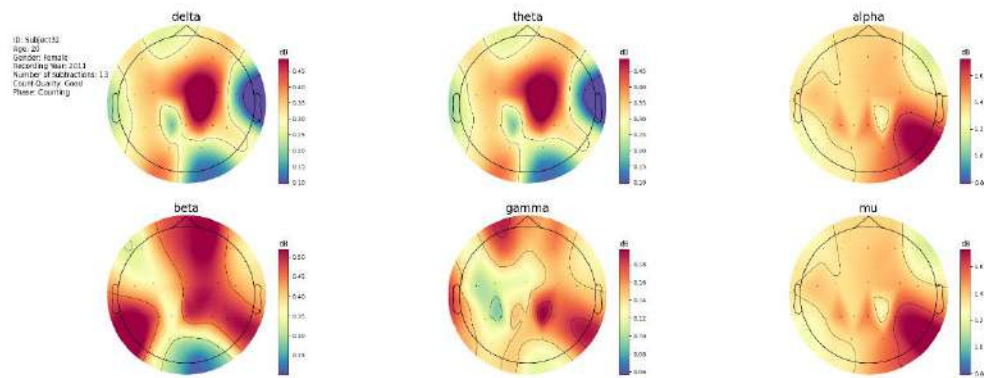
Слика 124 Мапирање мозга - прилог



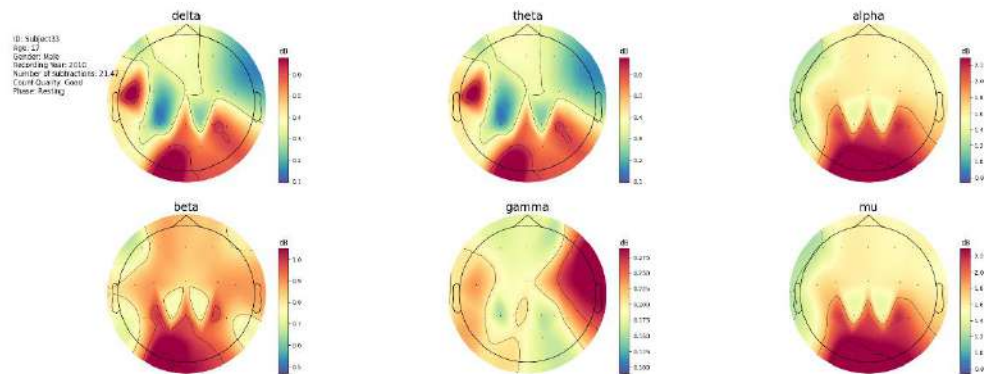
Слика 125 Мапирање мозга - прилог



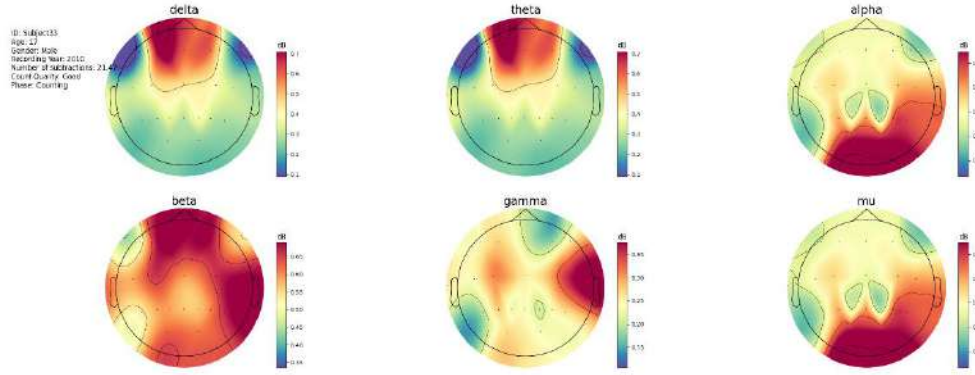
Слика 126 Мапирање мозга - прилог



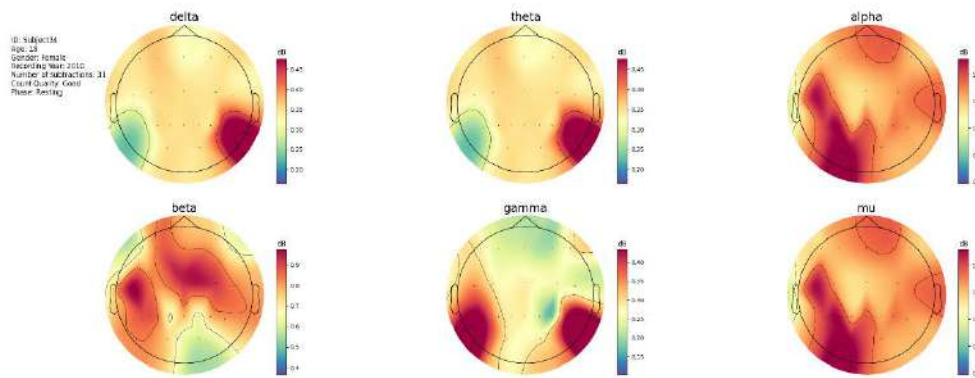
Слика 127 Мапирање мозга - прилог



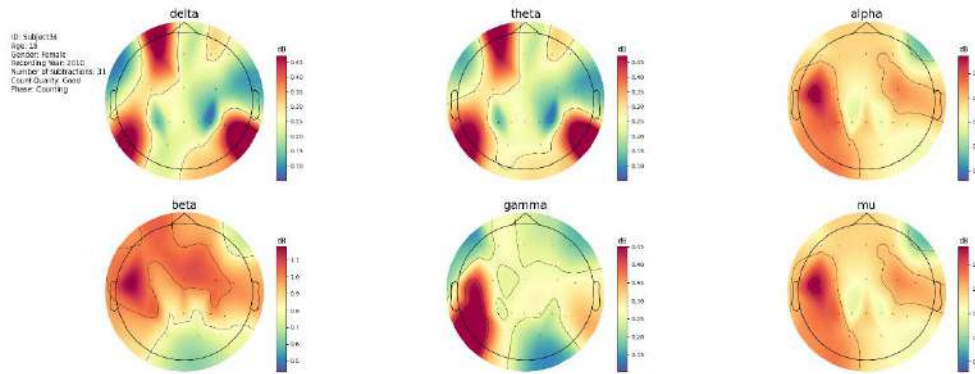
Слика 128 Мапирање мозга - прилог



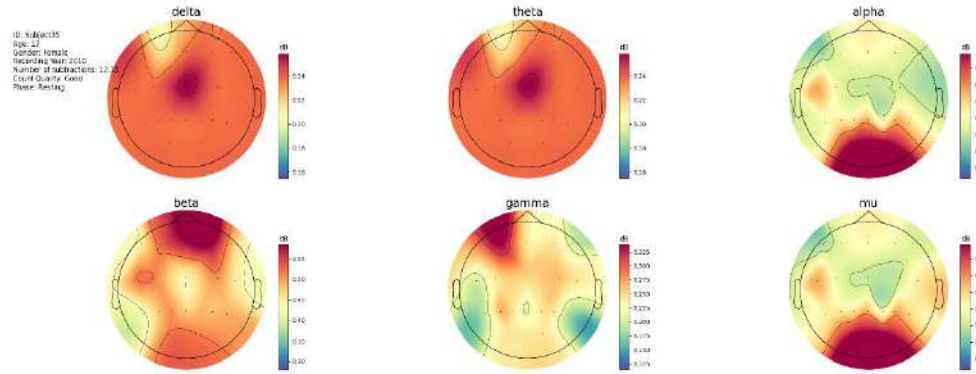
Слика 129 Мапирање мозга - прилог



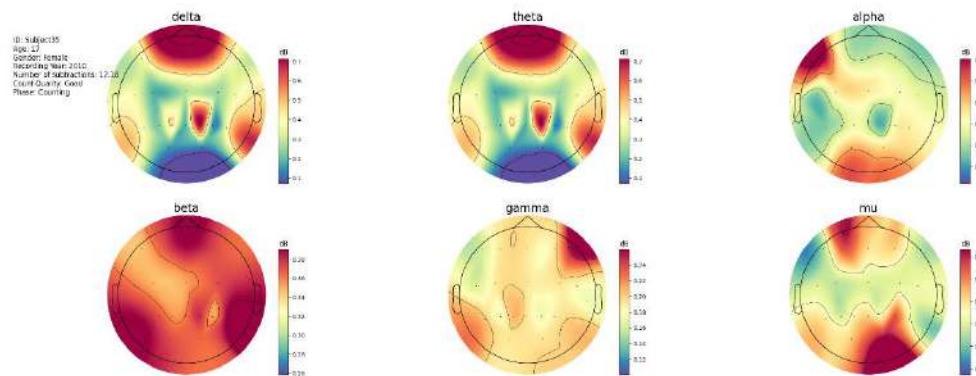
Слика 130 Мапирање мозга - прилог



Слика 131 Мапирање мозга - прилог



Слика 132 Мапирање мозга - прилог



Слика 133 Мапирање мозга – прилог

7.2 Кодови

7.2.1 Неурофидбек симулација

7.2.1.1 *MainUserInterfaceController.cs*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using MySql.Data.MySqlClient;

public class MainUserInterfaceController : MonoBehaviour
{
    public Dropdown dropID;
    public Dropdown dropLvl;

    private List<string> listID = new List<string>();
    public Button optionButton;
```

```
public Button dataButton;
public Button startButton;

public void Start()
{
    UpdateDropID();

    UpdateDropLvl();
    CheckEmptyFields();
}

public void UpdateDropID()
{
    string server = "127.0.0.1";
    string database = "mydb";
    string uid = "root";
    string password = "Koren123!";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
    database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

    //MySQLConnection connection = new MySqlConnection(connectionString);

    using (MySqlConnection connection = new
    MySqlConnection(connectionString))
    {
        connection.Open();
        MySqlCommand commandRead = connection.CreateCommand();

        commandRead.CommandText = "SELECT idExaminee FROM examinee;";

        using (MySql.Data.MySqlClient.MySqlDataReader reader =
        commandRead.ExecuteReader())
        {
            listID.Clear();
            listID.Add("Select ID:");
            while (reader.Read())
            {
                string tmpString = reader.GetString(0);
                //Debug.Log(tmpString);
                listID.Add(tmpString);
            }
        }
    }
}
```

```
    }

    dropID.ClearOptions();

    dropID.AddOptions(listID);
    connection.Close();
}

}

public void UpdateDropLvl()
{
    string server = "127.0.0.1";
    string database = "mydb";
    string uid = "root";
    string password = "Koren123!";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
    database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

    //MySQLConnection connection = new MySqlConnection(connectionString);

    using (MySqlConnection connection = new
    MySqlConnection(connectionString))
    {
        connection.Open();
        MySqlCommand commandRead = connection.CreateCommand();

        commandRead.CommandText = "SELECT expName FROM experimentType;";

        using (MySql.Data.MySqlClient.MySqlDataReader reader =
        commandRead.ExecuteReader())
        {
            listID.Clear();
            listID.Add("Select Experiment:");
            while (reader.Read())
            {
                string tmpString = reader.GetString(0);
                //Debug.Log(tmpString);
                listID.Add(tmpString);
            }
        }
    }
}
```

```
    }

    dropLvl.ClearOptions();

    dropLvl.AddOptions(listID);
    connection.Close();
}

}

public void CheckEmptyFields()
{
    if (dropID.value == 0)
    {
        optionButton.interactable = false;
        dataButton.interactable = false;
        startButton.interactable = false;
        dropLvl.interactable = false;
    }
    else if(dropID.value != 0 & dropLvl.value == 0)
    {
        optionButton.interactable = true;
        dataButton.interactable = true;
        startButton.interactable = false;
        dropLvl.interactable = true;
    }
    else if(dropID.value != 0 & dropLvl.value != 0)
    {
        optionButton.interactable = true;
        dataButton.interactable = true;
        startButton.interactable = true;
        dropLvl.interactable = true;
    }
}

public void StartSimulation()
{
    string server = "127.0.0.1";
    string database = "mydb";
    string uid = "root";
    string password = "Koren123!";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
    database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";
}
```

```
        using ( MySqlConnection connection = new
MySqlConnection(connectionString))
        {
            connection.Open();

            MySqlCommand getExpType = connection.CreateCommand();

            getExpType.CommandText = "SELECT idExperimentType, sceneName FROM
ExperimentType WHERE expName=@expName;";
            getExpType.Parameters.AddWithValue("@expName",
dropLvl.options[dropLvl.value].text);

            int expType;
            string sceneName;
            using( MySqlDataReader reader = getExpType.ExecuteReader())
            {
                if (reader.Read())
                {
                    expType = reader.GetInt32(0);
                    sceneName = reader.GetString(1);
                }
                else
                {
                    expType = -1;
                    sceneName = "";
                }
            }

            MySqlCommand command = connection.CreateCommand();

            command.CommandText = "INSERT INTO Experiments (idExaminee,
experimentType, experimentDateTime) VALUES (@idExaminee, @expType,
CURRENT_TIME);";
            command.Parameters.AddWithValue("@idExaminee",
dropID.options[dropID.value].text);
            command.Parameters.AddWithValue("@expType", expType);
            command.ExecuteNonQuery();

            connection.Close();

            SceneManager.LoadScene(sceneName);
        }
    }
```

```
}  
}
```

7.2.1.2 OptionEngine.cs

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.UI;  
using MySql.Data.MySqlClient;  
using TMPPro;  
  
public class OptionEngine : MonoBehaviour  
{  
    public Dropdown dropID;  
    public Dropdown verticalM;  
    public Dropdown horizontalM;  
    public Dropdown transversalM;  
  
    public Dropdown verticalR;  
    public Dropdown horizontalR;  
  
    public Dropdown dissapear;  
  
    public TMP_InputField mSpeed;  
    public TMP_InputField rSpeed;  
  
    public Slider sound;  
    public Slider music;  
  
    private string id;  
  
    void Start()  
    {  
  
    }  
  
    public void GetOptions()  
    {  
        id = dropID.options[dropID.value].text;  
  
        string server = "127.0.0.1";  
        string database = "mydb";  
    }  
}
```



```
string uid = "root";
string password = ""; //Uklonjeno zbog privatnosti
string connectionString;
connectionString = "SERVER=" + server + ";" + "DATABASE=" +
database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

//MySQLConnection connection = new MySqlConnection(connectionString);

using (MySqlConnection connection = new
MySqlConnection(connectionString))
{
    connection.Open();
    MySqlCommand commandRead = connection.CreateCommand();

    commandRead.CommandText = "SELECT Sound, Music, HorizontalMovement,
VerticalMovement, TransversalMovement, HorizontalRotation, VerticalRotation,
Dissapear, MovementSpeed, RotationalSpeed FROM SimOptions WHERE
Examinee_idExaminee = @fk;";
    commandRead.Parameters.AddWithValue("@fk", id);

    using (MySql.Data.MySqlClient.MySqlDataReader reader =
commandRead.ExecuteReader())
    {
        if (reader.Read())
        {
            sound.value = reader.GetFloat(0);
            music.value = reader.GetFloat(1);

            horizontalM.value = reader.GetInt32(2);
            verticalM.value = reader.GetInt32(3);
            transversalM.value = reader.GetInt32(4);

            horizontalR.value = reader.GetInt32(5);
            verticalR.value = reader.GetInt32(6);

            dissapear.value = reader.GetInt32(7);

            mSpeed.text = reader.GetString(8);
            rSpeed.text = reader.GetString(9);

        }
    }
}
```

```
        connection.Close();
    }
}

public void SetOptions()
{
    id = dropID.options[dropID.value].text;

    string server = "127.0.0.1";
    string database = "mydb";
    string uid = "root";
    string password = "";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
    database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

    //MySQLConnection connection = new MySqlConnection(connectionString);

    using (MySqlConnection connection = new
    MySqlConnection(connectionString))
    {
        connection.Open();
        MySqlCommand commandWrite = connection.CreateCommand();

        commandWrite.CommandText = "UPDATE SimOptions SET Sound=@sound,
    Music=@music, HorizontalMovement=@horizontalM, VerticalMovement=@verticalM,
    TransversalMovement=@transversalM, HorizontalRotation=@horizontalR,
    VerticalRotation=@verticalR, Dissapear=@dissapear, MovementSpeed=@mSpeed,
    RotationalSpeed=@rSpeed WHERE Examinee_idExaminee = @fk;";

        commandWrite.Parameters.AddWithValue("@sound", sound.value);
        commandWrite.Parameters.AddWithValue("@music", music.value);
        commandWrite.Parameters.AddWithValue("@horizontalM",
horizontalM.value);
        commandWrite.Parameters.AddWithValue("@verticalM", verticalM.value);
        commandWrite.Parameters.AddWithValue("@transversalM",
transversalM.value);
        commandWrite.Parameters.AddWithValue("@horizontalR",
horizontalR.value);
        commandWrite.Parameters.AddWithValue("@verticalR", verticalR.value);
```

```
        commandWrite.Parameters.AddWithValue("@dissapear", dissapear.value);
        commandWrite.Parameters.AddWithValue("@mSpeed",
float.Parse(mSpeed.text.Replace(".", ",")));
        commandWrite.Parameters.AddWithValue("@rSpeed",
float.Parse(rSpeed.text.Replace(".", ",")));
        commandWrite.Parameters.AddWithValue("@fk", id);

        commandWrite.ExecuteNonQuery();

        connection.Close();
    }
}
```

7.2.1.3 RegistrationEngine.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using MySql.Data.MySqlClient;
using TMPPro;
using UnityEngine.UI;

public class RegistrationEngine : MonoBehaviour
{
    public TMP_InputField inName;
    public TMP_InputField inLastName;
    public TMP_InputField inYear;
    public TMP_InputField inMonth;
    public TMP_InputField inDay;
    public TMP_InputField inOccupation;
    public TMP_Text infoText;
    public Button regButton;

    private string examineeID;

    private void Awake()
    {
        CheckEmptyRegisterField();
    }
    void Update()
    {
    }
}
```

```
public void RegisterExaminee()
{
    string server = "127.0.0.1";
    string database = "mydb";
    string uid = "root";
    string password = "";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
    database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

    //MySqlConnection connection = new MySqlConnection(connectionString);

    using (MySqlConnection connection = new
    MySqlConnection(connectionString))
    {

        //infoText.text = connection.ConnectionString;
        connection.Open();

        MySqlCommand command = connection.CreateCommand();
        command.CommandText = "INSERT INTO examinee (FirstName, LastName,
        BirthDate, Occupation) VALUES (@firstName, @lastName, @birthDate, @occupation)";
        command.Parameters.AddWithValue("@firstName", inName.text);
        command.Parameters.AddWithValue("@lastName", inLastName.text);
        command.Parameters.AddWithValue("@birthDate", inYear.text + "-" +
        inMonth.text + "-" + inDay.text);
        command.Parameters.AddWithValue("@occupation", inOccupation.text);
        //command.CommandTimeout = 3;

        command.ExecuteNonQuery();

        MySqlCommand commandRead = connection.CreateCommand();

        commandRead.CommandText = "SELECT idExaminee FROM examinee WHERE
        (FirstName=@firstName AND LastName=@lastName AND Occupation=@occupation)";
        commandRead.Parameters.AddWithValue("@firstName", inName.text);
        commandRead.Parameters.AddWithValue("@lastName", inLastName.text);
        commandRead.Parameters.AddWithValue("@occupation",
        inOccupation.text);

        using(MySql.Data.MySqlClient.MySqlDataReader reader =
        commandRead.ExecuteReader())
```

```
        {
            if (reader.Read())
            {
                examineeID = reader.GetString(0);
            }
            else
            {
                examineeID = "";
            }
        }

        infoText.text = "Examinee ID: " + examineeID;

        MySqlCommand commandInsertOption = connection.CreateCommand();
        commandInsertOption.CommandText = "INSERT INTO SimOptions
(Examinee_idExaminee) VALUES (@fk);";
        commandInsertOption.Parameters.AddWithValue("@fk", examineeID);
        commandInsertOption.ExecuteNonQuery();

        connection.Close();
    }
}

public void CheckEmptyRegisterField()
{
    if (inName.text == "" || inLastName.text == "" || inYear.text == "" ||
inMonth.text == "" || inDay.text == "" || inOccupation.text == "")
    {
        regButton.interactable = false;
    }
    else
    {
        regButton.interactable = true;
    }
}
}
```

7.2.1.4 Collectable.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
public class Collectable : MonoBehaviour
{
    public Animator animator;

    private void OnTriggerEnter(Collider other)
    {
        GameObject.Find("GameManager").GetComponent<GameManager>().OnCollectedBubble();

        animator.SetTrigger("Collide");

        Destroy(gameObject, 0.1f);
    }
}
```

7.2.1.5 CortexHandler.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using CortexAccess;
using Newtonsoft.Json.Linq;

public class CortexHandler : MonoBehaviour
{
    public PlayerMovement player;

    private DataStreamManager dsm;

    private void Start()
    {
        dsm = new DataStreamManager();
        dsm.AddStreams("dev");
        dsm.AddStreams("com");

        dsm.OnSubscribed += SubscribedOK;
        dsm.OnDeviceDataReceived += OnDeviceDataReceived;
        dsm.OnMentalCommandReceived += OnMentalCommandReceived;
        dsm.OnPerfDataReceived += OnPerfDataReceived;

        dsm.Start(Config.Licence, true);
    }
}
```



```
private void OnApplicationQuit()
{
    dsm.CloseSession();
}

private static void SubscribedOK(object sender, Dictionary<string, JArray> e)
{
    foreach (string key in e.Keys)
    {
        if (key == "dev")
        {
            ArrayList header = e[key].ToObject<ArrayList>();
            header.Insert(0, "Timestamp");
            string outyougo = "";
            foreach (var item in header)
                outyougo += item + ", ";

            Debug.Log(outyougo);
        }
        else if (key == "mot")
        {
            ArrayList header = e[key].ToObject<ArrayList>();
            string outyougo = "";
            foreach (var item in header)
                outyougo += item + ", ";

            Debug.Log(outyougo);
        }
        else if (key == "com")
        {
            ArrayList header = e[key].ToObject<ArrayList>();
            header.Insert(0, "Timestamp");
            string outyougo = "";
            foreach (var item in header)
                outyougo += item + ", ";

            Debug.Log(outyougo);
        }
        else if (key == "met")
        {
            ArrayList header = e[key].ToObject<ArrayList>();
            header.Insert(0, "Timestamp");
            string outyougo = "";
            foreach (var item in header)
```

```
        outyougo += item + ", ";
    }
    Debug.Log(outyougo);
}
}

private static void OnDeviceDataReceived(object sender, ArrayList data)
{
    string outyougo = "";
    foreach (var element in data)
        outyougo += element.ToString() + ", ";

    //Debug.Log(outyougo);
}

private void OnMentalCommandReceived(object sender, ArrayList data)
{
    string outyougo = "";
    foreach (var element in data)
        outyougo += element.ToString() + ", ";

    //Debug.Log(data);
    float commandInput = (float)Math.Round(float.Parse(data[2].ToString()),
2);

    player.MentalCommandLogger(data[1].ToString(), commandInput);

    if (data[1].ToString() == "neutral")
    {
        //player.SetXInput(0);
        //player.SetYInput(0);
        //player.SetZInput(0);
    }

    else if (data[1].ToString() == "push")
    {
        player.SetZInput(commandInput);
    }
    else if (data[1].ToString() == "pull")
    {
        player.SetZInput(-commandInput);
    }
}
```

```
else if (data[1].ToString() == "lift")
{
    player.SetYInput(commandInput);
}
else if (data[1].ToString() == "drop")
{
    player.SetYInput(-commandInput);
}

else if (data[1].ToString() == "right")
{
    player.SetXInput(commandInput);
}
else if (data[1].ToString() == "left")
{
    player.SetXInput(-commandInput);
}

else if (data[1].ToString() == "rotateLeft")
{
    player.SetYRInput(-commandInput);
}
else if (data[1].ToString() == "rotateRight")
{
    player.SetYRInput(commandInput);
}

else if (data[1].ToString() == "rotateClockwise")
{
    player.SetZRInput(commandInput);
}
else if (data[1].ToString() == "rotateCounterClockwise")
{
    player.SetZRInput(-commandInput);
}

else if (data[1].ToString() == "rotateForwards")
{
    player.SetXRInput(-commandInput);
}
else if (data[1].ToString() == "rotateReverse")
{
```

```
        player.SetXRInput(commandInput);
    }

    else if (data[1].ToString() == "disappear")
    {
        player.SetDisInput(commandInput);
    }

}

private void OnPerfDataReceived(object sender, ArrayList data)
{
    string outyougo = "";
    foreach (var element in data)
        outyougo += element.ToString() + ", ";

    Debug.Log(outyougo);
}
}
```

7.2.1.6 *GameManager.cs*

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class GameManager : MonoBehaviour
{
    public GameOptionEngine gameOptionEngine;
    public PlayerMovement player;
    public int bubbleScore = 0;
    public TMP_Text inputText;
    private int exaID;

    private void Awake()
    {
```

```
        exaID = player.GetExaID();
    }

    void Update()
    {
        inputText.text = bubbleScore.ToString();
    }

    public void OnCollectedBubble()
    {
        bubbleScore += 1;
    }

    public void PauseGame()
    {
        if
(GameObject.Find("Canvas").transform.GetChild(0).gameObject.activeSelf)
        {
            GameObject.Find("Canvas").transform.GetChild(0).gameObject.SetActive(
false);
            GameObject.Find("Canvas").transform.GetChild(1).gameObject.SetActive(
true);

            Time.timeScale = 0;

            Cursor.visible = true;
        }
        else
        {
            GameObject.Find("Canvas").transform.GetChild(0).gameObject.SetActive(
true);
            GameObject.Find("Canvas").transform.GetChild(1).gameObject.SetActive(
false);

            Time.timeScale = 1;
            Cursor.visible = false;
        }
    }
}
```

```
    }  
    public void ReturnToMainMenu()  
    {  
        Time.timeScale = 1;  
        player.SaveMentalLog();  
        SceneManager.LoadScene("UserInterface");  
    }  
}
```

7.2.1.7 GameOptionEngine.cs

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.UI;  
using MySql.Data.MySqlClient;  
using TMPPro;  
  
public class GameOptionEngine : MonoBehaviour  
{  
    public PlayerMovement player;  
    public Dropdown verticalM;  
    public Dropdown horizontalM;  
    public Dropdown transversalM;  
  
    public Dropdown verticalR;  
    public Dropdown horizontalR;  
  
    public Dropdown dissappear;  
  
    public TMP_InputField mSpeed;  
    public TMP_InputField rSpeed;  
  
    public Slider sound;  
    public Slider music;  
  
    private string id;  
  
    void Start()  
    {  
  
    }  
}
```

```
public void GetOptions()
{
    id = player.GetExaID().ToString();

    string server = "127.0.0.1";
    string database = "mydb";
    string uid = "root";
    string password = "Koren123!";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
    database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

    //MySQLConnection connection = new MySqlConnection(connectionString);

    using (MySqlConnection connection = new
    MySqlConnection(connectionString))
    {
        connection.Open();
        MySqlCommand commandRead = connection.CreateCommand();

        commandRead.CommandText = "SELECT Sound, Music, HorizontalMovement,
        VerticalMovement, TransversalMovement, HorizontalRotation, VerticalRotation,
        Dissapear, MovementSpeed, RotationalSpeed FROM SimOptions WHERE
        Examinee_idExaminee = @fk";
        commandRead.Parameters.AddWithValue("@fk", id);

        using (MySql.Data.MySqlClient.MySqlDataReader reader =
        commandRead.ExecuteReader())
        {
            if (reader.Read())
            {
                sound.value = reader.GetFloat(0);
                music.value = reader.GetFloat(1);

                horizontalM.value = reader.GetInt32(2);
                verticalM.value = reader.GetInt32(3);
                transversalM.value = reader.GetInt32(4);

                horizontalR.value = reader.GetInt32(5);
                verticalR.value = reader.GetInt32(6);

                dissapear.value = reader.GetInt32(7);

                mSpeed.text = reader.GetString(8);
            }
        }
    }
}
```



```
        rSpeed.text = reader.GetString(9);

    }

}

    connection.Close();
}
}

public void SetOptions()
{
    id = player.GetExaID().ToString();

    string server = "127.0.0.1";
    string database = "mydb";
    string uid = "root";
    string password = "Koren123!";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "DATABASE=" +
    database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

    //MySQLConnection connection = new MySqlConnection(connectionString);

    using (MySqlConnection connection = new
    MySqlConnection(connectionString))
    {
        connection.Open();
        MySqlCommand commandWrite = connection.CreateCommand();

        commandWrite.CommandText = "UPDATE SimOptions SET Sound=@sound,
    Music=@music, HorizontalMovement=@horizontalM, VerticalMovement=@verticalM,
    TransversalMovement=@transversalM, HorizontalRotation=@horizontalR,
    VerticalRotation=@verticalR, Dissapear=@dissapear, MovementSpeed=@mSpeed,
    RotationalSpeed=@rSpeed WHERE Examinee_idExaminee = @fk;";

        commandWrite.Parameters.AddWithValue("@sound", sound.value);
        commandWrite.Parameters.AddWithValue("@music", music.value);
        commandWrite.Parameters.AddWithValue("@horizontalM",
horizontalM.value);
        commandWrite.Parameters.AddWithValue("@verticalM", verticalM.value);
        commandWrite.Parameters.AddWithValue("@transversalM",
transversalM.value);
    }
}
```

```
        commandWrite.Parameters.AddWithValue("@horizontalR",
horizontalR.value);
        commandWrite.Parameters.AddWithValue("@verticalR", verticalR.value);
        commandWrite.Parameters.AddWithValue("@dissapear", dissapear.value);
        commandWrite.Parameters.AddWithValue("@mSpeed",
float.Parse(mSpeed.text.Replace(".", ",")));
        commandWrite.Parameters.AddWithValue("@rSpeed",
float.Parse(rSpeed.text.Replace(".", ",")));
        commandWrite.Parameters.AddWithValue("@fk", id);

        commandWrite.ExecuteNonQuery();

        player.UpdateParameters();
        connection.Close();
    }
}
}
```

7.2.1.8 *PlayerMovement.cs*

```
using MySql.Data.MySqlClient;
using UnityEngine;
using UnityEngine.InputSystem;
using UnityEngine.UI;
using System.Collections.Generic;
using System;

public class PlayerMovement : MonoBehaviour
{
    public Slider slider;

    private Vector2 movementInput;
    private Vector2 rotateInput;
    private float elevateInput;

    private Rigidbody rigidBody;

    public Transform groundCheck;
    public LayerMask groundMask;
    public float groundDistance = 0.4f;

    private float flySpeed;
    private float rotateSpeed;
```

```
public float gravity = -20f;
public float maxHeight = 50f;

private float xInput = 0f;
private float yInput = 0f;
private float zInput = 0f;

private float xRInput = 0f;
private float yRInput = 0f;
private float zRInput = 0f;

private float disInput = 0f;

public Animator animator;

private float soundVolume;
private float musicVolume;

private int horizontalMToggle;
private int verticalMToggle;
private int transversalMToggle;

private int horizontalRToggle;
private int verticalRToggle;

private int dissapearToggle;

public int exaID;
public float startTime;
public float elapsedTime;
private MentalLogs mentalLogs;
private void OnMove(InputValue value)
{
    movementInput = value.Get<Vector2>();
}

private void OnRotate(InputValue value)
{
    rotateInput = value.Get<Vector2>();
}

private void OnElevate(InputValue value)
{
    elevateInput = value.Get<float>();
}
```

```
private void OnInputPause( InputValue value)
{
    GameObject.Find("GameManager").GetComponent<GameManager>().PauseGame();
}

private void OnRestartOrientation( InputValue value)
{
    //Debug.Log("Restart");
    transform.localRotation = Quaternion.Euler(0,
transform.rotation.eulerAngles.y, 0);

}

void Awake()
{
    rigidBody = GetComponent<Rigidbody>();
    slider.minValue = -flySpeed;
    slider.maxValue = flySpeed;
    Cursor.visible = false;
    UpdateParameters();
    Debug.Log(horizontalMToggle);

    mentalLogs =new MentalLogs();
    startTime = Time.time;
}

private void Update()
{
    elapsedTime = Time.time - startTime;
}

void FixedUpdate()
{

    float xRealInput = ToggleValues(movementInput.x, xInput,
horizontalMToggle);
    float zRealInput = ToggleValues(movementInput.y, zInput,
transversalMToggle);
    float yRealInput = ToggleValues(elevateInput, yInput, verticalMToggle);

    float xRRealInput = ToggleValues(rotateInput.y, xRInput,
horizontalRToggle);
```

```
float zRRealInput = ToggleValues(0, zRInput, 0); //Kada se bude radio VR
dodacemo i ovo
float yRRealInput = ToggleValues(rotateInput.x, yRInput,
verticalRToggle);

float disRealInput = ToggleValues(0, disInput, dissapearToggle);

//Debug.Log(zRealInput);

float xVelocity = xRealInput * flySpeed;
float yVelocity = yRealInput * flySpeed;
float zVelocity = zRealInput * flySpeed;

float xRVelocity = -xRRealInput * rotateSpeed;
float yRVelocity = yRRealInput * rotateSpeed;
float zRVelocity = zRRealInput * rotateSpeed;

float stimer = 15;
rigidBody.velocity = transform.forward * zVelocity * stimer +
transform.up * yVelocity * stimer + transform.right * xVelocity * stimer;

rigidBody.transform.Rotate(new Vector3(xRVelocity, yRVelocity,
zRVelocity));

transform.localRotation =
Quaternion.Euler(transform.rotation.eulerAngles.x,
transform.rotation.eulerAngles.y, 0); // Da ispravimo zakrivljenje

float angle = Mathf.Atan2(xVelocity, zVelocity);
angle = Mathf.Rad2Deg * angle;

slider.value = yVelocity;

animator.SetFloat("XSpeed", xVelocity);
animator.SetFloat("ZSpeed", zVelocity);
animator.SetFloat("YSpeed", yVelocity);

GameObject UFO = gameObject.transform.GetChild(0).gameObject;
```

```
        Color bodyColor = UFO.GetComponent<Renderer>().material.color;
        Color baseColor =
UFO.transform.GetChild(0).gameObject.GetComponent<Renderer>().material.color;
        Color gabaritColor =
UFO.transform.GetChild(1).gameObject.GetComponent<Renderer>().material.color;
        Color lobeColor =
UFO.transform.GetChild(2).gameObject.GetComponent<Renderer>().material.color;

        bodyColor.a = 1f - (float)disRealInput;
        gabaritColor.a = 1f - (float)disRealInput;
        baseColor.a = 1f - (float)disRealInput;
        lobeColor.a = 1f - (float)disRealInput;

        UFO.GetComponent<Renderer>().material.color = bodyColor;
        UFO.transform.GetChild(0).gameObject.GetComponent<Renderer>().material.co
lor = gabaritColor;
        UFO.transform.GetChild(1).gameObject.GetComponent<Renderer>().material.co
lor = baseColor;
        UFO.transform.GetChild(2).gameObject.GetComponent<Renderer>().material.co
lor = lobeColor;

    }

    public void UpdateParameters()
    {
        string server = "127.0.0.1";
        string database = "mydb";
        string uid = "root";
        string password = "Koren123!";
        string connectionString;
        connectionString = "SERVER=" + server + ";" + "DATABASE=" +
database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

        using (MySqlConnection connection = new
MySqlConnection(connectionString))
        {
            connection.Open();

            MySqlCommand getExaID = connection.CreateCommand();
```

```
getExaID.CommandText = "SELECT idExaminee FROM Experiments ORDER BY  
idExperiments DESC LIMIT 1;";
```

```
using (MySqlDataReader reader = getExaID.ExecuteReader())  
{  
    if (reader.Read())  
    {  
        exaID = reader.GetInt32(0);  
    }  
    else  
    {  
        exaID = -1;  
    }  
}  
MySqlCommand getOptions = connection.CreateCommand();  
  
getOptions.CommandText = "SELECT Sound, Music, HorizontalMovement,  
VerticalMovement, TransversalMovement, HorizontalRotation, VerticalRotation,  
Dissapear, MovementSpeed, RotationalSpeed FROM SimOptions WHERE  
Examinee_idExaminee = @exaID;";  
getOptions.Parameters.AddWithValue("@exaID", exaID);  
  
using( MySqlDataReader reader = getOptions.ExecuteReader())  
{  
    if (reader.Read())  
    {  
        soundVolume = reader.GetFloat(0);  
        musicVolume = reader.GetFloat(1);  
  
        horizontalMToggle = reader.GetInt32(2);  
        verticalMToggle = reader.GetInt32(3);  
        transversalMToggle = reader.GetInt32(4);  
  
        horizontalRToggle = reader.GetInt32(5);  
        verticalRToggle = reader.GetInt32(6);  
  
        dissapearToggle = reader.GetInt32(7);  
  
        flySpeed = reader.GetFloat(8);  
        rotateSpeed = reader.GetFloat(9);  
    }  
}
```



```
    }  
}  
  
private float ToggleValues(float manualValue, float mentalValue, int  
toggleValue)  
{  
    float result = 0;  
    switch (toggleValue)  
    {  
        case 0:  
            result = 0;  
            break;  
        case 1:  
            result = manualValue;  
            break;  
        case 2:  
            result = mentalValue;  
            break;  
        case 3:  
            result = Mathf.Clamp(manualValue + mentalValue, -1, 1);  
            break;  
    }  
  
    return result;  
}  
  
public void SetXInput(float xInput)  
{  
    this.xInput = xInput;  
}  
public void SetYInput(float yInput)  
{  
    this.yInput = yInput;  
}  
public void SetZInput(float zInput)  
{  
    this.zInput = zInput;  
}  
  
public void SetXRInput(float xRInput)  
{  
    this.xRInput = xRInput;  
}
```

```
}
public void SetYRInput(float yRInput)
{
    this.yRInput = yRInput;
}
public void SetZRInput(float zRInput)
{
    this.zRInput = zRInput;
}

public void SetDisInput(float disInput)
{
    this.disInput = disInput;
}

public int GetExaID()
{
    return exaID;
}

public void MentalCommandLogger(string commandType, float value)
{
    value = (float)Math.Round(value, 2);

    if (commandType == "neutral")
    {
        //mentalLogs.neutralLog.value.Add(value);
        mentalLogs.neutralLog.timeStamp.Add(elapsedTime);
    }

    else if (commandType == "push")
    {
        mentalLogs.pushLog.value.Add(value);
        mentalLogs.pushLog.timeStamp.Add(elapsedTime);
    }

    else if (commandType == "pull")
    {
        mentalLogs.pullLog.value.Add(value);
        mentalLogs.pullLog.timeStamp.Add(elapsedTime);
    }

    else if (commandType == "lift")
    {
        mentalLogs.liftLog.value.Add(value);
        mentalLogs.liftLog.timeStamp.Add(elapsedTime);
    }
}
```

```
}  
else if (commandType == "drop")  
{  
    mentalLogs.dropLog.value.Add(value);  
    mentalLogs.dropLog.timeStamp.Add(elapsedTime);  
}  
  
else if (commandType == "right")  
{  
    mentalLogs.rightLog.value.Add(value);  
    mentalLogs.rightLog.timeStamp.Add(elapsedTime);  
}  
else if (commandType == "left")  
{  
    mentalLogs.leftLog.value.Add(value);  
    mentalLogs.leftLog.timeStamp.Add(elapsedTime);  
}  
  
else if (commandType == "rotateLeft")  
{  
    mentalLogs.rotateLeftLog.value.Add(value);  
    mentalLogs.rotateLeftLog.timeStamp.Add(elapsedTime);  
}  
else if (commandType == "rotateRight")  
{  
    mentalLogs.rotateRightLog.value.Add(value);  
    mentalLogs.rotateRightLog.timeStamp.Add(elapsedTime);  
}  
  
else if (commandType == "rotateClockwise")  
{  
    mentalLogs.rotateClockwiseLog.value.Add(value);  
    mentalLogs.rotateClockwiseLog.timeStamp.Add(elapsedTime);  
}  
else if (commandType == "rotateCounterClockwise")  
{  
    mentalLogs.rotateCounterClockwiseLog.value.Add(value);  
    mentalLogs.rotateCounterClockwiseLog.timeStamp.Add(elapsedTime);  
}  
  
else if (commandType == "rotateForwards")  
{  
    mentalLogs.rotateForwardsLog.value.Add(value);
```

```
        mentalLogs.rotateForwardsLog.timeStamp.Add(elapsedTime);
    }
    else if (commandType == "rotateReverse")
    {
        mentalLogs.rotateReverseLog.value.Add(value);
        mentalLogs.rotateReverseLog.timeStamp.Add(elapsedTime);
    }

    else if (commandType == "disappear")
    {
        mentalLogs.disappearLog.value.Add(value);
        mentalLogs.disappearLog.timeStamp.Add(elapsedTime);
    }

}

public void SaveMentalLog()
{
    mentalLogs.SaveExperiment();
}
}
```

7.2.2 Неурофидбек актуатор

7.2.2.1 Python апликација

```
import websocket
import json
import serial
import time

# Configure the serial port
serial_port = '/dev/ttyS6'
baud_rate = 9600
ser = serial.Serial(serial_port, baud_rate)

# Emotiv API token
client_id = "" # Izostavljeno zbog autorskih prava distribucije
client_secret = "" # Izostavljeno zbog autorskih prava distribucije

# Cortex API WebSocket URL
```

```
cortex_url = "wss://localhost:6868"

# Initialize movement variables
movement = {
    "LX": 0,
    "LY": 0,
    "RX": 0,
    "RY": 0
}

# Function to map power to range -50 to 50
def map_power_to_range(power):
    return int((power * 100) - 50)

# Function to send movement data via serial
def send_movement():
    serial_message =
f"LX_{movement['LX']}_LY_{movement['LY']}_RX_{movement['RX']}_RY_{movement['RY']}"
    print(f"Sending to serial port: {serial_message}")
    ser.write(serial_message.encode())

# Handle WebSocket messages
def on_message(ws, message):
    global movement
    data = json.loads(message)
    if 'com' in data:
        command_data = data['com'][0]
        action = command_data['action']
        power = command_data['power']
        value = map_power_to_range(power)

        if action == "push":
            movement["RY"] = value
        elif action == "pull":
            movement["RY"] = -value
        elif action == "left":
            movement["RX"] = -value
        elif action == "right":
            movement["RX"] = value
        elif action == "lift":
            movement["LY"] = value
        elif action == "drop":
            movement["LY"] = -value
        elif action == "rotate_left":
```

```
        movement["LX"] = -value
    elif action == "rotate_right":
        movement["LX"] = value

    # Send updated movement data
    send_movement()

def on_error(ws, error):
    print(f"Error: {error}")

def on_close(ws, close_status_code, close_msg):
    print("WebSocket connection closed")

def on_open(ws):
    print("WebSocket connection opened")

    # Send authorization request
    auth_request = {
        "jsonrpc": "2.0",
        "method": "authorize",
        "params": {
            "client_id": client_id,
            "client_secret": client_secret
        },
        "id": 1
    }
    ws.send(json.dumps(auth_request))

    # Subscribe to mental commands and cognitive data
    subscribe_request = {
        "jsonrpc": "2.0",
        "method": "subscribe",
        "params": {
            "streams": ["com", "met", "eeg", "dev"]
        },
        "id": 2
    }
    ws.send(json.dumps(subscribe_request))

    # Initialization sequence
    movement["LY"] = -50
    movement["RY"] = -50
    send_movement()
    time.sleep(1)
```

```
movement["LY"] = 50
movement["RY"] = 50
send_movement()
time.sleep(1)

movement["LY"] = 0
movement["RY"] = 0
send_movement()

if __name__ == "__main__":
    websocket.enableTrace(True)
    ws = websocket.WebSocketApp(
        cortex_url,
        on_open=on_open,
        on_message=on_message,
        on_error=on_error,
        on_close=on_close
    )
    ws.run_forever()

    # Close the serial port on exit
    ser.close()
```

7.2.2.2 Arduino програм

```
#include <Servo.h>

Servo leftHorizontal;
Servo leftVertical;
Servo rightHorizontal;
Servo rightVertical;

int pwmLow = 1000;
int pwmHigh = 2000;
int pwmCenter = 1500;

void setup() {
    Serial.begin(9600);

    leftHorizontal.attach(9);
    leftVertical.attach(10);
    rightHorizontal.attach(11);
    rightVertical.attach(12); // Adjust pin number as necessary

    // Initialize PWM signals to center
```



```

    leftHorizontal.writeMicroseconds(pwmCenter);
    leftVertical.writeMicroseconds(pwmCenter);
    rightHorizontal.writeMicroseconds(pwmCenter);
    rightVertical.writeMicroseconds(pwmCenter);
}

void loop() {
    if (Serial.available() > 0) {
        String command = Serial.readStringUntil('\n');
        int lx = extractValue(command, "LX");
        int ly = extractValue(command, "LY");
        int rx = extractValue(command, "RX");
        int ry = extractValue(command, "RY");

        leftHorizontal.writeMicroseconds(map(lx, -50, 50, pwmLow, pwmHigh));
        leftVertical.writeMicroseconds(map(ly, -50, 50, pwmLow, pwmHigh));
        rightHorizontal.writeMicroseconds(map(rx, -50, 50, pwmLow, pwmHigh));
        rightVertical.writeMicroseconds(map(ry, -50, 50, pwmLow, pwmHigh));
    }
}

int extractValue(String data, String key) {
    int startIndex = data.indexOf(key + "_") + key.length() + 1;
    int endIndex = data.indexOf("_", startIndex);
    if (endIndex == -1) {
        endIndex = data.length();
    }
    return data.substring(startIndex, endIndex).toInt();
}

```

7.2.3 Мултиспектрално мапирање мозга

7.2.3.1 *main.py*

```

import customtkinter as ctk # Use customtkinter instead of tkinter
import tkinter as tk
from tkinter import filedialog
from PIL import Image, ImageTk
import os
import re
import csv
import json
import numpy as np
import pandas as pd
import threading
import time
from BrainMapImage import brainMappingImage, brainMapSS, prepare_foof,
plotBrainMap

```

```
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt
from multiprocessing import Pool
from foof.bands import Bands
import pickle
import warnings

warnings.filterwarnings('ignore')

class Experiment:
    def __init__(self, subject_ID, birth_year, experiment_type, markers=None,
interval_markers=None, data=None, metadata=None, expResults=None):
        self.subject_ID = subject_ID
        self.birth_year = int(birth_year)
        self.experiment_type = experiment_type
        self.markers = markers
        self.interval_markers = interval_markers
        self.data = data
        self.metadata = metadata
        self.expResults = expResults if expResults else {}

def create_experiments(data_folder, progress_bar):
    files = [f for f in os.listdir(data_folder) if f.endswith('.xlsx')]
    total_files = len(files)
    experiments = {} # Assuming this is your experiments dictionary

    for idx, filename in enumerate(files):
        progress = ((idx + 1) / total_files)
        progress_bar.set(progress) # Update the progress bar in the UI
        time.sleep(0.01)
        window.update() # Update the progress bar in the UI

        # Updated regex to match the actual format of the filenames
        match = re.match(r'\d+_(S\d{2})\.xlsx', filename)
        if match:
            subject_ID = match.group(1)

            # Generate unique IDs for resting and simulation states
            resting_unique_id = f"{subject_ID}RS" # RS = Resting State
            simulation_unique_id = f"{subject_ID}SP" # SP = Simulation Phase
```

```
file_path = os.path.join(data_folder, filename)

# Reading the Excel file
xls = pd.ExcelFile(file_path)
if len(xls.sheet_names) > 1:
    resting_state_data = pd.read_excel(xls, sheet_name=0) # First
sheet
    simulation_phase_data = pd.read_excel(xls, sheet_name=1) #
Second sheet

# Create or update experiment object for resting state
if resting_unique_id not in experiments:
    experiments[resting_unique_id] = Experiment(subject_ID,
birth_year, 'Resting')
    experiments[resting_unique_id].data = resting_state_data

# Create or update experiment object for simulation phase
if simulation_unique_id not in experiments:
    experiments[simulation_unique_id] = Experiment(subject_ID,
birth_year, 'Simulation')
    experiments[simulation_unique_id].data = simulation_phase_data
else:
    print(f"Warning: Not enough sheets in {filename} for complete
data extraction.")

progress_bar.stop()
# Construct the save path using the data_folder
save_path = os.path.join(data_folder, "data.pkl")

# Save the experiments dictionary to a file using pickle
with open(save_path, 'wb') as save_file:
    pickle.dump(experiments, save_file)

print(f"Experiments saved to {save_path}")
return experiments

def update_dropdown(experiments):
    # This function can be used to update the dropdown options
    # It should be called after loading data to refresh the dropdown menu
    experiment_names = list(experiments.keys())
    dropdown.configure(values=experiment_names)
    # if experiment_names:
    #     dropdown.set(experiment_names[0]) # Set the first item as the default
value
```

```
# Function to handle export operation
def export_brain_map():
    # Code to export the brain mapping data/video
    pass

# Ensure initial state of autoplay_running is set
global autoplay_running
autoplay_running = False

def toggle_autoplay():
    global autoplay_running
    print(slider_value.get())
    print(len(clusters))
    if slider_value.get() >= len(clusters):
        slider_value.set(0)
        scrollbar.configure(variable=slider_value)
        plotSingleCluster(0)
    if not autoplay_running and slider_value.get() < len(clusters):
        autoplay_button.configure(text="Pause")
        autoplay_running = True
        autoplay_next_step()
    else:
        autoplay_button.configure(text="Autoplay")
        autoplay_running = False

window.update()

def autoplay_next_step():
    global autoplay_running, max_slider_value, is_programmatic_update

    if autoplay_running:
        current_value = slider_value.get() + 1
        if current_value <= len(clusters):
            slider_value.set(current_value)
            plotSingleCluster(current_value)
            window.update()

            # Schedule the next step
            window.after(500, autoplay_next_step)
    else:
        # Reached the end or autoplay stopped
```

```
        autoplay_running = False
        autoplay_button.configure(text="Autoplay")
window.update()

def toggle_window_length_entry():
    if checkbox_whole_signal_var.get():
        entry_window_length.configure(state='disabled')
        entry_window_length.delete(0, last_index=300)
    else:
        entry_window_length.configure(state='normal')

def update_experiment_display(experiments):
    experiment_display.delete('1.0', ctk.END) # Clear existing text
    for experiment in experiments.values():
        display_text = f"Subject ID: {experiment.subject_ID}\t\t\t\t\t Birth
Year: {experiment.birth_year}\t\t\t\t\t Experiment Type:
{experiment.experiment_type}\n"
        experiment_display.insert(ctk.END, display_text)

def load_data():
    folder_selected = filedialog.askdirectory()
    experiment_display.delete('1.0', ctk.END) # Clear existing text
    if folder_selected:
        progress_bar.pack(side=ctk.BOTTOM, anchor=ctk.W, pady=10, padx=10)
        # Start the loading in a separate thread to avoid UI freezing
        experiment_display.insert(ctk.END, "Experiment loading started")
        time.sleep(0.01)
        experiment_list = create_experiments(folder_selected, progress_bar)
        update_dropdown(experiment_list)
        display_text = f"Successfully loaded {len(experiment_list)} experiments."
        progress_bar.pack_forget()
        experiment_display.delete('1.0', ctk.END) # Clear existing text
        experiment_display.insert(ctk.END, display_text)
    else:
        experiment_display.insert(ctk.END, "Invalid folder selected")

def load_experiments_from_pkl( ):
    global experiments
    file_path = filedialog.askopenfilename(filetypes=[("Pickle files", "*.pkl")])
    experiment_display.delete('1.0', ctk.END) # Clear existing text
    if file_path:
        progress_bar.pack(side=ctk.BOTTOM, anchor=ctk.W, pady=10, padx=10)
```

```
experiment_display.insert(ctk.END, "Loading experiments from file...")

with open(file_path, 'rb') as file:
    experiments = pickle.load(file)

update_dropdown(experiments)
display_text = f"Successfully loaded {len(experiments)} experiments."
progress_bar.pack_forget() # Hide the progress bar after loading
experiment_display.delete('1.0', ctk.END) # Clear existing loading text
experiment_display.insert(ctk.END, display_text)
else:
    experiment_display.insert(ctk.END, "No file selected")

global clusters, step_size, ch_names, selected_spectra, fg, info

def start_brain_mapping():
    global clusters, step_size, ch_names, selected_spectra, fg, info

    selected_experiment = dropdown_var.get()
    use_whole_signal = checkbox_whole_signal_var.get()
    discrete_window_length = entry_window_length_var.get()
    spectra_selection = {spectrum: var.get() for spectrum, var in
checkbox_spectra_vars.items()}
    selected_spectra = [spectrum for spectrum, selected in
spectra_selection.items() if selected]

    experiment_display.delete('1.0', ctk.END) # Clear existing text
    if selected_experiment != "Select an experiment":

        for widget in plot_frame.winfo_children():
            widget.destroy()
        exp = experiments[selected_experiment]

        eeg_data = exp.data[['RAW_TP9', 'RAW_AF7', 'RAW_AF8', 'RAW_TP10']]
        sampling_frequency = 256
        # sampling_frequency =
int(experiments[selected_experiment].metadata['sampling rate'][0][0])
        ch_names_raw = list(eeg_data.columns.values)
        ch_names = [name.split('_')[1] if '_' in name else name for name in
ch_names_raw]

        if any(spectra_selection.values()):
            subject_id_label.configure(text=f"Subject's ID: {exp.subject_ID}") #
Update subject ID
```

```

        subject_yob_label.configure(text=f"Year of Birth:
{exp.birth_year}") # Update year of birth
        tt = "Resting State" if exp.experiment_type == "Resting" else
"Simulation Phase"
        test_type_label.configure(text=f"Test Type: {tt}") # Update test
type
        if use_whole_signal:
            plot_options_frame.grid_forget()
            display_text = f"Selected experiment: {selected_experiment}" +
"\n"
            display_text = display_text + f"Use whole signal:
{use_whole_signal}" + "\n"
            display_text = display_text + f"Spectra selection: {'',
'.join(selected_spectra)}" + "\n"
            display_text = display_text + f"Frequency: {sampling_frequency}"
+ "\n"
            display_text = display_text + f"Channel names: {'',
'.join(ch_names)}" + "\n"
            display_text = display_text + f"Starting the brain mapping" +
"\n"
            experiment_display.insert(ctk.END, display_text)

        fg, info = prepare_foof(eeg_data.transpose().to_numpy(),
ch_names, sampling_frequency, [1, 45])
        for i, spectrum in enumerate(selected_spectra):
            band_def = bands[spectrum]
            fig = plotBrainMap(fg, info, band_def, spectrum)
            # fig = brainMapSS(eeg_data.transpose().to_numpy(), ch_names,
sampling_frequency, spectrum)

            canvas = FigureCanvasTkAgg(fig, master=plot_frame) # A new
canvas for each figure
            canvas_widget = canvas.get_tk_widget()
            # Position each canvas in the grid
            canvas_widget.grid(row=i // 3, column=i % 3)
            canvas.draw()

        elif discrete_window_length != "":
            plot_options_frame.grid(row=2, column=0, sticky="ew")
            display_text = f"Selected experiment: {selected_experiment}" +
"\n"
            display_text = display_text + f"Discrete window length:
{discrete_window_length}" + "\n"
            display_text = display_text + f"Spectra selection: {'',
'.join(selected_spectra)}" + "\n"

```



```
    window_length = int(discrete_window_length) * sampling_frequency
    overlap_percentage = 50 # or get this value from a user input
    overlap_decimal = overlap_percentage / 100

    step_size = int(window_length * (1 - overlap_decimal))
    num_samples = eeg_data.shape[0]

    # Create clusters (windows) of data
    clusters = [eeg_data.iloc[i:i + window_length] for i in
                range(0, num_samples - window_length + 1, step_size)]
    display_text += f"Number of clusters: {len(clusters)}" + "\n"
    experiment_display.insert(ctk.END, display_text)

    max_slider_value = len(clusters)-1
    number_of_steps = len(clusters)
    scrollbar.configure(from_=0, to=len(clusters) - 1,
number_of_steps=len(clusters), variable=tk.IntVar(value=0))

    plotSingleCluster(0)

    else:
        experiment_display.insert(ctk.END, "Enter the valid window length
or use the whole signal")
    else:
        experiment_display.insert(ctk.END, "Select at least one EEG signal
spectrum")

    else:
        experiment_display.insert(ctk.END, "No experiment selected")
        window.update()

global is_programmatic_update
is_programmatic_update = False

def plotSingleCluster(value):
    global clusters, step_size, ch_names, selected_spectra, fg, info,
is_programmatic_update

    scrollbar.configure(state='disabled')
    # Ignore if the update is programmatic
    if is_programmatic_update:
        return
```

```
if value is None:
    cluster_index = 0
else:
    # Convert the slider value to the nearest valid cluster index
    cluster_index = round(value)
    value = cluster_index

# Programmatic update to the slider value
is_programmatic_update = True
slider_value.set(cluster_index)
scrollbar.configure(variable=slider_value)
is_programmatic_update = False

sampling_frequency = 256

current_time = cluster_index * (step_size/1) / sampling_frequency
if cluster_index < len(clusters):
    end_time = current_time + len(clusters[cluster_index])/sampling_frequency
else:
    last_cluster_length = len(clusters[cluster_index-1]) # Number of samples
in the last cluster
    end_time = current_time + (last_cluster_length / sampling_frequency)

time_label.configure(text=f"Time: {current_time}s - {end_time}s") # Update
test type

cluster = clusters[cluster_index-1]
data = cluster.transpose().to_numpy()
fg, info = prepare_foof(data, ch_names, sampling_frequency, [1, 45])

for i, spectrum in enumerate(selected_spectra):
    band_def = bands[spectrum]
    fig = plotBrainMap(fg, info, band_def, spectrum)

    canvas = FigureCanvasTkAgg(fig, master=plot_frame)
    canvas_widget = canvas.get_tk_widget()
    canvas_widget.grid(row=i // 3, column=i % 3)
    canvas.draw()
window.update()
scrollbar.configure(state='normal')

def plotExpResults(experiment):
```

```
    result_frame.update_idletasks() # Update the frame to make sure we have the
latest size info
    frame_width = result_frame.winfo_width()
    frame_height = result_frame.winfo_height() // len(experiment.expResults) #
Divide by the number of images

    for i, image_key in enumerate(experiment.expResults):
        pil_image = experiment.expResults[image_key]

        # Convert PIL image to a NumPy array
        image_array = np.array(pil_image)

        tk_image = ImageTk.PhotoImage(pil_image)
        # Convert NumPy array to a matplotlib Figure
        fig = plt.figure() # You can adjust the size to fit your need
        ax = fig.add_subplot(111)
        ax.imshow(image_array)
        ax.axis('off') # This will hide the axis

        # Convert the matplotlib Figure to a Tkinter canvas object
        canvas = FigureCanvasTkAgg(fig, master=result_frame)
        # canvas = ctk.CTkCanvas(master=result_frame)
        canvas_widget = canvas.get_tk_widget()
        canvas_widget.grid(row=i, column=0, sticky="nsew")
        canvas.draw()

def display_performance():
    selected_experiment = dropdown_var.get()

# Set up the main application window
window = ctk.CTk()
window.title("Brain Mapper")
window.state("zoomed")
window.update()
# window.attributes('-fullscreen', True)

tabView = ctk.CTkTabview(window)

# tabView.pack(padx=20, pady=20)
tabView.grid(row=0, column=0, padx=20, pady=20)
tabView.pack(expand=True, fill="both")

tabView.add("Settings")
```

```
tabView.add("Brain Mapping")
tabView.add("Results")

# Modify your widgets to use customtkinter
label_file_explorer = ctk.CTkLabel(tabView.tab("Settings"), text="Control Panel",
width=100, height=4)
label_file_explorer.pack(pady=10, padx=10)

# Create a Text widget with a Scrollbar
experiment_display = ctk.CTkTextbox(tabView.tab("Settings"), height=15, width=50)
scroll = ctk.CTkScrollbar(tabView.tab("Settings"),
command=experiment_display.yview)
experiment_display.configure(yscrollcommand=scroll.set)

# Place the Text widget and the Scrollbar in the window
experiment_display.pack(side=ctk.LEFT, fill="both", expand=True)
scroll.pack(side=ctk.RIGHT, fill=ctk.Y)

# BUTTONS
button_explore = ctk.CTkButton(tabView.tab("Settings"), text="Load Data Folder",
command=load_data)
button_explore.pack(pady=10, padx=10)

button_explore_file = ctk.CTkButton(tabView.tab("Settings"), text="Load Data
File", command=load_experiments_from_pk1)
button_explore_file.pack(pady=10, padx=10)

button_exit = ctk.CTkButton(tabView.tab("Settings"), text="Exit", command=exit)
button_exit.pack(side=ctk.BOTTOM, pady=10, padx=10)

dropdown_var = ctk.StringVar(value="Select an experiment")
dropdown = ctk.CTkOptionMenu(tabView.tab("Settings"), variable=dropdown_var,
values=[])
dropdown.pack(pady=10, padx=10)

# Checkbox for selecting whole signal or discrete windows
checkboxbox_whole_signal_var = ctk.BooleanVar()
checkboxbox_whole_signal = ctk.CTkCheckBox(tabView.tab("Settings"), text="Use whole
signal", variable=checkboxbox_whole_signal_var, command=toggle_window_length_entry)
checkboxbox_whole_signal.pack(pady=10, padx=10)

# Text input for discrete window length
sig_len_label = ctk.CTkLabel(tabView.tab("Settings"), text="Signal window length:
")
sig_len_label.pack(pady=1, padx=10)
```

```
entry_window_length_var = ctk.StringVar()
entry_window_length = ctk.CTkEntry(tabView.tab("Settings"),
textvariable=entry_window_length_var)
entry_window_length.pack(pady=1, padx=10)

# EEG spectra frame and checkboxes
spectra_frame = ctk.CTkFrame(tabView.tab("Settings"), fg_color="#242424")
spectra_frame.pack(pady=10, padx=10)

label_spectra = ctk.CTkLabel(spectra_frame, text="EEG Spectra")
label_spectra.pack(pady=10, padx=10)

checkbox_spectra_vars = {spectrum: ctk.BooleanVar() for spectrum in ["Delta",
"Theta", "Alpha", "Beta", "Gamma", "Mu"]}
checkbox_spectra = {spectrum: ctk.CTkCheckBox(spectra_frame, text=spectrum,
variable=var)
                 for spectrum, var in checkbox_spectra_vars.items()}

for checkbox in checkbox_spectra.values():
    checkbox.pack(pady=10, padx=10)

# Button to start brain mapping
button_start_mapping = ctk.CTkButton(tabView.tab("Settings"), text="Start Brain
Mapping", command=start_brain_mapping)
button_start_mapping.pack(pady=10, padx=10)

# Add a progress bar to the settings tab
progress_bar = ctk.CTkProgressBar(tabView.tab("Settings"))
progress_bar.pack(side=ctk.BOTTOM, anchor=ctk.W, pady=10, padx=10)
progress_bar.pack_forget() # Initially hide the progress bar

##### Brain
Mapping

# Configure grid rows and columns
tabView.tab("Brain Mapping").grid_rowconfigure(0, weight=0) # subject_info_frame
row
tabView.tab("Brain Mapping").grid_rowconfigure(1, weight=1) # plot_frame row
tabView.tab("Brain Mapping").grid_columnconfigure(0, weight=1)

# Frame for the plot
```

```
plot_frame = ctk.CTkFrame(tabView.tab("Brain Mapping"))
plot_frame.grid(row=1, column=0, sticky="nsew")
plot_frame.grid_rowconfigure([0, 1], weight=1)
plot_frame.grid_columnconfigure([0, 1, 2], weight=1)

##### Frame for subject information
subject_info_frame = ctk.CTkFrame(tabView.tab("Brain Mapping"))
subject_info_frame.grid(row=0, column=0, sticky="ew")
# Assuming you have a total of 4 elements to display
NUM_ELEMENTS = 4

# Configure the grid for subject_info_frame to have equal weight for each column
for i in range(NUM_ELEMENTS):
    subject_info_frame.grid_columnconfigure(i, weight=1)

# Place each label in its respective column, centered and filling the space
subject_id_label = ctk.CTkLabel(subject_info_frame, text="Subject's ID: ")
subject_id_label.grid(row=0, column=0, sticky="nsew")

subject_yob_label = ctk.CTkLabel(subject_info_frame, text="Year of Birth: ")
subject_yob_label.grid(row=0, column=1, sticky="nsew")

test_type_label = ctk.CTkLabel(subject_info_frame, text="Data Type: ")
test_type_label.grid(row=0, column=2, sticky="nsew")

subject_score_label = ctk.CTkLabel(subject_info_frame, text="Subject Score: ")
subject_score_label.grid(row=0, column=3, sticky="nsew")

#####
### Plot options
plot_options_frame = ctk.CTkFrame(tabView.tab("Brain Mapping"))
plot_options_frame.grid(row=2, column=0, sticky="ew")
plot_options_frame.grid_rowconfigure(0, weight=1)

plot_options_frame.grid_columnconfigure(0, weight=1) # For Autoplay button
plot_options_frame.grid_columnconfigure(1, weight=1) # For Time label
plot_options_frame.grid_columnconfigure(2, weight=13) # For scrollbar
plot_options_frame.grid_columnconfigure(3, weight=2) # For Export button

# Autoplay/Pause Button
autoplay_var = ctk.BooleanVar(value=False) # Variable to toggle between autoplay
and pause
autoplay_button = ctk.CTkButton(plot_options_frame, text="Autoplay",
command=toggle_autoplay)
```

```
autoplay_button.grid(row=0, column=0, padx=10, pady=10, sticky="nsew")

# Time label
time_label = ctk.CTkLabel(plot_options_frame, text="Time: ")
time_label.grid(row=0, column=1, padx=10, pady=10, sticky="nsew")

# Horizontal Scrollbar

slider_value = tk.IntVar(value=0)
scrollbar = ctk.CTkSlider(plot_options_frame, orientation="horizontal",
variable=slider_value, command=plotSingleCluster)
# scrollbar = ctk.CTkSlider(plot_options_frame, orientation="horizontal",
variable=slider_value)
scrollbar.grid(row=0, column=2, padx=10, pady=10, sticky="nsew")

# Export Button
export_button = ctk.CTkButton(plot_options_frame, text="Export",
command=export_brain_map)
export_button.grid(row=0, column=3, padx=10, pady=10, sticky="nsew")

plot_options_frame.grid_forget()
window.update()
#####
# Result plot
# Configure grid rows and columns
tabView.tab("Results").grid_rowconfigure(0, weight=1) # First plot
tabView.tab("Results").grid_rowconfigure(1, weight=1) # Second plot
tabView.tab("Results").grid_columnconfigure(0, weight=1)

# Frame for the plot
result_frame = ctk.CTkFrame(tabView.tab("Results"))
result_frame.grid(row=0, column=0, sticky="nsew")
result_frame.grid_rowconfigure(0, weight=1)
result_frame.grid_rowconfigure(1, weight=1)
result_frame.grid_columnconfigure(0, weight=1)

#####
# Variables
experiments = {}
bands = Bands({'Delta': [1, 7], 'Theta': [3, 7], 'Alpha': [7, 14], 'Beta': [14,
30], 'Gamma': [30, 45], 'Mu': [8, 13]})
fg = None
info = None
clusters = []
```



```
selected_spectra = []

max_slider_value = 0
number_of_steps = 0
autoplay_running = False
```

```
# Run the application
window.mainloop()
```

7.2.3.2 *BrainMapImage.py*

```
import cProfile
import pstats
from mpl_toolkits.axes_grid1 import make_axes_locatable

# Run your script with cProfile
profiler = cProfile.Profile()
profiler.enable()
#####
#####

import sklearn.metrics
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
import pandas as pd
import numpy as np
import pickle
import os.path
# import mkl
from matplotlib import pyplot as plt
import pathlib
import numpy
from pyedflib import highlevel
import numpy as np
from matplotlib import pyplot as plt
from scipy.fft import fft, fftfreq
from os import listdir
from os.path import isfile, join
import pandas as pd
import csv
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
from matplotlib import cm, colors, colorbar
```

```
# Import MNE, as well as the MNE sample dataset
import mne
from mne import io
from mne.datasets import sample
from mne.viz import plot_topomap
# from mne.time_frequency import psd_welch

# FOOOF imports
from fooof import FOOOFGroup
from fooof.bands import Bands
from fooof.analysis import get_band_peak_fg
from fooof.plts.spectra import plot_spectrum

from pyedflib import highlevel
from mne.datasets import sample
from mne import read_evokeds

# np.warnings.filterwarnings('ignore', category=np.VisibleDeprecationWarning)

def check_nans(data, nan_policy='zero'):
    """Check an array for nan values, and replace, based on policy."""

    # Find where there are nan values in the data
    nan_inds = np.where(np.isnan(data))

    # Apply desired nan policy to data
    if nan_policy == 'zero':
        data[nan_inds] = 0
    elif nan_policy == 'mean':
        data[nan_inds] = np.nanmean(data)
    else:
        raise ValueError('Nan policy not understood.')

    return data

def brainMappingImage(sigs, ch_names, sampling_freq):

    n_channels = len(ch_names)

    ch_types = ['eeg']*n_channels

    info = mne.create_info(ch_names, ch_types=ch_types, sfreq=sampling_freq)
    info.set_montage('standard_1020')
```

```
raw = mne.io.RawArray(sigs,info)

# spectra, freqs = psd_welch(raw, fmin=0, fmax=45, tmin=0,
#                             n_overlap=250, n_fft=1000)
spectrum = raw.compute_psd(method="welch", fmin=0, fmax=45, tmin=0,
n_overlap=250, n_fft=1000)
spectra, freqs = spectrum.get_data(return_freqs=True)
fg = F000FGroup(peak_width_limits=[2, 30], min_peak_height=0.15,
                peak_threshold=0.1, max_n_peaks=10, verbose=False)

# Define the frequency range to fit
freq_range = [1, 45]

fg.fit(freqs, spectra, freq_range)

bands = Bands({'delta': [1, 7], # This should be made better
              'theta': [3, 7],
              'alpha': [7, 14],
              'beta': [14, 30],
              'gamma': [30, 45],
              'mu': [8, 13]})

fig, axes = plt.subplots(2, 3, figsize=(30, 10)) #facecolor='#2b2b2b'

vecPic = np.array([[0,0],[0,1],[0,2],[1,0],[1,1],[1,2]])
# fig.tight_layout(pad=0)

for ind, (label, band_def) in enumerate(bands):
    # Get the power values across channels for the current band
    band_power = check_nans(get_band_peak_fg(fg, band_def)[: , 1], 'mean')

    # Create a topomap for the current oscillation band
    im = mne.viz.plot_topomap(band_power, raw.info, cmap='Spectral_r',
contours=2,
                                axes=axes[vecPic[ind][0],vecPic[ind][1]],
show=False, ch_type='eeg', extrapolate='head')

    # Set the plot title
    axes[vecPic[ind][0],vecPic[ind][1]].set_title(label , {'fontsize': 20})
    norm = mpl.colors.Normalize(min(band_power), max(band_power))
```

```
        cbaxes =
fig.add_axes([axes[vecPic[ind]][0],vecPic[ind][1]].get_position().x1+0.01,
axes[vecPic[ind]][0],vecPic[ind][1]].get_position().y0, 0.008, 0.3])
        clb = fig.colorbar(mpl.cm.ScalarMappable(norm=norm, cmap='Spectral_r'),
                           cax=cbaxes, orientation='vertical')
        clb.ax.set_title('dB')

    return fig

def brainMapSS(sigs, ch_names, sampling_freq, s): # brain map single spectrum

    n_channels = len(ch_names)

    ch_types = ['eeg']*n_channels

    info = mne.create_info(ch_names, ch_types=ch_types, sfreq=sampling_freq)
    info.set_montage('standard_1020')

    raw = mne.io.RawArray(sigs,info)

    # spectra, freqs = psd_welch(raw, fmin=0, fmax=45, tmin=0,
    #                             n_overlap=250, n_fft=1000)
    spectrum = raw.compute_psd(method="welch", fmin=0, fmax=45, tmin=0,
n_overlap=250, n_fft=1000)
    spectra, freqs = spectrum.get_data(return_freqs=True)
    fg = FOOFGGroup(peak_width_limits=[2, 30], min_peak_height=0.15,
                    peak_threshold=0.1, max_n_peaks=10, verbose=False)

    # Define the frequency range to fit
    freq_range = [1, 45]

    fg.fit(freqs, spectra, freq_range)

    bands = Bands({'Delta': [1, 7], # This should be made better
                  'Theta': [3, 7],
                  'Alpha': [7, 14],
                  'Beta': [14, 30],
                  'Gamma': [30, 45],
                  'Mu': [8, 13]})

    fig, ax = plt.subplots(figsize=(30, 10)) #facecolor='#2b2b2b'
```

```
vecPic = np.array([[0,0],[0,1],[0,2],[1,0],[1,1],[1,2]])
# fig.tight_layout(pad=0)

# for ind, (label, band_def) in enumerate(bands):
    # Get the power values across channels for the current band

band_def = bands[s]

band_power = check_nans(get_band_peak_fg(fg, band_def)[: , 1], 'mean')

# Create a topomap for the current oscillation band
im = mne.viz.plot_topomap(band_power, raw.info, cmap='Spectral_r',
contours=2,
                        show=False, ch_type='eeg', extrapolate='head', axes=ax)

# Set the plot title
ax.set_title(s , {'fontsize': 20})
norm = mpl.colors.Normalize(min(band_power), max(band_power))

colorbar_axes = fig.add_axes([0.88, ax.get_position().y0, 0.03,
ax.get_position().height])
clb = fig.colorbar(mpl.cm.ScalarMappable(norm=norm, cmap='Spectral_r'),
ax=ax, cax=colorbar_axes, orientation='vertical')
clb.ax.set_title('dB')

return fig

def prepare_foof(sigs, ch_names, sampling_freq, freq_range):
    signal_length = sigs.shape[1]
    n_channels = len(ch_names)
    ch_types = ['eeg'] * n_channels
    info = mne.create_info(ch_names, ch_types=ch_types, sfreq=sampling_freq)
    info.set_montage('standard_1020')

    n_fft_min = sampling_freq
    n_fft_max = sampling_freq * 2
    n_fft = min(max(int(signal_length / 20), n_fft_min), n_fft_max)
    n_overlap = int(n_fft / 2)
    raw = mne.io.RawArray(sigs, info)
    spectrum = raw.compute_psd(method="welch", fmin=0, fmax=45, tmin=0,
n_overlap=n_overlap, n_fft=n_fft)
    spectra, freqs = spectrum.get_data(return_freqs=True)
    spectra = np.abs(spectra) + 1e-6 # Apply absolute value and ensure no zeros
```

```
# Handling Infs and converting them to NaNs for uniform handling
spectra[np.isinf(spectra)] = np.nan

# Ensure there are positive values greater than zero; if not, use 1e-6
positive_values = spectra[spectra > 0]
if positive_values.size > 0:
    min_positive_value = np.nanmin(positive_values)
else:
    # Fallback to a small positive value if no positive values found
    min_positive_value = 1e-6

# Replace NaNs with the minimum positive value found or the fallback value
spectra = np.where(np.isnan(spectra), min_positive_value, spectra)
spectra_logged = np.log10(spectra)

fg = F000FGroup(peak_width_limits=[2, 30], min_peak_height=0.15,
peak_threshold=0.1, max_n_peaks=10, verbose=False)
# fg.fit(freqs, spectra, freq_range)
fg.fit(freqs, spectra, freq_range)

print(f"Signal length: {signal_length}")
print(f"n_fft: {n_fft}")

return fg, raw.info

def plotBrainMap(fg, info, band_def, s):
    band_power = check_nans(get_band_peak_fg(fg, band_def)[: , 1], 'mean')
    fig, ax = plt.subplots(figsize=(30, 10))

    # Create a topomap for the current oscillation band
    im = mne.viz.plot_topomap(band_power, info, cmap='Spectral_r', contours=2,
show=False, ch_type='eeg',
                                extrapolate='head', axes=ax)

    ax.set_title(s, {'fontsize': 20})
    norm = mpl.colors.Normalize(min(band_power), max(band_power))
    colorbar_axes = fig.add_axes([0.88, ax.get_position().y0, 0.03,
ax.get_position().height])
    clb = fig.colorbar(mpl.cm.ScalarMappable(norm=norm, cmap='Spectral_r'),
ax=ax, cax=colorbar_axes,
                                orientation='vertical')
    clb.ax.set_title('dB')
```

```
return fig
```

7.2.4 Систем за класификацију и регресију

7.2.4.1 Систем за класификацију коњивних активносци сјања

```
import mne
import numpy as np
import os
import pandas as pd
import tqdm
from scipy.signal import spectrogram, stft
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, Flatten, Dense, Dropout, MaxPooling1D
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau,
ModelCheckpoint

import glob
from mne.io import read_raw_edf

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score, f1_score, roc_curve, auc
from tensorflow.keras.callbacks import EarlyStopping
from pandas import ExcelWriter

# Definisanje funkcije za dobijanje maske opsega
def get_band_mask(freqs, band):
    return (freqs >= band[0]) & (freqs <= band[1])

# Definisanje opsega frekvencija
bands = [(0.5, 4), (4, 8), (8, 12), (12, 30), (30, 45)]
band_names = ['delta', 'theta', 'alpha', 'beta', 'gamma']

# Priprema skladišta za sve podatke i labele
all_data = []
all_labels = []

# Dobijanje apsolutne putanje direktorijuma skripte
```



```
script_dir = os.path.dirname(os.path.abspath(__file__))

# Kreiranje direktorijuma 'Results' ako ne postoji
results_dir = os.path.join(script_dir, 'Results')
os.makedirs(results_dir, exist_ok=True)

# Definisanje putanja do EEG fajlova i CSV fajla
eeg_files = glob.glob(os.path.join(script_dir, 'eegMentalArithmetic/*_2.edf'))
subject_info_path = os.path.join(script_dir, 'eegMentalArithmetic/subject-
info.csv')
subject_info = pd.read_csv(subject_info_path)

# Obrada fajlova
for file in tqdm.tqdm(eeg_files, "File processing"):
    # Učitavanje EEG fajla
    eeg = mne.io.read_raw_edf(file, preload=True, verbose=False)

    base_filename = os.path.basename(file).split('.')[0]

    subject_id = base_filename.split('_')[0]
    label = subject_info[subject_info['Subject'] == subject_id]['Count
quality'].values[0]

    # Dobijanje EEG podataka
    data = eeg.get_data()

    freqs, time, Sxx = stft(data[0], fs=500, nperseg=500, noverlap=250,
detrend=False, return_onesided=True)
    num_stft_samples = Sxx.shape[1]
    # Inicijalizacija 3D niza za ovaj fajl
    file_data = np.zeros((len(band_names), data.shape[0], num_stft_samples))

    # Izračunavanje STFT za svaki kanal
    for ch_idx in range(data.shape[0]):
        freqs, time, Sxx = stft(data[ch_idx], fs=500, nperseg=500, noverlap=250,
detrend=False, return_onesided=True)
        Sxx = np.abs(Sxx) ** 2 # Uzimanje apsolutne vrednosti i kvadriranje da
bi se dobila snaga
        for band_idx, (band, band_name) in enumerate(zip(bands, band_names)):
            freq_mask = get_band_mask(freqs, band)
            if freq_mask.any(): # Proverava da li bilo koja frekvencija spada u
opseg
                power = Sxx[freq_mask, :].mean(axis=0) # Uzimanje proseka kroz
frekvencije u opsegu
                file_data[band_idx, ch_idx, :] = power
```

```
all_data.append(file_data)
all_labels += [label] * file_data.shape[2]

# Konvertovanje lista u NumPy nizove za lakšu manipulaciju
all_data = np.concatenate(all_data, axis=2)
all_labels = np.array(all_labels)

print(all_data.shape)
print(all_labels.shape)

# Normalizacija karakteristika
scaler = StandardScaler()
all_data = scaler.fit_transform(all_data.reshape(-1, all_data.shape[-1])).reshape(all_data.shape)

# Transponovanje podataka da kanali budu druga dimenzija, a vreme prva
all_data = all_data.transpose(2, 1, 0)

# Podela podataka na trening i test setove
X_train, X_test, y_train, y_test = train_test_split(all_data, all_labels,
test_size=0.2, random_state=42)

model = Sequential([
    Conv1D(32, kernel_size=3, activation='relu', input_shape=(21, 5)),
    MaxPooling1D(pool_size=2),
    Dropout(0.25),
    Conv1D(64, kernel_size=3, activation='relu'),
    MaxPooling1D(pool_size=2),
    Dropout(0.25),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Kreiranje callback-a za rano zaustavljanje
early_stopping = EarlyStopping(monitor='val_loss', patience=5)

# Treniranje modela i dobijanje istorije treniranja
history = model.fit(X_train, y_train, validation_split=0.2, epochs=200,
batch_size=64, callbacks=[early_stopping])
```

```
# Konvertovanje istorije u pandas DataFrame
history_df = pd.DataFrame(history.history)

# Čuvanje istorije u Excel fajl
history_file = os.path.join(results_dir, 'training_history.xlsx')
with ExcelWriter(history_file) as writer:
    history_df.to_excel(writer)

# Plotovanje tačnosti i gubitka
history_df[['accuracy', 'val_accuracy']].plot()
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.savefig(os.path.join(results_dir, 'Accuracy.png'))

plt.figure()
history_df[['loss', 'val_loss']].plot()
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.savefig(os.path.join(results_dir, 'Loss.png'))

# Matrica konfuzije
y_pred_prob = model.predict(X_test)
y_pred = (y_pred_prob > 0.5).astype("int32")
conf_mat = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(10,7))
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.savefig(os.path.join(results_dir, 'ConfusionMatrix.png'))

# Dodatne metrike
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
```

```
print(f'F1 Score: {f1}')

# ROC kriva i AUC
fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
roc_auc = auc(fpr, tpr)

plt.figure()
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (area = %0.2f)' %
roc_auc)
plt.plot([0, 1], [0, 1], color='grey', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.savefig(os.path.join(results_dir, 'ROC_Curve.png'))

from tensorflow.keras.utils import plot_model

plot_model(model, to_file=os.path.join(results_dir, 'model_plot.png'),
show_shapes=True, show_layer_names=True)

# Sačuvaj model u novom Keras formatu
model.save(os.path.join(results_dir, 'mojaMreza.keras'))

import netron

# Ispravka escape karaktera u putanji
model_path = os.path.join(results_dir, 'mojaMreza.keras')
model_path = model_path.replace('\\', '\\\\')
netron.start(model_path)
```

7.2.4.2 Сисѐм за рејресију коїниїивних сїања

```
import numpy as np
import pandas as pd
import os
import glob
from tqdm import tqdm
from scipy.signal import stft
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential # type: ignore
```

```
from tensorflow.keras.layers import Conv1D, Flatten, Dense, Dropout, MaxPooling1D
# type: ignore
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint # type:
ignore
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from pandas import ExcelWriter
import itertools
import pickle

# Define the bands
bands = [(0.5, 4), (4, 8), (8, 12), (12, 30), (30, 45)]
band_names = ['delta', 'theta', 'alpha', 'beta', 'gamma']

# Function to get band mask
def get_band_mask(freqs, band):
    return (freqs >= band[0]) & (freqs <= band[1])

# Define cognitive parameters and EEG channels
cognitive_params = [
    'PM.Boredom.Scaled', 'PM.Attention.Scaled', 'PM.VisualAttention.Scaled',
    'PM.CognitiveStress.Scaled', 'PM.Engagement.Scaled', 'PM.Excitement.Scaled',
    'PM.Stress.Scaled', 'PM.Relaxation.Scaled', 'PM.Interest.Scaled',
    'PM.Focus.Scaled'
]

eeg_channels = ['EEG.AF3', 'EEG.T7', 'EEG.Pz', 'EEG.T8', 'EEG.AF4']

# Get the absolute path of the script directory and the data directory
script_dir = os.path.dirname(os.path.abspath(__file__))
data_dir = os.path.join(script_dir, 'Data')

# Get all CSV files that do not end with '_intervalMarker.csv'
csv_files = glob.glob(os.path.join(data_dir, '*.csv'))
csv_files = [f for f in csv_files if not f.endswith('_intervalMarker.csv')]

# Function to fill cognitive parameters
def fill_cognitive_parameters(df, cog_columns):
    for col in cog_columns:
        if col in df.columns:
            df[col] = df[col].ffill().bfill()
    return df
```

```
# Check if processed data exists, otherwise process each file and create a list
of dataframes
data_file = os.path.join(data_dir, 'all_data.pkl')

if os.path.exists(data_file):
    with open(data_file, 'rb') as f:
        all_data = pickle.load(f)
else:
    all_data = []
    for file in tqdm(csv_files, desc="Processing CSV files"):
        df = pd.read_csv(file, header=1) # Specify that column names start from
the second row
        df_filled = fill_cognitive_parameters(df, cognitive_params)
        all_data.append(df_filled)
        with open(data_file, 'wb') as f:
            pickle.dump(all_data, f)

# STFT processing and synchronization
all_processed_data = []
all_labels = []

for df in all_data:
    eeg_data = df[eeg_channels].values.T
    cog_data = df[cognitive_params].values

    # Perform STFT to determine the number of STFT samples
    freqs, time, Sxx = stft(eeg_data[0, :], fs=128, nperseg=128, noverlap=64,
detrend=False, return_onesided=True)
    num_stft_samples = Sxx.shape[1] # Number of time segments

    # Initialize the 3D array for this recording
    file_data = np.zeros((len(band_names), eeg_data.shape[0], num_stft_samples))

    for ch_idx in range(eeg_data.shape[0]):
        freqs, time, Sxx = stft(eeg_data[ch_idx, :], fs=128, nperseg=128,
noverlap=64, detrend=False, return_onesided=True)
        Sxx = np.abs(Sxx) ** 2 # Take the absolute value and square it to get
the power
        for band_idx, (band, band_name) in enumerate(zip(bands, band_names)):
            freq_mask = get_band_mask(freqs, band)
            if freq_mask.any(): # Check if any frequencies are within the band
                power = Sxx[freq_mask, :].mean(axis=0) # Take the mean across
frequencies in the band
                file_data[band_idx, ch_idx, :] = power
```

```
# Synchronize with labels
step_size = 64
labels = []

for i in range(num_stft_samples):
    time_idx = i * step_size
    if time_idx < cog_data.shape[0]:
        labels.append(cog_data[time_idx])
    else:
        labels.append(cog_data[-1]) # Use the last available value if out of
range

    all_processed_data.append(file_data)
    all_labels.append(np.array(labels))

# Combine the processed data
all_processed_data = np.concatenate(all_processed_data, axis=2)
all_labels = np.concatenate(all_labels, axis=0)

# Normalize features
scaler = StandardScaler()
all_processed_data = scaler.fit_transform(all_processed_data.reshape(-1,
all_processed_data.shape[-1])).reshape(all_processed_data.shape)

# Transpose data to have samples as the first dimension and time_steps as the
second dimension
all_processed_data = all_processed_data.transpose(2, 1, 0)

# Define function to create and train model
def create_and_train_cnn_model(X_train, y_train, X_test, y_test, output_dir):
    input_shape = (X_train.shape[1], X_train.shape[2]) # Define the input shape

    model = Sequential([
        Conv1D(64, kernel_size=3, activation='relu', padding='same',
input_shape=input_shape),
        MaxPooling1D(pool_size=2, padding='same'),
        Dropout(0.25),
        Conv1D(128, kernel_size=3, activation='relu', padding='same'),
        MaxPooling1D(pool_size=2, padding='same'),
        Dropout(0.25),
        Conv1D(256, kernel_size=3, activation='relu', padding='same'),
        MaxPooling1D(pool_size=2, padding='same'),
        Dropout(0.25),
        Flatten(),
        Dense(128, activation='relu'),
```



```
        Dropout(0.5),
        Dense(y_train.shape[1], activation='linear')
    ])

    model.compile(optimizer='adam', loss='mean_squared_error',
metrics=['mean_squared_error'])

    # Create callbacks
    early_stopping = EarlyStopping(monitor='val_loss', patience=5)
    checkpoint = ModelCheckpoint(os.path.join(output_dir, 'best_model.keras'),
monitor='val_loss', save_best_only=True)

    # Fit the model and get the history
    history = model.fit(X_train, y_train, validation_split=0.2, epochs=100,
batch_size=64, callbacks=[early_stopping, checkpoint])

    # Convert the history to a pandas DataFrame
    history_df = pd.DataFrame(history.history)

    # Save the history to an Excel file
    with ExcelWriter(os.path.join(output_dir, 'training_history.xlsx')) as
writer:
        history_df.to_excel(writer)

    # Plotting RMSE and Loss (MSE)
    history_df['rmse'] = np.sqrt(history_df['mean_squared_error'])
    history_df['val_rmse'] = np.sqrt(history_df['val_mean_squared_error'])

    history_df[['rmse', 'val_rmse']].plot()
    plt.title('Model RMSE')
    plt.ylabel('Root Mean Squared Error')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Val'], loc='upper left')
    plt.savefig(os.path.join(output_dir, 'RMSE.png'))

    plt.figure()
    history_df[['loss', 'val_loss']].plot()
    plt.title('Model loss (MSE)')
    plt.ylabel('Loss (MSE)')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Val'], loc='upper left')
    plt.savefig(os.path.join(output_dir, 'Loss.png'))

    # Predict and evaluate the model
    y_pred = model.predict(X_test)
```

```
# Ensure the shapes match for evaluation
y_test = y_test.reshape(-1, y_pred.shape[1])

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')
print(f'R2 Score: {r2}')

# Plotting R2
plt.figure()
plt.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=4)
plt.xlabel('Measured')
plt.ylabel('Predicted')
plt.title(f'R2 Score: {r2}')
plt.savefig(os.path.join(output_dir, 'R2_Score.png'))

# Save model
model.save(os.path.join(output_dir, 'regression_model.keras'))

return mse, rmse, r2

def create_and_train_feedforward_model(X_train, y_train, X_test, y_test,
output_dir):
    input_shape = (X_train.shape[1] * X_train.shape[2],) # Define the input
shape

    model = Sequential([
        Flatten(input_shape=input_shape),
        Dense(512, activation='relu'),
        Dropout(0.25),
        Dense(256, activation='relu'),
        Dropout(0.25),
        Dense(128, activation='relu'),
        Dropout(0.25),
        Dense(y_train.shape[1], activation='linear')
    ])

    model.compile(optimizer='adam', loss='mean_squared_error',
metrics=['mean_squared_error'])
```

```
# Create callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
checkpoint = ModelCheckpoint(os.path.join(output_dir, 'best_model.keras'),
monitor='val_loss', save_best_only=True)

# Fit the model and get the history
history = model.fit(X_train.reshape(X_train.shape[0], -1), y_train,
validation_split=0.2, epochs=100, batch_size=64, callbacks=[early_stopping,
checkpoint])

# Convert the history to a pandas DataFrame
history_df = pd.DataFrame(history.history)

# Save the history to an Excel file
with ExcelWriter(os.path.join(output_dir, 'training_history.xlsx')) as
writer:
    history_df.to_excel(writer)

# Plotting RMSE and Loss (MSE)
history_df['rmse'] = np.sqrt(history_df['mean_squared_error'])
history_df['val_rmse'] = np.sqrt(history_df['val_mean_squared_error'])

history_df[['rmse', 'val_rmse']].plot()
plt.title('Model RMSE')
plt.ylabel('Root Mean Squared Error')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.savefig(os.path.join(output_dir, 'RMSE.png'))

plt.figure()
history_df[['loss', 'val_loss']].plot()
plt.title('Model loss (MSE)')
plt.ylabel('Loss (MSE)')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.savefig(os.path.join(output_dir, 'Loss.png'))

# Predict and evaluate the model
y_pred = model.predict(X_test.reshape(X_test.shape[0], -1))

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

```
print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')
print(f'R2 Score: {r2}')

# Plotting R2
plt.figure()
plt.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=4)
plt.xlabel('Measured')
plt.ylabel('Predicted')
plt.title(f'R2 Score: {r2}')
plt.savefig(os.path.join(output_dir, 'R2_Score.png'))

# Save model
model.save(os.path.join(output_dir, 'feedforward_model.keras'))

return mse, rmse, r2

def train_random_forest(X_train, y_train, X_test, y_test, output_dir):
    model = RandomForestRegressor(
        n_estimators=200,
        max_features='sqrt',
        max_depth=20,
        min_samples_split=5,
        min_samples_leaf=2,
        bootstrap=True,
        random_state=42
    )

    # Create a progress bar
    with tqdm(total=model.n_estimators, desc="Training RandomForest") as pbar:
        for i in range(model.n_estimators):
            temp_model = RandomForestRegressor(
                n_estimators=i+1,
                max_features='sqrt',
                max_depth=20,
                min_samples_split=5,
                min_samples_leaf=2,
                bootstrap=True,
                random_state=42
            )
            # Fit the temporary model
            temp_model.fit(X_train.reshape(X_train.shape[0], -1),
y_train.ravel())
```

```
        # Increment the progress bar
        pbar.update(1)

    # Fit the final model
    model.fit(X_train.reshape(X_train.shape[0], -1), y_train.ravel())

    y_pred = model.predict(X_test.reshape(X_test.shape[0], -1))

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)

    print(f'Mean Squared Error: {mse}')
    print(f'Root Mean Squared Error: {rmse}')
    print(f'R2 Score: {r2}')

    # Plotting R2
    plt.figure()
    plt.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=4)
    plt.xlabel('Measured')
    plt.ylabel('Predicted')
    plt.title(f'R2 Score: {r2}')
    plt.savefig(os.path.join(output_dir, 'R2_Score.png'))

    # Save model
    with open(os.path.join(output_dir, 'random_forest_model.pkl'), 'wb') as f:
        pickle.dump(model, f)

    return mse, rmse, r2

def train_linear_regression(X_train, y_train, X_test, y_test, output_dir):
    model = LinearRegression()
    model.fit(X_train.reshape(X_train.shape[0], -1), y_train)

    y_pred = model.predict(X_test.reshape(X_test.shape[0], -1))

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)

    print(f'Mean Squared Error: {mse}')
```

```
print(f'Root Mean Squared Error: {rmse}')
print(f'R2 Score: {r2}')

# Plotting R2
plt.figure()
plt.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=4)
plt.xlabel('Measured')
plt.ylabel('Predicted')
plt.title(f'R2 Score: {r2}')
plt.savefig(os.path.join(output_dir, 'R2_Score.png'))

# Save model
with open(os.path.join(output_dir, 'linear_regression_model.pkl'), 'wb') as
f:
    pickle.dump(model, f)

return mse, rmse, r2

def compute_and_plot_cognitive_params(all_processed_data, all_labels,
output_dir):
    # Define the EEG channels and bands
    bands = ['delta', 'theta', 'alpha', 'beta', 'gamma']
    channels = ['AF3', 'T7', 'Pz', 'T8', 'AF4']

    # Compute power for each band
    total_power = np.sum(all_processed_data, axis=0)
    band_powers = {band: all_processed_data[idx, :, :] for idx, band in
enumerate(bands)}

    # Initialize results
    results = {}

    # Iterate through all parameters
    for param in cognitive_params:
        param_values = []

        # Compute values based on the formulas
        if param == 'PM.Boredom.Scaled':
            engagement = band_powers['beta'] + band_powers['gamma']
            attention = band_powers['beta'] / total_power
            boredom = 1 - ((engagement + attention) / 2)
            param_values = boredom
```

```
elif param == 'PM.Attention.Scaled':
    attention = band_powers['beta'] / total_power
    param_values = attention

elif param == 'PM.VisualAttention.Scaled':
    visual_attention = 1 - (band_powers['alpha'] / total_power)
    param_values = visual_attention

elif param == 'PM.CognitiveStress.Scaled':
    cognitive_stress = (band_powers['theta'] + band_powers['beta']) /
total_power
    param_values = cognitive_stress

elif param == 'PM.Engagement.Scaled':
    engagement = (band_powers['beta'] + band_powers['gamma']) /
total_power
    param_values = engagement

elif param == 'PM.Excitement.Scaled':
    excitement = band_powers['gamma'] / total_power
    param_values = excitement

elif param == 'PM.Stress.Scaled':
    stress = band_powers['beta'] / band_powers['alpha']
    param_values = stress

elif param == 'PM.Relaxation.Scaled':
    relaxation = (band_powers['alpha'] + band_powers['theta']) /
total_power
    param_values = relaxation

elif param == 'PM.Interest.Scaled':
    engagement = (band_powers['beta'] + band_powers['gamma']) /
total_power
    attention = band_powers['beta'] / total_power
    interest = (engagement + attention) / 2
    param_values = interest

elif param == 'PM.Focus.Scaled':
    focus = band_powers['beta'] / total_power
    param_values = focus

# Plot and save the results
time_steps = np.arange(len(param_values))
plt.figure()
```

```
plt.plot(time_steps, param_values)
plt.title(f'{param} over Time')
plt.xlabel('Time Steps')
plt.ylabel(param)
plt.savefig(os.path.join(output_dir, f'{param}.png'))
plt.close()

results[param] = param_values

# Save results to CSV
results_df = pd.DataFrame(results)
results_df.to_csv(os.path.join(output_dir, 'cognitive_params.csv'),
index=False)

return results

# Select the desired cognitive parameters combinations and run the model
selected_combinations = cognitive_params
results = {}

for param in selected_combinations:
    y_data = all_labels[:, [cognitive_params.index(param)]]

    X_train, X_test, y_train, y_test = train_test_split(all_processed_data,
y_data, test_size=0.2, random_state=42)

    # CNN Model
    cnn_output_dir = os.path.join(script_dir, 'Results', 'CNN', param)
    os.makedirs(cnn_output_dir, exist_ok=True)
    mse, rmse, r2 = create_and_train_cnn_model(X_train, y_train, X_test, y_test,
cnn_output_dir)
    results[f'CNN_{param}'] = {'MSE': mse, 'RMSE': rmse, 'R2': r2}

    # Linear Regression Model
    lr_output_dir = os.path.join(script_dir, 'Results', 'LinearRegression',
param)
    os.makedirs(lr_output_dir, exist_ok=True)
    mse, rmse, r2 = train_linear_regression(X_train, y_train, X_test, y_test,
lr_output_dir)
    results[f'LinearRegression_{param}'] = {'MSE': mse, 'RMSE': rmse, 'R2': r2}

    # Random Forest Model
    rf_output_dir = os.path.join(script_dir, 'Results', 'RandomForest', param)
    os.makedirs(rf_output_dir, exist_ok=True)
```



```

    mse, rmse, r2 = train_random_forest(X_train, y_train, X_test, y_test,
rf_output_dir)
    results[f'RandomForest_{param}'] = {'MSE': mse, 'RMSE': rmse, 'R2': r2}

    # Feedforward Neural Network Model
    ffn_output_dir = os.path.join(script_dir, 'Results', 'FeedforwardNN', param)
    os.makedirs(ffn_output_dir, exist_ok=True)
    mse, rmse, r2 = create_and_train_feedforward_model(X_train, y_train, X_test,
y_test, ffn_output_dir)
    results[f'FeedforwardNN_{param}'] = {'MSE': mse, 'RMSE': rmse, 'R2': r2}

# Save results to a CSV file
results_df = pd.DataFrame(results).T
results_df.to_csv(os.path.join(script_dir, 'Results', 'summary.csv'))

```

7.2.5 Психолошко тестирање и евалуација

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Data;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.EventSystems;
using System.IO;
using UnityEditor;
using System.Linq;
//using UnityEditorInternal;
using SimpleFileBrowser;
//using UnityEngine.UIElements;
using Newtonsoft.Json;
using UnityEngine.Rendering.VirtualTexturing;
//using UnityEditor.SceneTemplate;
using UnityEngine.UIElements;
//using static
UnityEditor.Experimental.AssetDatabaseExperimental.AssetDatabaseCounters;
using static TestEngine;
using UnityEngine.Localization.Settings;
using UnityEngine.Localization.Tables;

public class TestEngine : MonoBehaviour
{
    Dropdown question;

```

```
bool timerFlag = false;
float startTime;
float elapsedTime;
string[] questionArray;
string[] answerArray;
string csvFilePath;
string resultsFilePath;

// Kreiraj listu za cuvanje podataka
List<string> questions = new List<string>();
List<string> answers = new List<string>();
List<List<string>> splitAnswers = new List<List<string>>();
List<string> domains = new List<string>();
List<string> sectors = new List<string>();
public RectTransform containerPanel;
public Text infoText;
public UnityEngine.UI.Button startTestButton;

public float questionSpacingX = 10f; // Razmak izmedju pitanja po x osi
public float questionSpacingY = 10f; // Razmak izmedju pitanja po y osi
private int questionIndex = 0;

int questionsPerRow = 2;
int questionsPerColumn = 3;

public GameObject testPanel;
public GameObject finishPanel;

public InputField inputTitle;
public InputField inputTestID;
public InputField inputSubjectID;

// Klasa za skladištenje podataka o pitanju
[System.Serializable]
public class QuestionData
{
    public string QuestionText { get; set; }
    public List<string> Answers { get; set; }
    public string Domain { get; set; }
    public string Sector { get; set; }
}
```

```
// Klasa za skladistenje podataka o odgovoru korisnika
[System.Serializable]
public class UserResponse
{
    public string QuestionText { get; set; }
    public string SelectedAnswer { get; set; }
    public string Domain { get; set; }
    public string Sector { get; set; }
}

[System.Serializable]
public class Domain
{
    public string Name { get; set; }
    public float Score { get; set; }

    public Domain(string name, float score)
    {
        Name = name;
        Score = score;
    }
}

[System.Serializable]
public class Sector
{
    public string Name { get; set; }
    public float Score { get; set; }
    public Dictionary<string, Domain> Domains { get; set; }

    public Sector(string name, float score)
    {
        Name = name;
        Score = score;
        Domains = new Dictionary<string, Domain>();
    }
}

[System.Serializable]
public class Test
{
    public string Title { get; set; }
    public string TestID { get; set; }
}
```

```
public string SubjectID { get; set; }
public float Score { get; set; }
public float ElapsedTime { get; set; }
public DateTime Date { get; set; }
public Dictionary<int, UserResponse> UserResponses { get; set; }
public Dictionary<int, QuestionData> QuestionDictionary { get; set; }

public Dictionary<string, Sector> Sectors { get; set; }

}
Test testResults = new Test();
Test loadedResults;

public void TestTimer()
{
    if (!timerFlag)
    {
        startTime = Time.time;
        timerFlag = true;
    }
    else
    {
        elapsedTime = Time.time - startTime;
        timerFlag = false;
    }
}

public List<InputField> inputFields;
int inputSelected;
private void Start()
{
    inputSelected = 0;
    SelectInputField();
    ToggleFullScreen();
    //TextNode rootNode = TraverseAndCollectTextElements(gameObject);

    //string jsonData = JsonConvert.SerializeObject(rootNode,
Formatting.Indented);

    //File.WriteAllText(Application.dataPath + "/Language.json", jsonData);
```

```
}
public void Update()
{
    if (EventSystem.current.currentSelectedGameObject != null)
    {
        if (EventSystem.current.currentSelectedGameObject.transform.name !=
inputFields[inputSelected].name && EventSystem.current.currentSelectedGameObject
!= null)
        {
            if (inputFields.FindIndex(inputField => inputField.name ==
EventSystem.current.currentSelectedGameObject.transform.name) != -1)
            {
                inputSelected = inputFields.FindIndex(inputField =>
inputField.name == EventSystem.current.currentSelectedGameObject.transform.name);
            }
        }
        int previousInputSelected = inputSelected;

        if (Input.GetKeyDown(KeyCode.Tab) && Input.GetKey(KeyCode.LeftShift))
        {
            inputSelected--;
            if (inputSelected < 0) inputSelected = inputFields.Count - 1;
        }
        else if (Input.GetKeyDown(KeyCode.Tab))
        {
            inputSelected++;
            if (inputSelected >= inputFields.Count) inputSelected = 0;
        }

        // Ako se trenutno odabrano polje promenilo, selektuj novo polje.
        if (inputSelected != previousInputSelected)
        {
            SelectInputField();
        }
    }

    if (Input.GetKeyDown(KeyCode.F3))
    {
        ToggleFullScreen();
    }
}

void SelectInputField()
{
    // Obezbedi da brojac ostane u granicama indeksa inputFields liste.
```

```
        inputSelected = Mathf.Clamp(inputSelected, 0, inputFields.Count - 1);

        // Postavi fokus na trenutno odabrano polje.
        inputFields[inputSelected].Select();
    }

    public void StartTest()
    {
        startTestButton.gameObject.SetActive(false);
        infoText.text = "";
        //Debug.Log(csvFilePath);
        if (File.Exists(csvFilePath))
        {

            // Ucitaj linije iz .csv fajla
            string[] lines = File.ReadAllLines(csvFilePath);

            // Preskoci prvu liniju sa zaglavljem (ako postoji)
            for (int i = 1; i < lines.Length; i++)
            {
                string[] fields = lines[i].Split(',');

                if (fields.Length >= 4)
                {
                    // Ucitaj podatke iz polja i dodaj ih u odgovarajuće nizove
                    questions.Add(fields[0]);
                    answers.Add(fields[1]);
                    domains.Add(fields[2]);
                    sectors.Add(fields[3]);
                }
                else
                {
                    Debug.LogWarning("Linija " + i + " nema dovoljno polja.");
                }
            }

        }
        else
        {
            Debug.LogError("Fajl ne postoji na putanji: " + csvFilePath);
        }

        foreach (string answer in answers)
        {
```

```
        string[] parts = answer.Split(';').Select(s => s.Trim()).ToArray();
        List<string> subList = new List<string>(parts);
        splitAnswers.Add(subList);
    }

    testResults.QuestionDictionary = new Dictionary<int, QuestionData>();
    for (int i = 0; i < questions.Count; i++)
    {
        string questionText = questions[i];
        List<string> answers = splitAnswers[i];
        string domain = domains[i];
        string sector = sectors[i];

        // Kreiranje objekta QuestionData koji ce sadrzati sve informacije o
        pitanju
        QuestionData questionData = new QuestionData
        {
            QuestionText = questionText,
            Answers = answers,
            Domain = domain,
            Sector = sector
        };

        // Dodavanje podataka u rechnik, koristeći broj pitanja kao ključ
        pitanja pocinje od 1
        testResults.QuestionDictionary.Add(i + 1, questionData); // Broj

        //testResults.Date = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
        testResults.Date = DateTime.Now;
        testResults.Title = inputTitle.text;
        testResults.TestID = inputTestID.text;
        testResults.SubjectID = inputSubjectID.text;

        testResults.UserResponses = new Dictionary<int, UserResponse>();
        testResults.Sectors = new Dictionary<string, Sector>();

        foreach (string sectorName in sectors.Distinct().ToList())
        {
            Sector newSector = new Sector(sectorName, 0f);

            foreach (string domainName in domains.Distinct().ToList())
            {
```

```
        Domain newDomain = new Domain(domainName, 0f);
        newSector.Domains.Add(domainName, newDomain);

    }
    testResults.Sectors.Add(sectorName, newSector);

}

    TestTimer();
    SetQuestions();
}

public void LoadTest()
{
    FileBrowser.ShowLoadDialog((paths) =>
    {
        if (paths.Length > 0)
        {
            csvFilePath = paths[0];

            //Debug.Log("Izabrali ste fajl: " + csvFilePath);
            infoText.text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT load
file success");
            startTestButton.gameObject.SetActive(true);

        }
        else
        {
            //Debug.Log("Otvaranje fajla je otkazano.");
            startTestButton.gameObject.SetActive(false);
            infoText.text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT load
file fail");

        }
    },
    null, FileBrowser.PickMode.Files, false, null, null, "Load test file",
"Select"); //
    FileBrowser.SetFilters(true, new FileBrowser.Filter("CSV Files", ".csv"),
new FileBrowser.Filter("All Files", "*.*"));
    FileBrowser.SetDefaultFilter(".csv"); // Postavi "csv file" kao
podrazumevanu vrednost
```



```
}
public void SetQuestions()
{
    question =
gameObject.transform.Find("Question").GetComponent<Dropdown>();

    float questionWidth = 1f / questionsPerRow;
    float questionHeight = 1f / questionsPerColumn;

    int questionsAdded = 0;
    int maxQuestions = questionsPerColumn * questionsPerRow;

    if (containerPanel.childCount > 0)
    {
        GetAnswers();
    }

    foreach (Transform child in containerPanel)
    {
        Destroy(child.gameObject);
    }

    if (questionIndex >= questions.Count )
    //////////////////////////////////////
    ////////////////////////////////////// EXPERIMENT FINISHED
    {

        FinishTest();
        return;
    }

    float xOffset = questionWidth*0.05f;
    float yOffset = questionHeight*0.25f;

    for (int row = 0; row < questionsPerColumn; row++)
    {
```

```
        for (int col = 0; col < questionsPerRow; col++)
        {
            if (questionIndex >= questions.Count | questionsAdded >=
maxQuestions)
            {
                return;
            }
            Dropdown newQuestion = Instantiate(question,
containerPanel.transform);

            float anchorMinX = (col * questionWidth) + xOffset;
            float anchorMaxX = ((col + 1) * questionWidth) - xOffset; //
Oduzimanje xOffset
            float anchorMinY = 1 - ((row + 1) * questionHeight) + yOffset -
yOffset ; // Dodavanje yOffset
            float anchorMaxY = 1 - (row * questionHeight) - yOffset -
yOffset; // Oduzimanje yOffset

            // Postavi poziciju pitanja koristeći anchor
            newQuestion.GetComponent<RectTransform>().anchorMin = new
Vector2(anchorMinX, anchorMinY);
            newQuestion.GetComponent<RectTransform>().anchorMax = new
Vector2(anchorMaxX, anchorMaxY);

            // Postavi offset za anchor na (0, 0)
            newQuestion.GetComponent<RectTransform>().offsetMin =
Vector2.zero;
            newQuestion.GetComponent<RectTransform>().offsetMax =
Vector2.zero;

            // Kreiraj listu opcija odgovora
            List<Dropdown.OptionData> options = new
List<Dropdown.OptionData>();
            foreach (string answer in splitAnswers[questionIndex])
            {
                options.Add(new Dropdown.OptionData(answer));
            }

            // Dodaj opcije u Dropdown
            newQuestion.AddOptions(options);
        }
    }
}
```

```
        // Postavi tekst pitanja
        newQuestion.GetComponentInChildren<Text>().text =
questions[questionIndex]; // Postavi tekst u Text komponenti unutar Dropdown-a

        newQuestion.gameObject.SetActive(true);
        newQuestion.name = "Question_" + (questionIndex+1).ToString(); //
Dodaj redni broj pitanja u ime objekta
        questionIndex++;
        questionsAdded++;
    }
}

private void FinishTest()
{
    finishPanel.gameObject.SetActive(true);
    testPanel.gameObject.SetActive(false);
    TestTimer();
    testResults.ElapsedTime = elapsedTime;

    testResults.Score = 0f;
    foreach (Sector sector in testResults.Sectors.Values)
    {
        sector.Score = 0f;
        foreach (Domain domain in sector.Domains.Values)
        {
            sector.Score += domain.Score;
        }
        if (sector.Name != "0")
        {
            testResults.Score += sector.Score;
        }
    }
}

public void GetAnswers()
{
    foreach (Transform child in containerPanel.transform)
    {
        Dropdown dropdown = child.GetComponent<Dropdown>();
```

```

        if(dropdown != null)
        {
            string dropdownName = dropdown.gameObject.name;
            int startIndex = dropdownName.LastIndexOf("_");
            if (startIndex != -1 && startIndex < dropdownName.Length - 1)
            {
                string questionNumberStr = dropdownName.Substring(startIndex
+ 1);

                if (int.TryParse(questionNumberStr, out int questionNumber))
                {
                    // Pronalazenje odabrane opcije
                    string selectedAnswer = dropdown.captionText.text;

                    // Pronalazenje odgovarajućeg pitanja na osnovu rednog
broja

                    if
(testResults.QuestionDictionary.TryGetValue(questionNumber, out QuestionData
questionData))
                    {
                        // Dodavanje odgovora u recnik ispitanikovih odgovora
                        testResults.UserResponses.Add(questionNumber, new
UserResponse
                        {
                            QuestionText = questionData.QuestionText,
                            SelectedAnswer = selectedAnswer,
                            Domain = questionData.Domain,
                            Sector = questionData.Sector
                        });

                        int answerScore = dropdown.value;
                        if (selectedAnswer == "not relevant") { answerScore =
0; }

                        testResults.Sectors[questionData.Sector].Domains[ques
tionData.Domain].Score += answerScore;
                    }
                }
            }
        }
    }
}

```

```
public void SaveTestResults()
{
    FileBrowser.SetFilters(true, new FileBrowser.Filter("JSON Files",
".json"), new FileBrowser.Filter("All Files", "*.*"));
    FileBrowser.SetDefaultFilter(".json");
    FileBrowser.ShowSaveDialog((path) =>
    {
        if (!string.IsNullOrEmpty(path[0]))
        {
            string jsonData = JsonConvert.SerializeObject(testResults,
Formatting.Indented); //Stara verzija
            //string jsonData = JsonUtility.ToJson(testResults); // Ovo smece
i dalje ne radi

            // Upisivanje JSON stringa u fajl
            File.WriteAllText(path[0], jsonData);

            //Debug.Log("Podaci su sacuvani u JSON fajl: " + path);
            //infoText.text = "Test results saved successfully";
            infoText.text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT save
results success");
        }
        else
        {
            //Debug.Log("Cuvanje podataka otkazano.");
            //infoText.text = "Test results save canceled";
            infoText.text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT save
results canceled");
        }
    },
    null, FileBrowser.PickMode.Files, false, null, null, "Save results",
"Save");
}

public Text textSubjInfo;
public void LoadTestResults()
{
```

```
FileBrowser.SetFilters(true, new FileBrowser.Filter("JSON Files",
".json"), new FileBrowser.Filter("All Files", "*.*"));
FileBrowser.SetDefaultFilter(".json");
FileBrowser.ShowLoadDialog((string[] paths) =>
{
    if (paths.Length > 0)
    {
        string path = paths[0];

        try
        {
            string jsonData = File.ReadAllText(path);

            loadedResults =
JsonConvert.DeserializeObject<Test>(jsonData);

            string subjectID =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT
subject ID"); ;
            string testID =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT test
ID"); ;
            string testTitle =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT test
title"); ;

            textSubjInfo.text = subjectID + ": <b>" +
loadedResults.SubjectID + " </b>" +
            testID + ": <b>" + loadedResults.TestID + " </b>" +
testTitle + ": <b>" + loadedResults.Title + "</b>";

            DisplayResults();

            //infoText.text = "Test results loaded successfully";
            infoText.text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT load
results success");
        }
        catch (Exception e)
        {
            Debug.LogError("Error loading test results: " + e.Message);
            //infoText.text = "Error loading test results";
        }
    }
}
```

```
        infoText.text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT load
results failed");

    }
}
else
{
    //infoText.text = "Load test results canceled";
    infoText.text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT load
results canceled");

}
},
null, FileBrowser.PickMode.Files, false, null, null, "Load results",
"Load");
}
public Dropdown ddChocies;
public GameObject templateQA;
public GameObject scrollContainer;
public void DisplayResults()
{
    int selectedOption = ddChocies.value;

    if (scrollContainer.transform.childCount > 0)
    {
        foreach (Transform child in scrollContainer.transform)
        {
            Destroy(child.gameObject);
        }
    }
    scrollContainer.transform.GetComponent<GridLayoutGroup>().constraint =
GridLayoutGroup.Constraint.Flexible;

    switch (selectedOption)
    {
        case 0:
            DisplayScore();
            break;
        case 1:
            DisplayQuestions();
            break;
        case 2:
            DisplayGraphic();
    }
}
```

```
        break;
    default:
        Debug.LogWarning("Nepoznat indeks.");
        break;
    }
}

public void DisplayScore()
{
    int fieldCounter = 0;
    int numSectors = loadedResults.Sectors.Count;
    int numDomains = 0;
    foreach (Sector sector in loadedResults.Sectors.Values)
    {
        int tmpNumDomains = sector.Domains.Count;
        if (tmpNumDomains > numDomains) { numDomains = tmpNumDomains; }
        GameObject newSectorScore = Instantiate(templateQA,
scrollContainer.transform);
        fieldCounter++;

        // Postavi tekst pitanja
        newSectorScore.transform.Find("Question").GetComponent<Text>().text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT
sector") + ": <b>" + sector.Name + "</b> " +
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT
score") + ":";
        newSectorScore.transform.Find("Answer").GetComponent<Text>().text =
sector.Score.ToString();
        newSectorScore.transform.Find("Answer").GetComponent<Text>().alignmen
t = TextAnchor.UpperCenter;

        newSectorScore.gameObject.SetActive(true);
        newSectorScore.name = "Sector_" + sector.Name;
        foreach (Domain domain in sector.Domains.Values)
        {
            GameObject newDomainScore = Instantiate(templateQA,
scrollContainer.transform);
            fieldCounter++;

            newDomainScore.transform.Find("Question").GetComponent<Text>().te
xt = LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT
sector") + ": <b>" + sector.Name + "</b>, " +
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT
```



```
domain") + ": <b>" + domain.Name + " </b>" +
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT
score") + ":";

        newDomainScore.transform.Find("Answer").GetComponent<Text>().text
= domain.Score.ToString();
        newDomainScore.transform.Find("Answer").GetComponent<Text>().align
ment = TextAnchor.UpperCenter;

        newDomainScore.gameObject.SetActive(true);
        newDomainScore.name = "Sector_" + sector.Name + "_Domain_" +
domain.Name;
    }
}

GameObject allScore = Instantiate(templateQA, scrollContainer.transform);
fieldCounter++;

// Postavi tekst pitanja
allScore.transform.Find("Question").GetComponent<Text>().text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT all
score");
allScore.transform.Find("Answer").GetComponent<Text>().text =
loadedResults.Score.ToString();
allScore.transform.Find("Answer").GetComponent<Text>().alignment =
TextAnchor.UpperCenter;

allScore.gameObject.SetActive(true);
allScore.name = "Overall score";

scrollContainer.transform.GetComponent<GridLayoutGroup>().constraint =
GridLayoutGroup.Constraint.FixedColumnCount;
scrollContainer.transform.GetComponent<GridLayoutGroup>().constraintCount
= numDomains + 1;

}

public void DisplayQuestions()
{
    for (int qaCounter = 0; qaCounter < loadedResults.UserResponses.Count;
qaCounter++)
    {
```

```
        GameObject newQA = Instantiate(templateQA,
scrollContainer.transform);
        newQA.transform.Find("Question").GetComponent<Text>().text =
loadedResults.UserResponses[qaCounter + 1].QuestionText;
        newQA.transform.Find("Answer").GetComponent<Text>().text =
loadedResults.UserResponses[qaCounter + 1].SelectedAnswer;
        newQA.gameObject.SetActive(true);
        newQA.name = "Question_" + (qaCounter + 1).ToString();
    }
}
public void DisplayGraphic()
{

}

public void QutiApp()
{
    Application.Quit();
}
public void ClearInput()
{
    //infoText.text = "Please load the Test file";
    infoText.text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "IT load
results please");
    startTestButton.gameObject.SetActive(false);
    csvFilePath = "";
    if (scrollContainer.transform.childCount > 0)
    {
        textSubjInfo.text = "";
        foreach (Transform child in scrollContainer.transform)
        {
            Destroy(child.gameObject);
        }
    }
}

void OnEnable()
{
    Application.logMessageReceived += HandleLog;
}

void OnDisable()
{
```

```
        Application.logMessageReceived -= HandleLog;
    }

    void HandleLog(string logText, string stackTrace, LogType type)
    {
        if (type == LogType.Error || type == LogType.Exception)
        {
            string logFilePath = "error.log";
            File.AppendAllText(logFilePath, logText + "\n" + stackTrace + "\n");
        }
    }

    private void ToggleFullScreen()
    {
        if (Screen.fullScreenMode == FullScreenMode.ExclusiveFullScreen)
        Screen.fullScreenMode = FullScreenMode.Windowed;
        else Screen.fullScreenMode = FullScreenMode.ExclusiveFullScreen;
    }

[System.Serializable]
public class TextNode
{
    public string Name; // Promenljiva (field) umesto svojstva (property)
    public string Text;
    public Dictionary<string, TextNode> Children { get; set; }

    public TextNode(string name, string text)
    {
        Name = name;
        Children = new Dictionary<string, TextNode>();
        Text = text;
    }
}

private TextNode TraverseAndCollectTextElements(GameObject obj)
{
    TextNode node = new TextNode(obj.name, "");

    // Check if the object has a Text component
    Text textComponent = obj.GetComponent<Text>();
    if (textComponent != null)
    {
        // Set the text for this node
    }
}
```

```
        node.Text = textComponent.text;
    }

    // Recursively check child objects
    foreach (Transform child in obj.transform)
    {
        TextNode childNode =
        TraverseAndCollectTextElements(child.gameObject);
        if (!string.IsNullOrEmpty(childNode.Name))
        {
            node.Children.Add(child.name, childNode);
        }
    }

    return node;
}
}
```

7.2.5.1 *LocaleSelector.cs*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Localization.Settings;
using UnityEngine.Localization.Tables;
using UnityEngine.UI;

public class LocaleSelector : MonoBehaviour
{
    public Dropdown dropdownLanguage;
    public Dropdown dropdownDisplay;

    public void Start()
    {
        StartCoroutine(SetLocale(dropdownLanguage.value));
    }

    public void ChangeLocale()
    {
        StartCoroutine(SetLocale(dropdownLanguage.value));
    }

    IEnumerator SetLocale(int _localeID)
    {
        yield return LocalizationSettings.InitializationOperation;
    }
}
```

```
        LocalizationSettings.SelectedLocale =
LocalizationSettings.AvailableLocales.Locales[_localeID];
        UpdateDds();
    }

    public void UpdateDds()
    {
        // Language Dropdown
        dropdownLanguage.options[0].text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "Dd
En");
        dropdownLanguage.options[1].text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "Dd
In");
        dropdownLanguage.options[2].text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "Dd
Ru");
        dropdownLanguage.options[3].text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "Dd
Sr");
        dropdownLanguage.RefreshShownValue();

        // Display Dropdown
        dropdownDisplay.options[0].text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "Dd
Score");
        dropdownDisplay.options[1].text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "Dd
Questions");
        dropdownDisplay.options[2].text =
LocalizationSettings.StringDatabase.GetLocalizedString("Language Table", "Dd
Graphics");
        dropdownDisplay.RefreshShownValue();
    }
}
```

7.3 Списак објављених научних радова

M23 - Petrović, N., Mandić, S., Borojević, S., Gazivoda, N., Sovilj, P., 2024. Cognitive Phenomena Measurement with Time Window-Based Multispectral Brain Mapping. *Technology and Health Care*, 32(2), pp.799-808.

M23 - Novakovic, D., Sovilj, P., Petrovic, N., Milovanovic, M., Makal, J. and Walendziuk, W., 2020. Measurement of Event-Related Brain Potentials (ERP) Amplitude and Latency Based on Digital Stochastic Measurement over Interval. *Elektronika ir Elektrotehnika*, 26(2), pp.59-68.

M33 - Tsompou, P.I., Potsika, V.T., Petrovic, N., Pezoulas, V.C., Siogkas, P.K., Tsakanikas, V.D., Pleouras, D.S., Papafaklis, M., Nikopoulos, S., Sakellarios, A.I. and Fotiadis, D.I., 2022, September. Computational modeling of atherosclerotic plaque progression through an efficient 3D agent-based modeling approach. In 2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI) (pp. 01-04). IEEE.

M33 - M.A. AboArab MA, V.T. Potsika, N.J. Petrović and D.I. Fotiadis. (2022). DECODE cloud platform: A new cloud platform to combat the burden of peripheral artery disease. PACET 2022 PAhellenic Conference on Electronics and Telecommunications, 2-3 December 2022, Tripoli, Greece.

M33 - N.J. Petrović, V. T. Potsika, M. AboArab, M. Zacharopoulou, L. Tscheuschner, A. Chatziioannou, F. Sigala and D.I. Fotiadis. (2022). Tool for peripheral artery segmentation and reconstruction from angiography images. 13th HSTAM International Congress on Mechanics, 24-27 August 2022, Patras, Greece.

M33 - Petrović, N., Borojević, S., 2022. Multispectral brain mapping and its significance for validation of results of Neurofeedback experiments. In 11th International Student Scientific and Practical Conference "Cross-Cultural Communication in the Modern World". December 2022, Penza, Russian Federation.

M33 - Petrović, N., Borojević, S., 2022. INFLUENCE OF RUSSIAN SONGS ON THE COGNITIVE AND EMOTIONAL STATES OF THE SUBJECT. In 11th International Student Scientific and Practical Conference "Cross-Cultural Communication in the Modern World". December 2022, Penza, Russian Federation.

M63 - Petrović, N., Pejić, D., Urekar, M., Novaković, Đ, Gazivoda, N. (2019). Simulacioni model stohastičkog flash A/D konvertora. In Etran conference. June 2019.

7.4 Биографија кандидата

Никола Петровић рођен је 18. октобра 1994. године у Републици Србији.

Основну школу и Средњу машинску школу завршио је у Новом Саду, смер Машински техничар за роботiku, од 2009. до 2013. године. Дипломски рад на тему „Аутоматизовани флексибилни систем обраде“ одбранио је као техничар за роботiku.

На Факултету техничких наука Универзитета у Новом Саду уписао је ОАС Биомедицинско инжењерство, студирао је од 2013. до 2017. године. Дипломски рад на тему „Експериментална платформа за истраживање менталне концентрације заснованог на мерењу ЕЕГ сигнала“ одбранио је са стеченим звањем инжењера. Мастер студије на истом факултету завршио је 2018. године, с мастер радом на тему „Развој мерно-информационог система за истраживање когнитивних појава заснованог на мерењу ЕЕГ сигнала“.

На Факултету техничких наука уписао је докторске студије 2018. Године, студијски програм Енергетика, електроника и телекомуникације, саветник проф. др Платон Совиљ. Године 2020. почео је и докторске студије на истом факултету, смер Биомедицинско инжењерство.



Радно искуство:

- Универзитет у Новом Саду, Факултет техничких наука, Департман за енергетику, електронику и телекомуникације, Катедра за електроенергетику и примењено софтверско инжењерство 2023 - : асистент.
- Одељење за науку о материјалима и инжењерству, Универзитет у Јањини, 2021/2022: истраживач у раној фази, стипендиста и докторанд Фондације „Marie Skłodowska-Curie Action“.
- Метакогнис Истраживачки центар, 2020-2021: научни истраживач. Развој и имплементација програма у Matlab, Python, LabView, Unity и C#. Организација задатака и подзадатака на више пројеката и истраживачких програма.
- Компанија „Continental Automotive“, 2019-2020: инжењер за развој софтвера. Развој и имплементација функционалних програма у Matlab, Simulink и MBDS.
- Универзитет ЕДУКОНС, Факултет информacionих технологија, 2018/2019: сарадник у настави. Обавезе су укључивале припрему и извођење вежби из курсева „Рачунарска интелигенција“ и „Базе података“, као и курсеве у програмским језицима Rstudio, Matlab и MySQL Workbench.

План третмана података

Назив пројекта/истраживања
Методe мерења електричних активности мозга у контексту истраживања квантитативне електроенцефалографије, неурофидбека и неуропластичности
Назив институције/институција у оквиру којих се спроводи истраживање
а) Факултет техничких наука Универзитета у Новом Саду б)
Назив програма у оквиру ког се реализује истраживање
Докторске академске студије – Факултет техничких наука Универзитета у Новом Саду
1. Опис података
<p><i>1.1 Врста студије</i></p> <p><i>Укратко описати тип студије у оквиру које се подаци прикупљају</i> <u>Подаци су прикупљени из више јавно доступних научних извора који су цитирани у докторској дисертацији.</u></p> <p>_____</p> <p>_____</p> <p>_____</p> <p>1.2 Врсте података а) квантитативни б) квалитативни</p> <p>1.3. Начин прикупљања података а) анкете, упитници, тестови б) клиничке процене, медицински записи, електронски здравствени записи в) генотипови: навести врсту _____ г) административни подаци: навести врсту _____ д) узорци ткива: навести врсту _____ ђ) снимци, фотографије: EDF, CSV, JSON, JPG, MKV _____ е) текст, навести врсту _____ ж) мапа, навести врсту _____ з) остало: из јавно доступних научних извора _____</p> <p>1.3 Формат података, употребљене скале, количина података</p> <p>1.3.1 Употребљени софтвер и формат датотеке: а) Excel фајл, датотека .xlsx, .csv _____ б) SPSS фајл, датотека _____ с) PDF фајл, датотека .pdf _____ д) Текст фајл, датотека .docx _____ е) JPG фајл, датотека .jpg _____ ф) Остало, датотека .edf, .json, .mkv _____</p>

1.3.2. Број записа (код квантитативних података)

- а) број варијабли 26
 б) број мерења (испитаника, процена, снимака и сл.) 36 испитаника, на сваком је извршено два мерења што значи да је укупан број мерења 72

1.3.3. Поновљена мерења

- а) да
 б) не

Уколико је одговор да, одговорити на следећа питања:

- а) временски размак између поновљених мера је: мерења су извршена непосредно једно за другим
 б) варијабле које се више пута мере односе се на: сигнале ЕЕГ канала испитаника
 в) нове верзије фајлова који садрже поновљена мерења су именоване као: инкрементална промена последње цифре у називу фајла(нпр: Subject00_1 и Subject00_2)

Напомене: _____

Да ли формати и софтвер омогућавају дељење и дугорочну валидност података?

- а) Да
 б) Не

Ако је одговор не, образложити _____

2. Прикупљање података

2.1 Методологија за прикупљање/генерисање података

2.1.1. У оквиру ког истраживачког нацрта су подаци прикупљени?

- а) експеримент, навести тип _____
 б) корелационо истраживање, навести тип _____
 ц) анализа текста, навести тип _____
 д) остало, навести шта: из јавно доступних научних извора

2.1.2 Навести врсте мерних инструмената или стандарде података специфичних за одређену научну дисциплину (ако постоје).

Мерење је вршено уређајем Neurocom EEG са 23 канала, а подаци су сачувани у EDF формату.

2.2 Квалитет података и стандарди

2.2.1. Третман недостајућих података

- а) Да ли матрица садржи недостајуће податке? Да **Не**

Ако је одговор да, одговорити на следећа питања:

- а) Колики је број недостајућих података?

- б) Да ли се кориснику матрице препоручује замена недостајућих података? Да Не
 в) Ако је одговор да, навести сугестије за третман замене недостајућих података

2.2.2. На који начин је контролисан квалитет података? Описати

Квалитет података је контролисан од стране аутора истраживања, а читав поступак је објашњен у јавно доступном извору који је цитиран.

2.2.3. На који начин је извршена контрола уноса података у матрицу?

Подаци су учитавани помоћу стандардне Python библиотеке за учитавање и обраду EDF фајлова, која је наведена у раду, а контрола је извршена директном инспекцијом и увидом у уčitане податке и поређењем са подацима из извора.

3. Третман података и пратећа документација

3.1. Третман и чување података

3.1.1. Подаци ће бити депоновани у _____ репозиторијум.

3.1.2. URL адреса <https://physionet.org/content/eegmat/1.0.0/>

3.1.3. DOI <https://doi.org/10.3390/data4010014>

3.1.4. Да ли ће подаци бити у отвореном приступу?

а) Да

б) Да, али после ембарга који ће трајати до _____

в) Не

Ако је одговор не, навести разлог _____

3.1.5. Подаци неће бити депоновани у репозиторијум, али ће бити чувани.

Образложење

3.2 Метаподаци и документација података

3.2.1. Који стандард за метаподатке ће бити примењен? Метаподаци су сачувани табеларно у CSV формату

3.2.1. Навести метаподатке на основу којих су подаци депоновани у репозиторијум.

Године, Пол, Година мерења, Број рачунских операција, Квалитет рачунања

Ако је потребно, навести методе које се користе за преузимање података, аналитичке и процедуралне информације, њихово кодирање, детаљне описе варијабли, записа итд.

3.3 Стратегија и стандарди за чување података

3.3.1. До ког периода ће подаци бити чувани у репозиторијуму? Подаци су трајно јавно доступни

3.3.2. Да ли ће подаци бити депоновани под шифром? Да **Не**

3.3.3. Да ли ће шифра бити доступна одређеном кругу истраживача? Да **Не**

3.3.4. Да ли се подаци морају уклонити из отвореног приступа после извесног времена?

Да **Не**

Образложити

4. Безбедност података и заштита поверљивих информација

Овај одељак МОРА бити попуњен ако ваши подаци укључују личне податке који се односе на учеснике у истраживању. За друга истраживања треба такође размотрити заштиту и сигурност података.

4.1 Формални стандарди за сигурност информација/података

Истраживачи који спроводе испитивања с људима морају да се придржавају Закона о заштити података о личности (https://www.paragraf.rs/propisi/zakon_o_zastiti_podataka_o_licnosti.html) и одговарајућег институционалног кодекса о академском интегритету.

4.1.2. Да ли је истраживање одобрено од стране етичке комисије? Да **Не**

Ако је одговор Да, навести датум и назив етичке комисије која је одобрила истраживање

4.1.2. Да ли подаци укључују личне податке учесника у истраживању? Да **Не**

Ако је одговор да, наведите на који начин сте осигурали поверљивост и сигурност информација везаних за испитанике:

- а) Подаци нису у отвореном приступу
- б) Подаци су анонимизирани
- ц) Остало, навести шта

5. Доступност података

5.1. Подаци ће бити

а) јавно доступни

б) доступни само уском кругу истраживача у одређеној научној области

ц) затворени

Ако су подаци доступни само уском кругу истраживача, навести под којим условима могу да их користе:

Ако су подаци доступни само уском кругу истраживача, навести на који начин могу приступити подацима:

5.4. Навести лиценцу под којом ће прикупљени подаци бити архивирани.

Подаци тренутно нису архивирани под лиценсом већ су јавно доступни

6. Улоге и одговорност

6.1. Навести име и презиме и мејл адресу власника (аутора) података

Ivan Seleznov, isel@ciklum.com

6.2. Навести име и презиме и мејл адресу особе која одржава матрицу с подацима

Ivan Seleznov, isel@ciklum.com

6.3. Навести име и презиме и мејл адресу особе која омогућује приступ подацима другим истраживачима

Подаци су јавно доступни
