



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ANALÝZA A KLASIFIKACE DAT ZE SNÍMAČE MOZKOVÉ
AKTIVITY**

DATA ANALYSIS AND CLASIFICATION FROM THE BRAIN ACTIVITY DETECTOR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN JILEČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÖKE, Ph.D.

BRNO 2019

Zadání diplomové práce



21897

Student: **Jileček Jan, Bc.**
Program: Informační technologie Obor: Počítačová grafika a multimédia
Název: **Analýza a klasifikace dat ze snímače mozkové aktivity**
Data Analysis and Clasificacion from the Brain Activity Detector
Kategorie: Zpracování řeči a přirozeného jazyka

Zadání:

1. Nastudujte projekt openbci.com. Seznamte se s dodaným HW. Nastudujte dostupné knihovny a literaturu o snímání aktivity mozku (EEG).
2. Vyberte nebo navrhnete datový set, a posbírejte dostatek dat (EEG signály).
3. Analyzujte získaná data, navrhnete a implementujte techniky pro extrakci vhodných parametrů a klasifikaci tříd (machine learning). Vyhodnoťte dosažené výsledky.
4. Zhodnoťte, úroveň poznání ve světě a v rámci openbci komunity. Zhodnoťte dosažené výsledky v rámci vaší práce. Diskutujte možnosti využití openbci pro ovládání jednoduchých aplikací nebo her.
5. Zhodnoťte výsledky a navrhnete směry dalšího vývoje.
6. Vytvořte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Dále dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a část bodu 3 ze zadání.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Szóke Igor, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 10. května 2019

Abstrakt

V této práci zkoumám snímač mozkové aktivity OpenBCI Ultracortex IV EEG ve formě headsetu, problematiku nahrávání EEG dat, neurofeedbacku a metody strojového učení z nasbíraných dat mozkové aktivity sensorimotorické části mozku, ve které sleduji EEG signatury záměru pro pohyb levé a pravé ruky. Výstupem je několik datových sad pro trénování a testování, nástroj pro sběr dat mozkové aktivity, demo neurofeedback aplikace a statistické údaje vycházející z analýzy nasbíraných dat. Čerpám zde z existujících výzkumů, se kterými porovnávám dosažené výsledky. Použité metody a jejich úspěšnost popíšu a vyhodnotím jejich efektivitu.

Abstract

This thesis aims to implement methods for recording EEG data obtained with the neural activity sensor OpenBCI Ultracortex IV headset. It also describes neurofeedback, methods of obtaining data from the motor cortex for further analysis and takes a look at the machine learning algorithms best suited for the presented problem. Multiple training and testing datasets are created, as well as a tool for recording the brain activity of a headset-wearing test subject, which is being visually presented with cognitive challenges on the screen in front of him. A neurofeedback demo app has been developed, presented and later used for calibration of new test subjects. Next part is data analysis, which aims to discriminate the left and right hand movement intention signatures in the brain motor cortex. Multiple classification methods are used and their utility reviewed.

Klíčová slova

OpenBCI, Ultracortex, BCI, ovládání myšlenkami, EEG, biofeedback, mozek, soustředění, pozornost, prefrontální kortex, motorický kortex, python, unity, lstreamer, gui, openvibe, systém 10-20, mozkové vlny, DWT, wavelet, gradient boosting, random forests, support vector machines

Keywords

OpenBCI, Ultracortex, BCI, mind-controlled game, EEG, biofeedback, brain, focus, attention, prefrontal cortex, motor cortex, meditation, python, unity, lstreamer, gui, openvibe, 10-20 system, brain waves, DWT, wavelet, gradient boosting, random forests, support vector machines

Citace

JILEČEK, Jan. *Analýza a klasifikace dat ze snímače mozkové aktivity*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szöke, Ph.D.

Analýza a klasifikace dat ze snímače mozkové aktivity

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně pod vedením pana Ing. Igora Szöke, Ph.D. První fáze projektu pro vytváření datové sady proběhla v týmové spolupráci se Zdeňkem Jelínkem a Petrem Ullrichem. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Jileček
22. května 2019

Poděkování

Tímto bych rád poděkoval panu Ing. Igoru Szökemu, Ph.D. a Doc. Dr. Ing. Janovi Černockému za odborné vedení a týmovým kolegům Petru Ullrichovi a Zdeňku Jelínkovi za skvělou spolupráci.

Obsah

1	Úvod	3
1.1	Motivace	3
1.2	Přínos	3
2	Teoretický základ, cíle a existující výzkumy	5
2.1	Základy neuroinformatiky	5
2.2	Problematika měření EEG pomocí BCI	7
2.2.1	Prefrontální kortex	7
2.2.2	Motorická kůra	9
2.2.3	Mozkové vlny	10
2.3	Existující výzkumy	12
2.3.1	Existující Hardware/Headware	16
3	Sběr dat	20
3.1	OpenBCI Ultracortex	20
3.2	OpenBCI GUI a Hub	22
3.2.1	Instalace a spuštění	22
3.2.2	Moduly	23
3.2.3	Processing	24
3.3	Visual Prompter, generátor datové sady	25
3.4	Formát dat	28
3.4.1	JSON vstupní data	28
3.4.2	Fáze	29
3.4.3	L-streamer data	29
3.4.4	Formát EEG streamu	30
3.4.5	Nahrávání webkamery	30
3.4.6	QT GUI	31
3.5	Sound Prompter	31
3.6	Spektogram alfa vln	32
3.7	OpenVIBE	32
3.8	Metodologie sbírání dat	33
3.9	Použité datové sady	35
4	Demo aplikace pro ExcelFit	36
4.1	Aplikace NeuroBuddha	36
4.1.1	Neurofeedback	37
4.2	Architektura hry	37

5	Data pre-processing - předzpracování dat	39
5.1	Algoritmy	39
5.1.1	Extrakce rysů	39
5.2	Metody redukce dimenzionality	42
5.3	Klasifikátory v balíku scikit-learn	43
5.3.1	Random Forests	43
5.3.2	Gradient Boosting Classifier	45
5.3.3	Support Vector Machines	46
5.3.4	Metrika ROC a AUC	47
5.4	Způsoby zpracování dat	48
5.4.1	Architektura frameworku pro experimenty a formát dat	48
5.4.2	Moduly	48
5.4.3	Použití	49
6	Experimentální testování	51
6.1	Gradient Boosting	51
6.1.1	Otevřené/zavřené oči	51
6.1.2	Sensorimotorika	52
6.2	Random Forests	59
6.3	Support Vector Machines	65
6.4	Testy zvukových datasetů	66
6.5	Výsledky	67
6.6	Diskuze a srovnání	68
7	Závěr	70
8	Přílohy	71
	Literatura	72

Kapitola 1

Úvod

1.1 Motivace

V této práci zkoumám možnosti ovládní hry/aplikace pomocí pro uživatele cenově dostupného headsetu pro čtení mozkových EEG vln. EEG neboli elektroencefalografie je metoda pro zaznamenávání elektrické aktivity mozku skrz detekci povrchového napětí na pokožce hlavy, kam probíhají elektrické impulsy probíhající při přenášení informací mezi neurony v mozku. Díky stále se rozvíjející oblasti Brain Computer Interfaces (BCI), neboli rozhraní pro komunikaci mezi mozkiem a počítačem, a prohlubujícím se neurologickým znalostem o struktuře mozku můžeme využívat BCI efektivněji, rychleji a levněji. Pro to je zde využít relativně levný BCI hardware pro měření EEG impulsů, OpenBCI Ultracortex IV (dále UCIV), který se cenově pohybuje v řádech desítek tisíc korun, kdežto „profesionální“ EEG hardware pro vědecké výzkumy je cenově v řádech stovek tisíc až milionů korun. UCIV si tak může dovolit široké spektrum lidí. Práce je zaměřena na ovládní her pomocí detekce představovaného pohybu [Obr. 3.7] nebo také „záměru“ pro pohyb v centrech mozku zodpovědných za provádění těchto akcí, konkrétně sensorimotorického kortexu [Sekce 2.2.2], a jeden experimentální způsob ovládní her pomocí detekce soustředěné pozornosti uživatele [Sekce 4.1.1]. Oba způsoby lze využít pro ovládní hardware zařízení (robotická paže, vozíky, letadla, auta..) i software aplikací (psaní na obrazovku, navigace v aplikacích..), využitelné jsou pak ve více oblastech. Pro detekci představovaného pohybu levé a pravé ruky pomocí OpenBCI UCIV neexistuje mnoho výzkumů, a ty existující nevykazují přílišnou úspěšnost. V této práci se zaměřuji také na experimentální použití a případné zlepšení dosavadních výsledků.

1.2 Přínos

Pro UCIV jsem v této práci navrhl a implementoval nástroj pro sběr EEG dat účastníků při prováděných testovacích sezeních [Kapitola 3.8] dále využívaný i dvěma kolegy v týmu při počáteční fázi vývoje. Navrhl a implementoval jsem framework pro klasifikaci nasbíraných dat [Sekce 5.3], který umožňuje provádění hromadných experimentů na velké škále parametrů pro jednotlivé klasifikační algoritmy [Kapitola 6], pro implementované metody k předzpracování dat [Sekce 5], a pro jednotlivé datové sady [Tabulka 3.1]. Analýzu a trénování dat provádím z důvodu relativně velké časové/výpočetní náročnosti na cloudových službách Metacentrum.¹

¹<https://metavo.metacentrum.cz/cs/>

V této práci používám OpenBCI Ultracortex IV s 8 elektrodami a zakoupeným vodivým gelem pro lepší vodivost pokožky hlavy. Headset jsem vyzkoušel na 12 testovacích sadách získaných od testerů UCIV headsetu, od nichž jsem nasbíral data pomocí mnou vytvořeného nástroje pro sběr dat, které jsem pojmenoval Visual Prompter (sběr dat se zavřenými očima) a Sound Prompter (dává zvukovou instrukci při zavřených očích, aby se testovací subjekt mohl orientovat). Data jsou sbírána při předložené výzvě pro představování si pohybu levé nebo pravé ruky. Dalším segmentem je sběr a kalibrace dat pro jednoduchou verzi detekce pozornosti, také známou jako neurofeedback. Pro tento účel jsem provedl výzkum nejlepších metod pro detekci soustředění a vyvinul hru ve frameworku Unity, kterou lze ovládat myslí. Tuto hru jsem prezentoval na konferenci ExcelFit.

Nasbíraná data pro záměr pohybu levé a pravé ruky jsem zpracoval, analyzoval a podle nastudovaných materiálů jsem vyzkoušel různé metody získání relevantních dat a experimentoval jsem s vlastními metodami. Pro extrakci relevantních rysů pro následné trénování klasifikátorů získanými daty jsem použil analýzu spektra a statistických údajů o vstupním signálu pomocí Discrete Wavelet transform (DWT) (Diskrétní Wavelet transformace) [Sekce 5.1.1]. Použil jsem více kombinací těchto metod, a to včetně hlubší statistické analýzy transformovaného signálu, metod pro snížení dimenze dat (PCA, ICA, ...), experimentaci s časovým oknem (pohyb okna/sliding window a jeho velikost), míru rozkladu pomocí DWT a parametry jednotlivých klasifikátorů. Výsledky experimentů jsem porovnal mezi sebou a ty nejlepší také s existujícími výzkumy.

Kapitola 2

Teoretický základ, cíle a existující výzkumy

Brain Computer Interfaces (BCI), neboli rozhraní pro komunikaci mezi mozkem a počítačem, dnes existují hlavně ve vědeckých sférách a BCI hardware je na této úrovni cenově velmi nedostupný (\$20 000-\$50 000) ¹.

Dobrou alternativou pro běžné uživatele, co chtějí experimentovat s BCI nebo pro pomoc uživatelům neschopným pohybu jsou tak levnější BCI zařízení, jakým je např. OpenBCI UCIV, používaný v této práci ². Headset podporuje až 16 elektrod pro rozmístění po ploše hlavy a je k dispozici ve více konfiguracích. Verze headsetu Print it yourself, obsažena v balíčku DIY Neurotechnologist's Starter Kit zakoupeného VUT FIT za cca \$945 ³, používá 8 elektrod [Obr. 2.1].

2.1 Základy neuroinformatiky

Neuroinformatika je rozsáhlý obor zabývající se zpracováním dat z oblastí neurověd pomocí výpočetních nástrojů. Pro porozumění dále prezentovaným výsledkům je nejprve třeba porozumět základním pojmům, které v této práci používám.

Brain Computer Interface BCI je zařízení, které nevyžaduje ovládání pomocí fyzické aktivity, ale spoléhá čistě na čtení mozkových vzruchů přes elektrody rozmístěné na hlavě. Toto zařízení lze pak použít pro ovládání široké škály dalšího hardware, jako jsou počítače, vozíky, robotické paže, pro implementaci robotického vidění atp. BCI zařízení jsou často používána pro výzkum, mapování, asistenci, augmentaci nebo opravu kognitivních nebo sensorimotorických funkcí [29].

Elektroencefalografie (EEG) je metoda pro zaznamenávání elektrické aktivity mozku skrz detekci povrchového napětí na pokožce hlavy, kam probíhají elektrické impulsy probíhající při přenášení informací mezi neurony v mozku. K získání elektrického výstupu jsou použity elektrody, které jsou umístěny na pokožku hlavy (v některých případech se používá vodivý

¹<https://sapienlabs.co/emotiv-epoc-and-openbci-ultracortex-style-and-function/>

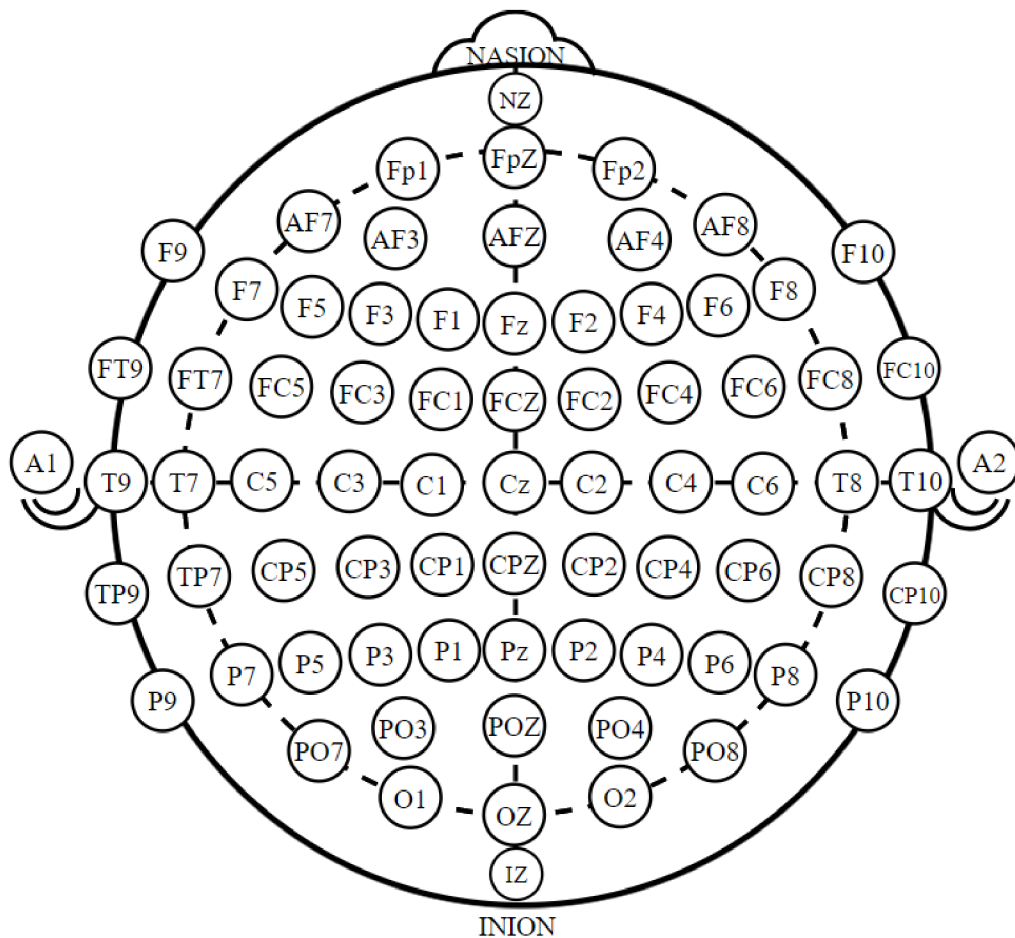
²<https://openbci.com/>

³<https://shop.openbci.com/collections/frontpage/products/d-i-y-neurotechnologists-starter-kit>

gel, jelikož např. ženy mají menší vodivost kůže než muži [24]). Tyto elektrody pak čtou mozkové impulsy ve formě signálu, který odpovídá části mozku, která jej vysílá.

Systém 10-20 Elektrody jsou podle mezinárodně uznávaného **Systému 10-20** rozmístěny na odpovídající pozice na hlavě. Čísla 10 a 20 v názvu značí vzdálenost 10 % a 20 % od přední/zadní strany hlavy a levé/pravé strany hlavy. Každá pozice má přiřazenou jinou oblast mozkové kůry ⁴.

Pro měření pohybových aktivit jsou ve výzkumech [21] použita různá rozložení elektrod v 10-20 systému. Ať už jde o BCI systémy s 60-, 64- nebo 128-kanálovým vstupem, pro analýzu mozkových vln pro detekci pohybových událostí zůstává nejdůležitější trojice C3, Cz a C4, jelikož jsou tyto elektrody umístěny přímo nad sensorimotorickým kortexem [Sekce 2.2.2] zodpovědným za ovládání pohybových aktivit.



Obrázek 2.1: Systém 10-20

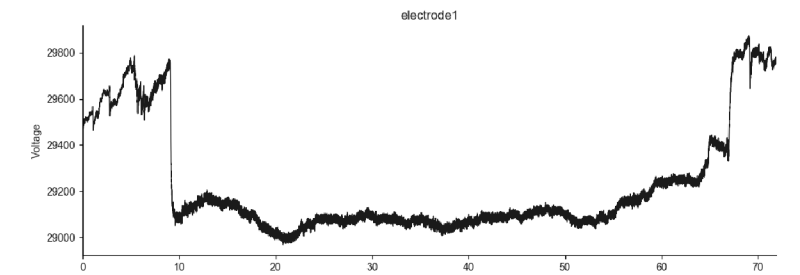
V případě headsetů s 8 elektrodami bývají tyto pozice zastoupeny následovně: ⁵:

- Kanál 1 - FP1 – Prefrontální kortex

⁴[https://en.wikipedia.org/wiki/10%E2%80%9320_system_\(EEG\)](https://en.wikipedia.org/wiki/10%E2%80%9320_system_(EEG))

⁵<http://docs.openbci.com/Headware/01-Ultracortex-Mark-IV#ultracortex-mark-iv-assembly-instructions-electrode-location-overview>

- Kanál 2 - FP2 – Prefrontální kortex



Obrázek 2.2: Napětí z elektrody FP2 UCIV při meditaci a zavřených očích - lze vidět pokles při zavření očí

- Kanál 3 - C3 – Centrální
- Kanál 4 - C4 – Centrální
- Kanál 5 - P7 – Parietální
- Kanál 6 - P8 – Parietální
- Kanál 7 - O1 – Occipitalis (Zde experimentuji i s pozicí Cz místo O1)
- Kanál 8 - O2 – Occipitalis

2.2 Problematika měření EEG pomocí BCI

V této části se zabývám problémy při používání BCI na testovacích subjektech a tím, jak zvolit nejvhodnější strategii pro obdržení co nejčistších dat z mozku uživatele. Ke zvolení nejlepší strategie je nejprve nutno prozkoumat sledovaný systém, v tomto případě mozek. Části mozku, kterými se v této práci zabývám, jsou prefrontální kortex a motorická kůra. Prefrontální kortex je zodpovědný hlavně za soustředěnou pozornost a proto ho využívám pro ovládání jednoduché hry založené na detekci míry soustředění [Sekce 4.1]. Další analyzovanou oblastí je motorická kůra, zodpovědná za vznik a zpracování impulzů k pohybu končetin, ale také k pouhému představování pohybu, který chce člověk provést. Na této části mozku provádím rozsáhlý výzkum a nabytá data nadále analyzuji pro jejich pozdější utilizaci (zejména pro využití výsledků klasifikace EEG signatur pro ovládání hry).

2.2.1 Prefrontální kortex

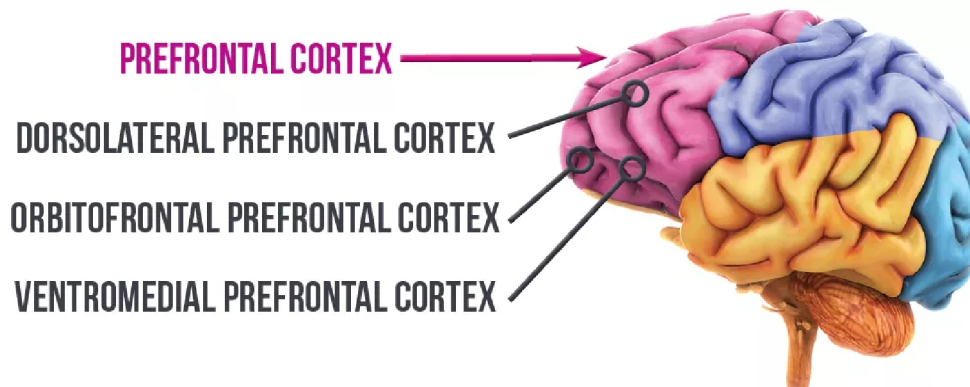
Prefrontální kortex (PFC) je částí mozku sídlící v přední části lebky, za čelem. Napětí z této části čtou elektrody FP1 a FP2.

PFC je zodpovědný za plánování komplexních kognitivních úloh, projevování vlastní osobnosti, rozhodovací procesy a tlumení společensky nepřijatelného chování, sebekontrolu, potlačování impulzů a schopnost dokončování úloh. Také zajišťuje vůli jednat v souladu se svými vnitřními cíli a přesvědčeními a udržuje tzv. „pracovní paměť“, je tak absolutně nezbytný pro manažerské a vůdcovské schopnosti. PFC je hlavním centrem soustředěné pozornosti a jako takový spoléhá nejvíce na neurotransmiter a katecholamin dopamin, který je hlavním aktérem při závislostech, chuti k jídlu a očekávání odměny [37]. Důležitou úlohou

Tabulka 2.1: Tabulka elektrod, oblastí a jejich kognitivních funkcí

<p>FP1 N1P PREFRONTAL logická pozornost organizace plánování rozhodování dokončování úloh rychlá provozní paměť</p>	<p>FP2 N2P PREFRONTAL emoční pozornost posuzování cítění svého já sebekontrola zadržování impulzů</p>	<p>C3 N3P CENTRAL integrace sensorimotoriky ovládání pravé horní části reflexní reakce psaní pravou rukou</p>
<p>C4 N4P CENTRAL integrace sensorimotoriky ovládání levé horní části uklidňující reakce psaní levou rukou</p>	<p>P7 M5P PARIETAL verbální logika porozumění rozpoznání slov zpracování sluchových vjemů</p>	<p>P8 N6P PARIETAL porozumění emocím rozpoznání tváří rozpoznání symbolů zpracování sluchových vjemů</p>
<p>O1 N7P CENTRAL vizuální zpracování prostorové části rozpoznání barev a vzorů vnímání pohybu vnímání černé a bílé vnímání hran</p>	<p>O2 N8P OCCIPITAL vizuální zpracování levé části prostoru rozpoznání vzorů vnímání pohybu vnímání černé a bílé vnímání hran</p>	

PFC je také tlumení přichozivších signálů od amygdaly, která vysílá emocionální signály a tzv. “fight or flight response” (bojuj nebo uteč) - úkolem PFC je pak tyto signály regulovat (např. pokud člověk stojí v zoo a od lva ho odděluje sklo, amygdala vysílá signál k útěku, kdežto PFC tento signál racionálně potlačí).



Obrázek 2.3: PFC [12]

2.2.2 Motorická kůra

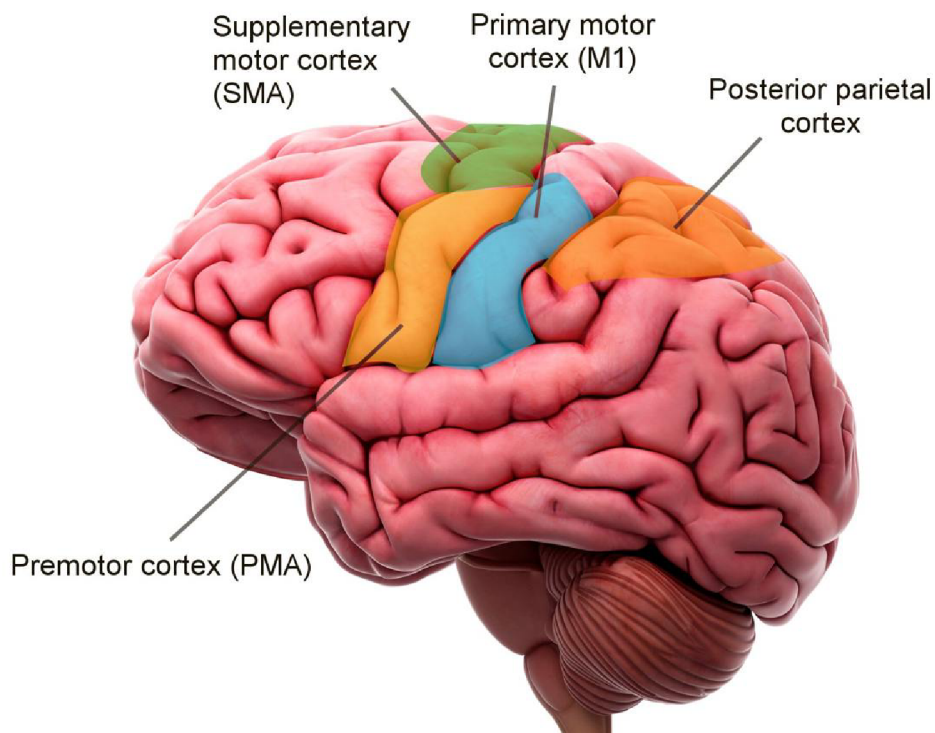
Motorická kůra je část mozkové kůry mající na starosti plánování, ovládání a provádění vědomých pohybů. Obecně se nachází v přední mozkové kůře. ⁶

V mozku je za pohyb a koordinaci zodpovědný právě primární motorický kortex (sensorimotorický kortex) a mozeček. Při koordinaci pohybu jsou neurony v těchto částech vzrušeny a začnou zasílat nervové signály, které jsou schopny aktivovat svaly. Tyto signály jsou tak silné, že probíjí na pokožku hlavy, kde je následně je schopný detekovat EEG headset. V této práci využívám data z elektrod hlavně na pozicích C3, Cz a C4, které pokrývají právě sensorimotorický kortex.

Motorická kůra má 3 hlavní části:

- Primární kortex - je zodpovědný za vytváření neuronových impulsů a jejich zasílání
- Premotorický kortex - jeho funkce není plně prozkoumána, pravděpodobně je však jeho funkcí příprava na pohyb, synchronizace smyslových vjemů a pohybu a prostorová orientace při uchopování předmětů
- Doplňkový kortex - jeho funkce také není plně známa, má však řadu navrhovaných funkcí, jako plánování pohybu, plánování sekvencí pohybu a koordinaci obou stran těla (např. žonglování)

⁶https://en.wikipedia.org/wiki/Motor_cortex



Obrázek 2.4: Lokace motorického kortexu v mozku [1]

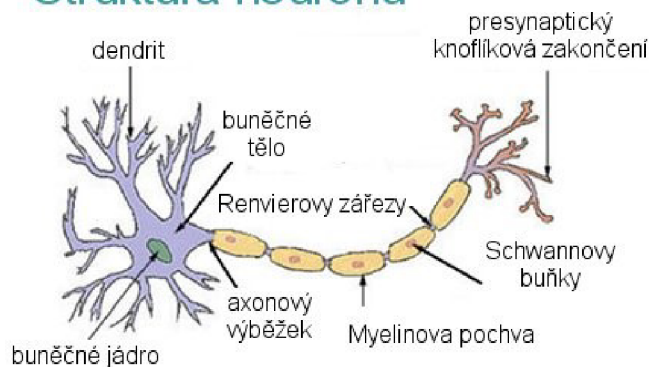
2.2.3 Mozkové vlny

Mozkové vlny mají různou frekvenci, na základě které můžeme přibližně určit, v jakém stavu se zkoumaný mozek nachází. Pokud jsou v mozku dominantní pomalé mozkové vlny, resp. vlny nízké frekvence, člověk se může cítit pomalý neb unavený. Naopak čím vyšší frekvence, tím je obecně člověk soustředěnější a ve střehu. Mozkové vlny se měří v Herzích (Hz). Většina mozkových vln se vyskytuje ve frekvenčním rozsahu od 0.5 do 100 Hz ⁷. Každé vlnové spektrum je symptomem jiného stavu mozku a každá mozková oblast má jiné „rozpoložení“ těchto vln. Pro tento výzkum jsou nejdůležitější tzv. Mu rytmy [Sekce 2.2.3] přítomné při pohybových událostech. Další sledované rytmy jsou Alfa, Beta a Theta - pomocí určení jejich poměru lze detekovat soustředěnou pozornost. [Sekce 4.1.1]

V sensorimotorickém kortexu jsou sledované hlavně rolandické Mu rytmy (7.5-12.5 Hz) a centrální Beta rytmy (>13 Hz) [25], které jsou relevantní pro BCI výzkumy řešící rozlišení mezi úmyslem k pohybu pravou/levou rukou, a tedy i pro tuto diplomovou práci. Při představování a provádění pohybových aktivit je odezva těchto částí mozku způsobena neurochemickými změnami, které se projevují vzájemnou interakcí synapsí a vnitřních membránových vlastností lokálních neuronů [21]. Mu a Beta rytmy jsou potlačeny během desynchronizace (Event-Related Desynchronization - ERD). Při pohybu dojde v sensorimotorických centrech mozku k potlačení právě Mu a centrálních Beta rytmů v lokalizovaných oblastech velkého množství neuronů. Opakem této události je synchronizace, kdy se tyto frekvence projeví silněji [21].

⁷<https://brainworksneurotherapy.com/what-are-brainwaves>

Struktura neuronu



Obrázek 2.5: Struktura neuronu

Nyní se podíváme na podtřídy frekvenčního spektra rozdělené na základě vlnové frekvence.

Alfa vlny – 8-13 Hz

Vyskytují se při zavřených očích a relaxaci. Aktivita alfa vln zmizí při soustředěné pozornosti, při stresových situacích nebo mentálních úkonech. Vyskytují se také při některých meditačních stavech, při momentech „bytí tady a teď“. Všeobecně se také mohou vyskytovat při dívání se na televizi, tzv. denním snění a relaxaci.

Mu vlny - 7.5 - 12.5 Hz

Mu vlny mohou být zastoupeny na frekvencích od 7.5-12.5 Hz, ale hlavně se vyskytují v rozmezí 9-11 Hz [38]. Velkou část spektra sdílí s Alfa vlnami, které se však projevují zejména ve vizuálním kortexu, jež je zastoupen v zadní části mozku, kdežto Mu vlny jsou především v sensorimotorickém kortexu. Mu vlny jsou potlačeny při provedení motorické činnosti, jako jsou pohyby rukou nebo nohou. S trénováním lze tyto rytmy potlačit i při pouhé představě daného pohybu. Toto potlačování vln je zvané desynchronizace, neboť jsou EEG vlny způsobovány velkým množstvím neuronů, které se spustí ve stejný moment. ⁸

Beta vlny - > 13 Hz

Beta vlny jsou přítomny pod vlivem některých drog. Jsou také ukazatelem normálního bdělého stavu, řešení kognitivních úkolů, běžné konverzace nebo soustředěné aktivity mozku. Pro mozek je tato frekvence velice vyčerpávající.

Theta vlny 3.5-7.5 Hz

Takzvané spánkové vlny, vyskytující se při REM spánku (Rapid Eye Movement). Jsou dominantní i při hypnóze a meditaci, nebo smyslovém soustředění na vnitřní stavy. Jsou ukazatelem probíhající živé vizualizace, intuice a probíhajících nevědomých informací. V této práci je využívám pro detekci soustředěné pozornosti.

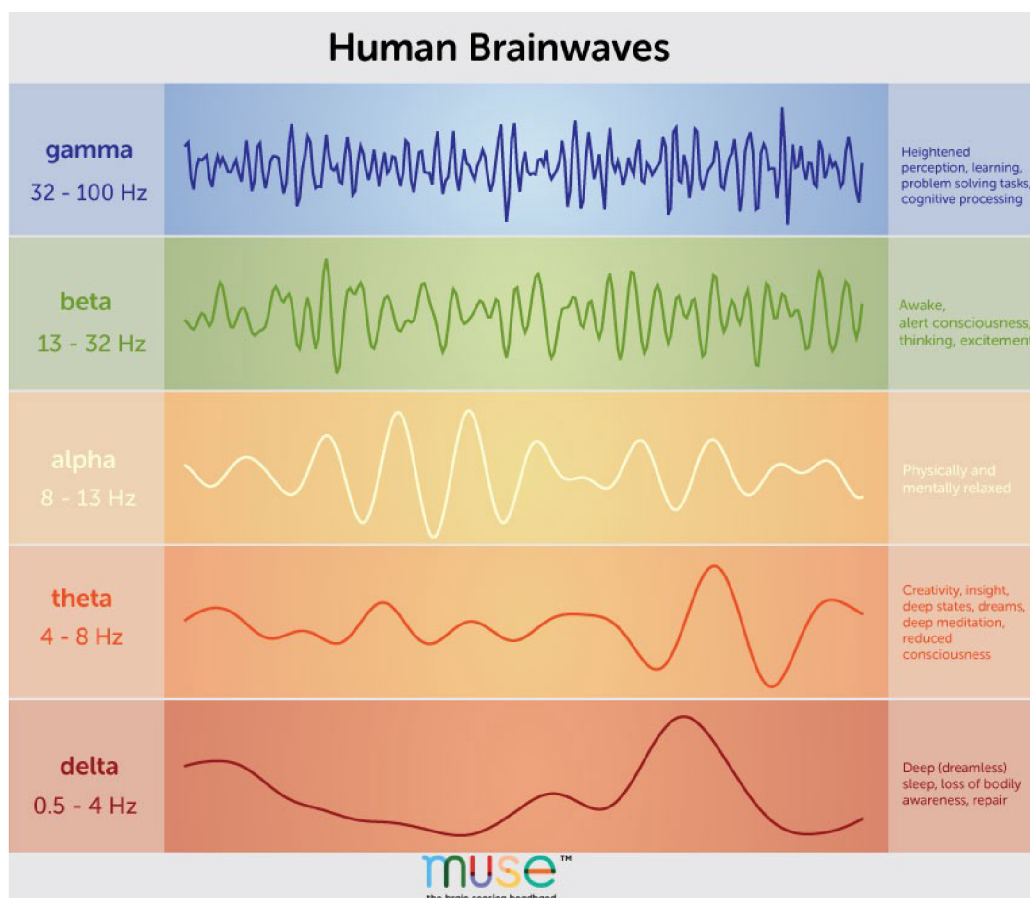
⁸https://en.wikipedia.org/wiki/Mu_wave

Delta vlny < 3 Hz

Přítomny při hlubokém spánku, u dětí a batolat, té nejhlubší meditaci a bezesném spánku. Potlačují externí povědomí a jsou zdrojem empatie. V tomto stavu také dochází k léčení a regeneraci.

Gamma 38 – 42 Hz

Řadí se mezi nejrychlejší vlny. Jsou to také ty nejjemnější vlny – mysl musí být utišena pro přístup k informacím, jež tyto vlny přenášejí. Gamma vlny jsou aktivní hlavně při stavech universální lásky, altruismu a vyšších stavech vědomí. Gamma frekvence je nad aktivací neuronů, takže jak jsou ve skutečnosti generovány, je pro vědce zatím záhadou.



Obrázek 2.6: Mozkové vlny[4]

2.3 Existující výzkumy

Po extrakci pásem z nasbíraných dat ze sensorimotorického kortexu (hlavně Mu a Beta pásem) je dalším krokem získání hlavních rysů (features), které nejvíce souvisí se sledovanou událostí (jak se změní mozkové vlny např. při provedení pohybu pravou rukou a jak při pohybu levou). Správná metoda získání těchto rysů je nejdůležitější částí tohoto výzkumu. V existujících výzkumech uvedených v dalším odstavci jsou použity k extrakci těchto rysů různé taktiky, mezi nimi Rychlá Fourierova transformace, Power Spectral Density (PSD),

statistiky jako standardní odchylka, nebo např. Discrete Wavelet transform (DWT - Diskrétní wavelet transformace [Sekce 5.1.1]). Zvolení těch správných metod je klíčové pro správnou klasifikaci dat a případné zlepšení úspěšnosti klasifikačních metod. Na zpracovaná data jsou pak používány metody redukce dimensionalit jako je PCA nebo ICA. Mým cílem je naleznout tu správnou metodu pro problém rozlišování záměru k pohybu pravé/levé ruky pomocí UCIV hardware a optimalizace nalezených metod v kapitole 6 pro maximalizaci dosažení přesnosti detekce. Tyto algoritmy popisují dále v kapitole 5.3

Pro detekci levé a pravé ruky neexistuje pro UCIV mnoho výzkumů, a ty existující nemají přílišnou úspěšnost [39] [16] (další uvedeny níže). Provedl jsem hledání na serverech s vědeckými pracemi ncbi.com, sciencedirect.com, researchgate.net. Použil jsem vyhledávač PRIMO VUT ⁹, který mi umožnil stahovat i pro veřejnost běžně nedostupné výzkumy a uskutečnil jsem všeobecné hledání přes vyhledávač Google. Níže popisují hlavní body nalezených prací. Mým hlavním záměrem v této práci je využít existující řešení prezentované v nalezených výzkumech, vylepšit je, implementovat vlastní metody pro analýzu a klasifikaci dat a porovnat dosažené výsledky pro UCIV headset a diskutovat možnosti využití pro ovládání dalšího hardware a software.

Ve svojí práci použil Febriliyan Samopa [39] produkt od firmy OpenBCI, a totiž jejich Headwear Spiderclaw V1 s 16 elektrodami. Pro testování použil software OpenVIBE ¹⁰ a s ním dodávané scénáře pro CSP. Se všemi testovacími subjekty se dostal na úspěšnost okolo 60 %.

Práce Benjamin Estermanna [20] je zaměřena na sensorimotorický kortex a pro detekci použil DWT a FB-CSP, Filter Bank Common Spatial Pattern ¹¹ a jako klasifikační metodu LDA, Linear Discriminant Analysis ¹². V této kombinaci dosáhl Benjamin na 60 % úspěšnost při klasifikaci 4 tříd (levá ruka, pravá ruka, nohy, jazyk) při použití UCIV s 16 elektrodami. Měl 3 testovací uživatele. Pro testování subjektů použil software pro prezentaci pokynů. Text prezentuje první 2 sekundy od počátku testovacího segmentu. Výzva (levá, pravá, noha, jazyk) je na obrazovce po 5 sekund, a je následovaná 2 sekundami pauzy. Pro extrakci rysů v prvním pokusu použil DWT, pomocí které analyzoval frekvenční spektrum a zvolil frekvence 12, 16, 20, 24 a 28 Hz. Jako trénovací data použil segment vzorků od 0.7 do 4s po startu výzvy. Jako klasifikátor použil zmíněný LDA. U prvního a třetího testovacího subjektu se dostal na přesnost klasifikace 40 %, u druhého na 32 %. V druhém pokusu použil zmíněný FBCSP a jako klasifikátor opět LDA. Subjekt 1 dosáhl 62 % přesnosti, Subjekt 2 56 % a Subjekt 3 46 %.

Dalším výzkumníkem je Michal Hlinka [24], který používá k detekci také jedno z levnějších BCI zařízení, headset Enobio [Sekce 2.3.1] s 32 EEG elektrodami. Jeho cílem bylo nasbírání praktických dat během jediného trénovacího sezení testovacího subjektu, otestování zda simulace pohybu pomocí videa v trénovací fázi zlepšila úspěšnost při trénování a porovnání původních a vlastních OpenVIBE klasifikátorů. Jeho testovací uživatelé podávali vesměs pozitivní zpětnou vazbu, hraní a celkový průběh experimentů si užívali. Teoretická úspěšnost klasifikace u při hraní hry byla 64 % a reálná úspěšnost při hraní 42 %. Jeho výsledky jsou založeny pouze na jediném testovacím sezení.

Hlavními poznatky z práce Michala Hlinky, užitečnými pro tuto práci, jsou hlavně následující:

⁹<https://primo.lib.vutbr.cz>

¹⁰<http://openvibe.inria.fr/>

¹¹<https://ieeexplore.ieee.org/document/4634130>

¹²https://en.wikipedia.org/wiki/Linear_discriminant_analysis

- Autor očekával určité ruchy a šum z headsetu, ale neočekával, že headset bude tak citlivý na pohyby. Díky tomu se rozhodl, že všechny experimenty pro trénovací sadu bude nutné provádět s otevřenýma očima a bez jakýchkoliv pohybů hlavy nebo těla.
- Autor se pokoušel omezit ruchy, a to přidáním meditačních a pozornostních hodnot jako vstup do neuronové sítě, a to pro každý frekvenční rozsah. To se prokázalo jako horší možnost než využívání pouze 5 vstupních rozsahů (alfa - gamma).
- Ve všech kategoriích klasifikace bylo dosaženo dobrých výsledků, kromě klasifikátoru K-NN, K-nearest neighbors, jež není vhodný pro zpracování více rozměrných dat.
- Mrkání je charakteristické zvýšením delta, theta a alfa frekvence a snížení gamma frekvence - proto není pro klasifikaci mrkání nutné používat složité metody; metody, které používám v této práci navíc mrkání, svalové záchvěvy a podobné ruchy odfiltrují.

V jiné studii [43] autoři použili OpenBCI pro akvizici dat a zaměřili se na mozkové rytmy Mu [Sekce 2.2.3] a ke klasifikaci použili MLP, Multilayer perceptron¹³. V práci experimentovali s metodami neurofeedbacku [Sekce 4.1.1], různými strategiemi před-zpracování dat a úpravami MLP klasifikátoru. Použili binární klasifikaci mezi stavem relaxace a používáním pravé ruky. Zjistili, že úspěšnost se zvyšuje při delší prezentaci testovaného pohybu před samotným nahráváním testovacích dat. Jejich průměrná úspěšnost při detekci této jednostranné činnosti je 77 %.

Další je práce Mohita Agarwala [16]. Používá OpenBCI hardware a OpenVIBE software. Jeho práce nespécifikuje pracovní prostředí. Mohit se dostal na průměrnou úspěšnost 53.4 %, v závislosti na rozmístění elektrod, se kterým experimentoval. Nejlepší pozice jsou podle jeho testů C3, Cz a C4 [Sekce 2.1].

Ke klasifikaci EEG signálů byla použita např. i Metoda podpurných vektorů (SVM) [Sekce 5.3]. Ve svojí práci [32] používá Xiaou Li Metodu podpurných vektorů s více jádry. Používá 3 fáze: předzpracování signálu, zvolení vhodných rysů pomocí DWT a v poslední fázi zkoumá schopnosti upravené SVM ke klasifikaci EEG signálů. Jeho SVM používá více polynomiálních a RBF (Radial basis function - Radiální bázová funkce) jader. Jeho metoda se prokázala být lepší, než klasifikace pomocí SVM s jedním jádrem.

Metoda podpurných vektorů byla použita i v práci Homri Inese „EEG classification using Support Vector Machine“ [25]. Ve svojí práci analyzoval EEG data pohybů levé a pravé ruky a aplikoval různé wavelet transformace na segmenty EEG dat určených pro extrakci rysů. SVM použil pro klasifikaci takto získaných dat a výsledky porovnal s výsledky klasifikátorů MLP a LDA. EEG signály nahrál z kanálů C3 a C4 a analyzoval pouze rytmy Mu na 8-13 Hz a centrální beta rytmy na 13-30 Hz. Popisuje také již zmíněné koncepty ERD a ERS. Wavelet transformace, které použil jsou následující: Daubechies4, Daubechies6, Daubechies8, Daubechies10, Daubechies12, Coiflet2, Coiflet4, Coiflet5, Symmlet5, Symmlet6, Symmlet7, Symmlet8, Symmlet9 a Symmlet10. (tyto wavelety by mohly být díky svým charakteristikám použity i k mému výzkumu).

Jeho EEG signál má vzorkovací frekvenci 128 Hz a tento signál rozkládá do 3 detailních rozložení (D1-D3) [Sekce 5.1.1] a jedné finální aproximace A3. Jelikož se Mu a Beta vlny vyskytují na frekvenčním rozsahu 8-30 Hz, soustředí se právě na rozklady 2. a 3. úrovně [Sekce 5.1.1].

Každý z takto rozložených segmentů EEG signálu je následně popsán vektorem 12 rysů reprezentujících statistické parametry vypočítané z koeficientů rozkladů D2 a D3.

¹³https://en.wikipedia.org/wiki/Multilayer_perceptron

Statistické parametry, které Ines sleduje jsou: průměr absolutních hodnot koeficientů, jejich směrodatná odchylka a výkon. Tyto jsou vypočítány pro každou elektrodu, tedy C3 a C4, dohromady je tedy datový bod tvořen 4 x 3 hodnotami.

Pro klasifikaci pomocí SVM [Sekce 5.3] zvolil nelineární radiální bázovou funkci RBF a parametrem $c=6$ a $\gamma=0.5$. Jako metriku pro měření přesnosti použil základní vzorec (správně klasifikované/všechny klasifikované)*100. Problém v tomto vidím s nezapočítáváním False Positives, které velice ovlivňují pravou výslednou přesnost a použitelnost. V mojí práci pracuji jak s Recall rate (True Positive/False positives), tak s přesností (zmíněný vzorec). Jednotlivé výsledky také porovnávám podle metriky AUC, které popisují v [Sekce 5.3.4].

Tímto modelem proběhla klasifikace dat čtených pohyblivým oknem, načítajícím segmenty vzorků a nasbírané hodnoty byly zprůměrovány na posledních 10 vzorků. Podobnou metodu používám i v jednom z experimentů v kapitole 6, kde provádím trénování na datech čtených pohyblivým oknem po sekundách.

Na nejvyšší úspěšnost klasifikace se Ines dostal u vzorků 466-760, cca 3.5s-6s po prezentování výzvy k představování si pohybu. Nejlepší wavelet transformace jsou v jeho případě DB10, Coiflet5 a Symmlet5 s úspěšnostmi 89.30 %, 89.30 % a 90 %.

Jako další pokus byly použity klasifikátory MLP a LDA, s přesnostmi 89.30 %, 87.86 % pro Coiflet5 a Symmlet10. Oba předčil SVM s 90 % a Symmlet5.

Jeho závěrem je, že úspěšnost klasifikace závisí na soustředění testovacího subjektu během průběhu testování.

Ve výzkumu [21] se výzkumníci v čele se Saugatem Bhattacharyyem zabývali analýzou výkonnosti klasifikačních algoritmů pro klasifikaci EEG signálů představovaného pohybu levé/pravé ruky.

Nejprve výzkumníci extrahovali před-zpracovaný EEG signál a použili ho jako trénovací data pro klasifikátory. K extrakci použili Wavelet transformace DWT, Power Spectral Density PSD a průměrný výkon. Jako klasifikátory použili Linear Discriminant Analysis LDA, Quadratic Discriminant Analysis QDA, K-nejbližších sousedů KNN, lineární SVM, SVM s jádrem RBF a naivní Bayesův klasifikátor.

Použili dva přístupy, kde v prvním klasifikovali všechny rysy a v druhém použili techniku pro redukci dimenzionality PCA [Sekce 5.2]. Došli k tomu, že SVM s RBF jádrem dalo nejlepší výsledky v kombinaci s PCA redukovanou trénovací sadou.

Bhattacharyya použil testovací data z BCI Competition 2003. Data byla nahrána od jediného testovacího subjektu, 25-leté ženy. K nahrávání EEG dat použil zesilovač G.tec a Ag/AgCl elektrody umístěné na pozice C3, Cz a C4. Nahrávání probíhalo v 7 sezeních po 40 výzvách s více pauzami mezi nimi, vše během jediného dne. Segment zde následuje podobnou metodu jako používá [20] Benjamin Estermann, a totiž první 2 sekundy ticho, v 2. sekundě je prezentována vizuální instrukce a je přehrán zvukový signál. Nahrávání probíhá 6 sekund. Pro klasifikaci takto nasbíraných dat nakonec použil Bhattacharyya pouze elektrody C3 a C4, jelikož je elektroda Cz nepotřebná, resp. nepodává signifikantní výsledky při detekci levé a pravé ruky [27]. Adaptivní autoregresivní koeficienty AAR parametry elektrod C3 a C4 spolu s PCA byly použity jako rysy. Použitá vzorkovací frekvence je zde 128 Hz.

Rozklady pomocí DWT provedl Bhattacharyya až na úrovni D3 (7.81-15.62Hz) a jako wavelet funkci použil Daubechies4 (db4) [Sekce 5.1.1]. Pro PSD přístup zvolil Welchovu metodu s Hammingovým oknem délky 64 vzorků. Přibližná PSD byla počítána pro pásma 8-25Hz, na které se vyskytují jak Mu, tak Beta vlny.

Z každé výzvy (trvajících 6 sekund) byl pro obě elektrody C3 a C4 vytvořen vektor rysů o velikosti 102 hodnot (Wavelet D3), z PSD na pásmu 8-25 Hz pak 768 hodnot a průměrný

výkon tvoří 1 reprezentující hodnotu. Ve výsledku má tak jeden datový bod velikost 871 (x 140 výzev). Pomocí PCA byla dimenzionalita zredukována na 91 x 140.

Pro trénovací data, kde neredukoval dimenzionalitu pomocí PCA se dostal s jednotlivými klasifikátory na tyto přesnosti klasifikace: 79.29 % s LDA a QDA, 76.43 % s K-NN, 80.71 % s lineárními SVM, 82.14 % se SVM s RBF jádrem a 80 % s naivním Bayesovým klasifikátorem. S PCA jsou výsledky přibližně o 1-5% lepší.

Ve výzkumu [34] „A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update“ analyzuje Francouzský výzkumník Fabien Lotte použité metody pro klasifikaci EEG signálů ve velmi rozsáhlém výzkumu. Jeho hlavní poznatky jsou:

- Pro nejlepší výkon klasifikace EEG dat by měly být použity adaptivní nežli statické klasifikátory
- Metody pro Deep Learning nejsou dle jeho závěrů efektivní pro klasifikaci EEG signálu z BCI, jelikož jsou trénovací data v těchto výzkumech velmi omezena. Mělké konvoluční neuronové sítě jsou slibnější.
- sLDA (Shrinkage LDA) algoritmus by měl být vždy použit místo klasického LDA, jelikož je efektivnější a robustnější na omezených trénovacích datech
- Pokud máme omezená vstupní data je nejlepší použít sLDA, Riemannovu geometrii (účinná pro klasifikaci v BCI) nebo velmi slibnou Metodu náhodných lesů (Random Forests) [Sekce 5.3.1]

V práci „EEG signal analysis for Seizure detection using Discrete Wavelet Transform and Random Forest“ [22] se výzkumníci zabývají detekcí epileptických signatur v mozku. K tomu používají DWT koeficienty a klasifikátory MLP a Metod náhodných lesů. K jejich výzkumu se ukázala být nejvhodnější Metoda náhodných lesů a to s přesností 98 %. Jako trénovací data použili veřejně přístupnou databázi z Bonnské univerzity. Data obsahují záznam z jedné elektrody se vzorkovací frekvencí 173Hz. Použili DWT rozklady A5 a D3-D5. Jejich výsledky naznačují, spolu s poznatky Lotteho [34], že Metoda náhodných lesů by se dala úspěšně využít i pro můj výzkum [Sekce 5.3.1].

2.3.1 Existující Hardware/Headware

Mimo OpenBCI Ultracortex na trhu zaujímají roli i alternativní EEG headsety.

Neurosky MindWave

MindWave Mobile 2 měří vlny EEG spektra. Má podobnou funkcionalitu jako OpenBCI headset, avšak obsahuje pouze jednu elektrodu, která má fixní pozici na čele (FP1). Proto je tento headset použitelný pouze pro měření neuronové aktivity v prefrontálním kortexu. Používá jednu AAA baterii, se kterou vydrží 8 hodin v provozu ¹⁴. Výzkumník Alf Inge Wang ho použil ve své práci [41] pro ovládání aplikace.

¹⁴<https://store.neurosky.com/pages/mindwave>



Obrázek 2.7: Neurosky MindWave [11]

Muse

Muse je další EEG headset. Má 4 senzory, z toho 2 na čele, 2 za ušima a 3 referenční elektrody. S cenou okolo 269 USD je zaměřen na běžné obyvatelstvo, se svým designovým uživatelským rozhraním, aplikací pro mobily a hlavní proklamovanou použitelností pro kvalitnější meditační činnost. Je zaměřen na běžné uživatele, kteří si např. chtějí dělat statistiky z jejich meditačních sezení. Dbá hlavně na pohodlí nošení, které je velmi důležité pro delší používání.¹⁵



Obrázek 2.8: Muse [3]

¹⁵<https://choosemuse.com/what-it-measures/>

Emotiv EPOC

Emotiv EPOC se pyšní 14 EEG senzory (oblasti AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) , 9-osým akcelerometrem a 12 hodinovou výdrží. Vzorkovací frekvence je 256 vzorků za sekundu. EPOC lze pořídit za 799 USD.¹⁶



Obrázek 2.9: Emotiv EPOC [6]

Enobio

Enobio EEG headset of firmy Neuroelectrics lze zakoupit ve verzích s 8, 20 nebo 32 senzory (verze 32 byla použita v [24]). Vzorkovací frekvence je u tohoto headsetu až 125 Hz, u akcelerometru pak 100 SPS (Samples per second). Pro rozmístění senzorů na ploše hlavy je využíván EEG systém 10-10.¹⁷



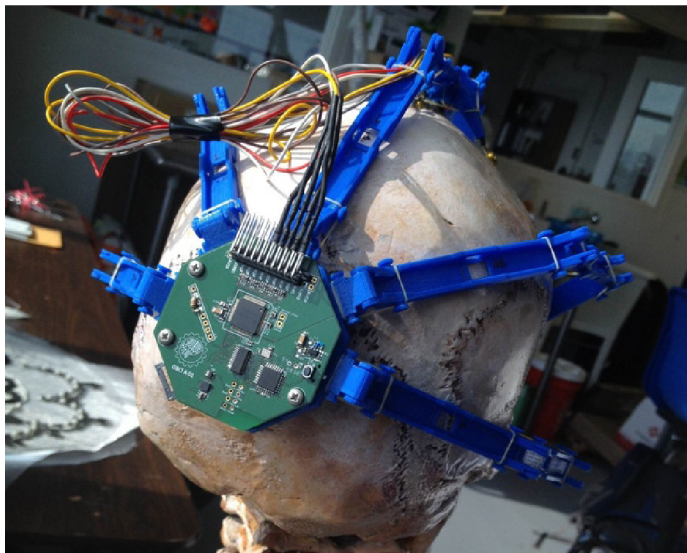
Obrázek 2.10: Neuroelectrics Enobio [7]

¹⁶<https://www.emotiv.com/product/emotiv-e poc-14-channel-mobile- eeg/#tab-description>

¹⁷<https://www.neuroelectrics.com/products/enobio/enobio-8-5g/>

OpenBCI Spiderclaw V1

Toto BCI zařízení bylo použito ve výzkumu [39]. Jedná se o jeden z produktů firmy OpenBCI, který používá stejný čip jako pro tuto práci používaný UCIV. Pozice elektrod jsou zastoupeny podle Systému 10-20 [Sekce 2.1]. Ve zmíněném výzkumu však byl Spiderclaw nevhodný pro tvar hlavy testovaných subjektů, a elektrody proto výzkumníci připevnili manuálně přímo na hlavu subjektu.



Obrázek 2.11: Spiderclaw [15]

Kapitola 3

Sběr dat

OpenBCI je open-source BCI (Brain Computer Interface), neboli rozhraní pro komunikaci mezi mozkem a počítačem. Všeobecně se jedná o víceoborové téma, spadající jak do neurologie, psychologie a zpracování signálu, tak do strojového učení. Můžeme říci, že se jedná o systém umělé inteligence, který má více fází, a totiž: obdržení signálu, zpracování signálu, získání charakteristik ze signálu, klasifikaci dat a výstup (ovládání PC, robota, real-time biofeedback. . .). Pro sběr dat je použit následující tok:

- Presentování instrukcí testovacímu subjektu
- Stream EEG dat z UCIV pomocí LSL
- Uložení do CSV
- Extrakce rysů a statistická analýza
- Případná redukce dimensionality dat
- Klasifikace do tříd

3.1 OpenBCI Ultracortex

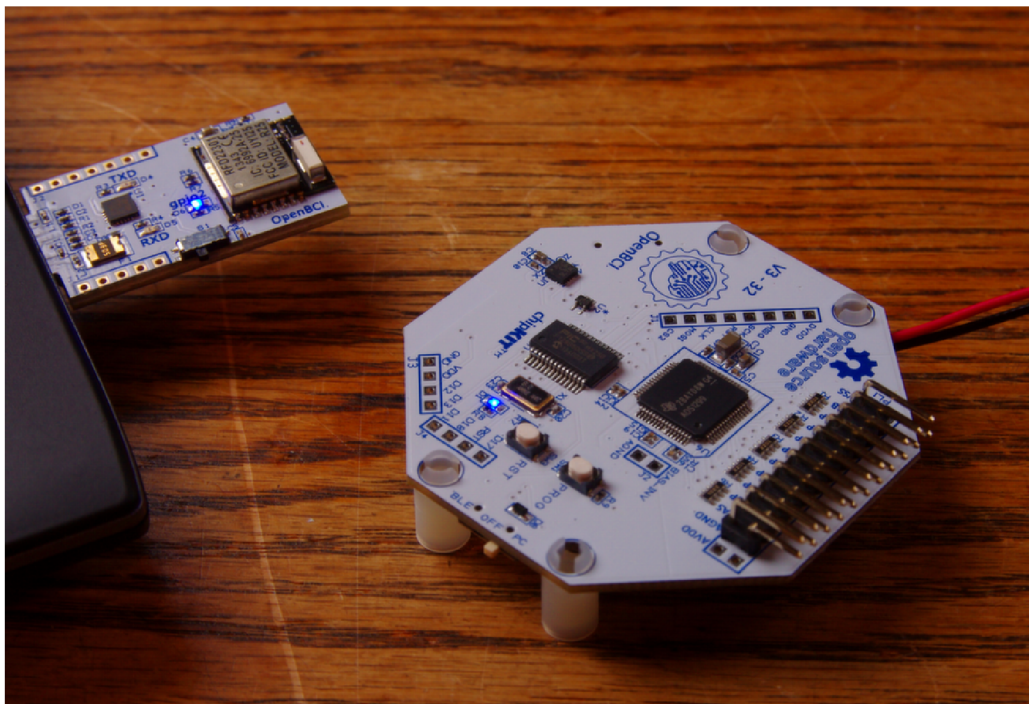
V této diplomové práci je užito OpenBCI Ultracortexu Mark IV. Jedná se o open-source EEG headset použitelný v kombinaci s OpenBCI systémem. Skeleton headsetu je tisknutelný na 3D tiskárně. Je schopen zaznamenávat mozkovou aktivitu kvalitativně na úrovni hardwaru používaného při vědeckém výzkumu. Hlavní součástí 8-kanálového OpenBCI headsetu je Cyton board (deska). Deska používá RFDuino moduly pro bezdrátovou komunikaci pomocí Bluetooth. RFDuina jsou založena na Nordic nRF51822 SoC Radio IC. Deska může také komunikovat s Android nebo iOS zařízením přes BLE, což vyžaduje RFDuino-BLE knihovnu. Díky svojí příznivé ceně byl zvolen pro tento výzkum.



Obrázek 3.1: OpenBCI Ultracortex na testovacím subjektu

Hardwarová specifikace Cyton desky je následující:

- Napájena 3-6V baterií
- Mikrokontroler PIC32MX250F128B s chipKIT UDB32-MX2-DIP bootloaderem
- Analogový front end ADS1299
- 3-směrný akcelerometr LIS3DH
- Bluetooth Low Energy RFDuino
- Slot na MicroSD kartu
- Regulátor napětí (3V3, +2.5V, -2.5V)
- Piny pro elektrody



Obrázek 3.2: Dongle a Cyton deska [5]

Velmi důležitou součástí je přiložený USB dongle. Jeho specifikace je:

- Napájení přes USB konektor
- Bluetooth LE RFDuino
- FTDI USB<>Serial IC (FT231XQ-R)
- Obnovitelná pojistka

Elektrody

OpenBCI má 8 elektrod, které slouží jako vstupní detektory mozkových vln. Na 3D vytisknutém skeletu headsetu jsou umístěny v rozpoložení 10-20 [Sekce 2.1]

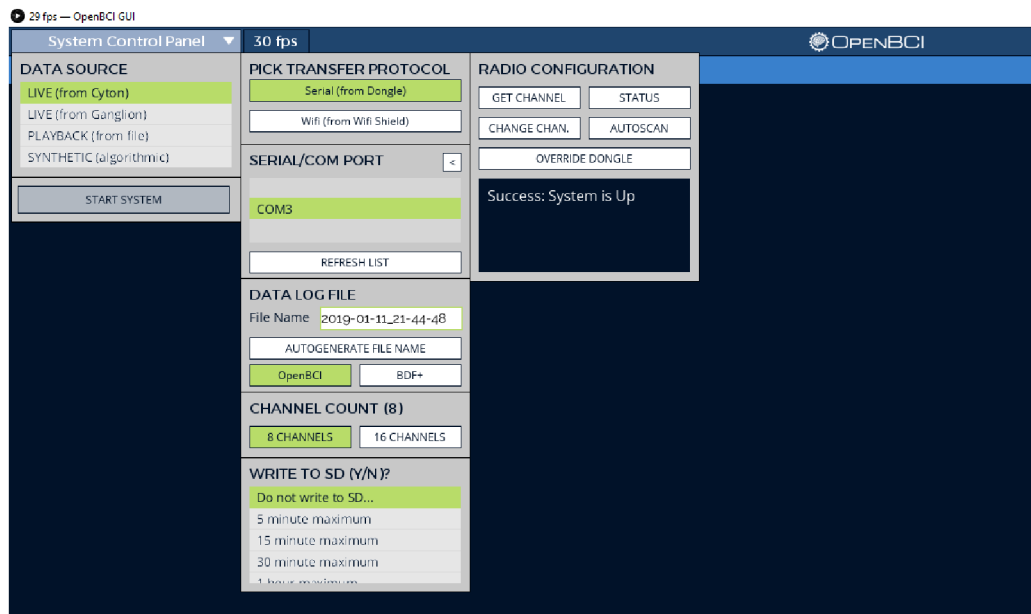
3.2 OpenBCI GUI a Hub

OpenBCI Ultracortex spolupracuje se Softwarem OpenBCI GUI. Ten slouží jako mnohoúčelový nástroj pro vizualizaci a filtrování dat, ať už přijatých v reálném čase nebo načtených ze souboru. Pro spolupráci OS a GUI je nutné použít přiložený software OpenBCI Hub, který má roli prostředníka mezi zmíněnými zařízeními. GUI může být spuštěno jako samostatná aplikace nebo pomocí Processing (na Javě založený programovací jazyk a nástroj)

3.2.1 Instalace a spuštění

Pro spuštění OpenBCI GUI je nutný FTDI USB ovladač. Po jeho instalaci dojde k připojení USB dongle, který se ve Správci zařízení načte jako nový COM sériový port (klasicky COM3). Na EEG headsetu musí být sepnutý spínač do pozice ON (horní poloha). Po

startu OpenBCI Hubu pod root právy a následném spuštění OpenBCI GUI dojde k vyzvání uživatele ke konfiguraci vstupního zařízení. Je třeba nastavit desku Cyton, datový zdroj “Live from Cyton”, správce vstupního portu nastavit na COMx a jakmile je vše správně nastaveno, stačí spustit pomocí “Start System”.



Obrázek 3.3: openBCI GUI spuštění

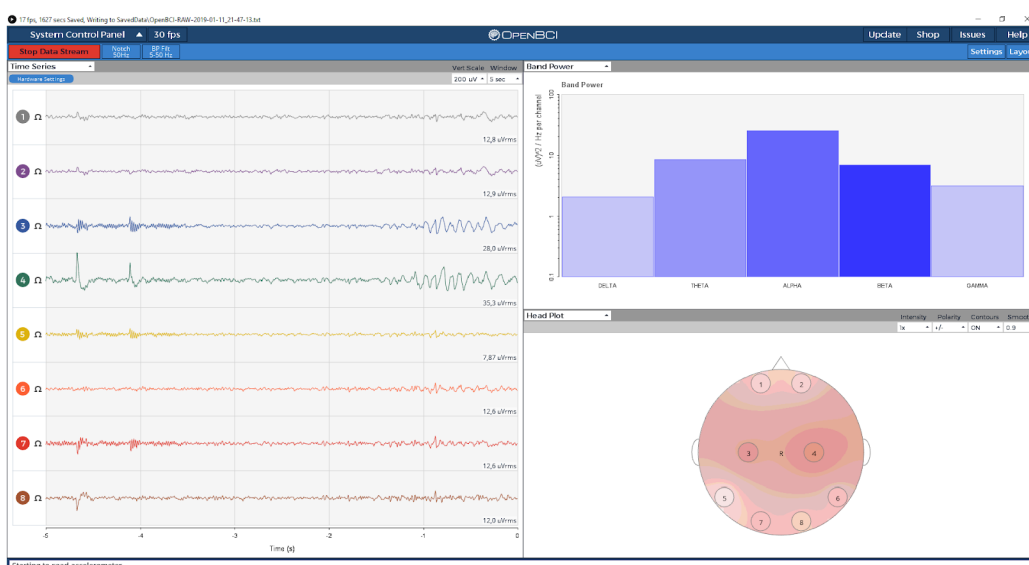
V dalším kroku lze započít se streamování dat z headsetu. V GUI je možnost provádět, mimo jiné, filtrování dat (low pass etc.) a nastavování frekvence (50 Hz). Všechna přijatá data jsou klasicky ukládána do složky SavedData.

3.2.2 Moduly

OpenBCI GUI se skládá z widgetů, které interpretují vstupní EEG data, zobrazují různé statistiky a jinak modulují obdržená data. Na obrázku níže je zobrazen tzv. “focus widget”, dále pak “FFT Plot” a “Time Series”. Time Series zobrazuje elektrické impulsy přicházející z EEG headsetu, FFT plot pak provádí spektrální analýzu a vykresluje frekvenční rozložení, které následně využívá Focus Widget, jehož funkcí je zobrazování poměru mezi alfa a beta vlnami. Pokud jsou ve správném poměru, znamená to, že uživatel je v soustředěné pozornosti. Moduly jsou psány v jazyce Processing, pseudo-Java jazyku. Jako takové je lze upravovat a vytvářet vlastní. V případě jejich použití je však nutno použít OpenBCI GUI z Processing aplikace, samostatná zkompileovaná verze má integrované statické moduly.



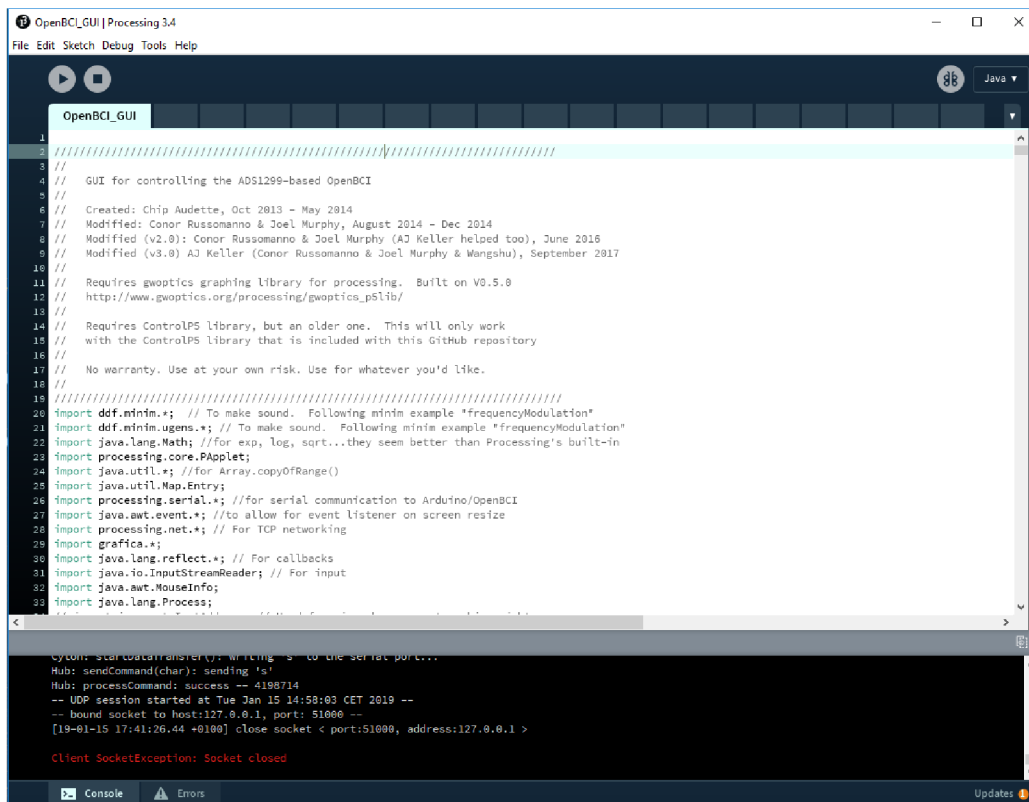
Obrázek 3.4: Focus widget, FFT plot a Time Series widgety



Obrázek 3.5: Moduly pro zobrazení vstupních EEG dat z každé elektrody, rozdělení na alfa - gamma frekvence a mozková aktivita v jednotlivých částech mozku vyobrazena na avatru hlavy viděné shora.

3.2.3 Processing

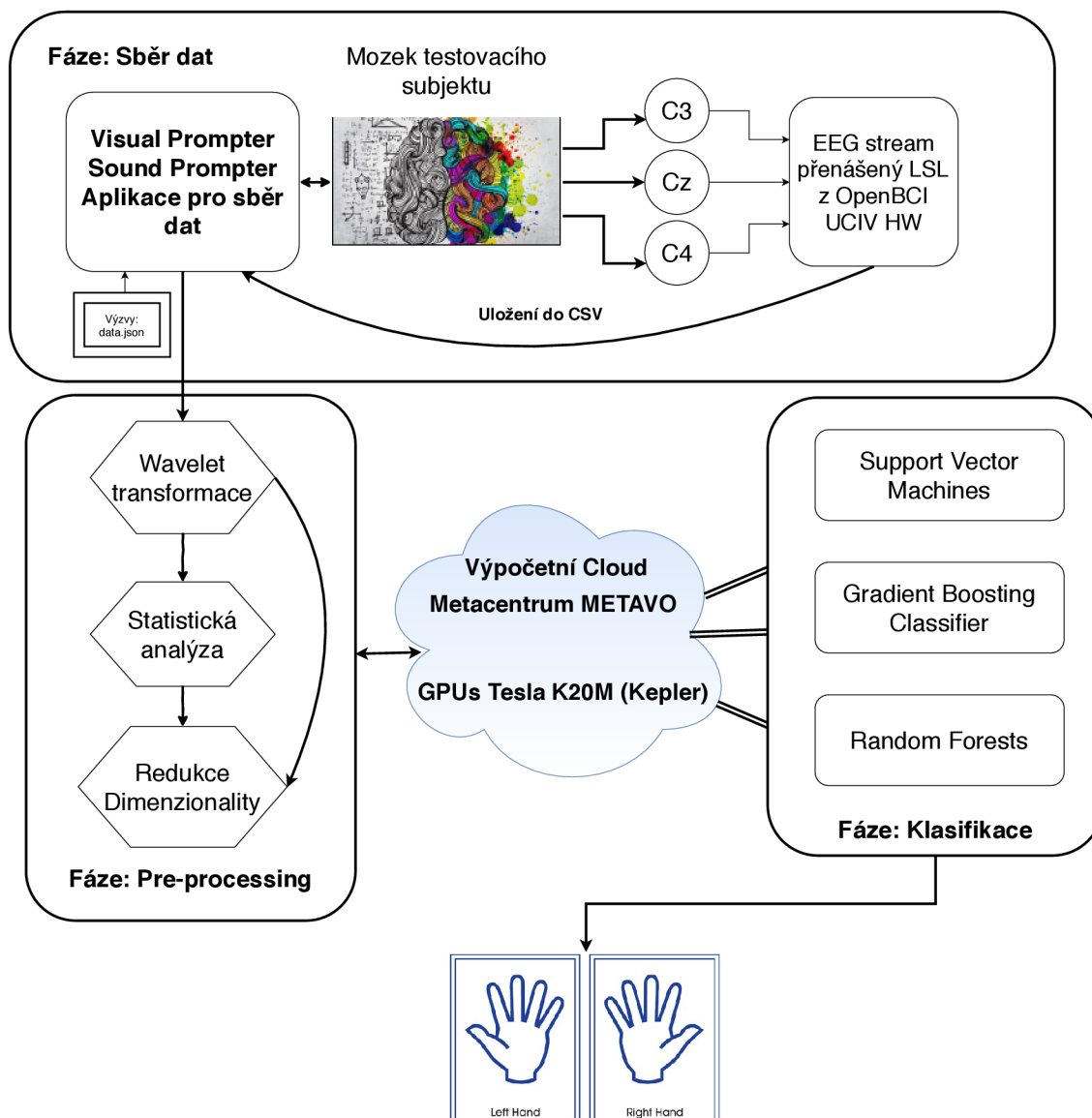
Processing je softwarový “náčrtník” a programovací jazyk. Slouží jako “bootstrap” skeleton pro spuštění dalších programů a používání knihoven. V této práci použijí Processing pro modifikaci existujících modulů pro neurofeedback hru.



Obrázek 3.6: Processing aplikace

3.3 Visual Prompter, generátor datové sady

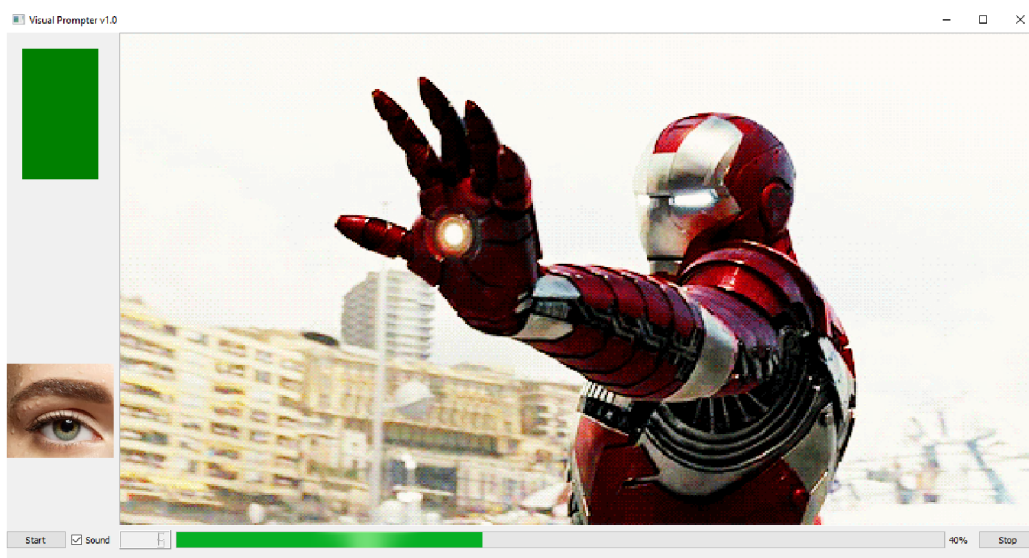
Pro získání prvotního datasetu pro levou a pravou ruku bylo nutné vytvořit nástroj, který jsem potom použil pro vytvoření několika datových sad (kolegové z týmu nástroj také použili a založili na něm další výzkum). Nástroj slouží jako základní stavební kámen celé diplomové práce, jelikož data z něj jsou dále analyzována a klasifikována.



Obrázek 3.7: Flowchart postupu

Vytvořil jsem proto základní multifunkční aplikaci pro sběr dat, kterou jsem pojmenoval Visual Prompter. Prvotní alpha verze, kterou jsme v týmu používali, se skládá z jednoduchého “prompteru” (předkládá jednotlivé “challenge” (výzvy)), modulu pro záznam z kamery a hlavně modulu pro čtení EEG dat. Níže popíšu detaily implementace a úskalí, na která jsem při vývoji narazil. Program jsem psal v Pythonu 3. Aplikace je k dispozici na mém GitHubu ¹

¹https://github.com/janjilecek/DP_files



Obrázek 3.8: Visual Prompter s akční scénou

Z existujících prací, které jsem zmínil výše, jsem již věděl, že nebude nejvhodnější používat pouze grafické šipky jako předloženou výzvu. Uživatel si musí celou činnost představit sám, což může být obtížně pro nesoustředěnou mysl a velice náchylné k chybám. Proto jsem použil kombinaci filmových GIFů, které zobrazovaly pohyb končetin. Někjaké obrázky jsem našel i na youtube, resp. vytvořil jsem GIF pro přehrávání jako video v Qt frameworku.

Aplikace se tedy skládá ze čtyř částí:

- modul pro předkládání výzev
- modul kamery
- modul pro ukládání dat
- modul pro načítání EEG dat

Obsahuje i další funkcionalitu, jako zobrazování probíhajícího času, možnost přehrávat zvuk (tibetský zvon pro začátek nové výzvy), barevnou indikaci začátku nové výzvy a také indikátor toho, zda má uživatel při výzvě mít otevřené nebo zavřené oči.

Na obrázku je první verze, probíhající výzva pro představování si pohybu levou rukou. Pro tuto činnost se nejvíce hodí bojové filmy, kde je více než hojné zastoupení možných úderů jak levou, tak pravou rukou. Scény, které měly jasně zřetelný a replikovatelný pohyb (resp. představení si pohybu) paže, jsem izoloval z vícero filmů a youtube videí do GIF obrázků.

Do datasetu jsem vložil i meditační výzvu, kdy uživatel musí 60 sekund meditovat. Výzva je započata a ukončena zazněním tibetského zvonu, v beta verzi jsem však změnil zvukový signál za kratší, jelikož by dlouho znějící tóny mohly narušovat data v navazující výzvě.

3.4 Formát dat

Pro aplikaci jsem vytvořil vlastní formát dat. Pro načítání výzev jsem zvolil vstupní formát json, ve kterém jsou specifikované detaily a proměnné ovládající jednotlivé fáze průběhu výzev. Soubor je implicitně pojmenován data.json.

3.4.1 JSON vstupní data

Klíče json pole jsou v první alfa verzi pouze tři, a to pro pravou ruku, levou ruku a pro meditaci. Jak můžeme vidět na obrázku výše, hodnotami je asociativní pole s následujícími proměnnými.

```
"right_hand": [
  {
    "index": 4,
    "name": "do_it",
    "stages_duration": [4,5,0],
    "text": "Zvedni pravou ruku, otevrene oci",
    "stages_data": ["audio/4.mp3", "challenges/pause.png"],
    "eyes": 1
  },
  {
    "index": 5,
    "name": "think_open",
    "stages_duration": [6,5,0],
    "text": "Mysli na zvedani prave ruky \n oci otevrene",
    "stages_data": ["audio/5.mp3", "challenges/pause.png"],
    "eyes": 1
  },
  {
    "index": 6,
```

Obrázek 3.9: data.json

- “right_hand” třída výzvy
- “index” je unikátní celočíselný index
- “name” drží individuální jméno výzvy (obojí zmíněné je později použito ve jménech výstupních souborů)
- “stages_duration” specifikuje časování první, druhé a třetí fáze v sekundách
- “text” je textová instrukce zobrazená v první fázi
- “stages_data” drží jména obrázků prezentovaných ve druhé a třetí fázi, v tomto pořadí (beta verze pak obsahuje i název zvukových souborů)
- “eyes” specifikuje, zda bude výzva nastavena jako výzva pro zavřené nebo otevřené oči (boolean ano, ne)

3.4.2 Fáze

Každá prezentovaná výzva se skládá ze tří úrovní (fází). Na levé straně GUI [Obrázek 3.8] je zobrazen obrázek oka, reprezentující typ výzvy - buď pro otevřené, nebo zavřené oči. Po dokončení úrovně je přehráván zvuk oznamující postup do další úrovně nebo konec výzvy. Zvukový výstup lze ovládat zaškrtnutím políčkem - zvuk je však nutný vždy při průběhu meditačního úkolu, jelikož ten musí proběhnout se zavřenými očima. Uživatel by bez zvukové signalizace nebyl schopen určit, kdy má s meditací přestat, a docházelo by k nepřesnostem. Jednotlivé výzvy jsou prezentovány náhodně, a v programu jsem implementoval mechanismus, který zajišťuje neopakující se výzvy (resp. nesmí se opakovat za posledních 5 pokusů. Pokud se opakují, hledá se další výzva, která vyhovuje parametrům).

První úroveň

Tato úroveň uživateli prezentuje text s popisem následujícího úkolu, který bude muset uživatel vykonat. Nejvhodnější jsou krátké texty jako “Představuj si zvedání pravé ruky”. Doba trvání fáze lze specifikovat v poli “stages_duration” v sekundách (první fáze -> první index).

Druhá úroveň

V této úrovni je testovacímu subjektu přehráván pohyblivý obrázek GIF, který je načten na základě proměnných ve vstupním json souboru. Uživatel si prohlédne přehrávaný obrázek a může se začít mentálně připravovat na fázi další. Poté co se doposud červeně svítící světlo (plnicí roli semaforu) přepne na zelenou barvu, třetí úroveň začíná.

Třetí úroveň

Třetí a poslední úroveň. V této fázi by si uživatel měl představovat “scénář”, neboli situaci, kterou si prohlédl v předchozím kroku. Měl by si tedy představovat pohyb paže/meditovat na základě předchozího kroku. Tato úroveň má, stejně jako úrovně předchozí, nastaveno časování ve vstupním souboru.

Tato konečná fáze je specifická tím, že při ní probíhá ukládání dat.

3.4.3 L-streamer data

Po připojení USB dongle je třeba ještě dostat data do Visual Prompteru. K tomu používám knihovnu OpenBCI_Python², která umožňuje uživateli používat různé moduly pro komunikaci s donglem. Její součástí je i modul LSL, tzv. Lab Streaming Layer. Lab Streaming Layer je unifikovaná kolekce knihoven, které se starají o síťovou komunikaci, časovou synchronizaci, přístup v (téměř) reálném čase a také centralizovaný sběr, zpracování a zápis dat.

²https://github.com/OpenBCI/OpenBCI_Python

```

Select Windows PowerShell
Administrator: Windows PowerShell

Closing Serial...
deactivating Plugins...
user.py exiting...
PS E:\OpenBCI\utils\OpenBCI_Python> python .\user.py --port COM3 --add
streamer_lsl

-----user.py-----
board type: OpenBCI Cyton (v3 API)
port: COM3

-----SETTINGS-----
notch filtering: True
user.py: Logging Disabled.

-----INSTANTIATING BOARD-----
connecting to V3 at port COM3
serial established...
openBCI v3 8-16 channel
on Board ADS1299 Device ID: 0x3E
I2SDH Device ID: 0x33
firmware: v3.1.1
335
no daists:
8 EEG channels and 3 AUX channels at 250.0 Hz.

-----PLUGINS-----
found plugins:
streamer_osc ]
sample_rate ]
print ]
streamer_tcp ]
noise_test ]
streamer_lsl ]
udp_server ]
csv_collect ]

activating [ streamer_lsl ] plugin...
creating LSL stream for EEG. Name:OpenBCI-EEG- ID:openbci_eeg_id1-
data type: float32, 8channels, at 250.0Hz
6818.06299061 [37113.39453125, 35338.51171875, -24252.873846875, -30257.64464375, 58448.5589
9375, -11348.94148625, 5808.0532268125, -6451.25]
6818.07033035 [37011.6315625, 35111.39546875, -24339.439453125, -30261.9375, 57957.635, -11355
.2980990625, 5820.1572265625, -6434.44148625]
6818.07431752 [37151.01171875, 35358.1528125, -24256.220783125, -30272.463359375, 58389.86328
125, -11357.686640625, 5837.188564375, -6358.12744148625]
6818.079230767 [37315.8359375, 35705.1484375, -24263.19921875, -30293.408203125, 59109.03125,
-11392.1823398625, 5813.965828125, -6387.9497878125]
6818.083297216 [37275.8046875, 35669.5715625, -24270.462890625, -30276.66796875, 59113.725662
5, -11378.756238125, 4998.0288859375, -6432.630859375]
6818.086245287 [37121.8671875, 35348.07421875, -24241.94148625, -30238.958884375, 58427.972656
25, -11342.3231553125, 5825.055783125, -6448.431640625]
6818.090247109 [37835.9453125, 35128.203125, -24221.289625, -30232.544921875, 57968.37898625,
-11337.488283125, 5852.8798828125, -6489.51984286875]
6818.094253767 [37366.2475625, 35307.07421875, -24227.30878125, -30256.505859375, 59414.98828
125, -11368.785625, 5861.84326171875, -6363.63889375]
6818.098233558 [37329.283125, 35718.67578125, -24244.4453125, -30268.6484375, 59138.46875, -11
372.5673828125, 5945.526271875, -6367.48986328125]
6818.102238828 [37275.8846875, 35688.6815625, -24259.197285625, -30259.9921875, 59112.02734375
, -11362.3974689375, 5822.21337898625, -6414.97314453125]
6818.107227176 [37215.4229375, 35328.8398625, -24248.04765625, -30238.353515625, 59413.0859375
, -11335.3623846875, 5845.34765625, -6421.4775398625]
6818.111288532 [37835.28515625, 35116.3528125, -24233.345703125, -30225.236328125, 57947.5664
9625, -11330.658203125, 5971.745121875, -6392.084989375]
6818.114388841 [37171.0625, 35368.81640625, -24226.294921875, -30245.798828125, 58388.80878125
, -11354.8283125, 5871.18685515625, -6368.9158398625]
6818.118980259 [37233.48875, 35732.121875, -24249.87953125, -30265.423828125, 59111.53515625
, -11373.215828125, 5849.28125, -6364.81123846875]
6818.122290762 [37282.0625, 35666.8878125, -24265.92578125, -30256.70783125, 59897.98818125, -
11363.22469375, 5820.8028871875, -6412.4265625]
6818.1264615 [37118.37898625, 35327.73828125, -24251.338878125, -30232.43359375, 58404.0585937
5, -11342.99609375, 5814.345783125, -6449.88154286875]
6818.130488387 [37828.2189375, 35314.72989375, -24244.833884375, -30231.13671875, 57938.336937
5, -11342.83884375, 5831.9148625, -6422.6386484375]
6818.135216029 [37175.5884375, 35373.22265625, -24228.552734375, -30247.916815625, 58396.6132
8125, -11358.33884375, 5852.1883898625, -6372.38871889375]
6818.139771684 [37339.2578125, 35732.66815625, -24248.15625, -30265.3125, 59122.734375, -11378
.982421875, 5835.46826171875, -6388.79726328125]
6818.142647987 [37228.80315625, 35888.19148625, -24267.6815625, -30265.821484375, 59131.117187
5, -11379.138671875, 5882.25341796875, -6443.484296875]
6818.146244029 [37106.421875, 35328.1952125, -24262.99846875, -30250.73828125, 59434.3671875,
-11366.72528125, 609.382734375, -6477.883189375]
6818.150468029 [37848.1484375, 35128.11328125, -24228.8398625, -30224.488846875, 57977.25, -11
339.41921875, 5841.369148625, -6423.95849889375]

```

Obrázek 3.10: L-streamer data

Po naklonování repozitáře knihovny můžeme spustit samotný LSL server. To provedeme přes hlavní skript `user.py` python .

`user.py --port COM3 --add streamer_lsl` a následně příkazem `/start` započneme přenos dat z donglu (resp. získávaných z aktivního Cyton boardu), načtež skript začne přeposílat vzorky z COM3 portu (nebo jakéhokoliv jiného portu, na kterém komunikuje náš USB dongle). Vlákno v mém Visual Prompteru následně zachytává tyto vzorky a dále je zpracovává, to vše pomocí knihovny `pylsl`, nainstalované přes nástroj `pip`.

3.4.4 Formát EEG streamu

Zasílaný EEG stream obsahuje nejen EEG data z jednotlivých elektrod, ale také data z akcelerometru. V knihovně `pylsl` se k datům přistupuje přes funkci přidělení streamu - `resolve_stream`. Ve funkci můžeme specifikovat buď EEG nebo AUX, kde EEG značí EEG stream a AUX značí stream dat z akcelerometru.

JSON formát

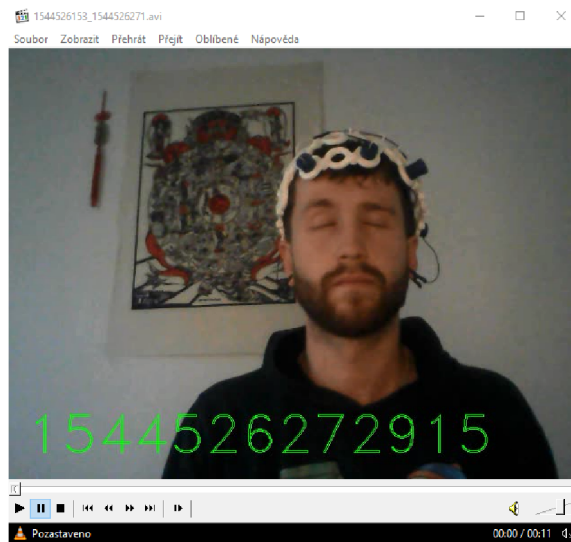
EEG data ukládám do json souboru ve vlastním formátu. Jedná se o asociativní JSON pole mající časový údaj v milisekundách jako klíče a hodnoty pak obsahují EEG data (z 8 elektrod) a data z akcelerometru (3 osy). EEG data jsou uložena pod subklíčem "e" a AUX data jsou uložena pod subklíčem "a". Tato data jsou zapisována po dokončení každé jednotlivé výzvy do json souboru, který je v názvu identifikován indexem výzvy, hlavní třídou proběhlé výzvy (např. `left_hand`), ukazatelem otevřených očí (0 nebo 1), hlavním názvem výzvy (např. `think_open`) a časovým razítkem v milisekundách.

3.4.5 Nahrávání webkamery

Další vlákno aplikace je určeno pro nahrávání dat z hlavní webkamery počítače. Vytvořil jsem třídu `WebcamRecorder`, která pomocí knihovny `OpenCV` zajišťuje nahrávání videa. Video je při provádění výzev důležité, jelikož uživatel programu může při probíhající výzvě mrkat, nebo jinak pohybovat hlavou, což by mohlo narušit ukládaná data. Nahrávání obličeje uživatele je tedy prováděno hlavně z důvodu případné kontroly chyb v datech.

Nahrávané video je ukládáno do souboru ve formátu avi, za použití MJPG kodeku a snímkové frekvence 30 snímků za sekundu. Výška a šířka snímku je pak nastavena na nejvyšší možnou, co použitá webkamera umožňuje. Zvuková data nahrávána nejsou.

Do každého snímku je navíc “vypálen” časový údaj v milisekundách, z důvodu snadnější synchronizace EEG dat a videa při manuální kontrole objevivších se chyb, nebo kontroly mrkání testovacího subjektu. K zapisování času je opět použita knihovna OpenCV.



Obrázek 3.11: Jeden z testovacích subjektů: Demonstrace vodoznaku časového údaje

3.4.6 QT GUI

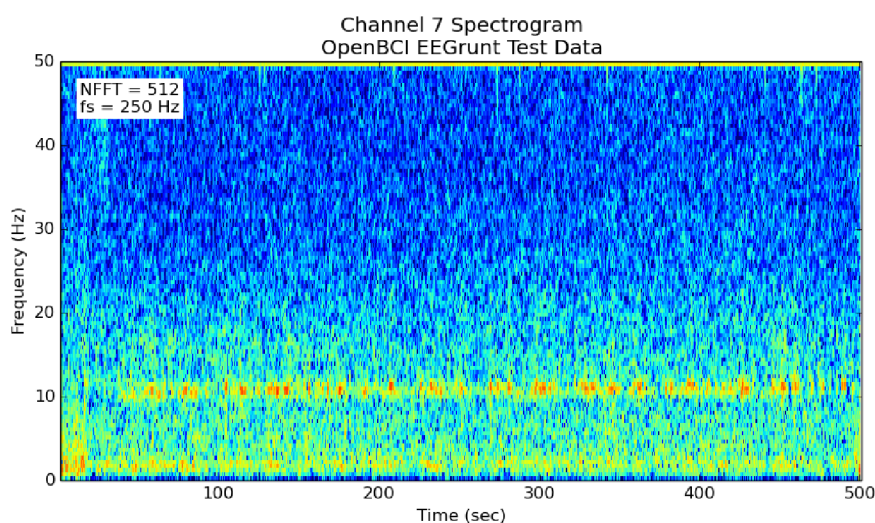
Aplikace je napsána v Qt5, konkrétně používám Python 3 knihovnu PyQt5. Ke komunikaci mezi součástmi grafické aplikace jsou použity klasické QT konstrukce, tedy signály a sloty. Aplikace se skládá z modulů Camera, Challenge, EEG, GUI, ProgressBarThread a Prompter. GUI součást drží všechny ostatní pohromadě, její ovládací prvky jsou přes signály navázány na hierarchicky podnáležící vlákna, které si navzájem předávají kontrolu pomocí již zmíněných signálů a slotů. Ze zmíněných součástí nebyl doposud popsán pouze ProgressBarThread, ten slouží pro ovládání časování celé aplikace, a zobrazování časového postupu jednotlivých výzev pomocí grafických prvků (tedy číslo udávající sekundy a lišta zobrazující procentuální časový postup výzvy).

3.5 Sound Prompter

Další vytvořená verze aplikace se nazývá Sound Prompter (vytvořeno jako další větev VisualPrompter). Po konzultaci s vedoucími jsem Visual Prompter upravil tak, aby ho bylo možné používat i pouze se zavřenýma očima. Aplikace tak místo obrázků prezentuje pouze text, který je předčítán pomocí syntezátoru hlasu, který jsem použil k vygenerování krátkých promluv. Tyto promluvy jsou přehrávány uživateli v druhé fázi. Třetí fáze je pak označena krátkým pípnutím nízkého tónu (buben) a pro naznačení ukončení jejího provádění je přehrán tón vysoký (struna Sitaru).

3.6 Spektrogram alfa vln

Úkolem Visual Prompteru byl sběr dat pro další trénink a získávání signatur signálu charakteristického pro představování si pohybu levé nebo pravé končetiny. Další nezbytnou součástí a nutnou prerekvizitou pro další postup je klasifikace EEG signálu pro ovládání hry. Jeden z kroků, který si v této práci probereme, je rozložení vstupního signálu na frekvenční pásma (alfa, beta, theta etc.), pomocí metody vlnkové/wavelet transformace [Sekce 5.1.1] a výstup těchto funkcí použijí pro základní neurofeedback hru [Sekce 4.1]. Pokud je alfa a beta frekvence ve správném poměru, prefrontální kortex je v procesu soustředění, a na základě této znalosti se dá vyprodukovat tímto výstupem ovládaná hra (např. pokud se uživatel soustředí, ve hře se začne zvedat předmět do vzduchu - je pouze na uživateli, jak dlouho ho tam vlastním soustředěním udrží.)



Obrázek 3.12: Spektrogram s viditelnou dominancí alfa frekvence

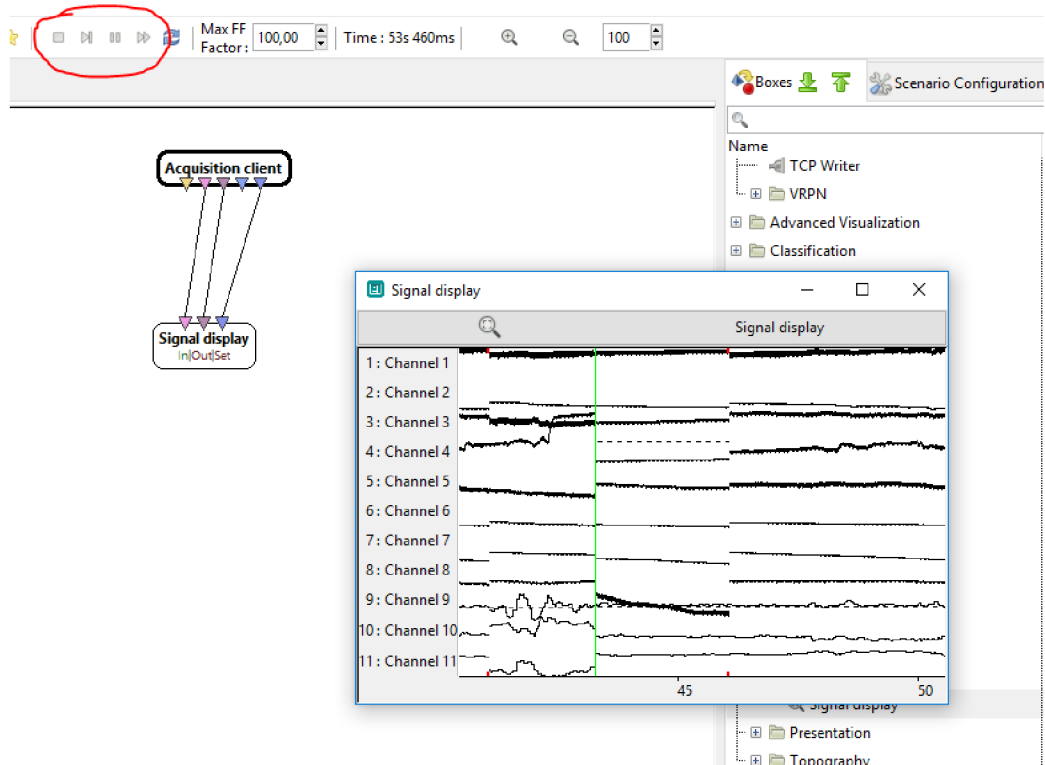
3.7 OpenVIBE

OpenVIBE je softwarová platforma pro návrh, testování a používání BCI. Umožňuje zpracování neurálních dat v reálném čase. Tato data pak dokáže dále filtrovat, zpracovávat, klasifikovat a vizualizovat. Od verze 2.2.0 také obsahuje funkcionalitu pro dávkovou analýzu rozsáhlých datových sad. Používá se hlavně v medicínských oblastech (asistence postiženým, real-time biofeedback, neurofeedback, real-time diagnóza), v multimediích (VR, videohry), robotice a řadě dalších oborů souvisejících s BCI.

Spojení s OpenVIBE

OpenBCI headset je možné spojit s OpenVIBE. Po instalaci je třeba spustit akviziční server (`acquisition_server.cmd`), který najde připojený USB dongle. Po připojení k dongle lze spustit Designer aplikaci, která zajišťuje rozhraní pro drag'n'drop programování.

Jednoduchý program: "Acquisition client" slouží jako bod pro vstupní data, které pak zašle do modulu "Signal display". Po spuštění programu jsou vykreslována data z jednotlivých elektrod headsetu.

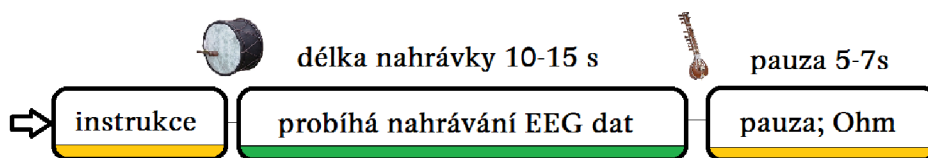


Obrázek 3.13: OpenVIBE

Pomocí OpenVIBE je možné vytvářet i velmi komplexní aplikace. Součástí toolkitu jsou i vzorové programy, ty však fungují s OpenVIBE jen ve verzi 1.0.0. Jedním z ukázkových příkladů je neurofeedback hra, kdy ji může testovací subjekt pomocí alfa/beta vln ovládat. Jedním z cílů této práce je vyvinout vylepšenou verzi na bázi zmíněné hry. Pro základní testování konceptů a navržených řešení pro budoucí vývoj v mojí Python aplikaci lze použít právě OpenVIBE [10].

3.8 Metodologie sbírání dat

Pro Sound Prompter byly navrženy 2 třídy, a to zvednutí levé ruky a zvednutí pravé ruky. Pro každou třídu existují 2 různé výzvy - myslí na činnost se zavřenýma očima a myslí na činnost s otevřenýma očima (provedení pohybu ve skutečnosti není žádoucí z důvodu vysoké citlivosti headsetu na pohyb). Mezera mezi jednotlivými výzvami je od 5 do 7 sekund, impuls k začátku soustředění na ruce je u Sound Prompteru dán zvukem bubnu a samotné trvání výzvy je od 10 do 15 sekund. Během výzvy je přehráván neutrální hlas a v GUI je zobrazena šipka napovídající směr. V případě Visual Prompteru je vše moderováno zelenou nebo červenou barvou “semaforu” pro začátek a konec činnosti / představování si, zvukové impulsy začátku a konce jsou ve Visual Prompteru volitelné. Pro ukončení soustředění je přehráván vyšší zvuk struny.



Obrázek 3.14: Graf průběhu výzvy

Pro nahrání co nejčistších EEG dat je nutné, aby testovací subjekt seděl v naprostém klidu a pohodlí a pokud možno při probíhající „výzvě“ nemrkal a nedělal různé artefakty-produkující činnosti, jako je svráštování obočí nebo mluvení. Každému subjektu jsem proto představil základní instrukce pro správnou formu testování.

Nejprve jsem však testoval EEG headset na sobě. Zjistil jsem, že headset OpenBCI Ultracortex IV s 8 diodami je vcelku nepohodlný na nošení, a sám jsem nevydržel celý segment déle jak 40 minut. Testování si začali stěžovat již po 5-10 minutách na velké nepohodlí a proto je nejdelší záznam, který jsem získal od jednoho z testovacích subjektů (Martin), 30 minut dlouhý.

Po několika desítkách pokusů s neurofeedback hrou jsem vyzkoušel koupit EEG gel pro zlepšení vodivosti pokožky hlavy a tak i přesnosti záznamu. Koupil jsem dvě 600ml láhve indiferentního gelu Topvet pro univerzální použití, ultrazvuk, sonograf a kavitaci. Gel je bez zápachu a barvy, a je určen jen pro vnější použití. Při vlastním testování na sobě jsem používal následující metodiku:

Oči zavřeny, po celou dobu. Pokud jsem z nějakého důvodu potřeboval otevřít oči nebo podobně (pohnout se, poškrabat se), činil jsem tak pouze během pauzy během výzev (které jsem nastavil na 5-7 sekund, náhodně zvolených).

Jako posed jsem zaujal meditační pozici s podsedákem a měkkou podložkou pod nohy. Ruce jsem měl položené nezkřížené na nohách, na kterých jsem měl další polštář pro zamezení „cross-talku“ mezi rukama a nohama (nohy jsou zastoupeny hlavně v části Cz, kdežto ruce v částech C3 a C4 –teoreticky jsem tak zvýšil přesnost snímání, končetiny spolu nekomunikovaly). Později v experimentu jsem přemístil ruce na kolena.

Na hlavě nasazen OpenBCI Ultracortex. Na všechny elektrody jsem nanesl indiferentní gel, přibližně půlku lžičky (elektroda dokonale pokryta gelem).

Datasety v této práci jsou pojmenovány buď jménem testovacího subjektu, nebo pomocí fonetické abecedy (například sada Echo).

Testovacím subjektům jsem dal podobnou instrukci, hlavně aby maximalizovali pohodlí při sezení, aby zvládli testování až do konce. Testování probíhalo v pokoji bez vnějších zvukových ruchů, a na monitoru byla maximalizovaná verze Sound/Visual Prompteru.

Jako instrukci jsem si nechal předkládat pouze hlas vyslovující „Left“ nebo „Right“. U testovacích subjektů jsem nechal i syntezátorem vygenerovanou větu s detailnější instrukcí.

Také bylo několik připomínek od testovaných. Například, že ve Visual Prompteru je text „Představuj si“ moc rozrušující; „při zavřených očích a jen zvuku by bylo představování si pohybu mnohem snazší“; a moc dlouhé mezery a repetitivní instrukce. První byl již vyřešen Sound Prompterem, prezentující pouze zvuk a umožňující mít zavřené oči a zbylé připomínky byly opraveny.

Tabulka 3.1: Datové sady

Jméno	Levá	Pravá	L trénovací	P trénovací	L testovací	R testovací	Typ	Dominantní ruka	Oči	EEG gel	Věk
Jan	4	12	2	8	2	4	Sound	Levá	Zavřené	Ne	24
Martin	40	39	28	28	12	11	Sound	Pravá	Zavřené	Ano	26
Dominika	5	9	3	6	2	3	Visual	Pravá	Otevřené	Ano	22
DominikaB	10	10	7	7	3	3	Visual	Pravá	Otevřené	Ano	22
Echo	36	51	25	36	11	15	Sound	Pravá	Zavřené	Ano	25
Victor	24	30	17	21	7	9	Sound	Pravá	Zavřené	Ano	25
Charlie	27	30	19	21	8	9	Sound	Pravá	Zavřené	Ano	25
Bravo	7	13	5	9	2	4	Visual	Pravá	Otevřené	Ano	25
Lima	23	21	16	15	7	6	Visual	Pravá	Otevřené	Ano	25
Foxtrot	17	36	12	25	5	11	Visual	Pravá	Otevřené	Ano	25
Zulu	3	3	2	2	1	1	Visual	Pravá	Zavřené	Ne	25
Oscar	9	7	6	5	3	2	Visual	Pravá	Zavřené	Ne	25

Pokud testovací subjekt udělal chybu (maximální počet chyb za jedno sezení byl 3, a to u Visual Prompteru), tak mi po skončení popsal, u jakých scén prohodil ruce.

Samotný headset měl výrobci doporučené rozložení (viz oficiální dokumentace a sekce OpenBCI v této práci), ve kterém však nebyla zastoupena elektroda Cz, jež je na samém vrcholu sensorimotorického kortexu, který je hlavním objektem tohoto výzkumu. Proto jsem jednu elektrodu O1 (normálně na temeni hlavy, detekující aktivitu vizuálního kortexu, který pro tuto práci není klíčový) použil právě pro pozici C4 (v metodách pro nahrávání dat jsem nastavil odpovídající indexy pro správné ukládání).

3.9 Použité datové sady

Celkově používám 24 datových sad (12 subjektů/sezení, u každého nahrána trénovací a testovací sada). V následující tabulce je vyobrazen počet výzev nahraných v datové sadě – každá výzva reprezentuje cca 13 sekund, celkem je tedy v těch datových sadách k dispozici přibližně 6500 datových bodů. U vizuálních datových sad Dominika a Lima byl v třetí fázi Visual Prompteru prezentován nápis „Imagine/Představuj si“, kdežto u sad DominikaB, Bravo a Foxtrot byla znovu přehrávána videa s pohyby prezentovanými ve fázi druhé [Sekce 3.4.2]

Kapitola 4

Demo aplikace pro ExcelFit

Ovládání hry pomocí levé a pravé ruky by nebylo v prostředí konference možné, jelikož tato metoda vyžaduje předtrénování na specifický mozek přímo před hraním samotné hry, pokud možno co nejlíže klidnému laboratornímu prostředí. Aplikace vodivého gelu na hlavu každého, kdo by chtěl vyzkoušet ovládání hry myslí by také byla komplikovaná.

Další částí vývoje je proto neurofeedback hra. Vyvinul jsem „Neurofeedback demo“ pro ExcelFit, kde jsem příchovivším umožnil ovládat hru alespoň pomocí soustředění, detaily o implementaci ovládání levou a pravou rukou jsem sdělil verbálně. Na konferenci měla nejlepší výsledky jedna neznámá dívka, vydržela udržet Buddhu ve vzduchu 38 sekund [Sekce 4.1]. Výsledná aplikace pak slouží jako kalibrační nástroj pro testovací subjekty před zahájením testovacího sezení, aby se seznámili s funkcionalitou UCIV headsetu.

4.1 Aplikace NeuroBuddha

Nejprve jsem provedl výzkum dostupných režimů neurofeedbacku (zda zvolit poměr theta/alfa, theta/beta nebo jiných pásem EEG vln), nejlepší rozmístění elektrod (FP1, FP2, Cz) a po nahrání zkušebních dat na testovacích subjektech jsem provedl základní kalibraci parametrů pro ovládání hry.

K vývoji jsem použil již dostupný software OpenVIBE, ve kterém jsem navrhl posílání dat z elektrod C3 a C4, na nichž provádím spektrální analýzu alfa, beta a theta vln. Data počítaná v reálném čase zasílám přes Lab Streaming Layer do mého herního skriptu. V OpenVIBE lze také specifikovat zasílání dat z elektrody na pozici Cz [Sekce 2.1], jež je podpořena výzkumy nejen pro ovládání končetin, a jelikož ADHD souvisí s tzv. „restless leg syndrome“, a tak i sensorimotorickým kortexem, je hlavním bodem výzkumů pro neurofeedback právě elektroda Cz v této práci používána pro detekci úmyslu pohnout končetinou [35].

Hra je psaná ve frameworku Unity¹, ve kterém se dnes vyvíjí AAA herní tituly. Zvolil jsem ho z důvodu předchozí zkušenosti s vývojem v něm. Unity je postaven na C# skriptech, a samotná obdržovací vrstva LSL [Sekce 3.4.3] existuje i jako externí knihovna pro jazyk C#, kterou zde používám. Z OpenVIBE tedy zasílám data, která v Unity přijímám a následně s nimi pracuji. Po vypočítání poměru alfa a beta vln je prováděna manipulace s herními objekty.

¹<https://unity.com/>

4.1.1 Neurofeedback

Další názvy pro neurofeedback jsou také EEG biofeedback nebo neuroterapie. Jedním z cílů této práce je navrhnout hru, která bude sloužit pro neurofeedback. Neurofeedback je metoda biofeedbacku, metody získávání zpětné vazby svého těla pomocí senzorů biologických funkcí, kdy se subjekt skrz zpětnou vazbu vědomě učí ovládat sledovanou “veličinu”, např. tlukot srdce nebo dech. Neurofeedback je pak nadstavbou této metody, používající EEG signály mozku pro zlepšení funkce zamýšlené oblasti mozku. V ADHD terapii, kdy je oslabený prefrontální kortex (PFC), se pak neurofeedback používá hlavně pro posílení funkce PFC, čímž se posiluje schopnost soustředěné pozornosti a síla vůle.

Analýza nejlepší taktiky neurofeedbacku

Poměr mezi beta a theta vlnami je dobrým indikátorem soustředěné pozornosti. Neurofeedback učí uživatele skrz zpětnou smyčku vědomě vytvářet správný poměr těchto vln a tak ho učí utvrzovat svoji pozornost [2] [9] [33] [31] [30].

Nalezl jsem vícero výzkumů na téma neurofeedback, hlavně jsem se zaměřoval na ADHD výzkumy a použitelnost jednotlivých frekvenčních pásem ve vzájemné kombinaci [35] [19].

Výzkum obsahuje přehledovou tabulku srovnávající theta/beta a alfa1/alfa2 poměry kontrolních subjektů a subjektů s různými podtypy ADHD. Nejpoužívanější protokoly jsou hlavně theta/beta a theta/alpha. Jejich cílem a potlačení theta vln a zvýšení alpha a beta vln.

4.2 Architektura hry

Hře jsem dal kódové označení NeuroBuddha, jelikož hlavní „postavou“ je socha Buddha, kterou hráč zvedá myslí do vzduchu. Hra obsahuje i časovač, aby dal hráči motivaci udržet Buddhu ve stavu levitace co nejdéle. Pro hru jsem použil nízko-polygonální modely, kvůli pěkné estetice a zároveň také lepšímu výkonu i na mém slabším notebooku, používaném pro prezentaci na ExcelFit.

Hra je zasazena v prostředí na louce, pro příjemné nerušivé prostředí. V pozadí jsou hory a vše je osvětleno simulací volumetrického světla, která dodává objektům tvrdé i měkké stíny. Používám ambient occlusion, aby měla scéna realistické míchání světla a stínu. Na přední kameru jsou aplikovány filtry pro vyvážení barev, hloubku ostrosti a chromatickou aberaci, pro dodání atmosférického nádechu prezentované scénérii. Uprostřed louky sedí socha Buddha, která má nad sebou a pod sebou neviditelný objekt značící spodní a dolní hranici. Na objekt sochy jsou také navázány zvukové události, které jsou spuštěny při počátku levitace, stálé levitaci a dopadu na zem. Socha také emituje zvuk dokreslující hudby (hudbu jsem vytvořil z vícero zvukových stop, většinou ambientní zvuky a zvuky lesní přírody).

Jakmile je detekován správný poměr alfa/theta/beta vln, na sochu je aplikována funkce LERP², která lineárně interpoluje mezi dvěma vektory, tím pádem vytváří plynulý pohyb. V tomto případě se jedná o dva Transform vektory, spodní a horní objekt, s vypnutým Mesh rendererem pro zajištění neviditelnosti. Pokud se kolizní box objektu Buddhy dotkne kolizního boxu horního objektu, začnou stopky inkrementovat čas a hráč je prezentován s zpětnou odezvou nejen ve formě časového údaje, ale také zvuku značícím levitaci. Vzal jsem různé nízko-vibrační zvuky a zvuk OHM (136.1 Hz) a vytvořil krátký zvuk navazující

²<https://docs.unity3d.com/ScriptReference/Vector3.Lerp.html>

začátek na konec pro možnost nerušivého loopingu. Jakmile je hráč rozrušen a poměr jeho mozkových vln přestane vyhovovat podmínce, socha přestane interpolovat k hornímu bodu od nynější pozice a spadne. To je opět značeno zpětnou odezvou, tedy jak vizuální tak zvukovou, kdy je při kontaktu dvou kolizních boxů na zemi přehráán zvuk žuchnutí/dopadu na zem.



Obrázek 4.1: NeuroBuddha v akci



Obrázek 4.2: NeuroBuddha na ExcelFit

Kapitola 5

Data pre-processing - předzpracování dat

Během vývoje jsem použil rozdílné způsoby ukládání dat z Visual/Sound Prompteru, a nebyly tak mezi sebou kompatibilní. Pro první verzi jsem použil formát json a formu Python slovníku (`json_dump`) pro ukládání nahraných dat. Způsob byl funkční, ale pro další „lidskou“ manipulaci se hodí spíše tabulky – proto jsem již Sound Prompter navrhl tak, abych mohl ukládat do tabulkového formátu csv. Jednotlivé sloupce jsou zde anotovány podobně jako je tomu u defaultního formátu OpenBCI GUI, které ukládá do `.txt` souboru. Přibližně v půlce vývoje jsem vytvořil skript pro transformaci mého starého formátu používaného Visual Prompterem do formátu `.csv`, který používám v Sound Prompter datech. Ten je následující:

```
timestamp;fp2;fp1;c3;o2;p7;p8;cz;c4;accX;accY;accZ;class  
1555760729.504407;-62570.22265625;-36627.9375;-60971.9609375;...  
1555760729.504407;-62570.22265625;-36627.9375;-60971.9609375;...
```

Jedná se o klasický csv formát, kde jsou jako oddělovače použity středníky. Soubor je tak snadno čitelný v tabulkových procesorech Excel a Calc a zároveň ho lze velice rychle načíst v Python skriptu pomocí frameworku `pandas`¹, který umožňuje i snadnou manipulaci, transformaci a filtraci dat pomocí `DataFrame` struktur. Sloupec `timestamp` drží časové razítko nahrání EEG vzorku, `fp2` a `fp1` data z prefrontálního kortexu, `c3/c4/cz` drží data z centrálního sensorimotorického kortexu, `P7` a `P8` z parietální části a `O2` z occipitální. Údaje `accXYZ` jsou hodnoty akcelerometru a `class` je label/třída nahrávané výzvy (0 pro levou ruku, 1 pro pravou).

5.1 Algoritmy

5.1.1 Extrakce rysů

Pro extrakci rysů používám Wavelet transform [Sekce], neboli vlnkovou transformaci. Je to efektivní způsob extrakce rysů a charakteristik EEG signálu a používá se i v oblastech jako je elektrokardiografie nebo elektromyografie. Dokážou pracovat jak s časovou, tak frekvenční doménou, což je také činí velice užitečným nástrojem právě pro EEG signály. Wavelety tak nemusí obětovat jednu ze složek, jak je tomu u Fourierovy transformace, která má vysoké

¹<https://pandas.pydata.org/>

rozlišení ve frekvenční doméně a nulové/nízké (dá se použít polyhblivého okna v rámci časové domény) v časové. Wavelety mají střední rozlišení v čase i frekvenci.

Ve srovnání s Fourierovou transformací, která používá řadu sinusoid s různými frekvencemi pro analýzu signálu, Diskrétní wavelet transformace (DWT) používá řadu wavelet funkcí, každou s jinou škálou. Sinusoida není lokalizována v čase, kdežto DWT funkce lokalizována je, a umožňuje tak získat nejen frekvenční, ale i časové informace.

DWT je tak ideálním analytickým nástrojem pro velice proměnlivý signál, jímž EEG rozhodně je. DWT analyzuje signály na různých úrovních rozlišení pomocí jejich rozkladu na hrubé (aproximace - A) a detailní (details - D) informace o signálu. Každá z úrovní je převzorkována na dvakrát menší frekvenci pomocí filtrů horní a dolní propusti. Horní propust poskytuje details, kdežto dolní propust poskytuje aproximace. Po prvním kroku, aplikaci filtrů a převzorkování, je celý proces opakován. Míra jednotlivých rozložení je tak určena rekurzivní iterací vzniklých aproximací. V existujících výzkumech pro sensorimotorický kortex se používají hlavně wavelet transformace z rodiny Daubechies (db), Biortogonální (Bior) a Symlets (sym), protože sdílejí charakteristiky vyžadované pro extrakci jak nízkofrekvenčních, tak vysokofrekvenčních složek EEG signálu [17]. V odkazované práci používají výzkumníci zejména wavelety Db4 a sym9 a 3. detailní rozklad pro signály elektrod C3 a C4.

Nejvýznamnější pozice pro ovládání rukou a nohou jsou právě zmíněné pozice v 10-20 systému [Obrázek 2.1], a totiž C3, Cz a C4 [23].

Dalším problémem, kterým se práce zabývá je odstranění artefaktů, jako jsou pohyby očí, mrkání nebo zatínání čelisti, a šumu ze vstupního signálu, které mohou narušit výslednou přesnost klasifikace [Sekce 5.3]. Wavelet transformace nevyžadují přílišné předzpracování dat, jelikož mají schopnost odstraňování šumu a očních artefaktů [28].

Sensorimotorický kortex, jež tyto 3 pozice odkazují, má důležitou roli v plánování pohybu a jeho provedení a proto je hlavním bodem této práce. V těchto částech se pak vyskytují tzv. Mu vlny, které jsou zastoupeny v Alfa pásmu, a to mezi 8-13 Hz (resp. 9-11 Hz) [Sekce 2.2.3] [40]. Jsou přítomny hlavně při plánování, pozorování nebo provádění pohybových úkonů [38]. V mojí práci používám pro rozdělení na frekvenční spektra právě DWT.

K tomuto účelu se používají algoritmy jako PCA nebo ICA (Primary component analysis a Independent component analysis), které zároveň snižují počet komponent/rysů [Sekce 5.2].

Při první fázi rozkladu se signál rozloží na nízko-frekvenční část a vysoko-frekvenční část. Testovací data jsou v mém výzkumu nahrávána se vzorkovací frekvencí 250 Hz. První aproximační rozklad A1 reprezentuje frekvence 0-125 Hz a detailní rozklad D1 na 125-250 Hz. Druhý rozklad přijme aproximační koeficienty předchozího a provede stejnou wavelet transformaci znovu. Dostane se tak rozklad A2 0-62.5 Hz a D2 62.5-125 Hz atd.

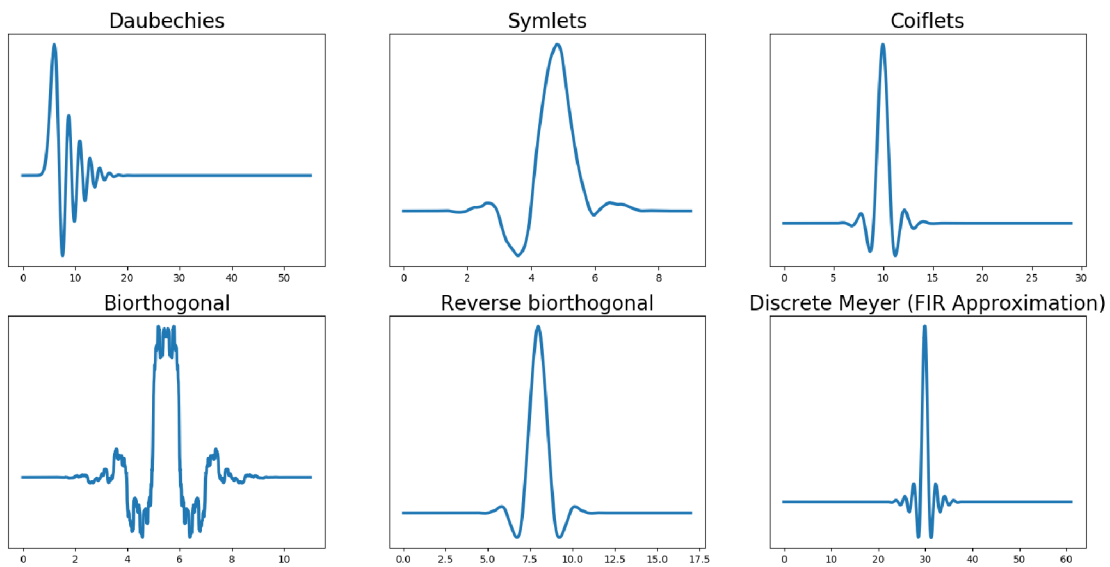
K tomuto rozkladu používám knihovnu PyWavelets ².

- Na segment signálu reprezentující 1 sekundu aplikuji funkci `dwt()`
- Ta vrátí dva seznamy koeficientů: aproximační a detailní
- Aproximační koeficienty jsou výsledkem DWT filtru dolní propusti
- Detailní koeficienty jsou výsledkem horní propusti
- Funkci `dwt()` aplikuji znovu na aproximační koeficienty z předchozího kroku

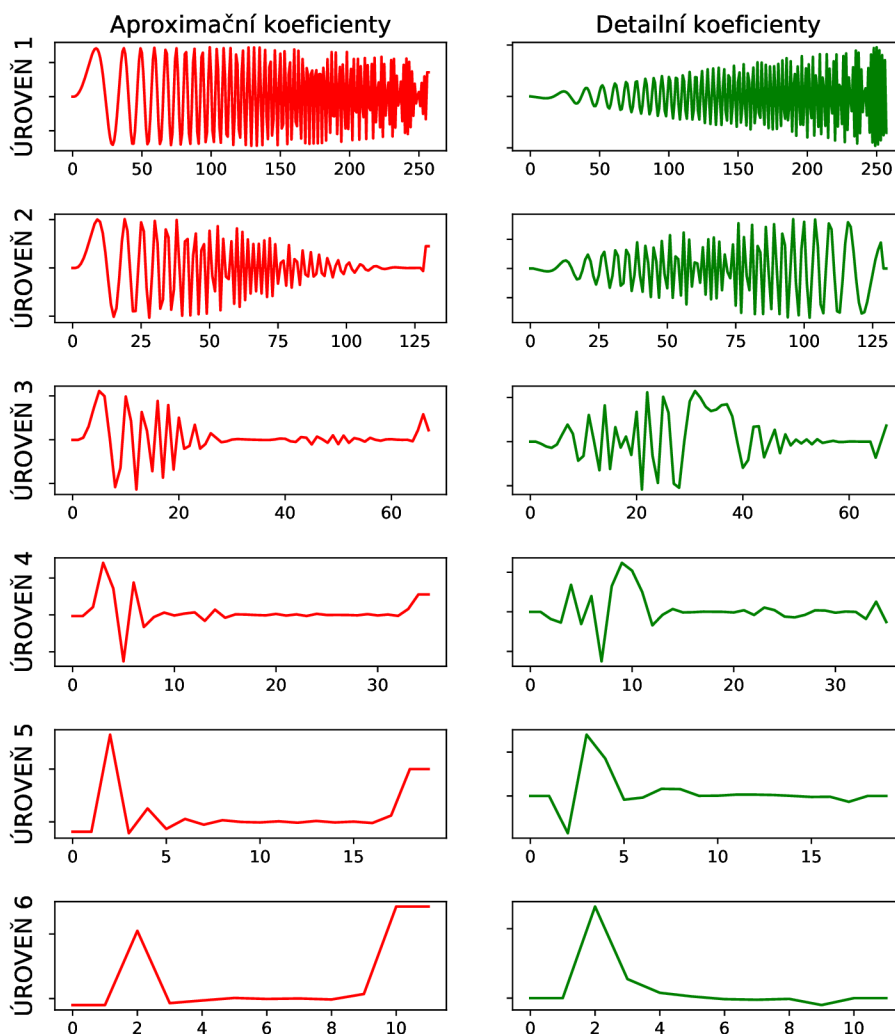
²<https://pywavelets.readthedocs.io/en/latest/>

- Dostávám další úroveň rozkladu

Pro testování používám 1-6 rozkladů [Obrázek 5.2], jak aproximačních tak detailních. V existujících výzkumech jsou více používány detailní rozklady 2-5 [17].



Obrázek 5.1: DWT rodiny



Obrázek 5.2: Rozklad pomocí waveletu Bior1.3

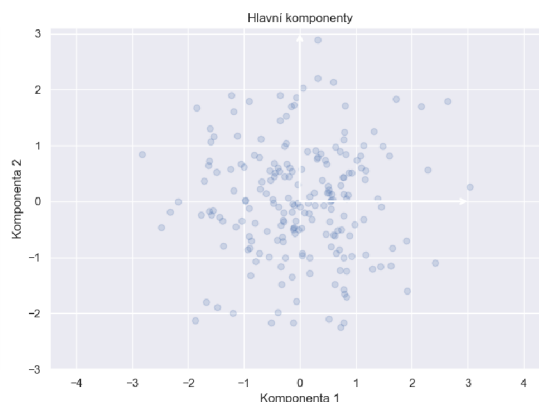
5.2 Metody redukce dimenzionality

Čím více u použitých datových sad roste počet rysů, tím obtížnější je pro klasifikátor správně generalizovat (a tak se učit) údaje, které je schopen použít pro správnou klasifikaci. Z toho důvodu se používají metody pro redukci dimenzionality rysů, které jsou schopny určit pouze ty rysy, jež jsou správnou klasifikaci nejdůležitější. V této práci používám PCA.

Principal Component Analysis (PCA), neboli analýza hlavních komponent je metoda sloužící k de Korelaci numerických dat. Používá se k redukci dimenzionality rysů (datové sady) s co nejmenší možnou ztrátou informace. Nejdůležitější rysy jsou zvoleny na základě množství změn, které způsobí ve výstupu. Rys, který ovlivňuje výstup nejvíce je zvolen jako první hlavní komponenta; druhý rys jako druhý hlavní komponenta atp. Takto zvolené hlavní komponenty mezi sebou nejsou nijak korelovány ³.

³<https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis>

Hlavní výhody PCA pro trénování a klasifikaci je zejména to, že se výrazně sníží čas potřebný pro trénování. Před aplikováním PCA je nutné nejprve normalizovat vstupní data (k tomu používám MinMaxScaler [4])



Obrázek 5.3: PCA na 2 komponenty

5.3 Klasifikátory v balíku scikit-learn

V mojí práci používám hlavně klasifikátory z Python balíku scikit-learn ⁵, na kterých provádím dále uvedené experimenty.

5.3.1 Random Forests

Random Forests (RF), metoda náhodných lesů, je nelineární metoda pro regresi a klasifikaci, která vytváří rozhodovací stromy a vrací nejčastější hodnotu tříd jednotlivých stromů.

Rozhodovací strom je základním stavebním kamenem náhodného lesa. O rozhodovacím stromu se dá přemýšlet jako o řadě otázek Ano/Ne (True/False) směřovaných na testovaná data, které algoritmus dovedou na kýžený výsledek. Náhodné lesy jsou tak velice snadný model na pochopení, jelikož dělají rozhodnutí podobně jako lidi. Jsou postaveny na rozhodnutích (zvaných rozdělení uzlů), která vedou k redukci tzv. Gini impurity (GI) ⁶ (další metrikou je entropie, která je v podstatě stejná jako GI, ale pro výpočty používá navíc logaritmické funkce. GI je používána pro snížení míry chybné klasifikace a entropie pro experimentální analýzu. Z toho důvodu zahrnuji v parametrech zkoušených u klasifikace RF pouze GI.). GI je metrika, kterou rozhodovací strom snaží minimalizovat při rozdělování uzlů. Reprezentuje pravděpodobnost, že náhodně zvolený vzorek v uzlu bude nesprávně klasifikován. Pokud se nespecifikuje u RF hloubka průchodu, může nastat přetrénování. To nastává, pokud je trénovací model velice přizpůsobivý a takřka si zapamatuje rozhodnutí dělaná na trénovacích datech tak, že se jim maximálně přizpůsobí. Problémem je i to, že se přizpůsobí i v datech přítomnému šumu. Naopak nepřizpůsobivý model dělá o datech „předsudky“ a v některých případech se nepřizpůsobí ani na samotná trénovací data. Pokud se tedy u stromu neomezí hloubka průchodu, jeho větve mohou stále růst a mít tak neomezenou flexibilitu. Jako alternativa k omezování hloubky stromu se nabízí spojení více rozhodovacích stromů do jediného modelu, Náhodného lesa.

⁴<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler>

⁵<https://scikit-learn.org/>

⁶<https://datascience.stackexchange.com/questions/10228/gini-impurity-vs-entropy>

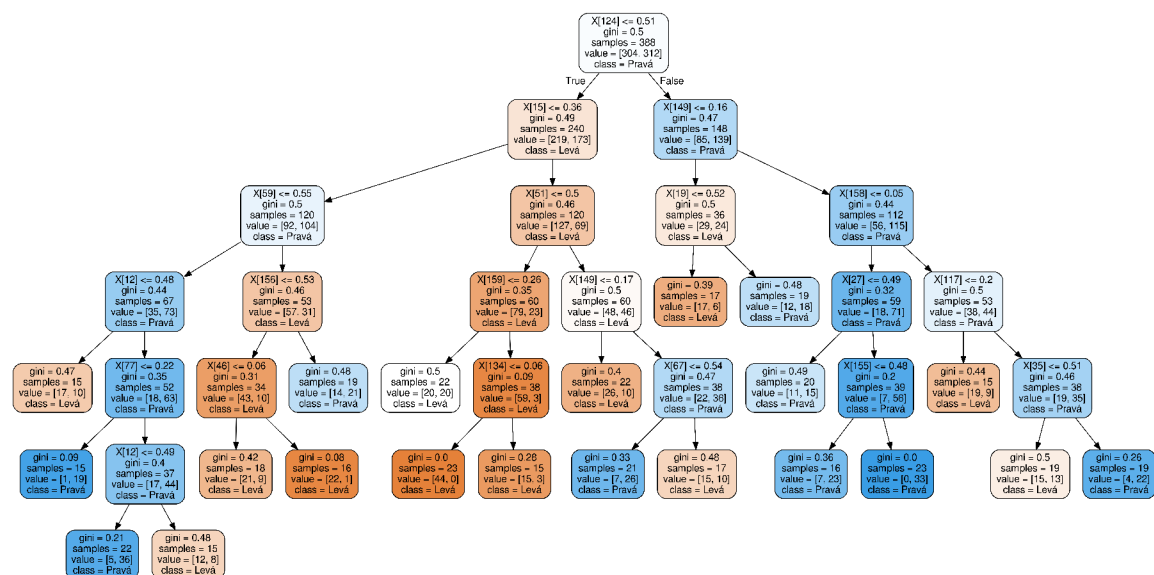
Náhodný les je tvořen více rozhodovacími stromy, při jejichž tvoření dochází k náhodnému vzorkování trénovacích dat, a stejně tak dochází k vybírání náhodných podmnožin rysů při dělení uzlů (při „rozhodnutí“). RF používá techniku bootstrappingu, což znamená, že některé vzorky budou použity vícekrát - resp. stejné stromy budou použity na náhodných kombinacích vzorků. Během testování jsou predikce uskutečněny pomocí zprůměrování predikcí každého z rozhodovacích stromů. Tento proces je znám jako agregace bootstrappingu [18]. Náhodné lesy tedy kombinují tisíce rozhodovacích stromů, trénují každý z nich na lehce jiné podmnožině dat, a rozdělují uzly při určitém počtu rysů. U scikit-learn implementace ⁷ je to druhá odmocnina z celkového počtu rysů.

Jedna ze zajímavých kvalit RF je, že dokáže analyzovat tzv. Feature Importance, neboli důležitost rysů. Každý RF model tak dokáže vrátit, jak jsou pro něj jednotlivé rysy důležité (model.feature_importances_).

V práci „EEG signal analysis for Seizure detection using Discrete Wavelet Transform and Random Forest“ [22] došli výzkumníci k závěru, že pro daný problém je neefektivnější metodou Random Forest Classifier, a to díky dosažené míře přesnosti klasifikace 98 %.

V mém výzkumu a experimentech se zabývám parametry:

- hloubka stromu (max_depth)
- minimální počet vzorků nutných pro rozdělení uzlu (min_samples_split)
- minimální počet vzorků vyžadovaných pro listový uzel (min_samples_leaf)
- počet rysů braných v potaz při rozhodování o nejlepším rozdělení uzlu (max_features)
- bootstrap
- počet stromů v lese (n_estimators)



Obrázek 5.4: Část náhodného lesa z frameworku během klasifikace

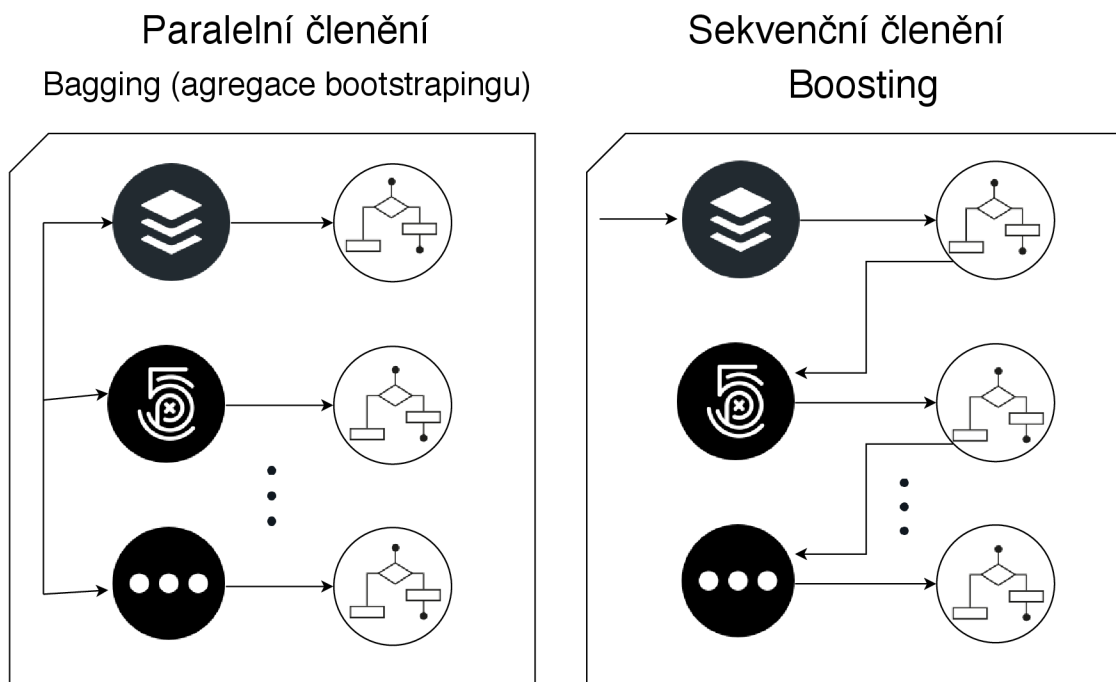
Všechny uzly kromě těch listových obsahují 5 částí [Obrázek 5.4]:

⁷<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier>

- Podmínka rysu
- GI - při postupu stromem k listům by se měla zmenšovat
- Počet přístupů k datům (kolikrát strom viděl vzorek)
- Počet vzorků pro každou třídu (zde Levá/Pravá)
- Třída, která v uzlu převažuje

5.3.2 Gradient Boosting Classifier

Gradient Boosting Classifier (GBC) produkuje podobně jako RF predikční model za použití rozhodovacích stromů, aplikován může být tento model ale i na jiné typy tzv. weak learner (špatný klasifikátor) klasifikátorů [13]. GBC tedy spojuje více těchto špatných klasifikátorů do jednoho, aby vytvořil jeden silný klasifikátor. Gradient Boosting Machines jsou obecně aplikovány na modely založené na rozhodovacích stromech, ale dají se použít na jakýkoliv typ weak learner modelu.

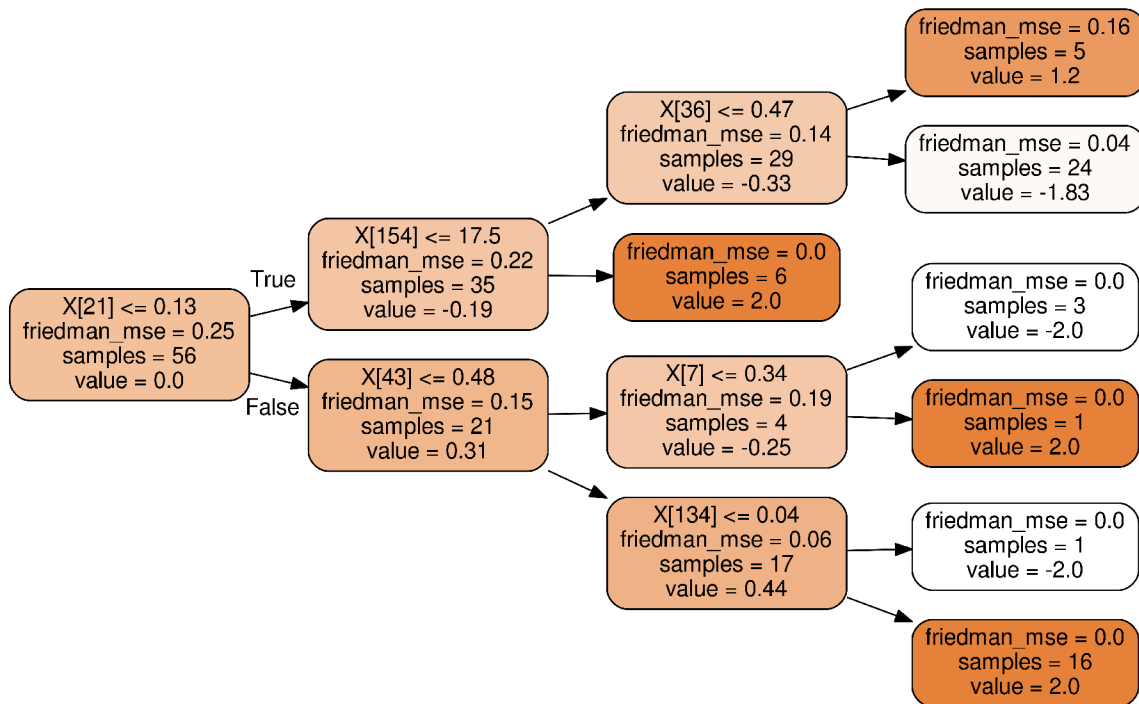


Obrázek 5.5: 3 různé vzorky dat - Bagging (RF) a Boosting (GBC)

Při Gradient boostingu dochází k sekvenční adaptaci slabých klasifikátorů na chybu předchozího modelu. Na počátku je pouze odhadovaná hodnota, která je následována sekvencí rozhodovacích stromů. Postup touto sekvencí pak tvoří jednotlivé kroky GB klasifikátoru. Konečná hodnota predikce je pak vážený průměr výsledků této sekvence. Podobně jako u RF se zde mj. dá ovlivňovat hloubka průchodu stromem a počet estimátorů. Na obrázku 5.7 jsou vidět i parametry jako `friedman_mse`, značící průměrnou chybu (mean squared error ⁸). V kapitole Experimenty dále používám např. parametr `subsample`, která

⁸<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier>

ovládá počet vzorků použitých pro jednotlivé stromy (pod-stromy, individuální klasifikátory). Pokud je < 1 , používá se Stochastic Gradient Boosting (volení náhodných vzorků), což vede k redukcí rozptylu (flexibilitnosti stromu) a zvýšení schopnosti generalizace.



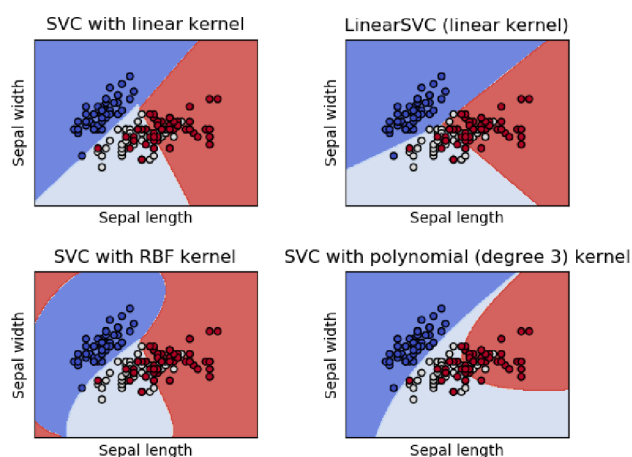
Obrázek 5.6: Snímek stromu GBC v mém frameworku

5.3.3 Support Vector Machines

Support Vector Machines, neboli Metoda podpůrných vektorů, byla původně navržena pro binární klasifikaci, ale může být upravena pro klasifikaci i více tříd. Lze u ní použít více jader, jako RBF nebo Poly. Výzkumníci došli se SVM také k velice dobrým výsledkům za použití RBF jádra [32].

SVM hledají optimální nadrovinu, která rozděluje více tříd tak, aby vzdálenost od okrajových bodů jednotlivých tříd byla co největší. Body ležící na okraji jsou nazývány podpůrné vektory [26].

Metoda existuje i v provedení **NuSVC**, kdy lze ovládat počet podpůrných vektorů pomocí proměnné Nu.



Obrázek 5.7: Jádra SVM [8]

5.3.4 Metrika ROC a AUC

Metrika ROC (resp. křivka ROC - Receiver Operating Characteristic Curve) je graf vyobrazující výkonnost klasifikačního modelu. Metrika Area Under Curve⁹, neboli AUC, se používá při nevyvážených klasifikačních problémech a jedná se o integrál plochy pod ROC.

ROC má parametry:

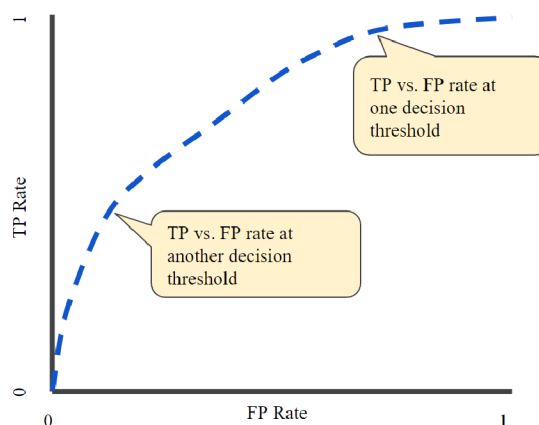
- Míra pravdivě pozitivních klasifikací (True Positive Rate - TPR)

$$TPR = \frac{TP}{TP + FN}$$

- Míra falešně pozitivních klasifikací (False Positive Rate - FPR)

$$FPR = \frac{FP}{FP + TN}$$

Křivka se mění podle poměru TPR a FPR, viz



Obrázek 5.8: ROC [14]

⁹<https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>

5.4 Způsoby zpracování dat

5.4.1 Architektura frameworku pro experimenty a formát dat

V této práci jsem vytvořil skript určený pro testování nasbíraných dat popsanych v tabulce 3.1.

```
test_segments = []
for custom_second in range(0,11):
    dummy_x, dummy_y, \
    X_test, y_test = get_datasets_ready(custom_second, electrodes,
                                       minApprox, maxApprox, minDets,
                                       maxDets, wantApprox, wavelet,
                                       doSeconds=True)

    evaluated = evaluate(model, X_test, y_test, X_train, y_train)
    test_segments.append(evaluated[0]) # AUC
plt.plot(test_segments)
plt.xlabel('seconds')
plt.show()
```

Obrázek 5.9: Ukázka kódu z frameworku (použit pro jeden z experimentů dále - Obrázek 6.5)

5.4.2 Moduly

Modul ModulesEEG obsahuje hlavní třídu, která zpracovává a stará se o většinu dat při běhu frameworku. Ve funkci `extractCsvChannelIntoArray` načítám data do pandas DataFrame podle oddělovače „;“.

Funkce `wavelet_decomposition` tvoří přibližně půlku všech volání ve skriptu (provedl jsem profilování aplikace). Pokud je specifikován testovací parametr pro použití koeficientů rozkladu DWT, je načten postupně specifikovaný trénovací a testovací dataset, který je dále zpracován pomocí frameworku pandas pro práci s csv soubory. Všechny hodnoty jsou rozděleny podle oddělovače buněk a načteny do pole s DataFrames. Pole je vyfiltrováno a rozděleno na základě sloupce „class“, který obsahuje identifikátor třídy nahrané výzvy.

V této části, `transform_input_data`, mohu specifikovat i počet useknutých sekund na počátku výzvy (po zaznění zvukového oznámení by teoreticky testovaný mohl mít ruchy ze doznívajícího zvuku, než se začne soustředit na pohyb končetin).

Dalším krokem je funkce `compute_class`, která přebírá většinu vstupních argumentů z počátku testu. Vstupními daty je iterováno podle velikosti vstupního okna (např. 250 = 1 sekunda - pro ovládání aplikace nebo hry bude použito okno právě jedné sekundy, testování probíhá s ním) a načtené hodnoty pro specifikované elektrody (např. C3 a C4) jsou podle parametrů zpracovány. Funkce `wavelet_decomposition` přebírá data ze vstupního okna a provede na nich wavelet transformaci podle specifikované wavelet transformace (např. sym9). Po rozkladu aproximací a detailů na 6 úrovní jsou zvoleny požadované rozklady, které jsou normalizovány pomocí `MinMaxScaler` na škálu 0.0-1.0 a spojeny do jediné posloupnosti jako jeden datový bod/sample.

Jedna z možností je také provedení statistické analýzy na vstupních datech. Na vzorku je tak počítána variance, standardní odchylka, medián, průměr a další. Takto vytvořená data však pozбудou velké množství informace a proto není příliš vhodná na klasifikaci, ale pouze pro experimenty.

Data jsou po transformaci načtena a pomocí funkce `shuffle` konzistentně promíchána (`seed = 48`). Testovací data jsou vůči trénovacím vždy v poměru 25-30 % a jejich průnik je prázdná množina.

V další části je spuštěna samotná klasifikace za použití zvoleného klasifikátoru. Ten je předán funkci `classify`, která provede fitting modelu na trénovací data. Následně je model zaslán do funkce `evaluate`, kde je spuštěna predikce na testovacích rysech a je spočítána míra false positive (FP) a true positive (TP). Další funkce, která převezme data, je `auc` z balíku `scikit.metrics`, počítající tzv. Area Under Curve (AUC) metriku pro FP a TP. Další metrikou je `score` trénovacího i testovacího modelu a je spočítána confusion matice. Z těchto metrik je vytvořena klasifikační zpráva, která je naformátovaná tak, aby šla uložit jako tabulkový formát `.csv`, kde je možné jednotlivé metriky a detaily o všech parametrech proběhlého experimentu dále použít pro filtraci podle veličin/metrik a případně manuální vytvoření grafu. Vytvoření automaticky anotovaného grafu je možné i pomocí funkce `my_plt`, která vykreslí graf se zvýrazněným nejlepším výsledkem.

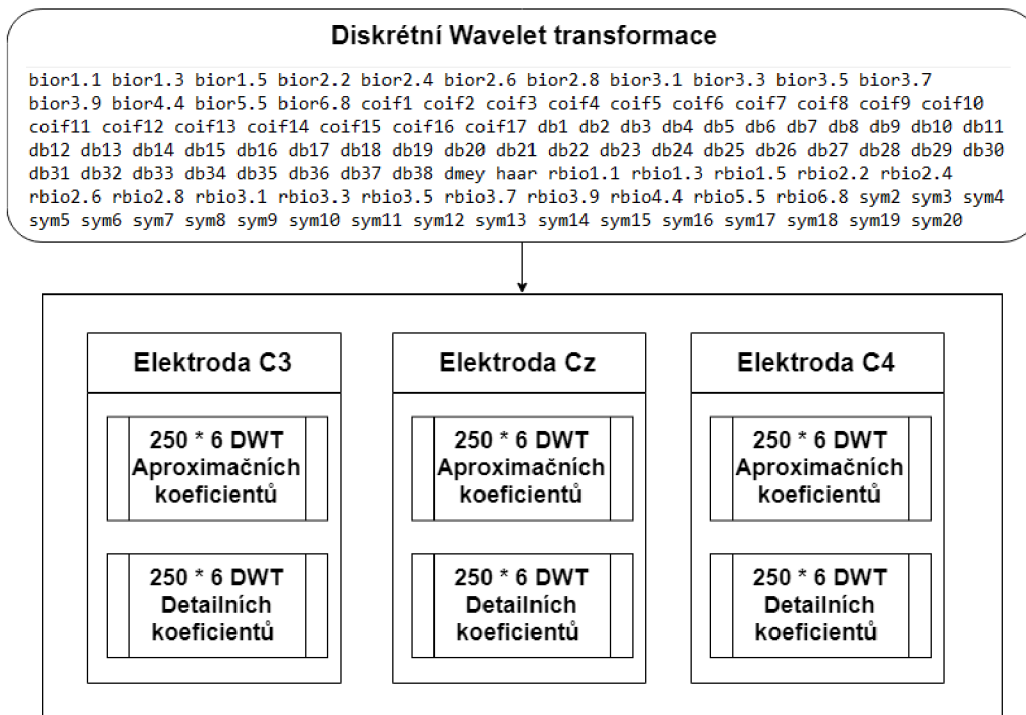
5.4.3 Použití

Při počátku testování specifikuji požadované parametry, napříč kterými je iterováno a na jejichž základě je transformována trénovací a testovací datová sada.

Mezi parametry patří: pole s diskrétními wavelet transformacemi, jména testovacích a trénovacích datasetů, počet estimátorů, spodní a horní index pole s wavelet aproximacemi, spodní a horní index pole s wavelet detaily, přepínač numerických statistik a wavelet koeficientu, přepínač pro použití aproximačních nebo detailních koeficientů, rozložení elektrod (C3, Cz, C4), velikost pohyblivého okna pro analýzu, velikost přesahu okna (`overlap`) a nakonec pole s klasifikátory, které budou použity v testu. Výstup je po každé iteraci zaznamenán do logovacího souboru i na standardní výstup a po dokončení testu je možné zapnout generování grafických statistik a grafů pomocí `matplotlib.pyplot` knihovny, díky které vytvářím dynamicky generované grafy se zvýrazněným nejlepším výsledkem (prezentováno níže) [Kapitola 6].

Při testování jednotlivých parametrů se ukázalo jako nejvíce časově náročné tvoření souborů. Dekompozice pomocí wavelet transformace zabírala často větší množství času než samotná klasifikace, proto jsem se rozhodl výsledný framework ještě upravit. Předgeneroval jsem soubory pro všechny wavelety a pro všechna rozložení sekund (`startDrop, endDrop`) a sjednotil jsem nahrání výzvy na 2500 vzorků (čistých 10 sekund, pokaždé).

Výsledný testovací soubor má strukturu vektoru s rozměry 9000x1 pro jednu sekundu záznamu [Obr. 5.10] (v případě statistik o DWT rozkladech počítám 5 statistických údajů o DWT rozkladu - medián hodnot, průměr hodnot, standardní odchylku, rozptyl a výkon. Pro představu stačí v diagramu nahradit 250 za 5, rozměr se změní na 180x1):



Obrázek 5.10: Trénovací sada po úpravě na všechny kombinace (A a D jsou v souboru sériově, pod sebou vyobrazeny pro úsporu místa)

Kapitola 6

Experimentální testování

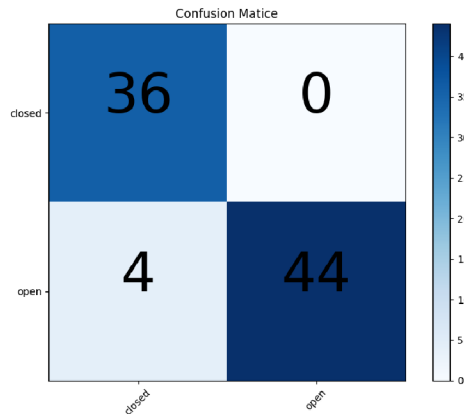
Pro experimenty jsem zvolil klasifikátory Random Forests [Sekce 6.2], Gradient Boosting [Sekce 6.1] a SVM [Sekce 6.3]. Výzkum [22] ukázal klasifikátor RF jako dobrou volbu pro práci s EEG daty, proto předpokládám, že i v mých datasetech bude vykazovat dobrou úspěšnost. GBC v kombinaci s DWT [Sekce 5.1.1] použil výzkum [42].

6.1 Gradient Boosting

Ve výzkumu [17] vyzkoušeli ke klasifikaci EEG signálu vícero DWT funkcí, a k nejlepším výsledkům se dostali s DWT z rodiny Symmlet, Sym9. Pro moje prvotní experimenty proto volím Sym9. Začínám s jednotlivými experimenty a postupně se dostanu ke složitým. Z důvodu časové náročnosti testování jsem prováděl experimenty pouze s datasetem Martin [Tabulka 3.1]. Jde o dataset EEG dat nahraných se zavřenými očima. (Martin má zkušenosti s meditací a zároveň díky jeho zálibě v posilování je schopen si jasněji představovat pohyby, tím pádem je ideálním kandidátem na testování představování si pohybu. Po testovacích sezeních poznamenal, že si nejlépe představoval pohyby jako pravý hák a levý hák). Jde také o druhý největší dataset se 56 nahranými výzvami pro testování a 23 pro testování (na první příčce je v této oblasti dataset Echo).

6.1.1 Otevřené/zavřené oči

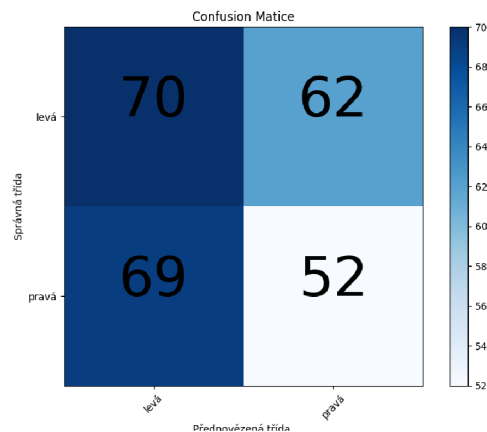
Nejprve provádím pokus s otevřenými/zavřenými očima. Vzal jsem sadu Foxtrot a Echo, kde u Foxtrot je 100 % výzev děláno s otevřenými očima (Visual Prompter) a Echo, kde je pár hodin od ukončení nahrávání Foxtrotu nahráváno na stejném subjektu (Autor) se zavřenými očima (Sound Prompter). Pro sadu, kterou jsem pojmenoval „Eyes“ tentokrát nepřikládám tabulku [Tabulka 3.1], pouze základní velikost souborů csv: test má 9423 KB a train 42418 KB. Na problém jsem použil defaultní parametry GBC klasifikátoru, kromě estimatorů nastavených na 1000. Rysy jsem použil všechny D1-D5, A1-A5 a pro urychlení klasifikace jsem zredukoval dimenzionalitu na 10 rysů pomocí PCA. Dostávám úspěšnost klasifikace 95,8%. Zkouším ještě Random Forests, počet estimatorů nastavuji stejný jako GBC - úspěšnost 97,6 %, přičemž klasifikace je mnohonásobně rychlejší (paralelní vs sekvencní model). Zkouším také SVC s RBF jádrem, dává úspěšnost 94,4 %. U všech klasifikátorů volím `random_state=48`, pro replikovatelnost výsledků.



Obrázek 6.1: Otevřené-zavřené oči (AUC 0.958)

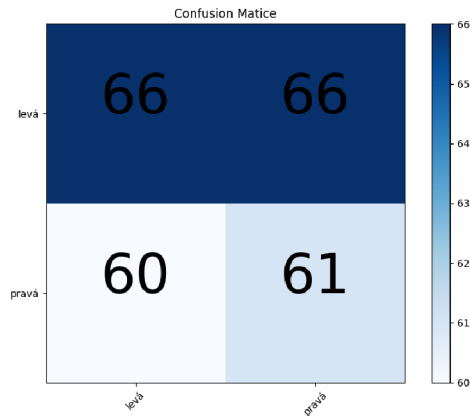
6.1.2 Sensorimotorika

Pro první experiment pro sensorimotoriku volím rozklady D4 a D5, jelikož ty zastupují sledované frekvence Mu a Beta (D5 je cca 7-15 Hz a D4 je cca 15-30 Hz). Aproximační rozklady v trénovací sadě prozatím ignoruji. Počet estimátorů volím 1000 a rychlost učení nastavuji na 0.05. Nijak neomezují sledovaný segment, tudíž se dá očekávat vysoká míra nepřesnosti.



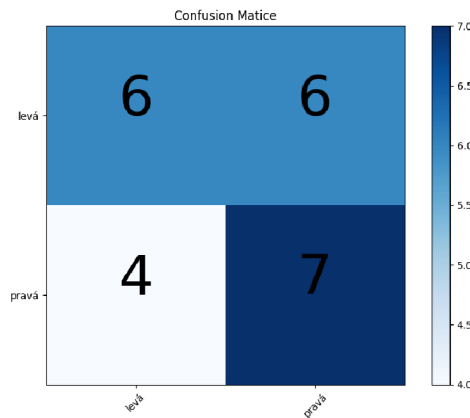
Obrázek 6.2: AUC 0.4800

První výsledek je 0.48, tzn. přibližně stejně jako náhoda. Zkousím použít redukci dimenzionality pomocí PCA=2 na stejný vzorek a dostávám se k vyváženějšímu výsledku, který ale stále značí velkou míru šumu.



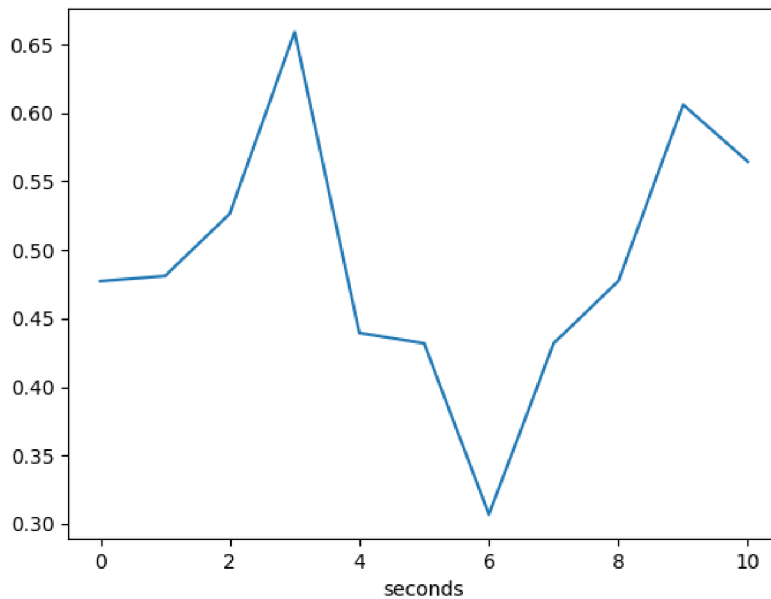
Obrázek 6.3: AUC 0.502

Při omezení segmentu na jenom určitou sekundu by se teoreticky měla úspěšnost zvýšit. V předchozích výzkumech [25] byly nejlepší výsledky okolo 4-té sekundy, zkouším tedy nastavit klasifikaci pouze na segmentu se 4. sekundou.



Obrázek 6.4: AUC 0.5681

Dle předpokladu se úspěšnost zvýšila, avšak počet vzorků se díky omezení na každou 4 sekundu velmi omezil a není tak statisticky příliš signifikantní. Je vidět, že v testovací sadě má Martin 11 výzev pro pravou ruku a 12 pro levou [Tabulka 3.1]. Zajímavé bude aplikovat model natrénovaný na celém 11 sekundovém segmentu (2750 vzorků z UCIV) na jednotlivé sekundy.



Obrázek 6.5: Model aplikovaný na sekundy

Jak je vidno na obrázku 6.5, Martin se nejvíc podle DWT Sym9 dokázal soustředit okolo třetí sekundy a těsně před koncem výzvy.

K testování jsem použil skvělé cloudové služby Metacentrum METAVO¹, dostupné studentům FITu. Na jejich cloudu jsem si zarezervoval stroj se dvěma grafickými kartami Tesla K20m běžící na Kepler architektuře, což mi umožňuje mnohonásobně rychlejší testování jednotlivých modelů, než na běžném stolním počítači.

V tabulce 6.1 jsou data z experimentu běžícího 3 dny na výpočetním cloudu METAVO. Celkově jsem dostal přes 78 tisíc výsledků. V tabulce je 42 nejlepších z nich. **Sec.** značí sledovanou sekundu, **lesApr** dolní hranici aproximačních rozkladů, **upApr** horní hranici (Aproximace však nejsou používány pokud je **Approx.** nastaven na **False**). **lesDet/upDet** značí dolní/horní hranici detailních rozkladů (tedy D5 a D6 např.), **elektrody** značí sledovanou konfiguraci (na prvním místě je pouze elektroda C4) a v posledním sloupci jsou vyobrazeny **Confusion matice** (horní řádek, spodní řádek).

Prvních 600 nejlepších výsledků jsem zanalyzoval a vybral 6 nejlepších wavelet transformací. Jsou to:

- bior3.1 x72
- bior1.1 x26
- bior3.3 x22
- db28 x16
- db1 x16
- rbio1.1 x14

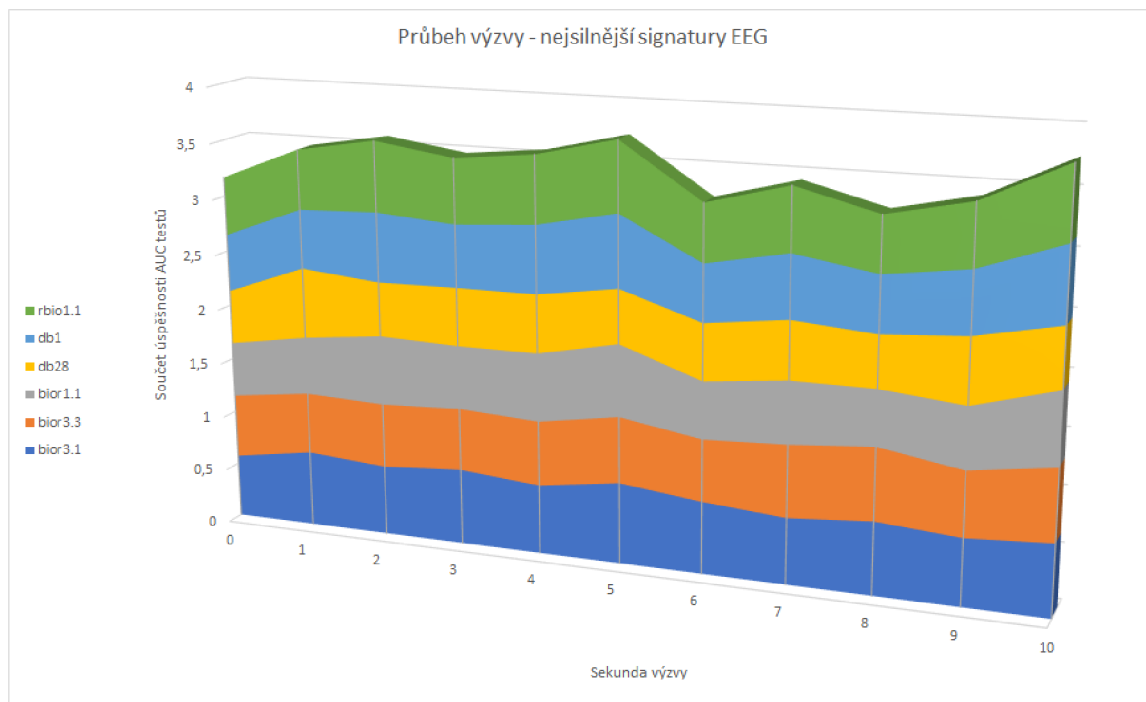
¹<https://metavo.metacentrum.cz>

Tabulka 6.1: Prvních 42 hodnot z výsledků 3-denního testu GBC čítajícího 78728 hodnot

AUC testu	wavelet	sec.	lesApr	upApr	lesDet	upDet	approx	el. (C3, Cz, C4)	Confusion matice
0.9583333333333333	sym13	4	2	5	4	5	False	[False, False, True]	[[11 1], [0 11]]
0.9583333333333333	sym13	4	3	5	4	5	False	[False, False, True]	[[11 1], [0 11]]
0.9583333333333333	sym13	4	4	5	4	5	False	[False, False, True]	[[11 1], [0 11]]
0.9128787878787878	bior3.1	5	3	5	2	5	True	[True, False, True]	[[11 1], [1 10]]
0.9128787878787878	bior3.1	5	3	5	2	5	True	[False, False, True]	[[11 1], [1 10]]
0.9128787878787878	bior3.3	8	2	5	3	5	True	[False, False, True]	[[11 1], [1 10]]
0.9128787878787878	db8	9	4	5	4	5	True	[True, False, False]	[[11 1], [1 10]]
0.875	sym4	6	4	5	4	5	True	[False, False, True]	[[9 3], [0 11]]
0.875	db4	8	4	5	4	5	True	[False, False, True]	[[9 3], [0 11]]
0.8712121212121213	bior2.2	10	2	5	4	5	True	[False, False, True]	[[10 2], [1 10]]
0.8712121212121213	bior3.1	5	2	5	2	5	True	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	bior3.1	5	3	5	3	5	True	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	bior3.1	5	4	5	3	5	True	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	bior3.1	5	4	5	4	5	True	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	rbio3.5	8	4	5	4	5	True	[False, False, True]	[[10 2], [1 10]]
0.8712121212121213	rbio2.4	4	3	5	4	5	True	[False, False, True]	[[10 2], [1 10]]
0.8712121212121213	rbio2.2	10	2	5	2	5	True	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	rbio2.2	10	2	5	2	5	False	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	rbio2.2	10	3	5	2	5	True	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	rbio2.2	10	3	5	2	5	False	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	rbio2.2	10	4	5	2	5	True	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	rbio2.2	10	4	5	2	5	False	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	coif17	4	2	5	4	5	False	[False, False, True]	[[10 2], [1 10]]
0.8712121212121213	coif17	4	3	5	4	5	False	[False, False, True]	[[10 2], [1 10]]
0.8712121212121213	coif17	4	4	5	4	5	False	[False, False, True]	[[10 2], [1 10]]
0.8712121212121213	db19	8	4	5	4	5	True	[True, False, True]	[[10 2], [1 10]]
0.8712121212121213	db1	10	3	5	2	5	True	[True, True, True]	[[10 2], [1 10]]
0.8712121212121213	db9	10	2	5	3	5	True	[True, True, True]	[[10 2], [1 10]]
0.8674242424242424	rbio5.5	7	2	5	4	5	False	[False, False, True]	[[11 1], [2 9]]
0.8674242424242424	rbio5.5	7	3	5	4	5	False	[False, False, True]	[[11 1], [2 9]]
0.8674242424242424	rbio5.5	7	4	5	4	5	False	[False, False, True]	[[11 1], [2 9]]
0.8636363636363636	db5	6	2	5	2	5	True	[False, False, True]	[[12 0], [3 8]]
0.8333333333333334	bior3.1	8	4	5	4	5	True	[False, False, True]	[[8 4], [0 11]]
0.8333333333333334	sym13	6	3	5	3	5	True	[False, False, True]	[[8 4], [0 11]]
0.8333333333333334	rbio1.3	10	4	5	3	5	True	[True, False, False]	[[8 4], [0 11]]
0.8295454545454546	bior1.1	0	4	5	4	5	True	[True, False, False]	[[9 3], [1 10]]
0.8295454545454546	bior1.3	4	2	5	2	5	False	[True, False, True]	[[9 3], [1 10]]
0.8295454545454546	bior1.3	4	3	5	2	5	False	[True, False, True]	[[9 3], [1 10]]
0.8295454545454546	bior1.3	4	4	5	2	5	False	[True, False, True]	[[9 3], [1 10]]
0.8295454545454546	bior3.7	10	4	5	3	5	True	[True, False, True]	[[9 3], [1 10]]
0.8295454545454546	bior3.9	10	2	5	4	5	False	[False, False, True]	[[9 3], [1 10]]

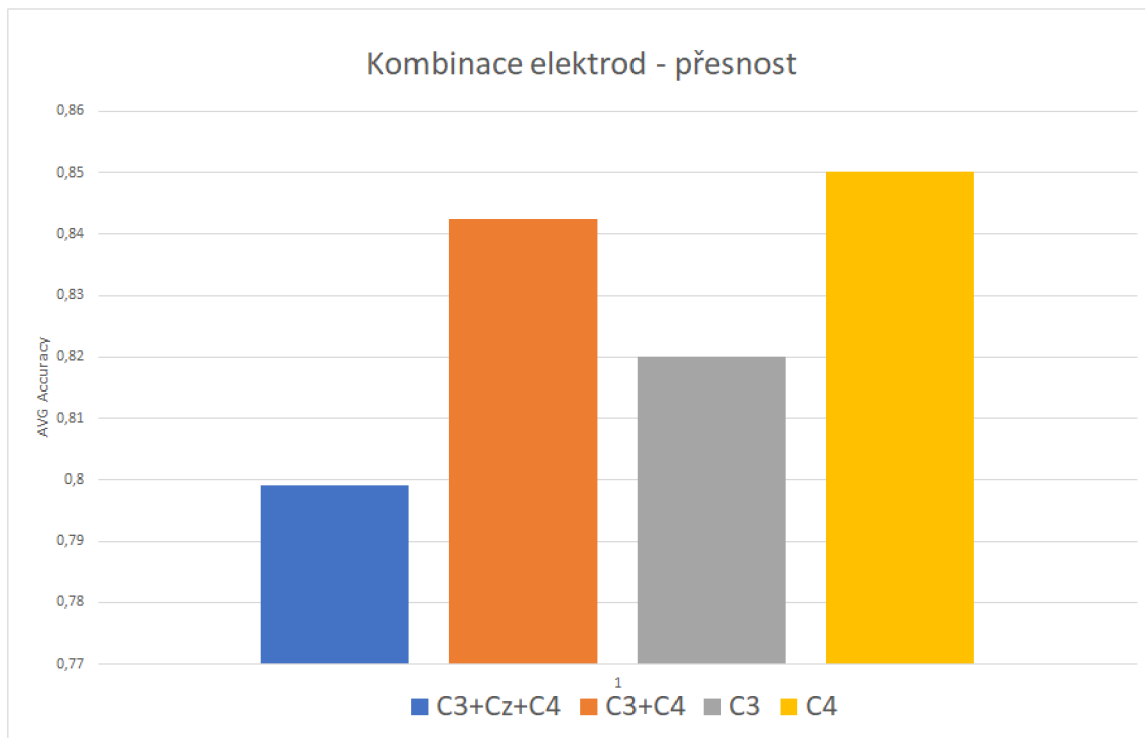
Jasně tedy převládá Biortogonální rodina a její Reverzní příbuzní. **Pro další výzkum proto volím výherce Bior3.1. Nejlépe Bior3.1 klasifikuje správně pro segment 5. sekundy a pro elektrody C3, Cz a C4 všechny zapnuté, a také pro C3 a C4 samotné.**

Analyzoval jsem průběh výzvy pomocí nejlepších 6 wavelet transformací. Na obrázku 6.6 lze vidět místa s nejsilnějšími signaturami EEG signálu pro levou a pravou ruku.



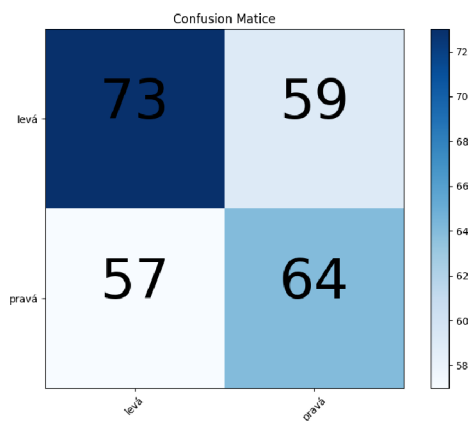
Obrázek 6.6: Nejsilnější signatury EEG pro pravou a levou ruku v čase

Důležitým poznatkem z testování je důležitost elektrod pro úspěšnost klasifikace sensori-motorického úkolu [Obrázek 6.7]. Podle grafu je zřejmé, že elektroda Cz opravdu nepřidává úspěšnosti klasifikace - v budoucím testování ji vynechávám. C4 má úspěšnost nejlepší, ale samostatná je nepoužitelná pro rozlišení levé a pravé ruky - dokázala by detekovat pouze impulsy pro levou ruku (při detekci pravé ruky je z většiny oblast pravé hemisféry vypnutá (hemisféry a končetiny jsou inverzně propojeny)). Nejlepší volbou proto zůstává C3 a C4.

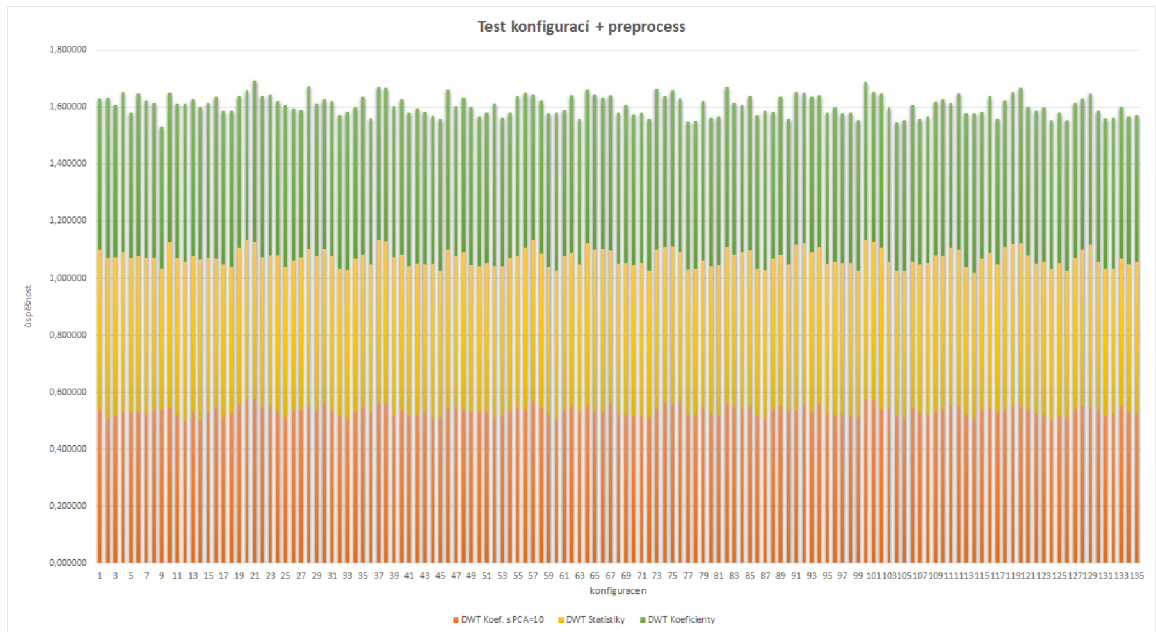


Obrázek 6.7: Kombinace elektrod

Průběh ukazuje, že klasifikace podle Sym9 není správná. Po změně Sym9 na Bior3.1 již vypadají výsledky klasifikace lépe [Obrázek 6.8] - zlepšení o 6 % oproti [Obrázek 6.2]



Obrázek 6.8: AUC 0.541

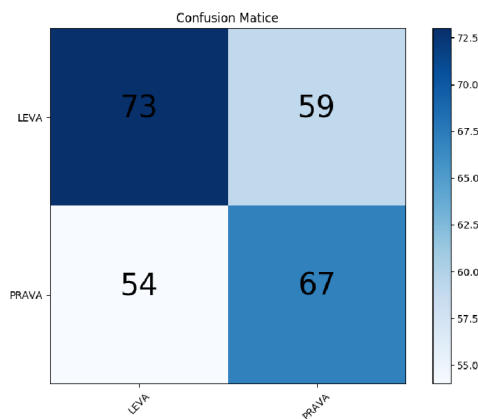


Obrázek 6.9: Konfigurace

Nejlepší úspěšnost při testu s koeficienty, statistikami a PCA [Obrázek 6.10] měla konfigurace 21 a 100 (číslo cyklu iterujícího přes kombinace uvedených parametrů), jejichž parametry jsou tyto:

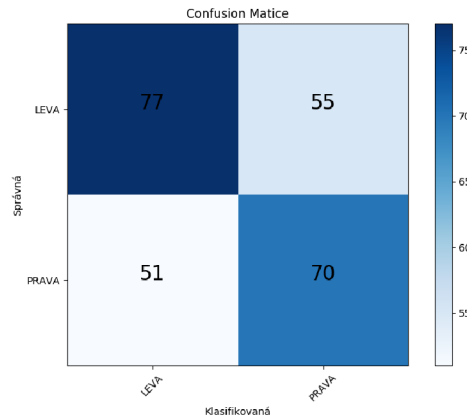
estimators = 100 a 1000 learning rate = 0.01 subsample = 0.2 max depth = 5 a 15

Je tedy vidět, že s parametry LR 0.01 a subsample 0.2 se dostane GBC k nejlepším výsledkům. Pokud nyní vyberu pouze D5 rozklady a aplikuji parametry:



Obrázek 6.10: GBC lr=0.01, est=100, depth=5, subsample=0.2 = AUC 0.55

Testy zjišťují, že při vypnutí všech parametrů (použití pouze samotných koeficientů) se dostávám na nejlepší úspěšnost při použití kombinace D4 a D5. AUC je v tomto případě 0.5809 a train score 0.82.



Obrázek 6.11: GBC lr=0.01, est=100, depth=5, subsample=0.2, koeficienty D4 a D5 = AUC 0.58

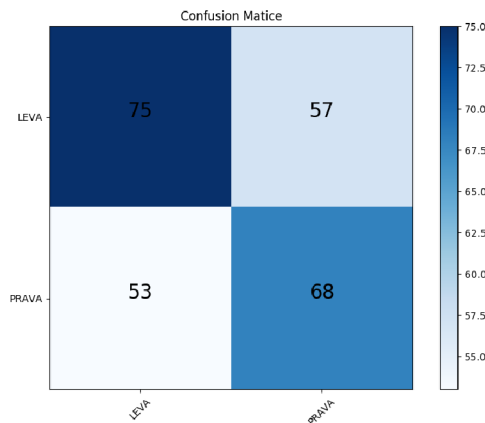
Vyzkouším další metodu klasifikace, Náhodné lesy.

6.2 Random Forests

Podobný postup jako u GBC jsem použil i u RF. Prvně testuji data D4-D5 pro celý segment (nevolím jednotlivé sekundy). Jako parametry RF používám, stejně jako u GBC, 1000 estimátorů a seed 48. Dostávám se k výsledku AUC 0.513, na první pokus horší než GBC ve stejném případě.

Zkouším použít pouze statistiky: výsledek AUC 0.536, nepřilíšná změna.

Úspěšnost 53 % stále není použitelná. Zkouším proto použít pouze rozklad D5 (7-15 Hz), na kterém se vyskytují Mu vlny [Sekce 2.2.3].



Obrázek 6.12: AUC 0.565

Výsledek naznačuje mírné zlepšení, a to na 56,5 %.

Testuji ještě wavelet koeficienty bez použití statistik. Dostávám se na AUC 0.57 %. S PCA=10 dostávám 0.56 %.

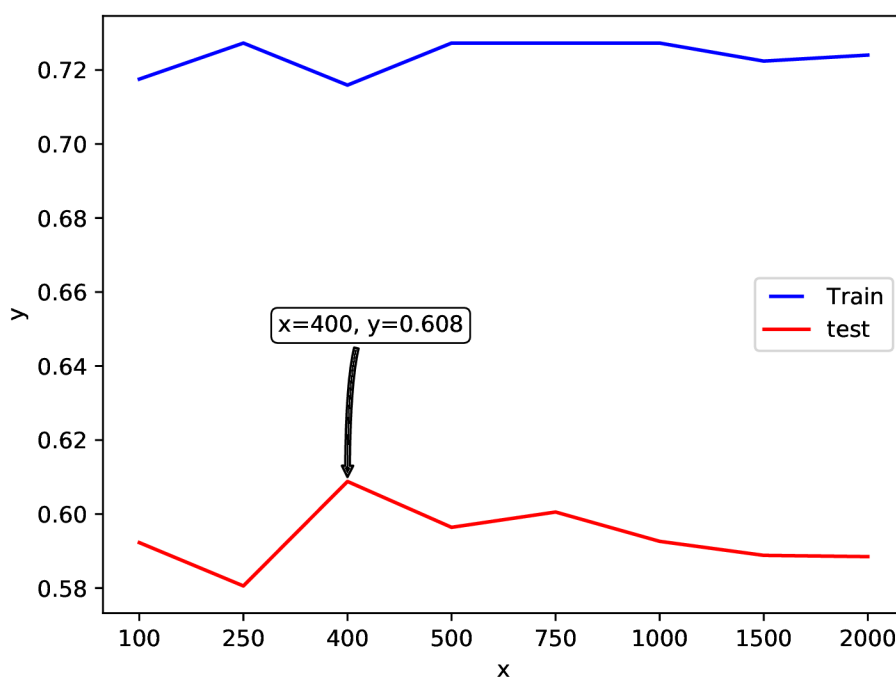
Používám funkci `RandomizedSearchCV`², která slouží k tzv. ladění hyperparametrů. Po zadání kýchžených parametrů pro vyladění začne prohledávat kombinace těchto parametrů a nalezne tu nejlepší pro předložený dataset. Po prohledání dochází optimizér k výsledku:

```
'min_samples_split': 5, 'min_samples_leaf': 20, 'n_estimators': 100
```

Dochází ke zlepšení AUC na 0.578 %.

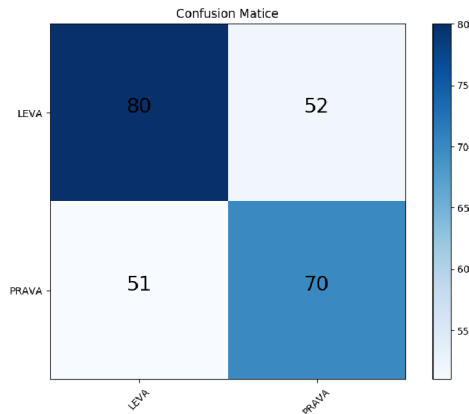
Po vrácení rozkladu D4, který obsahuje centrální Beta rytmy, (klasifikuji D4+D5) se klasifikační úspěšnost sníží na 0.561 %. Avšak po aplikaci ICA (komponenty=6) [36] se zvýší na 0.576 %. Pokud použiji pro vstup ICA před-zpracovaná data pomocí PCA=6, dostávám 0.592 %.

Pomocí skriptu procházím s touto konfigurací počet estimátorů/stromů [Obrázek 6.13]. Nejlepší se jeví být počet estimátorů=400.



Obrázek 6.13: x = počet estimátorů; y = úspěšnost klasifikace (na trénovací/testovací sadě)

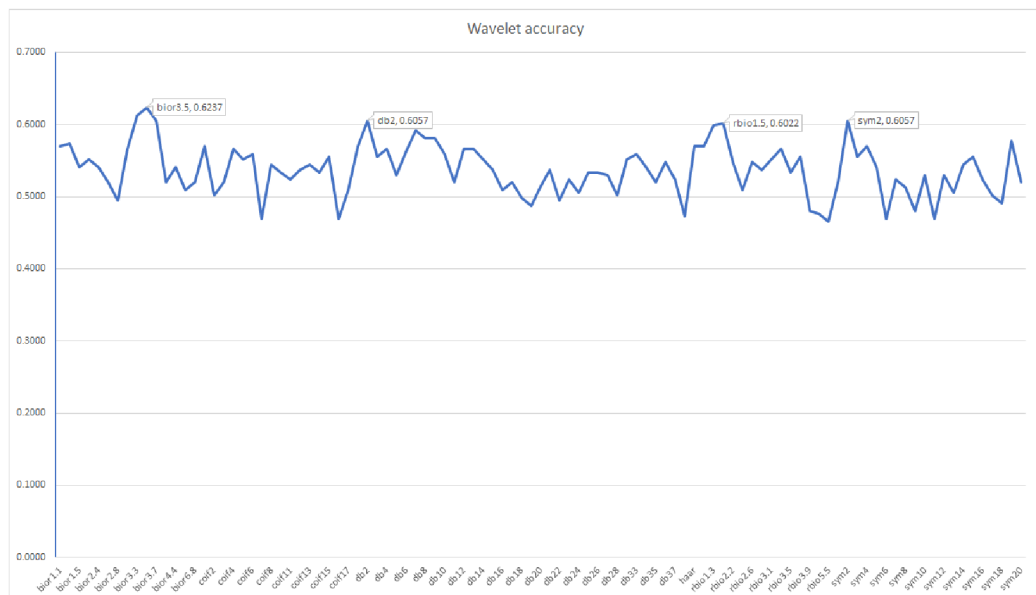
²https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV



Obrázek 6.14: AUC 0.592, PCA+ICA na DWT koeficientech D4 a D5

Zlepšení stále nejsou nijak razantní, zkusím proto upravenou testovací sadu. Dataset Martin je, stejně jako všechny ostatní datové sady v tabulce 3.1, „ořezán“ na prvních 11 sekund záznamu. Záznamy během výzev probíhaly v náhodném (uniform) rozložení od 10 do 15 sekund, ze záznamu je tedy potenciálně ořezána přibližně třetina dat.

Pro dataset MartinB jsem provedl testování RandomizedSearchCV, a vybírám nejlepší (metrika model.best_params_) - estimators = 390, depth=30, min samples split = 100, min samples leaf = 15 a spouštím hledání nejlepších wavelet transformací pro augmentovaný dataset MartinB. V tabulce s daty vizualizovanými do heatmapy 6.2 lze vidět, že nejúspěšnější jsou opět wavelety z rodiny Bior3.x (pro testování). Random_state jsem nechal náhodný. Na obrázku 6.15 jsou hodnoty vykresleny.



Obrázek 6.15: Všechny wavelet transformace

S kombinací nejlepších parametrů a DWT bior3.5 jsem provedl pokus s počtem estimatorů algoritmu RF ($n_estimators \leq 1000$). Seed pro promíchání datové sady, tak pro

Tabulka 6.2: Úspěšnost jednotlivých waveletů pro Random Forest

Wavelet	Train AUC	Test AUC	Wavelet	Train AUC	Test AUC
bior1.1	76.25	56.98	db20	84.94	51.25
bior1.3	77.75	57.34	db21	86.45	53.76
bior1.5	78.42	54.12	db22	86.62	49.46
bior2.2	76.75	55.19	db23	85.95	52.32
bior2.4	80.26	54.12	db24	82.94	50.53
bior2.6	81.93	51.97	db25	88.79	53.40
bior2.8	82.60	49.46	db26	87.29	53.40
bior3.1	77.09	56.63	db27	85.78	53.04
bior3.3	75.91	61.29	db28	86.12	50.17
bior3.5	77.92	62.36	db29	85.45	55.19
bior3.7	79.09	60.57	db33	83.77	55.91
bior3.9	82.10	51.97	db34	82.94	54.12
bior4.4	80.93	54.12	db35	82.60	51.97
bior5.5	81.60	50.89	db36	82.44	54.83
bior6.8	82.44	51.97	db37	79.76	52.32
coif1	79.26	56.98	db38	84.11	47.31
coif2	83.77	50.17	haar	76.25	56.98
coif3	89.63	51.97	rbio1.1	76.25	56.98
coif4	82.77	56.63	rbio1.3	77.92	59.85
coif5	83.44	55.19	rbio1.5	78.26	60.21
coif6	84.44	55.91	rbio2.2	80.93	54.83
coif7	84.28	46.95	rbio2.4	79.76	50.89
coif8	84.11	54.48	rbio2.6	81.60	54.83
coif9	81.10	53.40	rbio2.8	81.60	53.76
coif11	85.45	52.32	rbio3.1	79.76	55.19
coif12	84.94	53.76	rbio3.3	78.92	56.63
coif13	82.60	54.48	rbio3.5	81.43	53.40
coif14	83.77	53.40	rbio3.7	82.60	55.55
coif15	84.11	55.55	rbio3.9	83.27	48.02
coif16	83.77	46.95	rbio4.4	80.93	47.67
coif17	82.27	50.89	rbio5.5	83.27	46.59
db1	76.25	56.98	rbio6.8	81.27	51.97
db2	80.60	60.57	sym2	80.60	60.57
db3	78.09	55.55	sym3	78.09	55.55
db4	83.44	56.63	sym4	80.93	56.98
db5	81.77	53.04	sym5	80.93	54.12
db6	79.93	56.27	sym6	82.27	46.95
db7	85.45	59.13	sym7	87.45	52.32
db8	81.77	58.06	sym8	84.11	51.25
db9	81.93	58.06	sym9	78.26	48.02
db10	78.92	55.91	sym10	85.78	53.04
db11	80.76	51.97	sym11	80.93	46.95
db12	82.60	56.63	sym12	83.77	53.04
db13	79.09	56.63	sym13	81.93	50.53
db14	79.26	55.19	sym14	86.78	54.48
db15	83.77	53.76	sym15	80.43	55.55
db16	80.43	50.89	sym16	86.62	52.32
db17	79.26	51.97	sym17	81.10	50.17
db18	84.11	49.82	sym18	85.95	49.10
db19	86.12	48.74	sym19	85.78	57.70
			sym20	84.44	51.97

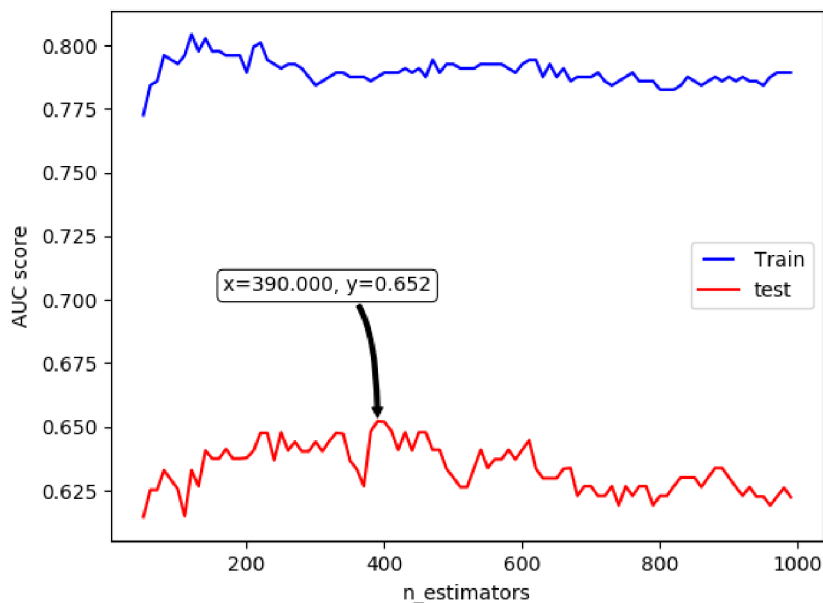
inicializaci náhodných stromů je zde náhodný. Maximální úspěšnost se tak proti předchozí změnila z důvodu rozdílné konfigurace (tyto experimenty běžely na výpočetním cloudu po dobu, kterou jsem věnoval ucelování parametrů. Fixní seed jsem začal používat po dokončení těchto experimentů). Na grafu 6.16 jsem použil hloubku 20, min split=100 a min leafs 15 [Sekce 5.3.1]. Modrá linie reprezentuje přesnost modelu na trénovací sadě, červená na testovací. Lze vidět, že trénování nedosahuje naprosté přesnosti, což může být znakem toho, že si model „nememoruje“ trénovací data (přetrénování) - může jít i o nedotrénování.

Na grafu 6.20 je vidět klesající trend, a úspěšnost testovací klasifikace zůstává stejná při 100 stromech, jako při 390 stromech, pouze úspěšnost na trénovací sadě značně klesla.

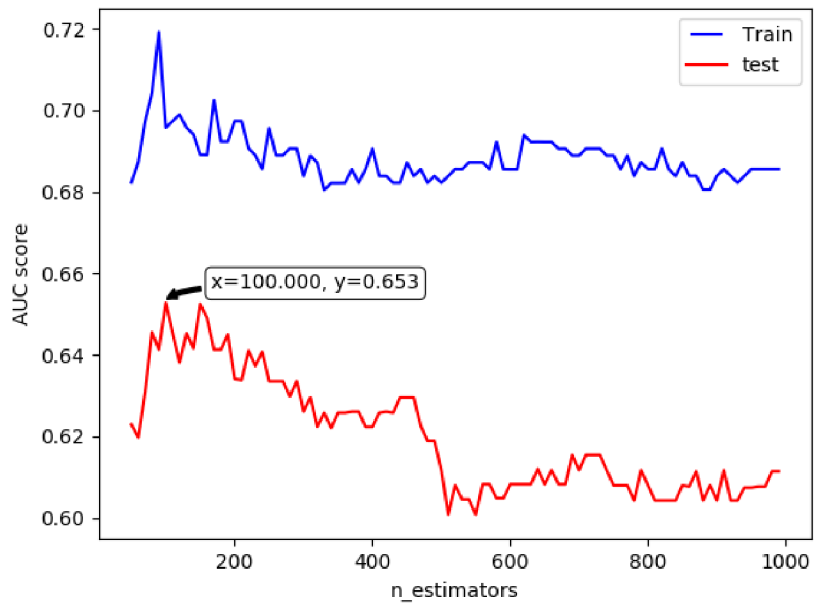
Graf 6.18 pak vykazuje zlepšení přesnosti u trénovací sady, přičemž se při přítomných parametrech testování příliš nezměnila úspěšnost klasifikace na testovací sadě.

Na grafu 6.19 jsem vyzkoušel ještě jednu konfiguraci, u té však poklesla přesnost testovací a trénovací úspěšnost se vyhoupla na 100 %, což je známkou přetrénování [Sekce 5.3.1].

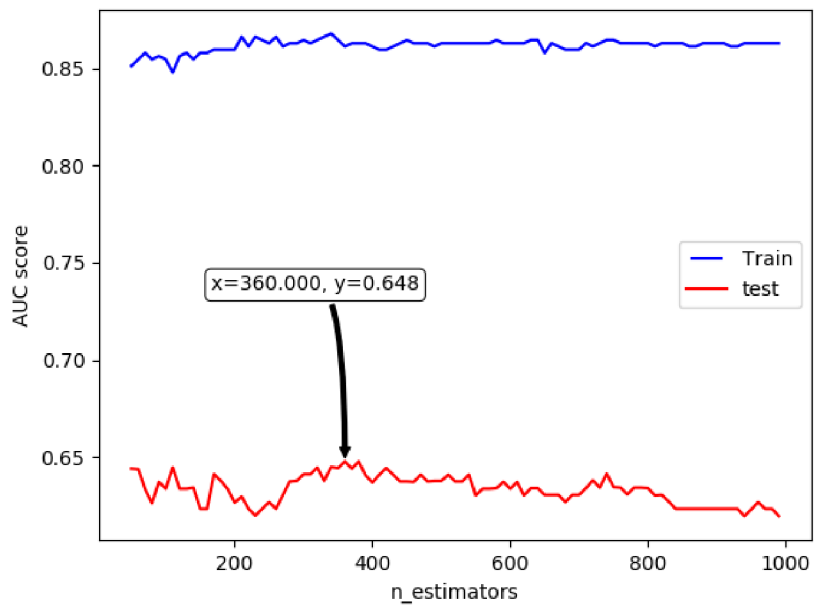
S Random Forests se tedy dostávám k průměrné úspěšnosti klasifikace 60 %. Otestuji ještě jednu klasifikační metodu - Podpůrné vektory.



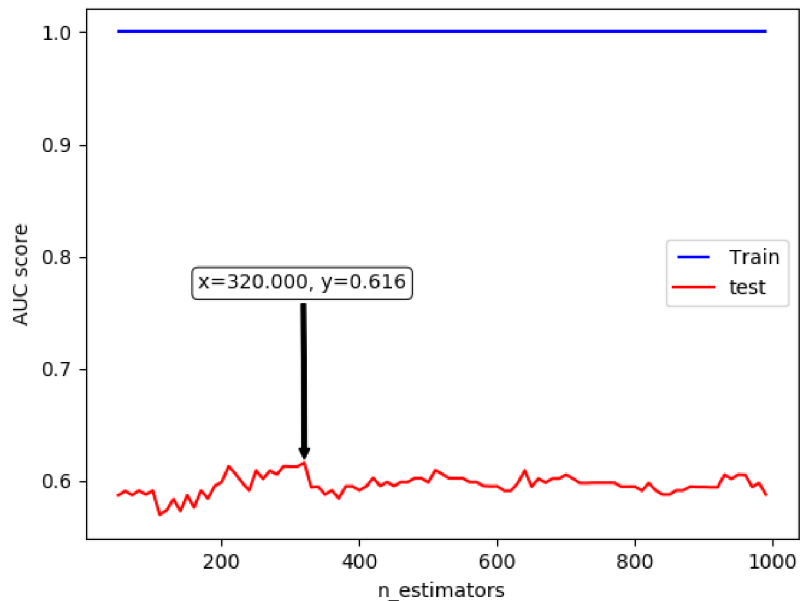
Obrázek 6.16: RF s Parametry: depth=20, min split=100, min leafs = 15, x=estimators, y=AUC testu



Obrázek 6.17: Parametry: depth=20, min split=200, min leafs = 20



Obrázek 6.18: Parametry: depth=30, min split=100, min leafs = 20



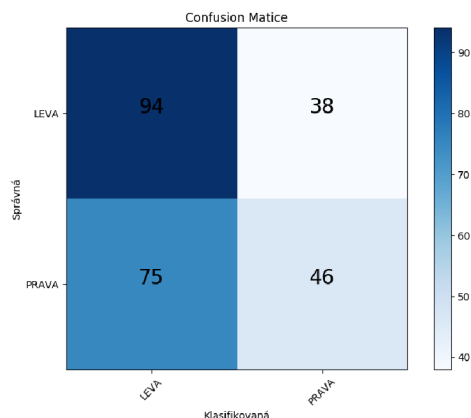
Obrázek 6.19: Parametry: depth=50, min split=5, min leafs = 5

6.3 Support Vector Machines

Support Vector Machines byly úspěšně použity např. ve výzkumu [25]. Autor zde klasifikuje data po sekundových oknech, používá implementaci libsvm.

Před další experimentací provádím GridSearchCV, pro nalezení nejlepších parametrů pro SVM. Hledám parametry gamma, degree a C. Grid Search určil nejlepší parametry pro dataset Martin jako: 'gamma': 0.1, 'degree': 3, 'C': 0.1

S těmito parametry a RBF kernelem je úspěšnost klasifikace dat z D4 a D5 (bez PCA, ICA, Aproximačních částí nebo statistik) = AUC 0.546 %. Při použití pouze dat z D5 se úspěšnost sníží na 0.5282 %. Při použití A5,A4 a D5,D4 je úspěšnost 0.563 %. Po použití PCA=6 se dostávám na horší výsledek, 0.551 % (a také velmi nevyvážený - 0.95 % pro třídu Levá a 0.15 % pro Pravá).

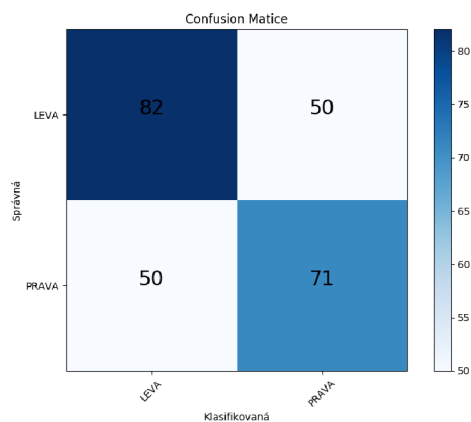


Obrázek 6.20: Parametry: 'gamma': 0.1, 'degree': 3, 'C': 0.1

Po vyzkoušení dalších parametrů pomocí skriptu (všechny kombinace PCA, ICA, statistik, rozložení D3-D5, A3-A5) se dostávám k nejlepšímu výsledku 0.55 %.

Na cloudu METAVO jsem spustil hledání parametrů pro implementaci SVM NuSVC, která umožňuje specifikovat počet podpůrných vektorů. Číslo Nu vyjadřuje zlomek (počet chyb při trénování)/(počet podpůrných vektorů).

Vyzkoušel jsem opět kombinace parametrů pro hledání Nu (v případě Nu 0.05 - 0.95 po kroku 0.05) a nejlepšího výsledku dosáhl nastavení Nu = 0.8 pro PCA aplikované na statistiky rozložení D4+D5, a to AUC 0.6039 (train score: 0.67). Druhé nejlepší nastavení je s Nu = 0.7, s ním se dostávám na AUC 0.5922 (train score: 0.76).



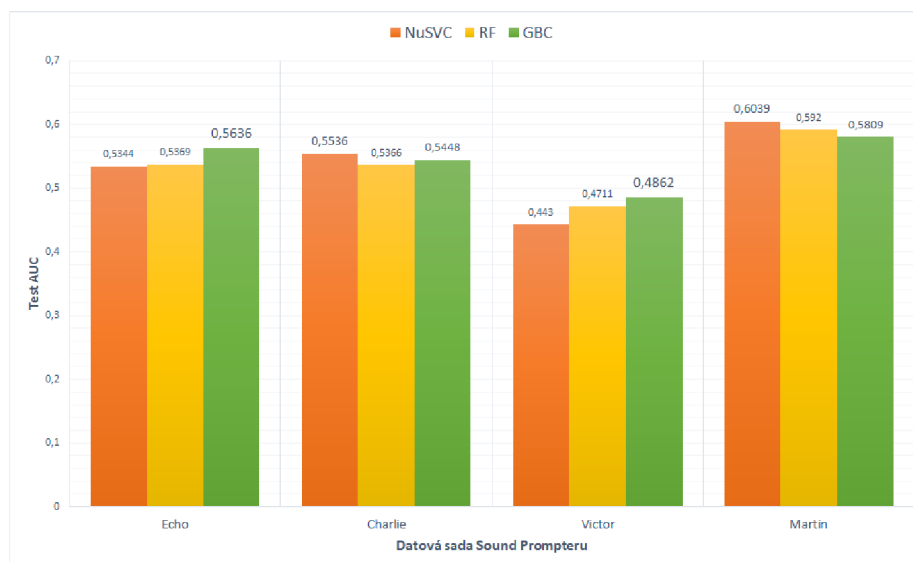
Obrázek 6.21: NuSVC s Nu=0.8, PCA a statistikami. Rozložení D4 a D5 (AUC 60 %)

6.4 Testy zvukových datasetů

Model s parametry nalezenými na datech ze Sound Prompter, kde musí mít uživatel zavřené oči, by dle předpokladu měl vykazovat menší úspěšnost na testovacích datech z Visual Prompter. To jsem úspěšně potvrdil při provedení testů na vizuálních datových sadách Foxtrot, Dominika a Jan, kde se úspěšnost nedostala přes 50 %.

Pro vizuální datové sady vykazují modely natrénované na pouze zvukových datech menší úspěšnost. Další v řadě jsou zvukové datové sady, kde by měla být úspěšnost vyšší, jelikož podmínky při trénování jsou zde pro testovací subjekty stejné.

Na obrázku 6.22 lze vidět úspěšnost jednotlivých algoritmů pro zvukové datové sady, a původní trénovací sadu, Martin. Random Forests byl použit, stejně jako ostatní klasifikátory, v konfiguraci, ve které dosahoval nejvyšší úspěšnosti.



Obrázek 6.22: Výsledky nejlepších verzí algoritmů na datových sadách + bázové

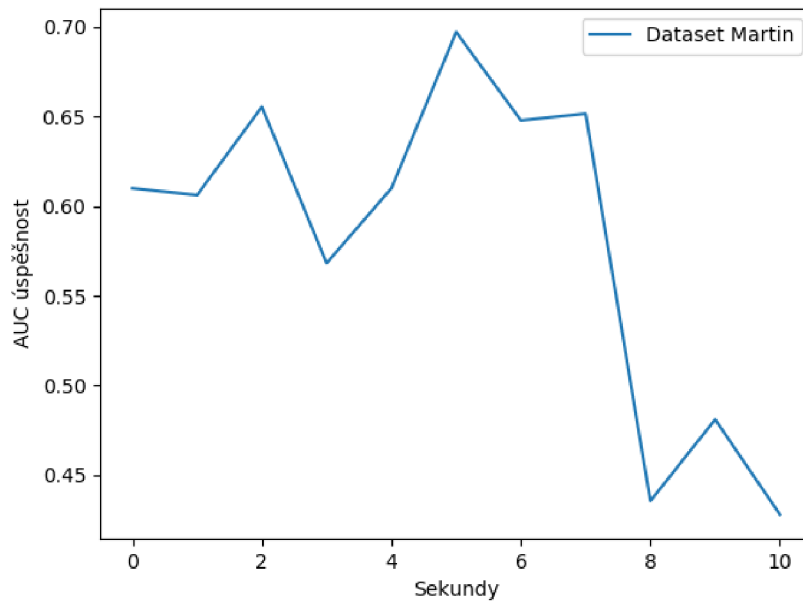
6.5 Výsledky

Na datasetu Martin jsem provedl rozsáhlé trénování různorodých modelů, jako jsou GBC, RF, SVC a NuSVC.

Pomocí funkcionality testovacího frameworku jsem po prvotním manuálním testování provedl hledání parametrů, které dosahují nejúspěšnějších výsledků. Použil jsem i metody GridSearchCV a RandomizedSearchCV z frameworku scikit.learn. Z důvodu výpočetní náročnosti hledání parametrů jsem používal výpočetní cloud METAVO. Po prohledání různých stavů kombinací parametrů jsem došel k následujícím výsledkům.

SVC, NuSVC - dosahuje nejlepších výsledků (60.3 %) při nastavení poměru podpůrných vektorů vůči počtu chyb na 0.8. Používá parametry D4+D5 (4. a 5. detailní rozklad reprezentující frekvenční pásmo 15-30 Hz (Beta) a 7-15 (Alfa, Mu)). Z metod redukce dimenzionality používá statistiky a na nich aplikované PCA.

GBC - Dosáhl nejlepších výsledků taktéž pro koeficienty D4 a D5, ale bez použití jakýchkoliv metod redukce dimenzionality. Parametry, co pomohly GBC klasifikovat s co nejvyšší přesností, jsou: počet estimátorů 100, rychlost učení 0.01, subsampling 0.2 a maximální hloubka průchodu 5. GBC tak dosahuje přesnosti 58.1 %.



Obrázek 6.23: GBC klasifikátor aplikovaný na segmenty po sekundách - dosahuje až k 70 %

RF - Náhodné stromy vykazují nejpřesnější výsledky také při použití koeficientů D4 a D5, zde s aplikovanou redukcí PCA a ICA (6 komponent). Dosahoval nejlepší výsledků při nastavení `n_estimator=400`, `min_samples_split=5` a `min_samples_leaf 20`. Nejvyšší dosažená přesnost pro RF je 59.2 %.

U výsledků testů na zvukových datasetech lze vidět pokles úspěšnosti klasifikace v datasetu Victor pod úroveň náhody, 50 %. To může být zapříčiněno tím, že nahrávaná data byla již několikrát pokus ten den a s únavou přišlo i zhoršení pozornosti.

6.6 Diskuze a srovnání

Provedl jsem testování rozličných parametrů, metod a klasifikátorů. Při porovnání datasetů Echo, Martin a Victor lze vidět, že při nahrávání dat může záležet na schopnosti soustředit se na úkon (předloženou výzvu) a také na schopnosti představit si pohyb (činnost, která je pro nesoustředěnou mysl obtížná). K lepší klasifikaci může také přispívat fyzická zdatnost testovaného subjektu, která přispívá ke schopnosti lépe si představit pohyb (záleží i na jejich míře koncentrace). Ke zlepšení výsledků by mohla přispět delší kalibrace testovacího subjektu před zahájením testovacího sezení - důkladnější seznámení s hardwarem, hlubší popis funkčnosti aplikace a popsat scénář při vyvstanuvších komplikacích (uživatel si představoval špatnou stranu atp.). Použití hry NeuroBuddha pro prvotní kalibraci uživatele a jeho seznámení s funkčností aplikace se jeví jako nejsnazší cesta.

Při pouhé detekci otevřených a zavřených očí si vedly všechny klasifikátory skvěle, GBC, RF i SVC dosáhly úspěšnosti přes 94 % i se základními parametry. Pokles napětí je totiž při zavřených očích velice zřetelný (Obrázek 2.2) a klasifikace je tak snadnější.

Testování bylo provedeno na datasetech ze Sound Prompter, tedy se zavřenýma očima. Dle očekávání proto není míra úspěšné klasifikace na datech z Visual Prompter (otevřené oči) lepší než náhoda.

Jako nejlepší DWT funkce se v testech s GBC a RF projeví ty z Biortogonální rodiny, konkrétně Bior3.x. Pro testování jsem použil Bior3.1. Wavelet Sym9, který jsem v počátečních testech použil na základě výzkumu používajícího jej pro detekci epileptických signatur, se neprokázal být efektivním pro tento problém.

Každý klasifikátor fungoval nejlépe s rozdílnými parametry. Ty byly z velké části nalezeny pomocí experimentace. U RF to byla kombinace metod pro redukci dimenzionality PCA a ICA, u NuSVC pak statistické údaje o wavelet koeficientech a na ně aplikovaná PCA, a u GBC nebylo nutné data připravit jinak, než pomocí rozkladu na wavelet koeficienty.

Při aplikování natrénovaného modelu na sekundová okna se dostávám na úspěšnost klasifikace od 60 % do 95 % (Tabulka 6.1, Obrázek 6.23), kde stabilně klasifikuje Bior3.1 a to s maximální úspěšností 70 %.

Pro ovládání her pomocí představování zvedání levé a pravé ruky lze využít klasifikátory po natrénování na větším vzorku dat, jelikož experimenty probíhaly primárně na datové sadě Martin. Zde jsem potvrdil, že úspěšnost detekce závisí na časovém segmentu, na kterém klasifikace probíhá - nejsilnější byly u Martina impulsy pro levou a pravou ruku na 5. sekundě. Proto lze usuzovat, že po delším učení a sbírání trénovacích dat se úspěšnost klasifikace dostane přes 70 %.

V porovnání s podobnými výzkumy jsem se dostal na přibližně stejné, a někdy i lepší výsledky.

Ve výzkumu [24] byl použit headset Enobio s 32 elektrodami, tedy 4x větším počtem, než zde použitého UCIV. Autor použil CSP filtry a OpenVIBE vestavěné klasifikátory. Úspěšnost jeho klasifikace je 42-64 %. V mém případě se dostávám na 59 % při průměrné klasifikaci a na 70 % v nejlepším případě, závislém na schopnostech testovacích subjektů soustředěně si představovat pohyb.

Studie [43] použila OpenBCI headset a výzkumníci klasifikovali s úspěšností 77 % mezi stavem relaxace a představováním pohybu pravé ruky, tedy problémem mírně odlišným od zde zkoumaného rozlišení mezi signaturami levého a pravého pohybu v mozku.

Práce [16] používá hardware OpenBCI a OpenVIBE software. Výzkumník experimentoval s pozicemi C3, Cz a C4 a dostal se na úspěšnost 53,4 %. V tomto případě dosahují lepších výsledků.

V jedné práci byl použit klasifikátor SVM [25] s úspěšností 90 % na segmenty testovacích dat. Výzkumník dosáhl s DWT Sym5 na úspěšnost 90% na segmentu cca. 5. sekundy, jeho metrika pro přesnost je však pouze True Positive, do vzorce nezapočítává False Positives, které jsou při reálné použitelnosti rozhodující. Já dosahuji na 70 % s Bior3.1 s metrikou AUC.

Kapitola 7

Závěr

V této diplomové práci jsem nastudoval projekt OpenBCI, problematiku EEG a snímání mozkové aktivity a zprovoznil headset OpenBCI Ultracortex IV pro použití na testování, sběr dat a ovládání jednoduché hry. Zhodnotil jsem existující výzkumy a zanalyzoval možnosti použití EEG pro ovládání her.

Vytvořil jsem nástroje Visual Prompter a Sound Prompter pro vytvoření vlastní datové sady. Těmito nástroji jsem získal data od více testovacích subjektů, celkem jsem nasbíral 12 datových sad. Tyto nástroje dále použili Petr Ullrich a Zdeněk Jelínek pro svoje bakalářské práce.

Nasbíraná data jsem analyzoval a po nastudování technik pro extrakci rysů a parametrů z EEG signálu jsem zvolil wavelet transformace. Po transformaci vstupních dat jsem použil ke klasifikaci model Gradient Boosting, Random Forests a Support Vector Machines.

Dále jsem vyvinul testovací framework, který umožňuje hledání parametrů pro co nej- přesnější klasifikaci testovací sady. Experimentálně získané výsledky jsem vyhodnotil a porovnal s existujícími pracemi. K výpočtům jsem použil výpočetní cloud Metacentrum METAVO, který mi značně urychlil výzkum.

V Unity frameworku jsem vyvinul na neurofeedbacku založenou hru NeuroBuddha, kterou je možné ovládat pouhou soustředěnou pozorností. Tu jsem prezentoval na konferenci ExcelFit a použil ji pro kalibraci testovacích subjektů před zahájením testovacího sezení.

Před začátkem testování jsem ověřil klasifikaci na detekci otevřených/zavřených očí, kdy jsem dosáhl úspěšnosti 97 % s GBC. Testy jsem dělal s DWT Bior3.1 z biortogonální rodiny, které jsem zvolil na základě jeho dobrých výsledků. Při klasifikaci EEG signatur úmyslu pro zvednutí levé a pravé ruky jsem se dostal k průměrné přesnosti klasifikace 58-60 % (metrika AUC) s klasifikátory GBC, RF a NuSVC. Pro klasifikaci jednotlivých sekund jsem dosáhl na přesnost 70 %.

Vzhledem k tomu, že u souvisejících výzkumů, se kterými jsem provedl srovnání, si vede OpenBCI Ultracortex IV velmi podobně, usuzuji, že přesnost klasifikace bude z velké míry ovlivněna právě UCIV headsetem a kvalitou provedení jeho hardware komponent, které nevykazují takovou přesnost, jako hardware pro neurologický výzkum. Vše je však vyváženo mnohonásobně menší cenou, která tyto nedostatky kompenzuje.

Výzkum a metody, které jsem zde prezentoval, mohou být dále využity pro použití na OpenBCI UCIV headsetu a k ovládání jednoduchých her, jako je tomu např. u neurofeedback hry NeuroBuddha. Po lepším zaškolení testovacích subjektů se dá teoreticky dojít k lepším výsledkům klasifikace, a tak by bylo ovládání pomocí EEG signatur úmyslu zvednout levou/pravou ruku spolehlivější.

Kapitola 8

Přílohy

Všechny zdrojové soubory jsou k dispozici v mém github repositáři https://github.com/janjilecek/DP_files. Jedná se pouze o zdrojové soubory, kromě Sound Prompteru nebou skripty bez souborů přiložených na DVD fungovat.

Seznam souborů na DVD disku:

- složka Datasetsy - obsahuje veškeré vytvořené datové sady
- Framework - testovací framework a generátor souborů
- NeuroBuddha - obsahuje spustitelnou hru a zdrojové kódy
- Sound Prompter - zvukový prompter
- Visual Prompter - visuální prompter
- video.mp4 - 30s video pro prezentaci na monitoru UPGM
- Požadavky a dokumentace.txt - návod ke spuštění
- SendLSL.py - skript pro mock EEG LSL
- zdrojové LaTeX soubory v zipu
- xjilec00.pdf - text této práce

Literatura

- [1] Brain.
URL <https://media.buzzle.com/media/images-en/photos/human-biology/brain/1200-71487927-labelled-human-brain.jpg>
- [2] Brain-Controlled Game Helps Kids With ADHD Improve Mental Focus - IEEE Spectrum.
URL <https://spectrum.ieee.org/the-human-os/biomedical/devices/brain-controlled-game-helps-kids-with-adhd-improve-mental-focus>
- [3] buddha.
URL https://choosemuse.com/app/uploads/2018/10/Buddha_What-It-Measures_Muse-Harley-Oliver.jpg
- [4] choosemuse.
URL <https://choosemuse.com>
- [5] Dongle.
URL <http://cdn.autodidacts.io/img/openbci/openbci-and-bluetooth-dongle-running--adjusted.png>
- [6] Emotivv.
URL <https://imotions.com/wp-content/uploads/2016/08/Emotiv-EPOC-EEG.png>
- [7] Enobio.
URL https://www.neurolite.ch/sites/default/files/styles/product/public/products/StarStim_2.png?itok=2uMtH43A
- [8] jadra.
URL
https://scikit-learn.org/stable/_images/sphx_glr_plot_iris_svc_0011.png
- [9] Mgr. Pavla Linhartová: Projekt fMRI neurofeedback je prvním svého druhu v České republice - Fakultní nemocnice Brno.
URL <https://www.fnbrno.cz/mgr-pavla-linhartova-projekt-fmri-neurofeedback-je-prvnim-sveho-druhu-v-ceske-republice/t6183>
- [10] motorImagery.
URL <http://openbci.com/community/openbci-crossing-swords-with-motor-imagery/>
- [11] neurosky.
URL

<https://d2drzakx2pq6fl.cloudfront.net/production/products/17716/large/neurosky-mindwave-mobile-eeg-sensor-w-slash-m-1784489348.jpg?1526552420>

- [12] PFC.
URL <https://i1.wp.com/www.thescienceofpsychotherapy.com/wp-content/uploads/2017/11/Screen-Shot-2017-11-18-at-1.49.10-PM.png?ssl=1>
- [13] PŘEDNÁŠKA KURZU MPOV. Technická zpráva.
URL http://midas.uamt.feec.vutbr.cz/POV/Lectures/09_Klasifikace_a_automaticke_trideni.pdf
- [14] roc auc.
URL <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>
- [15] Spiderclaw.
URL <https://openbci.com/headware/>
- [16] Agarwal, M.; Sivakumar, R.: THINK : Toward Practical General-Purpose Brain-Computer Communication *. doi:10.1145/2799650.2799659.
URL <http://dx.doi.org/10.1145/2799650.2799659>.
- [17] Al-Qazzaz, N. K.; Bin Mohd Ali, S. H.; Ahmad, S. A.; aj.: Selection of Mother Wavelet Functions for Multi-Channel EEG Signal Analysis during a Working Memory Task. *Sensors (Basel, Switzerland)*, ročník 15, č. 11, 11 2015: s. 29015–29035, ISSN 1424-8220 (Electronic), doi:10.3390/s151129015.
- [18] Alfaro, A.: Package 'adabag' Type Package Title Applies Multiclass AdaBoost.M1, SAMME and Bagging. Technická zpráva, 2018.
URL <https://cran.r-project.org/web/packages/adabag/adabag.pdf>
- [19] Bazanova, O. M.; Auer, T.; Sapina, E. A.: On the Efficiency of Individualized Theta/Beta Ratio Neurofeedback Combined with Forehead EMG Training in ADHD Children. *Frontiers in human neuroscience*, ročník 12, 2018: str. 3, ISSN 1662-5161 (Print), doi:10.3389/fnhum.2018.00003.
- [20] Benjamin Estermann: (PDF) Evaluation of Low-Cost EEG Hardware in a Motor Imagery Based Brain Computer Interface.
URL https://www.researchgate.net/publication/327797416_Evaluation_of_Low-Cost_EEG_Hardware_in_a_Motor_Imagery_Based_Brain_Computer_Interface
- [21] Bhattacharyya, S.; Khasnobish, A.; Konar, A.; aj.: Performance analysis of left/right hand movement classification from EEG signal by intelligent algorithms. 2011, doi:10.1109/CCMB.2011.5952111.
- [22] Bose, S.; Rama, V.; Warangal, N.; aj.: EEG signal analysis for Seizure detection using Discrete Wavelet Transform and Random Forest. In *2017 International Conference on Computer and Applications (ICCA)*, IEEE, 9 2017, ISBN 978-1-5386-2752-5, s. 369–378, doi:10.1109/COMAPP.2017.8079760.
URL <http://ieeexplore.ieee.org/document/8079760/>

- [23] Fatima, M.; Shafique, M.; Khan, Z.: Towards a Low Cost Brain-computer Interface for Real time Control of a 2 DOF Robotic Arm. 2015, doi:10.1109/ICET.2015.7389206.
- [24] Hlinka, M.: Mental Imagery based Brain-Computer Interface used in a simple Computer Game. 2017.
URL <https://is.muni.cz/th/ti88m>
- [25] Ines, H.; Slim, Y.; Noureddine, E.: EEG classification using support vector machine. In *10th International Multi-Conferences on Systems, Signals Devices 2013 (SSD13)*, 2013, s. 1–4, doi:10.1109/SSD.2013.6564011.
- [26] Jileček, J.: Rozpoznávání emocí pomocí konvolučních neuronových sítí. 2015.
- [27] Khasnobish, A.: K-Nearest Neighbor Classification of Left-Right Limb Movement Using EEG Data.
URL https://www.researchgate.net/publication/268206014_K-Nearest_Neighbor_Classification_of_Left-Right_Limb_Movement_Using_EEG_Data
- [28] Krishnaveni, V.; Jayaraman, S.; Anitha, L.; aj.: Removal of ocular artifacts from EEG using adaptive thresholding of wavelet coefficients. *Journal of Neural Engineering*, ročník 3, č. 4, 12 2006: s. 338–346, ISSN 1741-2560, doi:10.1088/1741-2560/3/4/011.
URL <http://www.ncbi.nlm.nih.gov/pubmed/17124338http://stacks.iop.org/1741-2552/3/i=4/a=011?key=crossref.b3bf41218b3130070cf895e15115d45c>
- [29] Krucoff, M. O.; Rahimpour, S.; Slutzky, M. W.; aj.: Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation. *Frontiers in neuroscience*, ročník 10, 2016: str. 584, ISSN 1662-4548, doi:10.3389/fnins.2016.00584.
URL <http://www.ncbi.nlm.nih.gov/pubmed/28082858http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5186786>
- [30] Leins, U.; Goth, G.; Hinterberger, T.; aj.: Neurofeedback for Children with ADHD: A Comparison of SCP and Theta/Beta Protocols. *Applied Psychophysiology and Biofeedback*, ročník 32, č. 2, 6 2007: s. 73–88, ISSN 1090-0586, doi:10.1007/s10484-007-9031-0.
URL <http://www.ncbi.nlm.nih.gov/pubmed/17356905http://link.springer.com/10.1007/s10484-007-9031-0>
- [31] Lévesque, J.; Beauregard, M.; Mensour, B.: Effect of neurofeedback training on the neural substrates of selective attention in children with attention-deficit/hyperactivity disorder: A functional magnetic resonance imaging study. *Neuroscience Letters*, ročník 394, č. 3, 2 2006: s. 216–221, ISSN 03043940, doi:10.1016/j.neulet.2005.10.100.
URL <http://www.ncbi.nlm.nih.gov/pubmed/16343769http://linkinghub.elsevier.com/retrieve/pii/S0304394005012036>
- [32] Li, X.; Chen, X.; Yan, Y.; aj.: Classification of EEG signals using a multiple kernel learning support vector machine. *Sensors (Basel, Switzerland)*, ročník 14, č. 7, 7 2014: s. 12784–12802, ISSN 1424-8220 (Electronic), doi:10.3390/s140712784.

- [33] Linden, M.; Habib, T.; Radojevic, V.: A controlled study of the effects of EEG biofeedback on cognition and behavior of children with attention deficit disorder and learning disabilities. *Biofeedback and self-regulation*, ročník 21, č. 1, 3 1996: s. 35–49. URL <http://www.ncbi.nlm.nih.gov/pubmed/8833315>
- [34] Lotte, F.; Bougrain, L.; Cichocki, A.; aj.: Computer Interfaces: A 10-year Update. *Journal of Neural Engineering*, 2018: str. 55, doi:10.1088/1741-2552/aab2f2{"i"}. URL <https://hal.inria.fr/hal-01846433>
- [35] Marzbani, H.; Marateb, H. R.; Mansourian, M.: Neurofeedback: A Comprehensive Review on System Design, Methodology and Clinical Applications. *Basic and clinical neuroscience*, ročník 7, č. 2, 4 2016: s. 143–158, ISSN 2008-126X (Print), doi:10.15412/J.BCN.03070208.
- [36] Naeem, M.; Brunner, C.; Pfurtscheller, G.: Dimensionality reduction and channel selection of motor imagery electroencephalographic data. *Computational intelligence and neuroscience*, ročník 2009, 2009: str. 537504, ISSN 1687-5273, doi:10.1155/2009/537504. URL <http://www.ncbi.nlm.nih.gov/pubmed/19536346><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2695957>
- [37] Oades, R. D.: Dopamine-serotonin interactions in attention-deficit hyperactivity disorder (ADHD). In *Progress in Brain Research*, ročník 172, Elsevier, 2008, ISBN 9780444532350, s. 543–565, doi:10.1016/S0079-6123(08)00926-6. URL <http://www.ncbi.nlm.nih.gov/pubmed/18772050><https://linkinghub.elsevier.com/retrieve/pii/S0079612308009266>
- [38] Pfurtscheller, G.; Brunner, C.; Schlögl, A.; aj.: Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks. *NeuroImage*, ročník 31, č. 1, 5 2006: s. 153–159, ISSN 1053-8119, doi:10.1016/J.NEUROIMAGE.2005.12.003. URL <https://www.sciencedirect.com/science/article/pii/S1053811905025140?via%3Dihub>
- [39] Suryotrisongko, H.; Samopa, F.: Evaluating OpenBCI Spiderclaw V1 Headwear's Electrodes Placements for Brain-Computer Interface (BCI) Motor Imagery Application. *Procedia Computer Science*, ročník 72, 2015: s. 398–405, ISSN 18770509, doi:10.1016/j.procs.2015.12.155. URL <https://linkinghub.elsevier.com/retrieve/pii/S1877050915036169>
- [40] Ulloa, E. R.; Pineda, J. A.: Recognition of point-light biological motion: Mu rhythms and mirror neuron activity. *Behavioural Brain Research*, ročník 183, č. 2, 11 2007: s. 188–194, ISSN 0166-4328, doi:10.1016/J.BBR.2007.06.007. URL <https://www.sciencedirect.com/science/article/pii/S0166432807003154?via%3Dihub>
- [41] Wang, A. I.; Larsen, E. A.: Using Brain-Computer Interfaces in an Interactive Multimedia Application. 2013, doi:10.2316/p.2012.790-046. URL <http://www.idi.ntnu.no/~alfw/publications/sea2012-bci-snake.pdf>
- [42] Wang, X.; Gong, G.; Li, N.: Automated Recognition of Epileptic EEG States Using a Combination of Symlet Wavelet Processing, Gradient Boosting Machine, and Grid

Search Optimizer. *Sensors*, ročník 19, č. 2, 2019, ISSN 1424-8220, doi:10.3390/s19020219.

- [43] Yohanandan, S. A.; Kiral-Kornek, I.; Tang, J.; aj.: A Robust Low-Cost EEG Motor Imagery-Based Brain-Computer Interface. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 7 2018, ISBN 978-1-5386-3646-6, s. 5089–5092, doi:10.1109/EMBC.2018.8513429. URL <https://ieeexplore.ieee.org/document/8513429/>