

# Analiza podataka u interakciji mozak-računalo tehnikama strojnog učenja

---

**Mustać, Mateo**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:211:117170>

*Rights / Prava:* [Attribution-NonCommercial 3.0 Unported / Imenovanje-Nekomercijalno 3.0](#)

*Download date / Datum preuzimanja:* **2025-03-13**



*Repository / Repozitorij:*

[Faculty of Organization and Informatics - Digital Repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN

Mateo Mustać

ANALIZA PODATAKA U INTERAKCIJI  
MOZAK-RAČUNALO TEHNIKAMA  
STROJNOG UČENJA

DIPLOMSKI RAD

Varaždin, 2024.

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Mateo Mustać**

**JMBAG: 0016138017**

**Studij: Informacijsko i Programsko inženjerstvo**

**ANALIZA PODATAKA U INTERAKCIJI MOZAK-RAČUNALO**  
**TEHNIKAMA STROJNOG UČENJA**

**DIPLOMSKI RAD**

**Mentor/Mentorica:**

Dr. sc. Božidar Kliček

**Varaždin, rujan 2024.**

Mateo Mustać

### **Izjava o izvornosti**

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor/Autorica potvrdio prihvaćanjem odredbi u sustavu FOI-radovi*

---

## Sažetak

U ovom diplomskom radu prikazana je analiza sučelja mozak-računalo izrađena preko algoritama strojnog učenja u programskom jeziku Python. Na samom početku rada objasniti ću samu povijest BCI, kako je započelo učitavanje putem EEG-a, kada je prvi put korišten naziv „sučelje mozak-računalo“. Zatim ću objasniti tehnologije koje se koriste u BCI danas, kakve su naravi, njihove prednosti i nedostatke. Potom ću objasniti metode strojnog učenja, koje vrste postoje, te svaku od njih posebno definirati i navesti prednosti i nedostatke za svaku, kao i prednosti i nedostatke u cjelini samog strojnog učenja. Primjena BCI-a je danas sve veća pa tako ću i objasniti primjenu u medicinskim i nemedicinskim svrhama. Na kraju rada objasniti će se napravljena analiza putem svakog algoritma posebno, kako se implementira u programskom jeziku i što dobiveni grafovi znače.

**Ključne riječi:** BCI; EEG; HCI; Analiza EEG signala; Strojno učenje; Tehnike obrade EEG signala; Primjena BCI tehnologija; Algoritmi strojnog učenja

## Sadržaj

1. Uvod .....	1
2. Metode i tehnike rada .....	2
3. Povijest sučelja mozak-računalo .....	3
3.1. Početci razvoja sučelja mozak-računalo.....	3
3.2. Postignuća u BCI razvoju .....	4
3.3. Evolucija sučelja mozak-računalo .....	4
4. Trenutne tehnologije u sučeljima mozak-računalo .....	6
4.1. Invazivni i neinvazivni BCI sustavi.....	6
4.2. Nosivi i implantabilni BCI sustavi.....	6
4.3. Strojno učenje i umjetna inteligencija u BCI sustavima.....	7
4.4. Hardverske komponente .....	7
4.4.1. EEG Uređaji .....	7
4.4.2. Neuralni implantat .....	8
4.4.3. Alati za akviziciju signala.....	8
4.5. Softverske komponente .....	9
4.5.1. Tehnike obrade signala .....	9
4.5.2. Metode ekstrakcije značajki.....	9
4.5.3. Klasifikacijski algoritmi.....	9
4.5.4. Duboko učenje u BCI .....	10
4.5.5. Napredak u neinvazivnim BCI sustavima .....	11
4.5.6. Studije slučaja suvremenih BCI sustava.....	11
5. Strojno učenje.....	12
5.1. Vrste strojnog učenja .....	12
5.1.1. Nadzirano strojno učenje.....	13
5.1.2. Strojno učenje bez nadzora.....	14
5.1.3. Polu-nadzirano strojno učenje .....	15

5.1.4. Učenje s pojačanjem .....	16
5.2. Prednosti strojnog učenja.....	17
5.3. Nedostatci strojnog učenja .....	18
6. Primjene BCI tehnologije .....	20
6.1. Primjene u medicini i rehabilitaciji.....	20
6.1.1. Neuroprotetika.....	20
6.1.2. Komunikacijski alati za paralizu.....	21
6.1.3. Kognitivna rehabilitacija.....	22
6.2. Nemedicinske primjene .....	23
6.2.1. Igre i zabava .....	24
6.2.2. Pametna okruženja .....	25
6.2.3. Pобољшanja interakcije čovjek-računalo .....	26
7. Analiza EEG skupa motoričkih pokreta .....	28
7.1. Preuzimanje podataka .....	28
7.2. Pregled skupa podataka .....	28
7.2.1. Prikupljanje podataka .....	29
7.2.2. Primjene.....	29
7.3. Format i struktura podataka .....	30
7.4. Pretprocesiranje podataka .....	30
7.4.1. Filtriranje podataka.....	33
7.4.2. Uklanjanje artefakata.....	34
7.4.3. Ponovno referenciranje .....	35
7.4.4. Epochiranje i detekcija događaja.....	35
7.4.5. Korekcija osnovne linije .....	36
7.4.6. Ekstrakcija značajki .....	36
7.5. Odabir značajki i klasifikacija.....	37
7.5.1. Odabir značajki .....	37
7.5.2. Klasifikacija .....	38
7.6. Analiza rezultata .....	44

7.6.1. IPMS Hipoteza .....	45
7.6.2. Konfuzijska matrica .....	45
7.6.3. Rezultati .....	46
7.6.4. Podrška vektorima (SVM) .....	48
7.6.5. Nasumične šume (RF) .....	48
7.6.6. Neuronske mreže (NN) .....	49
7.6.7. Stablo odlučivanja (DT) .....	50
7.6.8. Linearna diskriminantna analiza (LDA) .....	51
8. Zaključak .....	53
Popis literature .....	54
Popis slika .....	58
Popis tablica .....	59



# 1. Uvod

Sučelje mozak-računalo brzo je rastuće područje koje spaja neuroznanost, kognitivne znanosti i strojno učenje kako bi se uspostavio izravan komunikacijski kanal između ljudskog mozga i vanjskih uređaja. BCI sustavi posebno su korisni u pomaganju osobama s invaliditetom, omogućujući im upravljanje strojevima isključivo pomoću moždanih signala. Ova interakcija ovisi o točnom tumačenju i analizi moždane aktivnosti, koja se najčešće bilježi putem elektroencefalograma (EEG). EEG signali predstavljaju električnu aktivnost mozga i ključni su za detekciju različitih kognitivnih stanja, pokreta ili namjera.

Posljednjih godina tehnike strojnog učenja značajno su unaprijedile područje BCI-ja, poboljšavajući analizu i interpretaciju EEG podataka. Složenost i visoka dimenzionalnost EEG signala čine ručnu analizu izazovnom. Primjenom algoritama strojnog učenja, poput strojeva vektorske potpore (SVM), slučajnih šuma (Random Forests) i dubokog učenja, istraživači mogu učinkovitije i preciznije dekodirati moždane signale. Ovi algoritmi prepoznaju obrasce u EEG podacima koji odgovaraju različitim mentalnim ili motoričkim aktivnostima, čime se poboljšava komunikacija između mozga i vanjskih uređaja.

Rad je podijeljen na sedam dijelova: uvod (koji govori o samoj temi rada, navodi najvažnije izvore, te samu strukturu rada), metode i tehnike rada (gdje se spominju alati izrade, te proces izrade rada), povijest sučelja mozak-računalo (gdje se obrađuje sama misao sučelja mozak računalo i kako ona nastala), trenutne tehnologije sučelja mozak-računalo (gdje se opisuju trenutne vrste tehnologija i objašnjava se svaka od njih i njihove prednosti i nedostaci), strojno učenje (u kojemu se navode vrste strojnog učenja, one se opisuju i navode se prednosti i nedostaci), primjena sučelja mozak-računalo (u kojem se navodi primjena BCI u medicinske i nemedicinske svrhe), praktični dio rada (u kojem se objašnjava analiza rezultata dobivenih provedbom testiranja modela EEG skupa podataka).

## 2. Metode i tehnike rada

Program korišten za izradu praktičnog dijela ovog rada je Python programski jezik koji je važan alat u analizi skupova podataka sučelja mozak-računalo. On pruža obradu EEG podataka. Za učitavanje, analizu i obradu EEG podataka korišten je paket *pyedflib*, koji omogućuje rad s .edf datotekama. Korištene su metode strojnog učenja za klasifikaciju mentalnih stanja na temelju EEG signala spremjenih u .edf datoteke. Korišteni su algoritmi strojnog učenja: Potporni Vektorski Strojevi (SVM), Nasumične šume (RF), Neuronske mreže (NN), Stablo Odlučivanja (DT) i Linearna Diskriminantna Analiza (LDA). Korištenjem biblioteke *matplotlib*, izrađene su vizualizacije matrice konfuzije, krivulje učenja, stabla odlučivanja i scatter plot dijagrama. Implementirane su tehnike uklanjanja artefakata, filtriranja podataka.

Nakon što sam odabrao jedan od mnogih online BCI skupova podataka za analizu on je sadržavao EEG signale spremjene u .edf datoteke za 109 volontera i mnogo snimki za svakog.

Prilikom izrade analize koristeći se pythonom prvi put sam se susreo s implementacijom algoritama strojnog učenja. To je bila prva prepreka koju sam svladao proučavajući službenu Python dokumentaciju vezanu uz navedene algoritme. Kako se radi o velikom skupu podataka uzeo sam reprezentativni uzorak od 10 volontera i dvije snimke za svakog. Vizualizacija pojedinih algoritama predstavljala je svoje probleme, u prikazu ili formatu dobivenih filtriranih podataka, pa su neke vizualizacije smanjene kako nije bilo moguće dobiti grafikon na temelju trenutnih podataka.

Teoretski dio ovog rada je napravljen koristeći se različitim izvorima s interneta kao što su članci, online dokumenti, knjige. Korišten je alat Microsoft Word.

### 3. Povijest sučelja mozak-računalo

Sučelje mozak-računalo (eng. Brain-Computer interface, BCI) je suradnja između mozga i uređaja koji signalima iz mozga omogućuje usmjeravanje neke vanjske aktivnosti, kao što je kontrola pokazivača ili protetskog uda. Sučelje omogućuje izravnu komunikaciju između mozga i objekta kojim se upravlja. U slučaju kontrole kursora, na primjer, signal se prenosi izravno iz mozga do mehanizma koji usmjerava kursor, umjesto normalnim putem kroz tjelesni neuromuskularni sustav od mozga do prsta na mišu [1].

Očitavanjem signala iz niza neurona i korištenjem računalnih čipova i programa za prevođenje signala u akciju, BCI može omogućiti osobi koja pati od paralize da napiše knjigu ili upravlja motoriziranim invalidskim kolicima ili protetskim udovima samo putem misli. Trenutačni uređaji moždanog sučelja zahtijevaju promišljenu svjesnu misao; neke buduće primjene, kao što je protetska kontrola, vjerojatno će raditi bez napora. Jedan od najvećih izazova u razvoju BCI tehnologije bio je razvoj elektrodnih uređaja i kirurških metoda koje su minimalno invazivne. U tradicionalnom BCI modelu, mozak prihvaća ugrađeni mehanički uređaj i upravlja njime kao prirodnim dijelom svoje reprezentacije tijela. Velik broj trenutačnih istraživanja usmjeren je na potencijal neinvazivnog BCI-a [1].

#### 3.1. Početci razvoja sučelja mozak-računalo

Razvoj tehnologije sučelja mozak-računalo rezultat je desetljeća istraživanja koja spajaju neurologiju, inženjering i računalne znanosti. Povijest BCI-a započinje 1924. godine kada je njemački psihijatar Hans Berger prvi put zabilježio električnu aktivnost ljudskog mozga pomoću elektroencefalografije (eng. Electroencephalography, EEG). Ova revolucionarna tehnologija omogućila je analizu oscilatornih aktivnosti mozga, poput alfa valova, što je postavilo temelje za budući razvoj BCI sustava [2][3].

EEG je ubrzo postao ključan alat za dijagnostiku neuroloških poremećaja i istraživanje moždanih funkcija. U 1960-ima, španjolski neurofiziolog José M. Delgado razvio je prvi implantabilni uređaj za stimulaciju mozga, tzv. "stimoceiver", koji je omogućio preciznu kontrolu emocija i ponašanja kod životinja i ljudi. Iako su Delgadovi eksperimenti izazvali etičke kontroverze, njegovo istraživanje otvorilo je put daljnjem razvoju implantabilnih BCI uređaja [3].

Godine 1965. američki skladatelj Alvin Lucier prvi je upotrijebio EEG podatke za kontrolu glazbenih instrumenata, čime je demonstrirao da se moždana aktivnost može koristiti za upravljanje strojevima. Njegovo istraživanje predstavljalo je prvi korak prema konceptu BCI-

a kao sredstva za interakciju između mozga i vanjskih uređaja. Nekoliko godina kasnije, 1969., američki neuroznanstvenik Eberhard Fetz pokazao je da majmun može naučiti kontrolirati iglu na mjeracu isključivo svojom neuronskom aktivnošću. Ovo otkriće označilo je početak razvoja dvosmjernih BCI sustava, koji omogućuju povratnu informaciju između mozga i stroja [2][3].

## 3.2. Postignuća u BCI razvoju

Pojam "sučelje mozak-računalo" prvi je put upotrijebio Jacques J. Vidal 1973. godine u svojoj znanstvenoj publikaciji koja je detaljno opisala potencijale i izazove ovih sustava. Vidal je demonstrirao kako se vizualni evocirani potencijali mogu koristiti za pomicanje objekata na računalnom ekranu, čime je utemeljio osnove za današnje BCI sustave [3].

Tijekom 1970-ih i 1980-ih, brojni neuroznanstvenici nastavili su istraživati mogućnosti BCI-a, primjenjujući ovu tehnologiju za obnovu izgubljenih osjetila vida, sluha i pokreta. Godine 1978., američki istraživač William Doherty ugradio je BCI uređaj u vizualni korteks slijepa osobe, čime je djelomično obnovio pacijentov vid. Ovaj zahvat označio je važnu prekretnicu u razvoju medicinskih aplikacija BCI-a [2].

## 3.3. Evolucija sučelja mozak-računalo

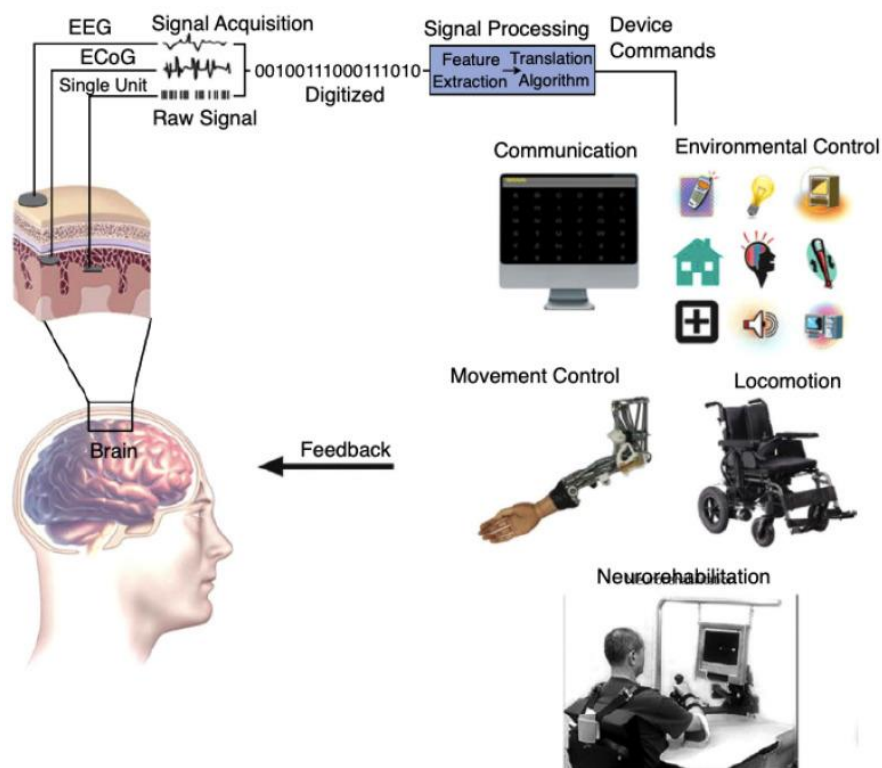
Danas, sučelja mozak-računalo doživjela su značajan napredak zahvaljujući razvoju neinvazivnih implantata poput Stentrodea, koji omogućuju sigurno i učinkovito korištenje BCI tehnologije u kliničkoj praksi. Na primjer, istraživači sa Sveučilišta Stanford razvili su BCI model koji omogućuje dekodiranje mentalnog pisanja, pomažući pacijentima s invaliditetom u komunikaciji. Ova tehnologija također se koristi za sintetiziranje govora kod osoba koje su izgubile sposobnost govora zbog paralize ili drugih neuroloških poremećaja [2][3].

Na Europskoj izložbi istraživanja i inovacija u Parizu u lipnju 2006., američki znanstvenik Peter Brunner sastavio je poruku jednostavno se koncentrirajući na zaslon. Brunner je nosio tijesno pripijenu (ali potpuno vanjsku) kapu opremljenu brojnim elektrodama. EEG aktivnost iz Brunnerova mozga bila je pokupljena elektrodama na kapici i informacijama korištenim, zajedno sa softverom, za identifikaciju specifičnih slova ili znakova za poruku [1].

Pokazani BCI Brunner-a temelji se na metodi koja se naziva Wadsworthov sustav. Kao i druge BCI tehnologije temeljene na EEG-u, Wadsworthov sustav koristi prilagodljive algoritme i tehnike usklađivanja uzoraka kako bi olakšao komunikaciju. Od korisnika i softvera očekuje se prilagodba i učenje, čineći proces učinkovitijim s vježbom. Tijekom prezentacije prikazana je poruka američkog neurobiologa koji sustav koristi za nastavak rada, unatoč tome

što boluje od amiotrofične lateralne skleroze (Lou Gehrigova bolest). Iako znanstvenik više ne može pomaknuti čak ni oči, uspio je poslati sljedeću e-poruku: "Ja sam neuroznanstvenik koji (sic) ne bi mogao raditi bez BCI-ja. Ovo pišem svojim EEG-om zahvaljujući Wadsworth-u Centru za istraživanje sučelja mozak-računalo" [1].

Povijest sučelja mozak-računalo jasno pokazuje da je riječ o multidisciplinarnom polju koje se neprestano razvija, kombinirajući napredak u neuroznanosti, računalnim znanostima i biomedicinskom inženjeringu. Danas, BCI tehnologija ne samo da pomaže u medicinskoj rehabilitaciji, već otvara nove mogućnosti za interakciju čovjeka i stroja, čineći budućnost interakcije s tehnologijom sve više nalik onome što je nekoć bila znanstvena fantastika.



Slika 1. Komponente BCI sustava i njegove komunikacijske metode - pojednostavljen prikaz (Izvor: National Library of Medicine, 2021)

Izvor:

[https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop\\_pmc/tileshop\\_pmc\\_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=7824107\\_brainsci-11-00043-g001.jpg](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=7824107_brainsci-11-00043-g001.jpg)

## 4. Trenutne tehnologije u sučeljima mozak-računalo

Područje sučelja mozak-računalo (BCI) doživjelo je značajan napredak posljednjih godina, potaknuto tehnološkim inovacijama i konvergencijom različitih znanstvenih disciplina. BCI tehnologije, koje uspostavljaju izravan komunikacijski kanal između mozga i vanjskih uređaja, značajno su se razvile, omogućujući širok raspon primjena, od medicinske rehabilitacije do kognitivnog poboljšanja.

### 4.1. Invazivni i neinvazivni BCI sustavi

Jedna od najvažnijih podjela u aktualnim BCI tehnologijama odnosi se na razliku između invazivnih i neinvazivnih sustava. Invazivni BCI uključuju implantaciju elektroda izravno u mozak, pružajući visokorezolucijske podatke i robusnu jasnoću signala. Ovi sustavi se primarno koriste u medicinskim primjenama, kao što je pomoć osobama s teškim neurološkim poremećajima. Primjeri uključuju uređaje za duboku stimulaciju mozga (eng. Deep Brain Stimulation, DBS) koji se koriste za liječenje Parkinsonove bolesti i drugih poremećaja pokreta. Ove tehnologije pokazale su značajnu učinkovitost, ali dolaze s visokim kirurškim rizicima i etičkim razmatranjima [5].

S druge strane, neinvazivni BCI sustavi oslanjaju se na vanjske senzore postavljene na vlasište, najčešće korištenjem EEG-a. Ovi sustavi su manje rizični i šire primjenjivi, iako općenito nude nižu kvalitetu signala i podložniji su šumovima. Nedavni napredak poboljšao je točnost i funkcionalnost neinvazivnih BCI sustava, čineći ih prikladnijima za primjene kao što su neuroproteze i komunikacijski uređaji za osobe s invaliditetom. Istražuju se novi pristupi, poput funkcionalne blisko infracrvene spektroskopije (eng. Functional Near-Infrared Spectroscopy, fNIRS) i magnetoencefalografije (eng. Magnetoencephalography, MEG), kako bi se povećale sposobnosti neinvazivnih BCI sustava [5].

### 4.2. Nosivi i implantabilni BCI sustavi

Trend prema nosivoj tehnologiji značajno je utjecao na razvoj BCI tehnologija. Nosivi BCI sustavi dizajnirani su za praktičnost i upotrebljivost, integrirajući se bez problema u svakodnevni život. Ovi uređaji su uglavnom neinvazivni i koriste se u primjenama od igara do praćenja mentalnog zdravlja. Na primjer, nosivi EEG uređaji sada mogu pratiti kognitivna stanja i pružati povratne informacije za upravljanje stresom ili poboljšanje fokusa. Iako nude manje detaljne podatke o moždanim signalima od implantabilnih uređaja, njihova

jednostavnost korištenja i neinvazivna priroda čine ih vrlo atraktivnima za potrošačke primjene. [4]

Implantabilni BCI sustavi ostaju na čelu najnovijih istraživanja zbog svoje sposobnosti pristupa dubljim strukturama mozga i pružanja podataka visoke preciznosti. Nedavne inovacije uključuju razvoj minimalno invazivnih tehnika, poput mikro elektrodnih mreža i optogenetike, koje imaju za cilj smanjenje invazivnosti ovih sustava uz održavanje visoke kvalitete podataka. Ove tehnologije obećavaju buduće primjene u područjima kao što su poboljšanje pamćenja i liječenje psihijatrijskih stanja [5].

### **4.3. Strojno učenje i umjetna inteligencija u BCI sustavima**

Strojno učenje (eng. Machine learning, ML) i umjetna inteligencija (eng. Artificial Intelligence, AI) igraju sve važniju ulogu u razvoju BCI sustava. Ove tehnologije ključne su za interpretaciju složenih neuronskih signala koje BCI sustavi prikupljaju, pretvarajući ih u naredbe koje se mogu koristiti za upravljanje vanjskim uređajima. Napredni ML algoritmi, kao što je duboko učenje, koriste se za poboljšanje točnosti i brzine obrade signala, omogućujući primjene u stvarnom vremenu. Na primjer, BCI sustavi pokretani AI tehnologijom razvijaju se za pomoć u neurorehabilitaciji prilagođavanjem neuronskim obrascima korisnika tijekom vremena, povećavajući učinkovitost tretmana [9].

Nadalje, integracija AI s BCI sustavima otvara put za personalizirane i adaptivne sustave. Ovi inteligentni BCI sustavi mogu učiti iz interakcija s korisnicima, kontinuirano poboljšavajući svoju izvedbu i smanjujući vrijeme obuke potrebno za učinkovito korištenje. Ova prilagodljivost ključna je za razvoj BCI sustava koji mogu zadovoljiti specifične potrebe korisnika s različitim neurološkim stanjima [9].

### **4.4. Hardverske komponente**

Tehnologije sučelja mozak-računalo mogu biti hardverske naravi ili softverske. Hardverske komponente u području sučelja mozak-računalo uvelike su pomogle u rastu i razvoju BCI tehnologije.

#### **4.4.1. EEG Uređaji**

EEG uređaji predstavljaju najčešće korištenu tehnologiju u neinvazivnim BCI sustavima. EEG uređaji mjere električnu aktivnost mozga putem elektroda postavljenih na vlasište, pružajući ključne podatke za analizu neuronskih signala. Glavne prednosti EEG

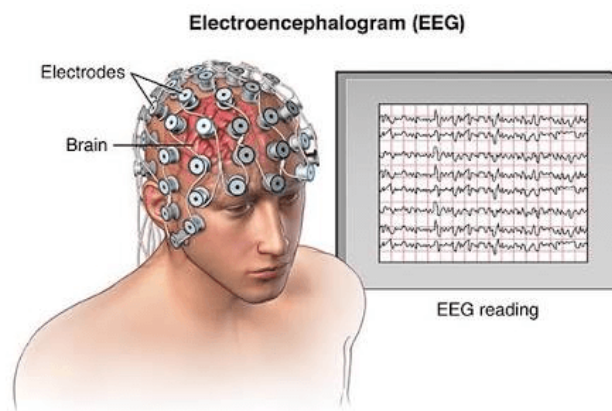
tehnologije uključuju relativno nisku cijenu, sigurnost i mogućnost primjene u stvarnom vremenu. Međutim, zbog toga što se signali prikupljaju izvan lubanje, oni su osjetljivi na šum i smetnje, što može utjecati na točnost rezultata . Nedavni napreci uključuju razvoj prijenosnih EEG uređaja koji omogućuju korištenje BCI sustava izvan laboratorijskog okruženja, čime se širi njihova primjena u svakodnevnom životu i zdravstvenoj skrbi [6].

#### 4.4.2. Neuralni implantat

Neuralni implantati su invazivni BCI uređaji koji se implantiraju izravno u mozak kako bi zabilježili neuronsku aktivnost visoke rezolucije. Ovi implantati pružaju preciznije podatke u usporedbi s neinvazivnim metodama, ali dolaze s većim rizikom i složenijom kirurškom procedurom. Neuralni implantati se obično koriste u medicinskim istraživanjima i za liječenje teških neuroloških stanja, kao što su epilepsija i Parkinsonova bolest . Iako nude superiornu kvalitetu signala, izazovi povezani s implantatima uključuju rizik od infekcije, odbacivanje uređaja i dugoročno održavanje [7].

#### 4.4.3. Alati za akviziciju signala

Alati za akviziciju signala ključni su za prikupljanje, obradu i prijenos podataka iz mozga u računalni sustav. Ovi alati uključuju senzore, pojačala i digitalne konvertere koji pretvaraju analogne neuronske signale u digitalne podatke za daljnju analizu. Visoka kvaliteta akvizicije signala presudna je za preciznost BCI sustava, jer bilo kakav gubitak ili izobličenje podataka može značajno utjecati na performanse sustava . Trenutni napredci u ovoj oblasti usmjereni su na poboljšanje osjetljivosti senzora i smanjenje veličine i težine uređaja kako bi se olakšala njihova upotreba u stvarnom svijetu [8].



Slika 2. EEG uređaj i njegova očitavanja. (Izvor: East Neurology)

Izvor: <https://eastneurology.com.au/eeg-electroencephalogram-brain-wave-tests/>



## 4.5. Softverske komponente

Danas, softverske komponente su sve više zastupljene i pomažu u rastu i razvoju sučelja mozak-računalo. Softverske komponente koriste algoritme strojnog učenja za brzu i pouzdanu obradu povratnih podataka u informacije, ovisno radi li se o medicinskoj ili nemedicinskoj uporabi.

### 4.5.1. Tehnike obrade signala

Tehnike obrade signala igraju ključnu ulogu u pretvaranju sirovih neuronskih podataka u korisne informacije za BCI sustave. Ove tehnike uključuju filtriranje šuma, pojačanje signala i izolaciju specifičnih obrazaca moždane aktivnosti. Kvalitetna obrada signala omogućuje precizniju interpretaciju neuronskih podataka i povećava točnost sustava u prepoznavanju korisnikovih namjera . Nedavne inovacije uključuju korištenje algoritama strojnog učenja za automatsko poboljšanje kvalitete signala i prilagodbu specifičnim korisničkim potrebama [12].

### 4.5.2. Metode ekstrakcije značajki

Metode ekstrakcije značajki usmjerene su na izdvajanje relevantnih informacija iz obrađenih neuronskih signala. Ove značajke mogu uključivati frekvencijske komponente, amplitude signala i vremenske uzorke koji su ključni za klasifikaciju različitih mentalnih stanja ili namjera korisnika. Ekstrakcija značajki je kritična za smanjenje složenosti podataka i poboljšanje učinkovitosti algoritama za klasifikaciju . Trenutno se istražuju nove metode koje koriste duboko učenje kako bi se automatski identificirale najvažnije značajke iz složenih neuronskih podataka [12].

### 4.5.3. Klasifikacijski algoritmi

Algoritmi za klasifikaciju koriste se za interpretaciju značajki iz neuronskih podataka i njihovo prevođenje u konkretne naredbe unutar BCI sustava. Uobičajeni algoritmi uključuju linearne diskriminantne analize (LDA), potporne vektorske strojeve (SVM) i neuronske mreže. Ovi algoritmi igraju ključnu ulogu u točnosti BCI sustava jer omogućuju brzo i precizno prepoznavanje korisnikovih namjera . Nedavni napredci u korištenju dubokih neuronskih mreža omogućili su značajna poboljšanja u performansama i prilagodljivosti BCI sustava [12].

#### 4.5.3.1. Linearna Diskriminantna Analiza (LDA)

LDA je jedan od najpopularnijih klasifikacijskih metoda korištenih u BCI-ima zbog svoje jednostavnosti i učinkovitosti. Funkcionira pronalaženjem linearne kombinacije značajki koja

najbolje razdvaja različite klase (npr. različita mentalna stanja). LDA je posebno učinkovita u slučajevima kada su podaci linearno odvojivi, a klase imaju slične kovarijacijske matrice. Međutim, njezina izvedba može se pogoršati kada se radi o nelinearnim podacima ili kada su prekršene pretpostavke normalnosti [12].

#### **4.5.3.2. Potporni Vektorski Strojevi (SVM)**

SVM-ovi su široko korišteni u BCI aplikacijama zbog svoje sposobnosti obrade visokodimenzionalnih podataka i nelinearnih odnosa. Mapiranjem ulaznih značajki u prostor veće dimenzije pomoću kernela, SVM-ovi mogu učinkovito razdvojiti klase koje nisu linearno odvojive u izvornom prostoru značajki. Ovo ih čini posebno korisnima u BCI-ima, gdje moždani signali često pokazuju složene, nelinearne obrasce [12].

#### **4.5.3.3. Umjetne Neuronske Mreže (ANN)**

ANN-i, a posebno duboke neuronske mreže (DNN), privukli su značajnu pažnju u BCI istraživanjima zbog svoje sposobnosti modeliranja složenih, nelinearnih odnosa u podacima. ANN-i se sastoje od više slojeva međusobno povezanih neurona koji zajedno rade na učenju obrazaca u ulaznim podacima. U kontekstu BCI-a, ANN-i su uspješno korišteni za zadatke poput klasifikacije motoričke imaginacije i prepoznavanja emocija. Glavni izazov s ANN-ima leži u potrebi za velikim količinama označenih podataka za treniranje, kao i njihovoj računalnoj složenosti [12].

### **4.5.4. Duboko učenje u BCI**

Duboko učenje, podskup strojnog učenja, revolucioniralo je mnoga područja, uključujući BCI-e. Modeli dubokog učenja, poput konvolucijskih neuronskih mreža (CNN) i rekurentnih neuronskih mreža (RNN), pokazali su veliki potencijal u poboljšanju performansi BCI sustava.

#### **4.5.4.1. Konvolucijske Neuronske Mreže (CNN)**

Konvolucijske neuronske mreže su posebno pogodne za obradu prostorno strukturiranih podataka, poput slika. U BCI-ima, CNN-i su primijenjeni na analizu EEG podataka, gdje automatski uče prostorne filtre koji mogu izdvojiti relevantne značajke iz moždanih signala. To smanjuje potrebu za ručnom ekstrakcijom značajki i omogućuje precizniju klasifikaciju kognitivnih stanja. CNN-i su uspješno korišteni u zadacima poput klasifikacije motoričke imaginacije i prepoznavanja emocija, nadmašujući tradicionalne pristupe strojnog učenja [13].

#### **4.5.4.2. Rekurentne Neuronske Mreže (RNN)**

Rekurentne neuronske mreže su dizajnirane za obradu sekvencijalnih podataka, što ih čini idealnim za analizu vremenskih serija poput EEG signala. RNN-i mogu uhvatiti vremenske ovisnosti u podacima, što je ključno za točno tumačenje moždanih signala koji se razvijaju tijekom vremena. Long short-term memory (LSTM) mreže, vrsta RNN-a, pokazale su se posebno učinkovitima u BCI aplikacijama jer mogu zadržati informacije tijekom duljih vremenskih razdoblja, omogućujući modeliranje složenih vremenskih obrazaca u moždanoj aktivnosti [13].

#### **4.5.5. Napredak u neinvazivnim BCI sustavima**

Napredak u neinvazivnim BCI sustavima omogućio je širu primjenu ovih tehnologija u svakodnevnom životu. Glavne prednosti neinvazivnih sustava uključuju njihovu sigurnost, lakoću korištenja i relativno nisku cijenu, što ih čini pristupačnijima za širu populaciju. Međutim, izazovi poput niske razlučivosti signala i osjetljivosti na šum i dalje ograničavaju njihove mogućnosti. Unatoč tome, kontinuirani razvoj novih tehnologija poput funkcionalne infracrvene spektroskopije (fNIRS) i poboljšanih EEG uređaja, obećava značajno poboljšanje u točnosti i pouzdanosti neinvazivnih BCI sustava [12].

#### **4.5.6. Studije slučaja suvremenih BCI sustava**

Studije slučaja suvremenih BCI sustava pružaju dragocjen uvid u stvarne primjene i performanse ovih tehnologija. Na primjer, BCI sustavi za kontrolu protetskih udova pokazali su se vrlo uspješnim u vraćanju funkcionalnosti pacijentima s gubitkom udova, omogućujući im da ponovno steknu određeni stupanj autonomije. Drugi primjeri uključuju BCI sustave za komunikaciju koji omogućuju osobama s motoričkim invaliditetom da komuniciraju putem računalnih sučelja jednostavnim fokusiranjem na određene misaone zadatke. Ove studije ilustriraju praktičnu primjenu i potencijal za poboljšanje kvalitete života putem BCI tehnologija [12].

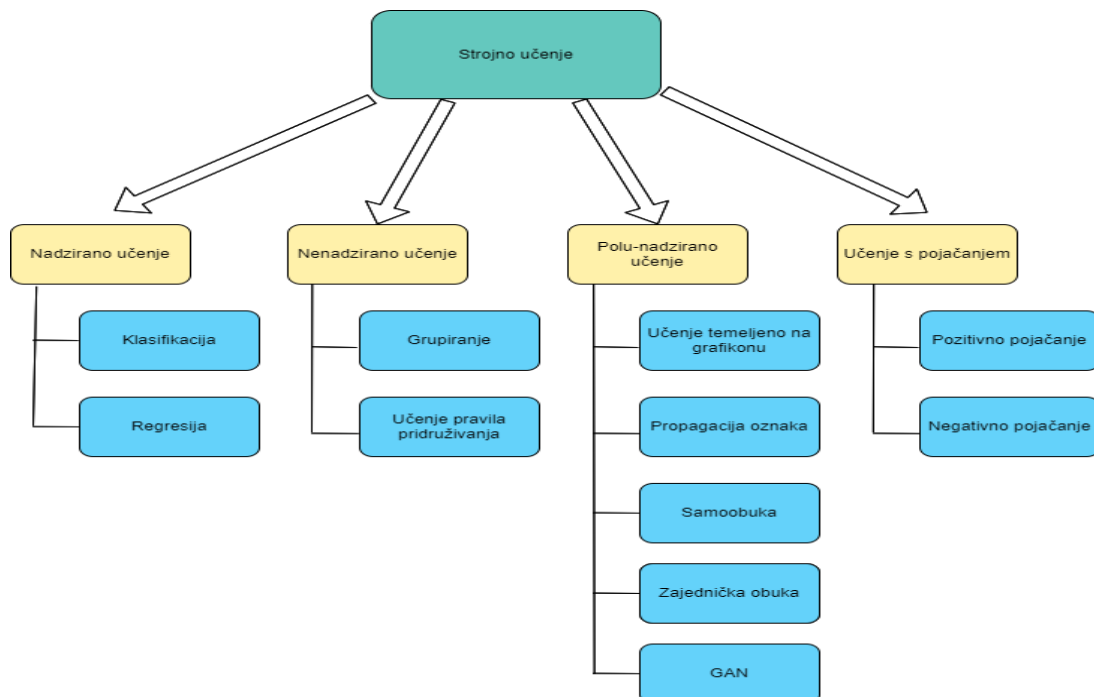
## 5. Strojno učenje

Strojno učenje (ML) grana je umjetne inteligencije (AI) koja se usredotočuje na razvoj modela i algoritama koji omogućuju računalima da uče iz podataka i poboljšavaju se iz prethodnog iskustva bez eksplicitnog programiranja za svaki zadatak. Jednostavnim riječima, ML uči sustave da razmišljaju i razumiju poput ljudi učeći iz podataka. Strojno učenje općenito je sustav obuke za učenje iz prošlih iskustava i poboljšanje performansi tijekom vremena. Strojno učenje pomaže u predviđanju golemih količina podataka. Pomaže u postizanju brzih i točnih rezultata za dobivanje profitabilnih prilika [9].

### 5.1. Vrste strojnog učenja

Postoji nekoliko vrsta strojnog učenja, od kojih svaka ima posebne karakteristike i primjene. Neki od glavnih tipova algoritama strojnog učenja su:

- Nadzirano strojno učenje
- Strojno učenje bez nadzora
- Polu-nadzirano strojno učenje
- Učenje s pojačanjem [9].



Slika 3. Vrste strojnog učenja (Vlastita izrada)

### **5.1.1. Nadzirano strojno učenje**

Nadzirano učenje definira se kao kada se model obučava na "označenom skupu podataka". Označeni skupovi podataka imaju i ulazne i izlazne parametre. U nadziranom učenju algoritmi uče mapirati točke između ulaza i ispravnih izlaza. Ima označene skupove podataka za obuku i validaciju. Nadzirano strojno učenje se dijeli na dvije glavne kategorije:

- Klasifikacija
- Regresija

#### **5.1.1.1. Klasifikacija**

Klasifikacija se bavi predviđanjem kategoričkih ciljnih varijabli, koje predstavljaju diskretne klase ili oznake. Na primjer, klasificiranje e-pošte kao željene ili neželjene pošte ili predviđanje ima li pacijent visok rizik od srčanih bolesti. Klasifikacijski algoritmi uče mapirati ulazna obilježja u jednu od unaprijed definiranih klasa. Neki od algoritama klasifikacije su:

- Logistička regresija
- Potporni vektorski stroj
- Slučajna šuma
- Stablo odluke
- K-najbližih susjeda (KNN)
- Naivni Bayes

#### **5.1.1.2. Regresija**

Regresija se bavi predviđanjem kontinuiranih ciljnih varijabli, koje predstavljaju numeričke vrijednosti. Na primjer, predviđanje cijene kuće na temelju njezine veličine, lokacije i pogodnosti ili predviđanje prodaje proizvoda. Regresijski algoritmi uče preslikati ulazne značajke na kontinuiranu numeričku vrijednost. Neki od algoritama regresije su:

- Linearna regresija
- Polinomska regresija
- Regresija grebena
- Laso regresija
- Stablo odlučivanja

#### **5.1.1.3. Prednosti nadziranog učenja**

- Modeli nadziranog učenja mogu imati visoku točnost jer se treniraju na označenim podacima.

- Proces donošenja odluka u modelima nadziranog učenja često je protumačiv.
- Često se može koristiti u unaprijed obučanim modelima što štedi vrijeme i resurse pri razvoju novih modela od nule.

#### 5.1.1.4. Nedostatci nadziranog učenja

- Ima ograničenja u poznavanju obrazaca i može se boriti s nevidljivim ili neočekivanim uzorcima koji nisu prisutni u podacima o obuci.
- To može biti dugotrajno i skupo jer se oslanja samo na označene podatke.
- To može dovesti do loših generalizacija na temelju novih podataka. [9][10]

### 5.1.2. Strojno učenje bez nadzora

Učenje bez nadzora je vrsta tehnike strojnog učenja u kojoj algoritam otkriva obrasce i odnose koristeći neoznačene podatke. Za razliku od nadziranog učenja, nenadzirano učenje ne uključuje osiguranje ciljanih izlaza algoritmu. Primarni cilj učenja bez nadzora često je otkrivanje skrivenih obrazaca, sličnosti ili klastera unutar podataka, koji se zatim mogu koristiti u različite svrhe, kao što je istraživanje podataka, vizualizacija, smanjenje dimenzionalnosti i još mnogo toga. Učenje bez nadzora se dijeli na dvije glavne kategorije:

- Grupiranje (eng. Clustering)
- Učenje pravila pridruživanja (eng. Association rule learning)

#### 5.1.2.1. Grupiranje

Grupiranje je proces grupiranja točaka podataka u klasterne na temelju njihove sličnosti. Ova je tehnika korisna za prepoznavanje uzoraka i odnosa u podacima bez potrebe za označenim primjerima. Neki od algoritama grupiranja su:

- K-Means algoritam klasteriranja
- Algoritam srednjeg pomaka
- DBSCAN algoritam
- Analiza glavnih komponenti
- Neovisna analiza komponenti

#### 5.1.2.2. Učenje pravila pridruživanja

Učenje pravila pridruživanja je tehnika za otkrivanje odnosa između stavki u skupu podataka. Identificira pravila koja ukazuju na prisutnost jedne stavke koja podrazumijeva prisutnost druge stavke s određenom vjerojatnošću. Neki od algoritama učenja pravila pridruživanja su:

- Apriorni algoritam
- Eclat
- FP-algoritam rast

### 5.1.2.3. Prednosti nenadziranog strojnog učenja

- Pomaže u otkrivanju skrivenih obrazaca i različitih odnosa među podacima.
- Koristi se za zadatke kao što su segmentacija kupaca, otkrivanje anomalija i istraživanje podataka.
- Ne zahtijeva označene podatke i smanjuje napor označavanja podataka.

### 5.1.2.4. Nedostatci nenadziranog strojnog učenja

- Bez upotrebe oznaka, može biti teško predvidjeti kvalitetu izlaza modela.
- Interpretabilnost klastera možda neće biti jasna i možda neće imati smislenu tumačenja.
- Ima tehnike kao što su autoenkoderi i smanjenje dimenzionalnosti koje se mogu koristiti za izdvajanje značajnih značajki iz neobrađenih podataka [9][10].

## 5.1.3. Polu-nadzirano strojno učenje

Polu-nadzirano učenje je algoritam strojnog učenja koji radi između nadziranog i nenadziranog učenja tako da koristi i označene i neoznačene podatke. Osobito je korisno kada je dobivanje označenih podataka skupo, dugotrajno ili zahtijeva resurse. Ovaj pristup je koristan kada je skup podataka skup i dugotrajan. Polu-nadzirano učenje odabire se kada označeni podaci zahtijevaju vještine i relevantne resurse kako bi se moglo trenirati ili učiti iz njih. Ove tehnike koristimo kada imamo posla s podacima koji su malo označeni, a veliki dio njih je neoznačen. Možemo koristiti nenadzirane tehnike za predviđanje oznaka, a zatim te oznake unijeti u nadzirane tehnike. Ova tehnika je uglavnom primjenjiva u slučaju skupova slikovnih podataka gdje obično sve slike nisu označene. Neki od algoritama polu-nadziranog učenja su:

- Polu-nadzirano učenje temeljeno na grafikonu
- Propagacija oznaka
- Zajednička obuka
- Samoobuka
- Generativne kontradiktorne mreže (GAN)

#### **5.1.3.1. Prednosti polu-nadziranog učenja**

- To dovodi do bolje generalizacije u usporedbi s nadziranim učenjem, budući da uzima i označene i neoznačene podatke.
- Može se primijeniti na širok raspon podataka.

#### **5.1.3.2. Nedostatci polu-nadziranog učenja**

- Polu-nadzirane metode mogu biti složenije za implementaciju u usporedbi s drugim pristupima.
- Još uvijek zahtijeva neke označene podatke koji možda nisu uvijek dostupni ili ih je lako dobiti.
- Neoznačeni podaci mogu na odgovarajući način utjecati na izvedbu modela.

#### **5.1.4. Učenje s pojačanjem**

Algoritam strojnog učenja s pojačanjem je metoda učenja koja je u interakciji s okolinom stvaranjem radnji i otkrivanjem pogrešaka. Pokušaj, pogreška i kašnjenje najrelevantnije su karakteristike učenja s pojačanjem. U ovoj tehnici, model nastavlja povećavati svoju izvedbu koristeći povratnu informaciju o nagradi kako bi naučio ponašanje ili obrazac. Ovi algoritmi su specifični za određeni problem, npr. Google auto koji samostalno vozi. Svaki put kada unosimo podatke, oni uče i dodaju podatke svom znanju, a to su podaci za obuku. Dakle, što više uči, to se bolje obučava, a time i doživljava. Neki od najčešćih algoritama s pojačanjem su:

- Q-učenje
- SARSA (eng. State-Action-Reward-State-Action)
- Duboko Q-učenje

Dvije glavne vrste strojnog učenja s pojačanjem su:

- Pozitivno pojačanje
- Negativno pojačanje

Pozitivno pojačanje nagrađuje agenta za poduzimanje željene akcije i potiče agenta da ponovi ponašanje, dok negativno pojačanje uklanja nepoželjni podražaj kako bi potaknuo željeno ponašanje. Također, odvraća agenta od ponavljanja zadatka.

#### **5.1.4.1. Prednosti strojnog učenja s pojačanjem**

- Ima autonomno donošenje odluka koje je dobro prilagođeno za zadatke i koje može naučiti donositi niz odluka, poput robotike i igranja igrica.



- Ova tehnika se preferira za postizanje dugoročnih rezultata koje je vrlo teško postići.
- Koristi se za rješavanje složenih problema koji se ne mogu riješiti konvencionalnim tehnikama.

#### 5.1.4.2. Nedostatci strojnog učenja s pojačanjem

- Vježbanje navedenih agenata učenja može biti računalno skupo i dugotrajno.
- Učenje s potkrepljenjem nije bolje od rješavanja jednostavnih problema.
- Zahtijeva puno podataka i puno računanja, što ga čini nepraktičnim i skupim [9][10].

## 5.2. Prednosti strojnog učenja

Strojno učenje ima brojne prednosti, ali isto tako i brojne nedostatke. Neke od prednosti strojnog učenja su:

- Poboljšana točnost i preciznost

Jedna od najznačajnijih prednosti strojnog učenja je njegova sposobnost poboljšanja točnosti i preciznosti u različitim zadacima. ML modeli mogu obraditi ogromne količine podataka i identificirati obrasce koje bi ljudi mogli zanemariti. Na primjer, u medicinskoj dijagnostici, ML algoritmi mogu analizirati medicinske slike ili podatke o pacijentima kako bi otkrili bolesti s visokim stupnjem točnosti.

- Automatizacija zadataka koji se ponavljaju

Strojno učenje omogućuje automatizaciju ponavljajućih i svakodnevnih zadataka, oslobađajući ljudske resurse za složenije i kreativnije pothvate. U industrijama kao što su proizvodnja i služba za korisnike, automatizacija vođena ML-om može se nositi s rutinskim zadacima kao što su kontrola kvalitete, unos podataka i upiti kupaca, što rezultira povećanom produktivnošću i učinkovitosti.

- Poboljšano donošenje odluka

Modeli strojnog učenja mogu analizirati velike skupove podataka i pružiti uvide koji pomažu u donošenju odluka. Prepoznavanjem trendova, korelacija i anomalija, strojno učenje pomaže tvrtkama i organizacijama u donošenju odluka temeljenih na podacima. Ovo je osobito vrijedno u sektorima poput financija, gdje se strojno učenje može koristiti za procjenu rizika, otkrivanje prijevara i investicijske strategije.

- Poboljšana sigurnost

Strojno učenje poboljšava sigurnosne mjere otkrivanjem i odgovaranjem na prijetnje u stvarnom vremenu. U kibernetičkoj sigurnosti, ML algoritmi analiziraju obrasce mrežnog prometa kako bi identificirali neobične aktivnosti koje ukazuju na kibernetičke napade. Slično tome, financijske institucije koriste strojno učenje za otkrivanje prijevvara praćenjem transakcija radi sumnjivog ponašanja.

- Smanjenje troškova

Automatiziranjem procesa i poboljšanjem učinkovitosti, strojno učenje može dovesti do značajnog smanjenja troškova. U proizvodnji, prediktivno održavanje vođeno strojnim učenjem pomaže identificirati probleme s opremom prije nego što postanu skupi kvarovi, smanjujući vrijeme zastoja i troškove održavanja. U službi za korisnike, chatbotovi koje pokreće ML smanjuju potrebu za ljudskim agentima, smanjujući operativne troškove.

- Skalabilnost

Modeli strojnog učenja mogu obraditi velike količine podataka i učinkovito se skalirati kako podaci rastu. Ova skalabilnost ključna je za tvrtke koje se bave velikim podacima, kao što su platforme društvenih medija i online trgovci. ML algoritmi mogu obrađivati i analizirati podatke u stvarnom vremenu, pružajući pravovremene uvide i odgovore [11][14][15].

### **5.3. Nedostatci strojnog učenja**

Postoje brojni nedostatci strojnog učenja, neki od njih su:

- Ovisnost o podacima

Modeli strojnog učenja zahtijevaju ogromne količine podataka za učinkovito treniranje. Kvaliteta, količina i raznolikost podataka značajno utječu na izvedbu modela. Nedovoljni ili pristrani podaci mogu dovesti do netočnih predviđanja i lošeg donošenja odluka. Osim toga, dobivanje i vođenje velikih skupova podataka može biti dugotrajno i skupo.

- Visoki računalni troškovi

Obuka modela strojnog učenja, posebno algoritama dubokog učenja, zahtijeva značajne računalne resurse. Često je potreban hardver visokih performansi kao što su GPU i TPU, što može biti skupo. Potrošnja energije povezana s treniranjem velikih modela također je znatna, što izaziva zabrinutost oko utjecaja na okoliš.

- Složenost i interpretabilnost

Mnogi modeli strojnog učenja, posebice duboke neuronske mreže, funkcioniraju kao crne kutije. Njihova složenost otežava tumačenje načina na koji dolaze do određenih odluka. Ovaj nedostatak transparentnosti postavlja izazove u područjima gdje je razumijevanje procesa donošenja odluka ključno, kao što su zdravstvo i financije.

- Sigurnosne ranjivosti

Modeli strojnog učenja osjetljivi su na kontradiktorne napade, gdje zlonamjerni akteri manipuliraju ulaznim podacima kako bi prevarili model da napravi netočna predviđanja. Ova ranjivost predstavlja značajne rizike u kritičnim aplikacijama kao što su autonomna vožnja, kibernetička sigurnost i otkrivanje financijskih prijevara.

- Održavanje i ažuriranja

Modeli strojnog učenja zahtijevaju kontinuirani nadzor, održavanje i ažuriranja kako bi se osiguralo da ostanu točni i učinkoviti tijekom vremena. Promjene u temeljnoj distribuciji podataka, poznate kao pomicanje podataka, mogu pogoršati performanse modela, zahtijevajući čestu ponovnu obuku i provjeru valjanosti [11][15].

## 6. Primjene BCI tehnologije

U novije vrijeme uloga sučelja mozak-računalo postaje sve značajnija. Njegova primjena je sve veća ovisno radi li se o primjeni u medicini ili u području zabave, pametnog okruženja.

### 6.1. Primjene u medicini i rehabilitaciji

Primjena BCI u medicini i rehabilitaciji je iznimno važna zato što pomaže i olakšava život osobama s teškim bolestima i rehabilitacija istih.

#### 6.1.1. Neuroprotetika

Neuroprotetika predstavlja ključno područje unutar sučelja mozak-računalo (BCI), koje omogućuje povrat funkcionalnosti osobama s tjelesnim oštećenjima, poput amputacija ili paralize. Kroz integraciju naprednih tehnologija, poput strojnog učenja i analize podataka, neuroprotetika omogućuje stvaranje sofisticiranih uređaja koji se mogu kontrolirati izravno putem moždanih signala. Ovaj odjeljak istražuje razvoj neuroprotetike, njezine ključne komponente te ulogu strojnog učenja u poboljšanju funkcionalnosti ovih uređaja [16].

##### 6.1.1.1. Razvoj neuroprotetike

Neuroprotetika se razvila iz ranih istraživanja u neuroznanosti i biomedicinskom inženjerstvu, s ciljem vraćanja izgubljenih funkcija kod osoba s različitim vrstama invaliditeta. Prvi funkcionalni neuroproteznih uređaji pojavili su se krajem 20. stoljeća, temeljeći se na tehnologiji koja je omogućavala osnovne funkcije poput pokreta ruku ili nogu. Razvoj elektroencefalografije (EEG) i elektrokortikografije (ECoG) omogućio je bilježenje i dekodiranje moždanih signala, čime su stvoreni temelji za sofisticirane BCI sustave.

U protekla dva desetljeća, napredak u tehnologiji i povećanje razumijevanja moždane funkcije doveli su do razvoja naprednijih neuroproteznih uređaja koji omogućuju finu motoričku kontrolu i osjetilni povratni signal (eng. Feedback). Ti uređaji omogućuju korisnicima ne samo kontrolu proteza već i osjećaj dodira, što značajno poboljšava njihovu funkcionalnost i kvalitetu života [17]

##### 6.1.1.2. Komponente Neuroproteznih Sustava

Suvremeni neuroproteznih sustavi sastoje se od nekoliko ključnih komponenti: senzora za bilježenje moždanih signala, algoritama za obradu i dekodiranje tih signala, te izvršnih

uređaja koji omogućuju fizičku izvedbu željene akcije. Senzori, poput EEG elektroda, igraju ključnu ulogu u snimanju moždane aktivnosti. Ovi senzori moraju biti visoko osjetljivi i pouzdani kako bi precizno prenijeli signale koji predstavljaju korisnikove namjere.

Jednom kada se moždani signali snime, dolazi do procesa obrade i dekodiranja. U ovoj fazi, algoritmi strojnog učenja igraju ključnu ulogu. Ovi algoritmi analiziraju moždane signale, prepoznaju obrasce i prevode ih u naredbe koje proteza može izvršiti. Preciznost ovog dekodiranja ključna je za osiguranje prirodnog i učinkovitog upravljanja protezom.

Izvršni uređaji u neuroprotezama uključuju umjetne udove ili druge mehaničke sustave koji provode dekodirane signale u pokrete ili druge akcije. Napredni izvršni uređaji također mogu pružiti povratne informacije korisniku, kao što su osjećaj pritiska ili temperature, što dodatno poboljšava funkcionalnost i korisničko iskustvo [16] [17].

### **6.1.1.3. Uloga strojnog učenja u neuroprotetici**

Strojno učenje ima ključnu ulogu u razvoju neuroproteznih uređaja, posebno u procesima dekodiranja moždanih signala i prilagodbe uređaja specifičnim potrebama korisnika. Algoritmi strojnog učenja omogućuju analizu složenih obrazaca u moždanim signalima i njihovu pretvorbu u precizne naredbe koje proteza može izvršiti.

Jedna od glavnih prednosti strojnog učenja u neuroprotetici je njegova sposobnost prilagodbe individualnim razlikama u moždanim signalima. Svaki korisnik ima jedinstven način na koji njegov mozak generira signale za određene pokrete, što znači da algoritmi moraju biti dovoljno fleksibilni da se prilagode tim razlikama. To se postiže korištenjem metoda nadziranog učenja, gdje se algoritmi treniraju na podacima specifičnim za svakog korisnika, što omogućuje visoku razinu personalizacije uređaja .

Osim toga, strojno učenje omogućuje kontinuirano učenje i prilagodbu tijekom vremena. Kako korisnik koristi protezu, algoritmi mogu prikupljati nove podatke i ažurirati svoje modele kako bi se poboljšala preciznost i učinkovitost upravljanja protezom. To rezultira uređajima koji ne samo da pružaju visoku razinu funkcionalnosti, već se i prilagođavaju promjenama u korisnikovim potrebama i preferencijama tijekom vremena [16] [17].

### **6.1.2. Komunikacijski alati za paralizu**

Osobe s teškom paralizom suočavaju se s velikim izazovima u komunikaciji. Tradicionalni alati i metode često nisu dovoljno učinkoviti u pružanju pouzdane komunikacije ovim pacijentima, što značajno utječe na njihovu kvalitetu života. Kroz napredak u tehnologijama sučelja mozak-računalo (BCI), sada su dostupni novi alati koji omogućuju

osobama s paralizom da komuniciraju putem analize moždanih signala, koristeći tehnike strojnog učenja [18].

#### **6.1.2.1. Razvoj komunikacijskih alata temeljenih na BCI**

Sučelja mozak-računalo omogućuju korisnicima kontrolu vanjskih uređaja isključivo pomoću moždanih signala, bez potrebe za mišićnom aktivnošću. Ovi sustavi funkcioniraju tako da snimaju moždane signale, najčešće pomoću EEG-a, i zatim te signale dekodiraju koristeći algoritme strojnog učenja kako bi prepoznali korisnikove namjere. Jedna od najpoznatijih aplikacija ove tehnologije je u komunikaciji, gdje korisnici mogu "tipkati" ili odabrati riječi i fraze jednostavno usmjeravanjem svojih misli .

Korištenjem metoda strojnog učenja, ovi sustavi postaju sve sofisticiraniji i točniji. Na primjer, algoritmi nadziranog učenja mogu se obučiti na pojedinačnim korisnicima kako bi prepoznali specifične obrasce moždanih aktivnosti povezane s određenim mislima ili namjerama, omogućujući personalizirane komunikacijske profile. To omogućava bržu i precizniju interakciju, smanjujući pogreške i povećavajući brzinu komunikacije [19].

#### **6.1.2.2. Tehnike strojnog učenja u komunikacijskim alatima**

Algoritmi strojnog učenja igraju ključnu ulogu u razvoju komunikacijskih alata za osobe s paralizom. Jedna od najvažnijih tehnika u ovom području je klasifikacija, gdje se moždani signali klasificiraju u različite kategorije koje odgovaraju određenim riječima ili akcijama. Algoritmi poput linearne diskriminantne analize (LDA) i potpornih vektorskih strojeva (SVM) često se koriste u ranim fazama razvoja ovih sustava. Ovi algoritmi omogućuju preciznu dekodifikaciju signala, ali zahtijevaju prethodno znanje o signalima i obuku na specifičnim podacima [19].

S pojavom dubokog učenja, učinkovitost i fleksibilnost komunikacijskih alata značajno su se poboljšali. Konvolucijske neuronske mreže (CNN) i rekurentne neuronske mreže (RNN) omogućuju složenije analize moždanih signala, pružajući dublje razumijevanje moždane aktivnosti i omogućujući prepoznavanje složenijih obrazaca. Ovi napredni algoritmi smanjuju potrebu za opsežnim prilagođavanjem sustava svakom korisniku, dok u isto vrijeme poboljšavaju brzinu i točnost komunikacije [19].

#### **6.1.3. Kognitivna rehabilitacija**

Kognitivna rehabilitacija predstavlja ključno područje u neurorehabilitaciji, koje se bavi obnavljanjem ili poboljšanjem kognitivnih funkcija kod pacijenata koji su pretrpjeli ozljede

mozga ili pate od neurodegenerativnih bolesti. Sučelja mozak-računalo (BCI) pružaju inovativne pristupe kognitivnoj rehabilitaciji, omogućujući personalizirane terapije koje su podržane strojnim učenjem i naprednim analizama moždanih podataka [20].

#### **6.1.3.1. Uloga strojnog učenja u kognitivnoj rehabilitaciji**

Strojno učenje ima značajnu ulogu u kognitivnoj rehabilitaciji, omogućujući analizu i interpretaciju složenih moždanih podataka kako bi se identificirali obrasci koji su povezani s kognitivnim funkcijama. Korištenjem metoda kao što su potporni vektorski strojevi (SVM), neuronske mreže i duboko učenje, istraživači mogu razviti modele koji prepoznaju specifične moždane aktivnosti povezane s različitim kognitivnim zadacima, poput memorije, pažnje i izvršnih funkcija. Ovi modeli omogućuju stvaranje personaliziranih terapija koje se prilagođavaju individualnim potrebama pacijenata, što je ključno za postizanje optimalnih rezultata rehabilitacije [20].

Jedan od ključnih aspekata strojnog učenja u ovom kontekstu je njegova sposobnost kontinuiranog učenja i prilagodbe. Kako pacijenti napreduju kroz rehabilitaciju, algoritmi strojnog učenja mogu prilagoditi svoje modele na temelju novih podataka, omogućujući terapije koje se razvijaju zajedno s pacijentovim napretkom. To rezultira dinamičnim pristupom koji može bolje odgovoriti na promjenjive potrebe pacijenata i pružiti učinkovitije terapijske intervencije [21].

#### **6.1.3.2. Primjena BCI-a u kognitivnoj rehabilitaciji**

BCI sustavi omogućuju pacijentima da sudjeluju u kognitivnoj rehabilitaciji kroz interaktivne zadatke i igre koje su osmišljene kako bi stimulirale specifične kognitivne funkcije. Ovi sustavi koriste podatke o moždanim signalima kako bi prilagodili težinu i sadržaj zadataka u stvarnom vremenu, omogućujući personalizirano iskustvo koje odgovara trenutnoj razini kognitivne funkcije pacijenta.

Strojno učenje je ključno za ovu vrstu prilagodbe jer omogućuje sustavima da prepoznaju suptilne promjene u moždanim signalima koje ukazuju na napredak ili poteškoće u izvršavanju zadataka [21].

## **6.2. Nemedicinske primjene**

Drugo veliko područje primjene BCI tehnologije je u nemedicinske svrhe, kao što su igre ili zabava gdje se putem uređaja virtualne stvarnosti (eng. Virtual Reality, VR) i putem misli kontroliraju igre ili virtualne okoline tj. pametno okruženje.

## 6.2.1. Igre i zabava

Primjena sučelja mozak-računalo (BCI) u industriji igara i zabave postala je sve popularnija, nudeći nove mogućnosti za interaktivnost i iskustva temeljena na moždanoj aktivnosti korisnika. Kroz integraciju naprednih tehnika strojnog učenja, BCI sustavi omogućuju korisnicima da kontroliraju igre i virtualne okoline samo putem svojih misli, bez potrebe za fizičkim unosom.

### 6.2.1.1. Uloga strojnog učenja u BCI igrama

Strojno učenje igra ključnu ulogu u razvoju BCI sustava za igre, omogućujući dekodiranje i interpretaciju moždanih signala u stvarnom vremenu. Korištenjem algoritama kao što su konvolucijske neuronske mreže (CNN) i rekurentne neuronske mreže (RNN), istraživači su u mogućnosti prepoznati složene obrasce u EEG signalima koji odgovaraju određenim mislima ili namjerama korisnika. Na taj način, BCI sustavi mogu omogućiti korisnicima da kontroliraju avatar, aktiviraju radnje ili čak sudjeluju u složenim zadacima unutar igre bez korištenja tradicionalnih kontrola [22].

Jedna od glavnih prednosti primjene strojnog učenja u BCI igrama je prilagodljivost sustava. Kako korisnici postaju iskusniji u korištenju BCI-a, algoritmi strojnog učenja mogu prilagoditi svoje modele kako bi poboljšali točnost i brzinu interakcije. Ovo omogućuje personalizirano iskustvo koje se prilagođava individualnim potrebama i sposobnostima svakog korisnika, čineći igre intuitivnijim i dostupnijim [23].

### 6.2.1.2. Primjene BCI-a u zabavi

Osim u igrama, BCI sustavi se sve više koriste u širem spektru zabavnih aplikacija. Na primjer, virtualna stvarnost (VR) i proširena stvarnost (eng. Augmented Reality, AR) integriraju BCI tehnologije kako bi omogućile korisnicima potpuno uranjanje u digitalna okruženja kontrolirana njihovim moždanim signalima. Korištenjem naprednih algoritama strojnog učenja, ovi sustavi mogu prilagoditi okruženja i scenarije u stvarnom vremenu, temeljeći se na emocionalnom stanju ili namjerama korisnika [24].

BCI također ima potencijal za korištenje u interaktivnim filmovima i glazbi, gdje korisnici mogu utjecati na tijek radnje ili promjene u glazbi putem svojih misli. Ove primjene otvaraju nove mogućnosti za stvaranje potpuno novih oblika zabave, gdje je korisnik ne samo pasivni promatrač već aktivni sudionik u kreativnom procesu [23].



## **6.2.2. Pametna okruženja**

Pametna okruženja su prostori opremljeni naprednim sensorima i računalnim sustavima koji mogu prilagoditi svoje funkcionalnosti i ponašanje u skladu s potrebama korisnika. Sučelja mozak-računalo (BCI) u kombinaciji s tehnikama strojnog učenja otvaraju nove mogućnosti za interakciju s ovim okruženjima, pružajući korisnicima intuitivan i personaliziran način upravljanja različitim aspektima njihovog okruženja [24].

### **6.2.2.1. Uloga strojnog učenja u pametnim okruženjima**

Strojno učenje igra ključnu ulogu u analizi podataka generiranih iz BCI sustava unutar pametnih okruženja. Pametna okruženja generiraju velike količine podataka iz različitih senzora, uključujući EEG signale iz BCI-a, a strojno učenje omogućuje analizu ovih podataka u stvarnom vremenu kako bi se prepoznali obrasci i donijele odluke na temelju korisničkih potreba i preferencija. Na primjer, algoritmi dubokog učenja mogu se koristiti za analizu EEG signala kako bi se identificirale specifične mentalne stanja korisnika, poput koncentracije, umora ili stresa, i zatim prilagodili osvjetljenje, temperaturu ili druge aspekte okruženja u skladu s tim [24].

Jedan od glavnih izazova u ovom kontekstu je obrada i analiza podataka u stvarnom vremenu. Strojno učenje omogućuje obradu velikih količina podataka brzo i učinkovito, što je ključno za pametna okruženja koja trebaju reagirati na promjene u okruženju ili stanju korisnika u stvarnom vremenu. Osim toga, adaptivni algoritmi strojnog učenja mogu se prilagoditi promjenama u korisnikovom ponašanju tijekom vremena, što omogućuje pametnim okruženjima da postanu sve "inteligentnija" i prilagodljivija [23].

### **6.2.2.2. Primjene BCI-a u pametnim okruženjima**

Primjena BCI-a u pametnim okruženjima omogućuje korisnicima da upravljaju raznim aspektima svog doma ili radnog prostora koristeći samo svoje moždane signale. Na primjer, korisnici s ograničenom pokretljivošću mogu koristiti BCI sustave za upravljanje kućnim aparatima, rasvjetom ili sigurnosnim sustavima bez potrebe za fizičkim naporom. Ova tehnologija može značajno poboljšati kvalitetu života korisnika, omogućujući im veću neovisnost i kontrolu nad svojim okruženjem [23].

Jedan od značajnih primjera je integracija BCI-a s pametnim domovima opremljenim Internet stvari (eng. Internet of Things, IoT) uređajima. Korištenjem strojnog učenja, ovi sustavi mogu naučiti korisnikove preferencije i prilagoditi funkcionalnosti doma na temelju njihove trenutne mentalne aktivnosti. Na primjer, BCI može detektirati kada je korisnik umoran i automatski smanjiti razinu osvjetljenja ili aktivirati uređaj za reprodukciju umirujuće glazbe [23].

### **6.2.3. Poboljšanja interakcije čovjek-računalo**

Interakcija čovjek-računalo (eng. Human-computer interaction, HCI) predstavlja ključnu komponentu u razvoju tehnologija koje omogućuju korisnicima učinkovitu i intuitivnu komunikaciju s računalnim sustavima. Sučelja mozak-računalo (BCI), u kombinaciji s tehnikama strojnog učenja, nude značajna poboljšanja u ovom području, omogućujući nove načine interakcije koji su prilagođeni individualnim potrebama i sposobnostima korisnika [25].

#### **6.2.3.1. Uloga strojnog učenja u Poboljšanju HCI sustava**

Strojno učenje igra ključnu ulogu u analizi i interpretaciji složenih podataka iz BCI sustava, što omogućuje preciznu i prilagodljivu interakciju između korisnika i računala. U kontekstu HCI, algoritmi strojnog učenja, kao što su duboke neuronske mreže (DNN) i potporni vektorski strojevi (SVM), koriste se za prepoznavanje obrazaca u EEG signalima, koji su ključni za dekodiranje namjera korisnika. Ovi algoritmi mogu kontinuirano učiti i prilagođavati se na temelju povratnih informacija, omogućujući sve preciznije prepoznavanje korisnikovih misli i namjera [25].

Jedan od glavnih ciljeva primjene strojnog učenja u HCI sustavima je smanjenje kognitivnog opterećenja korisnika. Umjesto da korisnik mora izvoditi složene radnje kako bi komunicirao s računalom, BCI sustavi s integriranim strojnim učenjem omogućuju intuitivniju interakciju temeljenu na prirodnim moždanim signalima. Na primjer, u sustavima za komunikaciju osobama s invaliditetom, strojno učenje omogućuje prepoznavanje jednostavnih misli kao što su "da" ili "ne", čime se omogućuje učinkovita i brza komunikacija bez potrebe za fizičkim unosom [26].

#### **6.2.3.2. Primjeri primjene**

Primjene BCI-a s integriranim strojnim učenjem u HCI-u su široke i raznolike. U području asistivnih tehnologija, na primjer, BCI sustavi omogućuju osobama s invaliditetom upravljanje računalima, komunikacijskim uređajima, pa čak i robotima, koristeći samo svoje moždane signale. Ovi sustavi koriste strojno učenje kako bi se prilagodili specifičnim potrebama svakog korisnika, omogućujući im veću autonomiju i poboljšavajući kvalitetu života. Još jedan primjer je primjena BCI-a u igrama, gdje strojno učenje omogućuje igračima da kontroliraju igre koristeći samo svoje misli. Kroz analizu EEG signala, algoritmi mogu prepoznati specifične obrasce moždane aktivnosti povezane s određenim radnjama u igri, omogućujući igračima da se u potpunosti urone u igru bez korištenja tradicionalnih kontrola [25].

U kontekstu rada i produktivnosti, BCI sustavi s integriranim strojnim učenjem mogu se koristiti za poboljšanje interakcije u složenim radnim okruženjima. Na primjer, u kontrolnim sobama gdje je potrebno brzo donošenje odluka, BCI može omogućiti operaterima da intuitivno upravljaju sustavima i donose odluke na temelju stvarnih moždanih signala, smanjujući rizik od pogrešaka i povećavajući učinkovitost [27].

## 7. Analiza EEG skupa motoričkih pokreta

EEG skup motoričkih sposobnosti je dostupan preko PhysioNet-a, jedan je od najčešće korištenih skupova podataka za proučavanje sučelja mozak-računalo (BCI) i elektroencefalografskih (EEG) signala povezanih s motoričkim pokretima i zamišljanjem pokreta. Objavljen kao dio PhysioNet-ove Neuro informatike, ovaj skup podataka pruža snažan resurs za istraživanja u neuroznanosti, kognitivnoj znanosti i strojnom učenju, osobito u domeni motoričke aktivnosti mozga.

### 7.1. Preuzimanje podataka

Prvo je potrebno preuzeti kompletan skup podataka putem linka <https://physionet.org/content/eegmmidb/1.0.0/#files-panel>. Može se preuzeti cijeli skup podataka ili samo određene datoteke koje su potrebne za analizu. Putem ručnog preuzimanja moguće je preuzeti .zip datoteku veličine 1.9 GB ili klikom na pojedine datoteke u direktorijima preuzeti samo jedan file.

Ovisno o operacijskom sustavu (eng. Operating System, OS) moguće je preuzeti uz prethodnu instalaciju alata wget:

```
wget -r -N -c -np https://physionet.org/files/eegmmidb/1.0.0/
```

Odlična metoda za preuzimanje velikih skupova podataka je alat rsync:

```
rsync -r -P physionet.org::physionet-data/eegmmidb/1.0.0/ ./lokalna_mapa/
```

Isto tako moguće je pristupiti podacima preko Google Oblaka (eng. Google Cloud Storage). Prijava na Googleov račun je potrebna [28].

### 7.2. Pregled skupa podataka

Ovaj skup podataka sadrži EEG zapise 109 ispitanika koji su sudjelovali u nizu zadataka povezanih s motoričkim pokretima i imaginacijom. Zadatci uključuju stvarne fizičke pokrete i zamišljene pokrete (motorna imaginacija), kao što su otvaranje i zatvaranje šake ili pomicanje stopala. Ove aktivnosti su ključne za razvoj BCI sustava koji prevode moždanu aktivnost u akcijske signale, s primjenama od asistivnih tehnologija za osobe s invaliditetom do neuroproteza.

Svaki ispitanik je sudjelovao u 14 eksperimentalnih sesija u kojima su ili izvodili motoričke zadatke ili zamišljali izvođenje istih. Skup podataka uključuje kontinuirane EEG

signale zabilježene s pomoću 64-kanalnog BCI2000 sustava, koji snima moždanu aktivnost s visokom vremenskom rezolucijom. Podaci su organizirani prema BIDS (eng. Brain Imaging Data Structure) formatu, što ih čini kompatibilnim s većinom modernih alata za analizu moždanog zamišljanja (eng. Neuroimaging). Ova struktura olakšava upravljanje eksperimentalnim podacima i osigurava da istraživači lako razumiju i ponovo koriste skup podataka [28].

### **7.2.1. Prikupljanje podataka**

EEG signali u ovom skupu podataka zabilježeni su s pomoću elektroda postavljenih na vlasištu prema 10-10 međunarodnom sustavu, koji je široko priznata metoda za prikupljanje EEG podataka. Ovaj sustav omogućuje sveobuhvatno pokrivanje kortikalnih područja mozga, osobito regija povezanih s motoričkom kontrolom poput motorne kore. Signali su uzorkovani pri 160 Hz, što omogućuje detaljan prikaz električne aktivnosti mozga tijekom motoričkih i imaginacijskih zadataka. Skup podataka također uključuje metapodatke poput oznaka događaja, koje ukazuju na to kada su se specifični zadatci dogodili, što omogućava istraživačima segmentaciju podataka za analizu određenih događaja [28].

### **7.2.2. Primjene**

EEG skup motoričkih pokreta neprocjenjiv je za istraživanja u nekoliko područja:

- Razvoj BCI sustava

Skup podataka podržava razvoj BCI sustava, osobito onih koji su usmjereni na upravljanje vanjskim uređajima (poput robotskih udova) putem motoričke imaginacije ili motoričkih pokreta.

- Analiza motoričke imaginacije

Razlika između stvarnih i zamišljenih motoričkih zadataka omogućuje istraživačima istraživanje kako se te aktivnosti prikazuju u mozgu, što je ključno za razumijevanje neuralnih mehanizama motoričke kontrole i za stvaranje učinkovitih BCI sustava.

- Strojno učenje u moždanom zamišljanju

Uz rastući interes za primjenu tehnika strojnog učenja na EEG podatke, ovaj skup podataka pruža velik, dobro strukturiran resurs za treniranje modela koji klasificiraju motoričku aktivnost mozga [28].

## 7.3. Format i struktura podataka

EEG skup podataka na PhysioNet platformi sadrži EEG zapise 109 ispitanika tijekom izvođenja motoričkih zadataka i imaginacije. Podaci su u formatu European Data Format (EDF), standardnom formatu za EEG podatke. Skup podataka uključuje metapodatke (npr. .txt i .rpt), te informacije o lokacijama elektroda i markerima događaja, što omogućuje analizu potencijala izazvanih događajima (eng. Event-Related Potential, ERP). Struktura skupa olakšava istraživanje strojnog učenja i BCI-ja kroz dobro dokumentirane oznake i višekanalne zapise.

Struktura skupa podataka obično uključuje:

- Sirove EEG datoteke: .edf datoteke koje pohranjuju EEG signale tijekom vremena.
- Metapodatke: Datoteke koje sadrže informacije o postavkama eksperimenta, položaju elektroda i detaljima zadatka (npr. početak zadatka, trajanje itd.).
- Oznake događaja: Anotacije koje označavaju specifične događaje, poput početka zadatka ili prezentacije podražaja [28].

## 7.4. Pretprocesiranje podataka

Pretprocesiranje EEG podataka iz odabranog skupa podataka predstavlja ključan korak prije provođenja bilo kakve analize, posebno kada se primjenjuju tehnike strojnog učenja. Cilj je pripremiti podatke za ekstrakciju značajki i treniranje modela tako da se podaci očiste i pravilno organiziraju. Osnovni koraci u pretprocesiranju su:

- Filtriranje podataka
- Uklanjanje artefakata
- Ponovno referenciranje
- Epochiranje i detekcija događaja
- Korekcija osnovne linije
- Ekstrakcija značajki

Pretprocesiranje podataka je moguće obaviti putem Python programa. Prije pokretanja bilo kakvog programa potrebno je instalirati MNE biblioteku i to se može napraviti kroz terminal sa komandom

```
pip install pyedflib mne
```

Također, potrebno je instalirati i sklearn (scikit-learn) paket, u kojemu se nalazi analiza nezavisne komponente (eng. Independent Component Analysis, ICA)

```
pip install scikit-learn
```

Prvo je potrebno uključiti određene biblioteke za rad s pretprocesiranjem

```
import pyedflib

import mne

from mne.preprocessing import ICA, create_eog_epochs
```

Zatim je potrebno učitati .edf datoteku

```
putanja_dat = 'datoteka.edf'

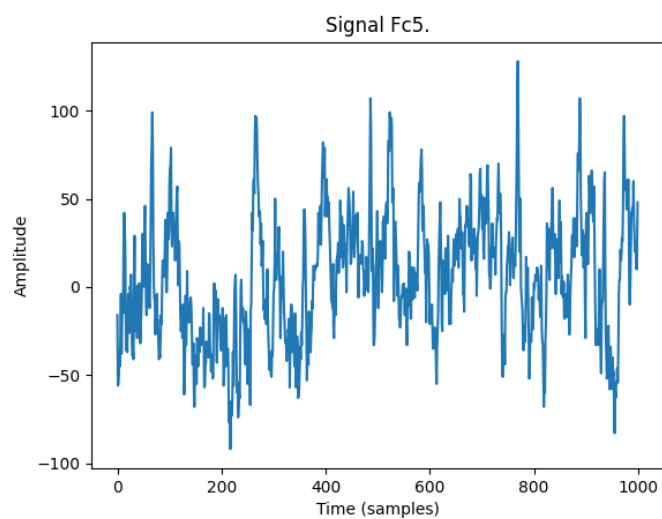
raw_data = pyedflib.EdfReader(putanja_dat)
```

Kako se u skupu podataka radi o podacima spremljenim u .edf datoteku potrebno je koristiti *pyedflib.EdfReader(datoteka)* koji u varijablu *raw\_data* sprema podatke.

```
Broj signala: 64
Oznake signala: ['Fc5.', 'Fc3.', 'Fc1.', 'Fcz.', 'Fc2.', 'Fc4.', '
fz.', 'Af4.', 'Af8.', 'F7..', 'F5..', 'F3..', 'F1..', 'Fz..', 'F2.
', 'Po7.', 'Po3.', 'Poz.', 'Po4.', 'Po8.', 'O1..', 'Oz..', 'O2..']
Stopa uzorkovanja: 160.0
```

Slika 4. Čitanje EDF datoteke

Kako se radi o velikom skupu podataka koji sadrži puno .edf datoteka, uzeo sam za primjer jednu koju ću analizirati od subjekta broj 1, prvo očitovanje nad kanalom Fc5.



Slika 5. Prikaz nefiltriranih podataka kanala Fc5.

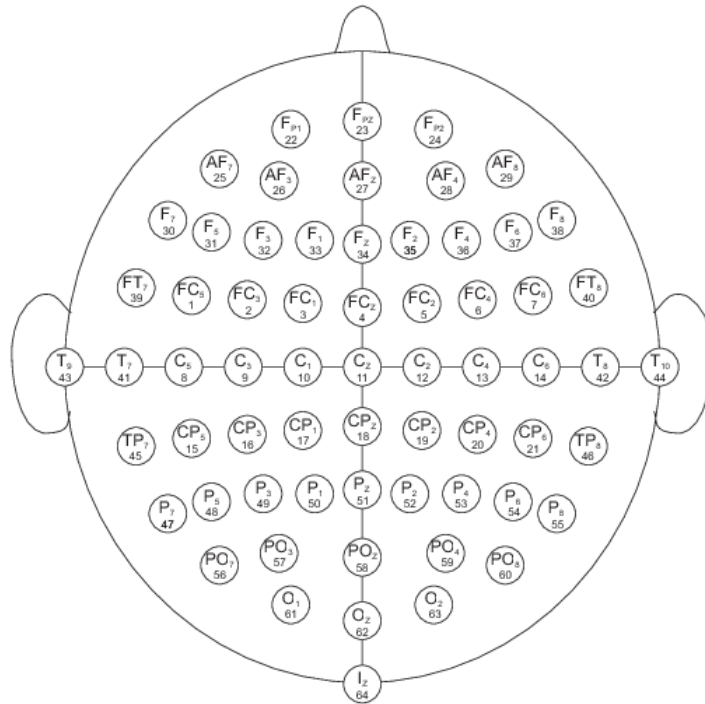
Moguće je postaviti i montažu lokacije elektroda, tj. ako je ona poznata (npr. standard\_1020).

```
montaža = mne.channels.make_standard_montage('standard_1020')  
raw_data.set_montage(montaža)
```

Standard\_1020 predložak sadrži 64 kanala elektroda i sirovi podatci moraju odgovarati predlošku kako bi se postavila montaža. Preko metode *rename\_channels* svi kanali se mogu preimenovati kako bi odgovarali predlošku [30].

```
raw_data.rename_channels({  
    'Fc5.': 'FC5', 'Fc3.': 'FC3', 'Fc1.': 'FC1', 'Fcz.': 'FCz', 'Fc2.':  
'FC2', 'Fc4.': 'FC4', 'Fc6.': 'FC6', 'C5..': 'C5', 'C3..': 'C3', 'C1..': 'C1',  
'Cz..': 'Cz', 'C2..': 'C2', 'C4..': 'C4', 'C6..': 'C6', 'Cp5.': 'CP5', 'Cp3.':  
'CP3', 'Cp1.': 'CP1', 'Cpz.': 'CPz', 'Cp2.': 'CP2', 'Cp4.': 'CP4', 'Cp6.':  
'CP6', 'Fp1.': 'Fp1', 'Fpz.': 'Fpz', 'Fp2.': 'Fp2', 'Af7.': 'AF7', 'Af3.':  
'AF3', 'Afz.': 'AFz', 'Af4.': 'AF4', 'Af8.': 'AF8', 'F7..': 'F7', 'F5..':  
'F5', 'F3..': 'F3', 'F1..': 'F1', 'Fz..': 'Fz', 'F2..': 'F2', 'F4..': 'F4',  
'F6..': 'F6', 'F8..': 'F8', 'Ft7.': 'FT7', 'Ft8.': 'FT8', 'T7..': 'T7',  
'T8..': 'T8', 'T9..': 'T9', 'T10.': 'T10', 'Tp7.': 'TP7', 'Tp8.': 'TP8',  
'P7..': 'P7', 'P5..': 'P5', 'P3..': 'P3', 'P1..': 'P1', 'Pz..': 'Pz', 'P2..':  
'P2', 'P4..': 'P4', 'P6..': 'P6', 'P8..': 'P8', 'Po7.': 'PO7', 'Po3.': 'PO3',  
'Poz.': 'POz', 'Po4.': 'PO4', 'Po8.': 'PO8', 'O1..': 'O1', 'Oz..': 'Oz',  
'O2..': 'O2', 'Iz..': 'Iz'  
})
```





Slika 6. Prikaz kanala elektroda

### 7.4.1. Filtriranje podataka

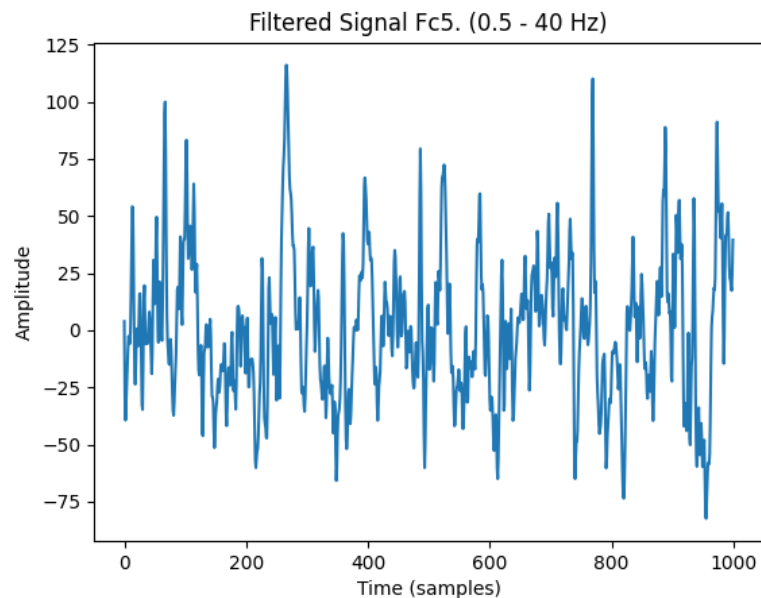
EEG podaci često sadrže različite frekvencije koje nisu povezane s moždanom aktivnošću, poput šumova uzrokovanih pokretima mišića ili smetnji iz okoline. Obično se primjenjuje band-pass filter kako bi se zadržale frekvencije između 0.5 i 40 Hz, koje su najrelevantnije za moždanu aktivnost povezanu s motoričkim zadacima.

```
def bandpass_filter(signal, lowcut, highcut, fs, order=5):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = butter(order, [low, high], btype='band')
    return filtfilt(b, a, signal)

lowcut = 0.5
highcut = 40

filtered_signals = np.array([bandpass_filter(sig, lowcut, highcut,
sampling_rate) for sig in signal_data])
```

Band-pass filter za uklanjanje šumova se primjenjuje putem *filter* metode kojoj se kao parametri proslijede najniža i najviša frekvencija. Interval frekvencija je tipično između 0.5 i 40 Hz [30].



Slika 7. Prikaz filtriranih podataka kanala Fc5. (0.5 - 40 Hz)

## 7.4.2. Uklanjanje artefakata

Uklanjanje artefakata ključan je korak u prethodnoj obradi EEG podataka. Uobičajeni artefakti, poput treptanja očiju, pokreta mišića i električnih smetnji, mogu iskriviti interpretaciju moždanih signala. Tehnike poput neovisne komponente analize (ICA) koriste se za razdvajanje i uklanjanje artefakata iz originalnog signala. Ovaj korak poboljšava točnost kasnije analize podataka, osiguravajući da se bilježi samo neuronska aktivnost. Uklanjanje artefakata bitno je za čiste podatke koji se mogu učinkovito koristiti u modelima strojnog učenja za primjene u sučelju mozak-računalo.

Neovisna komponentna analiza (ICA) se primjenjuje za uklanjanje artefakata treptanja oka.

```
ica = ICA(n_components=20, random_state=97, max_iter=800)
ica.fit(raw_data)
eog_indices, eog_scores = ica.find_bads_eog(raw_data,
ch_name='EOG 061')
```

```
ica.exclude = eog_indices  
raw_clean = ica.apply(raw_data)
```

Nakon filtracije podataka potrebno je učitati filtrirane sirove podatke u ICA, gdje se postavlja broj komponenti, random stanje i maksimalni broj iteracija. Metoda *fit* primjenjuje ICA-u nad podacima. Zatim je potrebno pronaći loše artefakte kao što su treptanje oka i isključiti ih, to se radi preko metode *find\_bads\_eog* gdje se kao parametri prosljeđuju sirovi podatci i identificiran EOG kanal za treptanje oka i putem metode *exclude* se pronađeni artefakti isključuju. Na kraju koraka je potrebno preko metode *apply* primijeniti pronađeno ICA rješenje [31].

### 7.4.3. Ponovno referenciranje

Ponovno referenciranje je proces promjene referentne elektrode korištene za EEG podatke. Ovo može značajno utjecati na omjer signal-šum podataka, čineći signale lakše interpretabilnim. Uobičajena metoda je metoda prosječnog referenciranja, koja koristi srednju vrijednost svih elektroda kao referentnu točku. Ponovno referenciranje pomaže u smanjenju električnog šuma i poboljšanju prostorne rezolucije EEG signala, čineći ih prikladnijima za primjenu strojnog učenja u BCI sustavima.

```
raw_data.set_eeg_reference('average', projection=True)
```

Preko metode *set\_eeg\_reference* potrebno je referencirati EEG podatke na prosjek svih elektroda [32].

### 7.4.4. Epochiranje i detekcija događaja

Epochiranje je segmentacija kontinuiranih EEG podataka u manje vremenske intervale, obično oko specifičnih događaja ili podražaja. Detekcija događaja identificira te trenutke od interesa, poput početka podražaja ili motoričke aktivnosti. Epochiranje omogućava detaljniju analizu moždanih odgovora na određene zadatke, olakšavajući korištenje algoritama strojnog učenja za klasifikaciju različitih vrsta moždane aktivnosti. Pravilno epochiranje i detekcija događaja ključni su za stvaranje strukturiranog skupa podataka za BCI.

```
events, event_ids = mne.events_from_annotations(raw_data)  
epochs = mne.Epochs(raw_data, events, event_id=event_ids, tmin=-0.2,  
tmax=0.5, baseline=None)
```

U ovom koraku potrebno je kreirati epohe (manje vremenske intervale) kako bi se lakše analizirali podatci, pa se preko metode *events\_from\_annotations* otkrivaju događaji iz anotacija i preko *mne.Epochs* kreiraju vremenski intervale za skup podataka kojem se prosljeđuju događaji kao i minimalno i maksimalno vrijeme intervala [33].

## 7.4.5. Korekcija osnovne linije

Korekcija osnovne linije primjenjuje se za uklanjanje pomaka i drugih niskofrekventnih artefakata iz EEG signala. Ovo uključuje oduzimanje srednjeg signala zabilježenog tijekom pre-stimulacijskog ili neutralnog razdoblja od podataka koji su povezani s događajem (ERP) ili EEG podataka od interesa. Korekcija osnovne linije osigurava da se promjene u EEG signalu pripisuju stvarnoj moždanoj aktivnosti, a ne vanjskim čimbenicima, čime se poboljšava točnost modela strojnog učenja u otkrivanju kognitivnih ili motoričkih stanja.

```
epochs.apply_baseline(baseline=(-0.2, 0))
```

Već se preko epochiranja i detekcije događaja može napraviti korekcija osnovne linije, ali ako bi se ona napravila naknadno, to bi se odradilo preko metode *apply\_baseline* koja prima i uklanja niskofrekventne pomake u epohama [33].

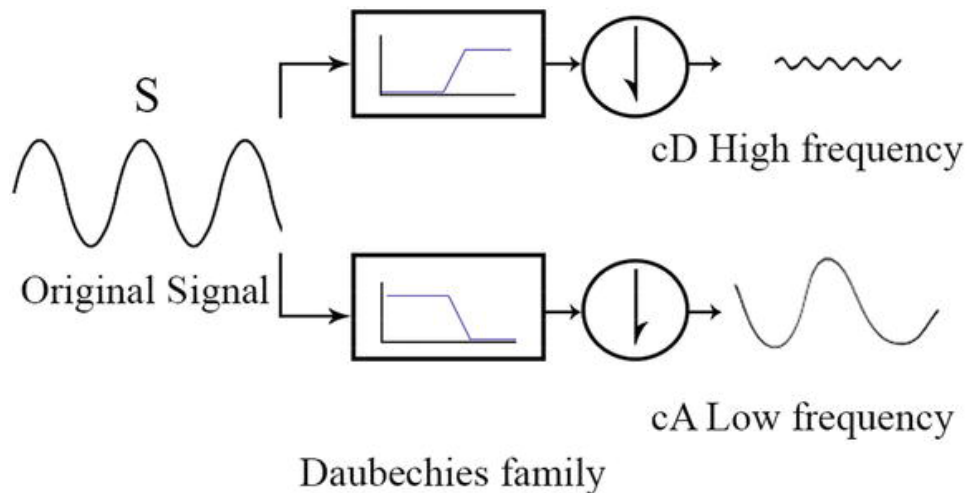
## 7.4.6. Ekstrakcija značajki

Ekstrakcija značajki pretvara sirove EEG podatke u skup informativnih atributa ili značajki koje se mogu koristiti u algoritmima strojnog učenja. Uobičajene značajke uključuju gustoće spektralne snage, frekvencijske pojaseve (npr. alfa, beta) i značajke u vremenskoj domeni. Cilj je smanjiti dimenzionalnost podataka uz zadržavanje ključnih informacija koje razlikuju različita moždana stanja. Ekstrakcija značajki ključni je korak u BCI sustavima, omogućujući klasifikaciju moždane aktivnosti u stvarnom vremenu korištenjem modela strojnog učenja.

```
def extract_features(signal, window_size):  
    n_windows = len(signal) // window_size  
    features = []  
  
    for i in range(n_windows):  
        window = signal[i*window_size:(i+1)*window_size]  
        mean = np.mean(window)  
        variance = np.var(window)  
        energy = np.sum(window**2)  
        features.append([mean, variance, energy])  
  
    return np.array(features)  
  
window_size = int(sampling_rate * 2)
```

```
features = [extract_features(signal_data[i], window_size) for i in range(n_signals)]
```

Programski kod prikazan iznad računa aritmetičku sredinu, varijancu i standardnu devijaciju. Putem funkcije *extract\_features* koja prima signal podataka i veličinu ekrana. Funkcija računa aritmetičku sredinu, varijancu i energiju za svaki signal podatka u 2-sekundom razdoblju [34].



Slika 8. Razdvajanje signala u niske i visoke frekvencije

izvor: <https://cdnintech.com/media/chapter/83501/1680006060/media/F4.png>

## 7.5. Odabir značajki i klasifikacija

Nakon pretprocesiranja podataka u analizi sučelja mozak-računalo (BCI), sljedeći korak je odabir značajki i klasifikacija.

### 7.5.1. Odabir značajki

Nakon pretprocesiranja, podaci su i dalje visoke dimenzionalnosti, što može uključivati šum i redundantne informacije. Kako bismo poboljšali performanse modela strojnog učenja, potrebno je izdvojiti relevantne značajke. Uobičajeni značajku uključuju:

- Značajke u vremenskoj domeni

Srednja vrijednost, varijanca ili statistički pokazatelji signala višeg reda.

- Značajke u frekvencijskoj domeni

Snaga spektralne gustoće, snaga unutar specifičnih frekvencijskih opsega (alfa, beta itd.).

- Prostorne značajke

Korištenje metoda poput uobičajenih prostornih obrazaca (CSP) za smanjenje dimenzionalnosti i izolaciju značajnih komponenti podataka.

## 7.5.2. Klasifikacija

Nakon izdvajanja značajki, sljedeći zadatak je klasificiranje stanja mozga ili namjera korisnika na temelju tih značajki. Postoje brojni algoritmi koji se mogu koristiti u fazi klasifikacije, a najpopularniji od njih su: Podrška vektorima, Nasumične Šume i Neuronske mreže.

### 7.5.2.1. Podrška vektorima (SVM)

Support Vector Machines (SVM) su strojno učiteljski algoritmi koji se često koriste za klasifikaciju EEG podataka u okviru sučelja mozak-računalo (BCI). SVM funkcionira tako da pronalazi hiper-ravninu koja optimalno razdvaja podatke u različite klase. SVM je posebno koristan u situacijama kada su podaci linearno separabilni, ali može se proširiti za nelinearne probleme korištenjem kernela, kao što su radijalne bazne funkcije (RBF).

Jedna od glavnih prednosti SVM-a je njegova robusnost u radu s visokodimenzionalnim podacima, poput EEG signala, i otpornost na pretreniranje. Međutim, algoritam može biti računalno intenzivan kada se koristi na vrlo velikim skupovima podataka, ali njegova preciznost u klasifikaciji često opravdava ovaj trošak. SVM se koristi za razne zadatke u BCI, uključujući klasifikaciju pokreta, zamisli pokreta, te emocionalnih stanja korisnika.

Kako bi se izradila SVM analiza Pythonu preko *scikit-learn*, koji se prethodno instalirao preko pip komande, potrebno je uvesti

```
import pyedflib

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, classification_report

from sklearn.model_selection import GridSearchCV
```

Kako su epohe već kreirane u ranijem koraku pretprocesiranja potrebno je stvoriti X i Y vrijednosti. X predstavlja sirove podatke i njegove značajke u matrici, dok Y predstavlja labelu

za klasifikaciju. X se preko Vectorizera i njegove metode *fit\_transform* kreira kojemu se prosleđuju podatci iz epohe. Metoda *epochs.get\_data()* izvlači EEG podatke u obliku polja [broj epoha, broj kanala, broj vremena]. Y predstavlja labelu vektora i kreira se preko *epochs.events* metode.

```
X = np.hstack(features)
Y = np.random.randint(0, 2, X.shape[0])
```

Zatim se kreiraju skupovi za treniranje i testiranje. Jedan za svaki X i Y znači *x\_train*, *x\_test*, *y\_train*, *y\_test*. Oni se kreiraju uporabom metode *train\_test\_split*.

```
X_train, X_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=42)
```

Kada su kreirani skupovi za treniranje i testiranje potrebno je inicirati SVM, pokrenuti treniranje modela i provesti predviđanje na testnom skupu

```
svm = SVC(kernel='linear')
svm.fit(X_train, y_train)
y_pred = svm.predict(X_test)
```

Kada su odrađeni svi koraci SVM analize potrebno je prikazati točnost modela [35].

```
accuracy = accuracy_score(Y_test, Y_pred)
print(f"Test Accuracy: {accuracy:.2f}")
print("Classification Report:")
print(classification_report(Y_test, Y_pred))
```

```
Test Accuracy: 0.17
Classification Report:
              precision    recall  f1-score   support

     0           0.00      0.00      0.00         3
     1           0.25      0.33      0.29         3

 accuracy                   0.17         6
 macro avg              0.12      0.17      0.14         6
 weighted avg           0.12      0.17      0.14         6
```

Slika 9. Klasifikacijska matrica za SVM

### 7.5.2.2. Nasumične šume (RF)

Random Forest (RF) je popularan algoritam za klasifikaciju EEG signala, poznat po svojoj otpornosti na pretreniranje i sposobnosti rada s nestrukturiranim podacima. RF je skup metoda stabla odluka, gdje svako stablo generira klasifikaciju, a konačna odluka se donosi na temelju glasanja svih stabala. RF koristi nasumične uzorke iz skupa podataka za izgradnju svakog stabla, čime se smanjuje varijanca i povećava preciznost.

Jedna od ključnih prednosti RF-a je njegova otpornost na šum u podacima i sposobnost da radi s velikim količinama EEG podataka bez gubitka točnosti. Zbog toga se RF često koristi za klasifikaciju kompleksnih moždanih stanja, kao što su razlike između motoričkih i nemotoričkih aktivnosti. RF je vrlo prilagodljiv i može se koristiti u različitim BCI primjenama, uključujući kontrolu neuroprostetike i analizu mentalnih stanja.

Random Forest algoritam u Pythonu se također implementira putem *scikit-learn*. Prvo je potrebno uključiti sami algoritam *RandomForestClassifier*.

```
import pyedflib

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score
```

Zatim je potrebno inicijalizirati sami model preko metode *RandomForestClassifier* kojoj se proslijeđuju broj procjenitelja i random stanje. Na temelju već prethodno kreiranih podataka za treniranje i testiranje provodi se treniranje modela i provodi se predviđanje nad testnim skupom. Na kraju se prikazuje točnost modela [36].

```
rf = RandomForestClassifier(n_estimators=100, random_state=42)

rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy * 100:.2f}%')
```

### 7.5.2.3. Neuronske mreže (NN)

Neuronske mreže (eng. Neural Network, NN) su složeni modeli strojnog učenja koji se sve češće koriste u klasifikaciji EEG podataka. Konvolucijske neuronske mreže (CNN) i rekurentne neuronske mreže (RNN) pokazale su izvanredne rezultate u prepoznavanju obrazaca u EEG signalima. CNN-ovi su posebno učinkoviti u analizi prostorno-vremenskih



podataka, kao što su EEG signali, dok su RNN-ovi pogodni za sekvencijalne podatke jer mogu zadržati informacije o prošlim stanjima kroz vremenske korake.

NN modeli mogu automatski učiti značajke iz EEG podataka, što ih čini korisnima za složene zadatke poput prepoznavanja zamisli pokreta ili emocionalnih stanja. Iako neuronske mreže zahtijevaju puno računalne snage i često duge periode treniranja, njihov potencijal u BCI sustavima je ogroman, posebno u kombinaciji s velikim datasetima i naprednim metodama optimizacije.

Algoritam Neuronski Mreža u Pythonu se provodi preko raznih biblioteka, te ako neke od njih nisu instalirane potrebno je instalirati putem komande

```
Pip install pyedflib numpy matplotlib scipy scikit-learn tensorflow
```

Nakon instalacije potrebno je uključiti potrebne klase za implementaciju algoritma neuronskih mreža.

```
import pyedflib
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import butter, filtfilt
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Flatten,
Dense
```

Kako su skupovi za treniranje i testiranje već izrađeni može se kreirati model neuronskih mreža koji se inicijalizira preko *Sequential()*. Modelu je također potrebno dodati filter, veličinu, aktivaciju i slično.

```
model = Sequential()
model.add(Conv1D(filters=32, kernel_size=3, activation='relu',
input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Zatim slijedi kompiliranje modela preko metode *compile* kojoj se prosljeđuju gubitak, optimizier i metrika. Isti model se putem metode *fit* trenira, metodi se prosljeđuju *x\_train*, *y\_train* (podatci za treniranje i testiranje), te broj epoha kao i veličina batch posla. Provodi se predviđanje modela kao i ispis točnosti modela [37].

```

model.compile(loss='binary_crossentropy',          optimizer='adam',
metrics=['accuracy'])

model.fit(X_train,      Y_train,      epochs=10,      batch_size=32,
validation_data=(X_test, Y_test))

loss, accuracy = model.evaluate(X_test, Y_test)

print(f"Točnost neuronske mreže: {accuracy:.2f}")

```

```

Epoch 1/10
1/1 ██████████ 1s 1s/step - accuracy: 0.5417 - loss: 6.8865 - val_accuracy: 0.3333 - val_loss: 25.7104
Epoch 2/10
1/1 ██████████ 0s 51ms/step - accuracy: 0.6667 - loss: 8.5586 - val_accuracy: 0.3333 - val_loss: 19.1947
Epoch 3/10
1/1 ██████████ 0s 54ms/step - accuracy: 0.9583 - loss: 0.8219 - val_accuracy: 0.1667 - val_loss: 13.3791
Epoch 4/10
1/1 ██████████ 0s 55ms/step - accuracy: 1.0000 - loss: 1.6697e-05 - val_accuracy: 0.3333 - val_loss: 10.2275
Epoch 5/10
1/1 ██████████ 0s 55ms/step - accuracy: 0.9583 - loss: 0.0030 - val_accuracy: 0.1667 - val_loss: 9.7034
Epoch 6/10
1/1 ██████████ 0s 54ms/step - accuracy: 1.0000 - loss: 3.6650e-05 - val_accuracy: 0.3333 - val_loss: 10.4985
Epoch 7/10
1/1 ██████████ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0026 - val_accuracy: 0.3333 - val_loss: 11.8625
Epoch 8/10
1/1 ██████████ 0s 53ms/step - accuracy: 0.9583 - loss: 0.0341 - val_accuracy: 0.3333 - val_loss: 12.8500
Epoch 9/10
1/1 ██████████ 0s 53ms/step - accuracy: 1.0000 - loss: 0.0020 - val_accuracy: 0.3333 - val_loss: 13.6694
Epoch 10/10
1/1 ██████████ 0s 53ms/step - accuracy: 1.0000 - loss: 6.2839e-04 - val_accuracy: 0.3333 - val_loss: 14.3800
1/1 ██████████ 0s 23ms/step - accuracy: 0.3333 - loss: 14.3800
Točnost neuronske mreže: 0.33

```

Slika 10. Prikaz provedbe algoritma neuronskih mreža i njegove točnosti

#### 7.5.2.4. Stablo Odlučivanja (DT)

Stablo odlučivanja (eng. Decision tree, DT) je popularan algoritam za nadzirano učenje, često korišten za klasifikacijske zadatke. U kontekstu podataka iz sučelja mozak-računalo (BCI), stabla odlučivanja su korisna zbog svoje razumljivosti i sposobnosti rješavanja nelinearnih odnosa podataka. Svaki unutarnji čvor predstavlja odluku temeljenu na značajki, dok završni čvorovi predstavljaju izlazne klase, što model čini intuitivnim i lako vizualiziranim.

U BCI skupovima podataka, stablo odluke može se koristiti za klasifikaciju moždanih signala u različita mentalna stanja, poput motornih zadataka ili stanja mirovanja. Značajke, poput spektralne gustoće snage, koriste se za treniranje stabla odluke kako bi se klasificirala korisnikova namjera.

Stablo odlučivanja se može implementirati u Pythonu putem *scikit-learn*. Već ranije je navedena biblioteka instalirana, pa je potrebno samo uključiti potrebne klase iz biblioteke.

```
import pyedflib

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score,
classification_report

from sklearn import tree
```

Potrebno je razdvojiti na podatke za treniranje i testiranje (*x\_train*, *x\_test*, *y\_train*, *y\_test*), zatim je potrebno inicijalizirati model putem konstruktora klase *DecisionTreeClassifier*. Model se preko metode *fit* trenira, metodi se prosljeđuju *x\_train*, *y\_train* pa se provjerava predviđanje modela kao i klasifikacijsko izvješće o stablu odlučivanja [38].

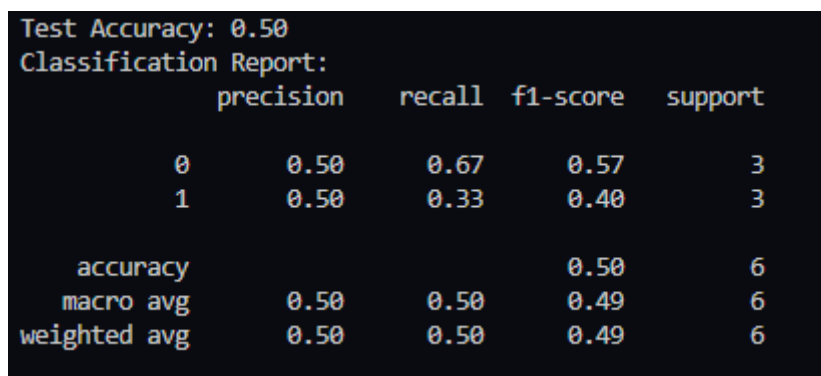
```
clf = DecisionTreeClassifier(random_state=42)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

print(f"Test Accuracy: {accuracy:.2f}")

print(classification_report(y_test, y_pred))
```



Test Accuracy: 0.50				
Classification Report:				
	precision	recall	f1-score	support
0	0.50	0.67	0.57	3
1	0.50	0.33	0.40	3
accuracy			0.50	6
macro avg	0.50	0.50	0.49	6
weighted avg	0.50	0.50	0.49	6

Slika 11. Klasifikacijski izvještaj (DT)

#### 7.5.2.5. Linearna Diskriminantna Analiza (LDA)

Linearna diskriminantna analiza (LDA) je klasifikacijska metoda koja pronalazi linearne kombinacije značajki koje najbolje razdvajaju dvije ili više klasa. LDA se često koristi u analizi podataka iz moždano-računalne interakcije (BCI) zbog svoje učinkovitosti pri radu s podacima

koji sadrže više dimenzija, kao što su EEG signali. Metoda radi tako da maksimizira razliku između klasa dok minimizira unutar-klasnu varijaciju.

Kod BCI podataka, LDA može klasificirati različite mentalne stanja ili zadatke na temelju EEG signala. Značajke, poput spektralne gustoće ili vremenskih karakteristika, koriste se za treniranje LDA modela. Nakon što model nauči optimalne razlike među klasama, može precizno predvidjeti novu nepoznatu klasu.

LDA se može implementirati u Pythonu preko biblioteke *scikit-learn*. Potrebno je izvesti brojne biblioteke, najvažnija od njih *LinearDiscriminantAnalysis*.

```
import pyedflib

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

from sklearn.metrics import accuracy_score
```

Potrebno je razdvojiti na podatke za treniranje i testiranje (*x\_train*, *x\_test*, *y\_train*, *y\_test*) putem klase *train\_test\_split* koja dijeli podatke 80%-20% u odnosu trening:test, zatim je potrebno inicijalizirati model putem konstruktora klase *LinearDiscriminantAnalysis*. Model se preko metode *fit* trenira, metodi se prosljeđuju *x\_train*, *y\_train* pa se provjerava predviđanje modela kao i klasifikacijsko izvješće o LDA [39].

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

lda = LinearDiscriminantAnalysis()

lda.fit(X_train, y_train)

y_pred = lda.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy * 100:.2f}%')
```

## 7.6. Analiza rezultata

U ovom poglavlju će se obaviti analiza rezultata dobivenih tijekom koraka predobrade, ekstrakcije značajki i klasifikacije podataka iz BCI skupa. Svaki korak u procesu obrade podataka pridonio je ukupnim performansama sustava, što se odrazilo na ključnim metrima poput točnosti i preciznosti. Kako se radi o velikom skupu podataka kao reprezentativne uzorke u svrhu provedbe algoritama i testiranja odabrano je prvih 10 korisnika. Kako za svakog

korisnika postoji snimljeno 14 snimaka odabrano su prve dvije radi prikladnosti. Sve snimke spremljene su u .edf datoteku. Primjer imenovanja izgleda S001R01.edf, S001R02.edf i tako sve do S010R01.edf. U svrhu analize rezultata ključna metrika za računanje točnosti modela bit će IPMS Hipoteza.

Algoritmi strojnog učenja uvelike su olakšali analizu BCI skupa, tj. EEG skupa podataka koji sadrži podatke o moždanim signalima. Algoritmi korišteni u analizi BCI skupa su: podrška vektorima (SVM), linearna diskriminantna analiza (LDA), nasumične šume (RF), neuronske mreže (NN) i stablo odlučivanja (DT).

Kanali koji sadrže glavne informacije i koji će biti ključni u radu algoritama su: FC1, FC2, FC4, FC6, FCz, C1, C2, C3, C4, C5, C6, i Cz.

U analizi će se koristiti razni grafikoni koji su prikladni za različite algoritme. Neki od njih su konfuzijska matrica, 1D scatter plot, krivulja učenja.

### **7.6.1. IPMS Hipoteza**

IPMS (eng. Immediat Previous Mental State) hipoteza odnosi se na koncept u interakciji mozak-računalo (BCI) i kognitivnoj neuroznanosti, fokusirajući se na utjecaj nedavnog mentalnog ili kognitivnog stanja osobe na trenutnu moždanu aktivnost i ponašanje. IPMS pretpostavlja da neposredno prethodno mentalno stanje ima mjerljiv učinak na to kako mozak obrađuje nove podražaje, što može utjecati na interpretaciju podataka u BCI aplikacijama. Ova hipoteza često se koristi u studijama koje uključuju analizu EEG podataka u stvarnom vremenu i tehnike strojnog učenja za adaptivne BCI sustave [40].

### **7.6.2. Konfuzijska matrica**

Konfuzijska matrica je tablica koja se koristi za procjenu performansi modela klasifikacije. Ona prikazuje stvarne klase i predviđene klase te pruža detaljan uvid u to koliko je model ispravno ili pogrešno klasificirao uzorke. Tablica sadrži četiri osnovne komponente:

1. True Positive (TP): Ispravno predviđene pozitivne klase.
2. True Negative (TN): Ispravno predviđene negativne klase.
3. False Positive (FP): Pogrešno predviđene pozitivne klase (tip I greška).
4. False Negative (FN): Pogrešno predviđene negativne klase (tip II greška).

Ova analiza pomaže u mjerenju točnosti, preciznosti, povratnosti i F1 mjere modela [41].

### 7.6.3. Rezultati

U tablici ispod prikazani su podatci točnosti modela za svaki od 5 korištenih algoritama strojnog učenja. Svaka od snimki spremljenih u .edf datotekama je pokrenuta samo jednom. Rezultati u tablici ne uzimaju u obzir IPMS hipotezu.

Tablica 1. Klasifikacija bez uzimanja u obzir IPMS hipotezu

Korisnici	SVM	RF	NN	DT	LDA
S001R01	60.0	50.0	60.0	33.3	40.0
S001R02	53.3	33.3	26.7	53.3	40.0
S002R01	80.0	80.0	60.0	66.7	66.7
S002R02	66.7	73.3	66.7	80.0	66.7
S003R01	60.0	66.7	26.7	20.0	40.0
S003R02	73.3	80.0	73.3	26.0	60.0
S004R01	53.3	50.0	53.3	26.7	33.3
S004R02	73.3	33.3	53.3	66.7	73.3
S005R01	93.3	33.3	60.0	73.3	60.0
S005R02	60.0	26.7	26.7	26.7	26.7
S006R01	53.3	46.7	46.7	26.7	53.3
S006R02	53.3	66.7	46.7	53.3	40.0
S007R01	60.0	93.3	66.7	40.0	53.3
S007R02	80.0	66.7	33.3	73.3	66.7
S008R01	73.3	53.3	60.0	33.3	60.0
S008R02	66.7	53.3	26.7	13.3	46.7
S009R01	53.3	40.0	60.0	66.7	46.7
S009R02	53.3	60.0	66.7	60.0	46.7
S010R01	73.3	50.0	33.3	86.7	53.3
S010R02	53.3	73.3	33.3	40.0	33.3

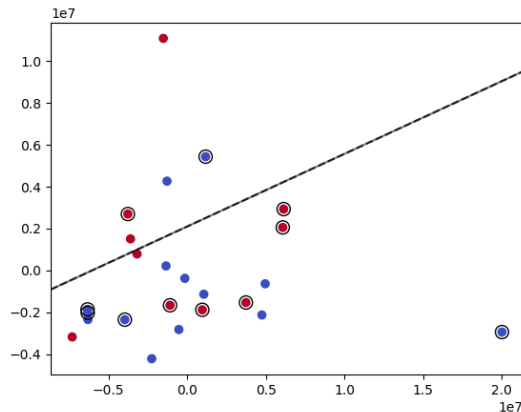
Rezultati u tablici ispod prikazuju točnost klasifikacije uzimajući u obzir IPMS Hipotezu.

*Tablica 2. Klasifikacija uzimajući u obzir IPMS hipotezu*

Korisnici	SVM	RF	NN	DT	LDA
S001R01	66.7	25.0	60.0	40.0	33.3
S001R02	73.3	73.3	60.0	73.3	66.7
S002R01	53.3	66.7	33.3	33.3	86.7
S002R02	86.7	66.7	46.7	73.3	60.0
S003R01	53.3	40.0	60.0	46.7	46.7
S003R02	53.3	60.0	73.3	93.3	66.7
S004R01	60.0	93.3	60.0	86.7	53.3
S004R02	53.3	86.7	53.3	60.0	46.7
S005R01	66.7	40.0	46.7	60.0	60.0
S005R02	66.7	50.0	60.0	73.3	80.0
S006R01	53.3	46.7	46.7	73.3	60.0
S006R02	66.7	53.3	80.0	80.0	73.3
S007R01	53.3	40.0	46.7	60.0	46.7
S007R02	66.7	80.0	40.0	46.7	33.3
S008R01	80.0	73.3	46.7	80.0	46.7
S008R02	53.3	33.3	40.0	73.3	73.3
S009R01	73.3	33.3	53.3	73.3	60.0
S009R02	53.3	46.7	66.7	60.0	46.7
S010R01	53.3	60.0	46.7	53.3	53.3
S010R02	60.0	33.3	60.0	86.7	26.7

## 7.6.4. Podrška vektorima (SVM)

Scatter plot dijagram s dvije različite skupine točaka, plavom i crvenom. One predstavljaju različite klase i neke od njih imaju crni obrub. One predstavljaju podskup točaka u SVM grafu. Isprekidana linija koja prolazi kroz dijagram predstavlja granicu odluke, linija bi trebala razdvojiti dvije klase (crvene i plave točke). Radi lakše analize i dobivanja rezultata prikazane su samo dvije vrste točaka dok se u BCI skupu EEG podataka tražilo 192 podatka (3\*64 kanala).

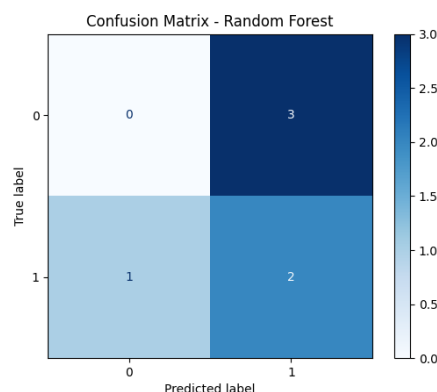


Slika 12. Scatter plot za SVM - 50.0 % točnost

Scatter plot graf iznad ima uspješnost SVM modela na predviđanju od 50.0 %. Neke od točaka na grafu su pogrešno klasificirane dok točke sa crnim obrubom predstavljaju marginu između dvije klase.

## 7.6.5. Nasumične šume (RF)

U analizi s pomoću algoritma nasumičnih šuma, prikazat će se dvije matrice konfuzije. Predviđanje točnosti modela jedne matrice je 33.3 % dok je predviđanje modela druge 50.0 %.

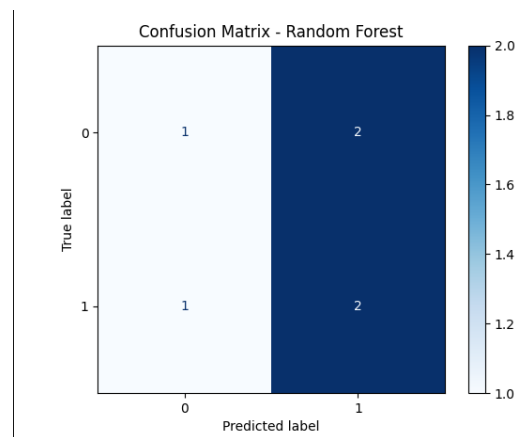


Slika 13. Konfuzijska matrica - 33.3 % točnost



Ova konfuzijska matrica iz modela nasumičnih šuma pokazuje sljedeće:

- Za pravu klasu 0: Model nije točno predvidio klasu 0 ni jednom, ali je pogrešno klasificirao 3 instance kao klasu 1.
- Za pravu klasu 1: Model je ispravno predvidio klasu 1 dva puta, ali je pogrešno klasificirao 1 instancu kao klasu 0.
- Ukupna izvedba pokazuje neravnotežu u predviđanjima klasa.



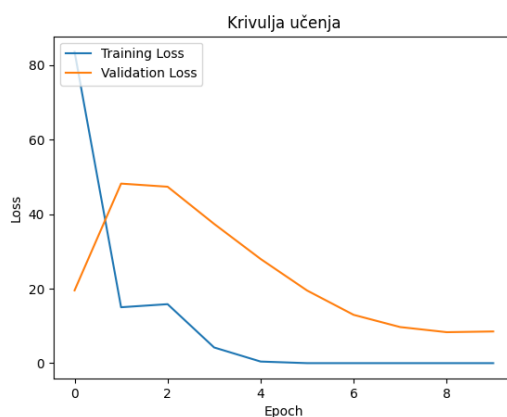
Matrica konfuzije za Random Forest model prikazuje performanse klasifikatora na dvije klase:

- Istinski pozitivni (TP): Klasa 0 predviđena kao 0 pojavljuje se 1 put.
- Lažno pozitivni (FP): Klasa 1 predviđena kao 0 pojavljuje se 1 put.
- Lažno negativni (FN): Klasa 0 predviđena kao 1 pojavljuje se 2 puta.
- Istinski negativni (TN): Klasa 1 predviđena kao 1 pojavljuje se 2 puta.

To pokazuje da klasifikator ima poteškoće s ispravnim klasificiranjem klase 0 te je pristran prema predviđanju klase 1.

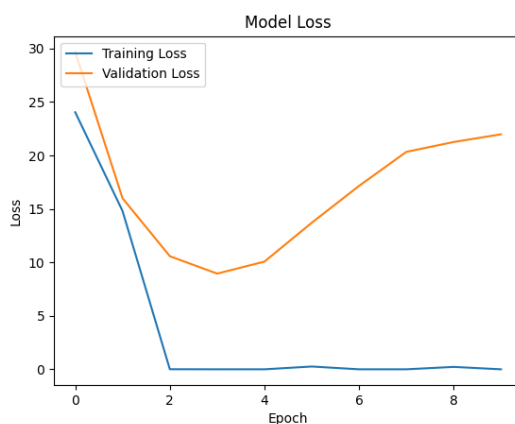
### 7.6.6. Neuronske mreže (NN)

Analiza putem algoritma neuronskih mreža koristit će se graf krivulje učenja za demonstraciju prikaza rezultata. Prva krivulja učenja ima točnost 67 % dok druga ima točnost 33 %. Krivulja učenja prikazuje gubitak podataka za testiranje i treniranje po epochama.



Slika 14. Krivulja učenja – 67 % točnost

- Trening gubitak: U početku se smanjuje, a zatim stabilizira na nuli ili blizu nje, što znači da model dobro pristaje na trening podatke.
- Validacijski gubitak: Počinje visok, ali se postupno smanjuje kroz epohe, što ukazuje na dobru generalizaciju na validacijske podatke, bez znakova prenapučenosti (eng. Overfitting).



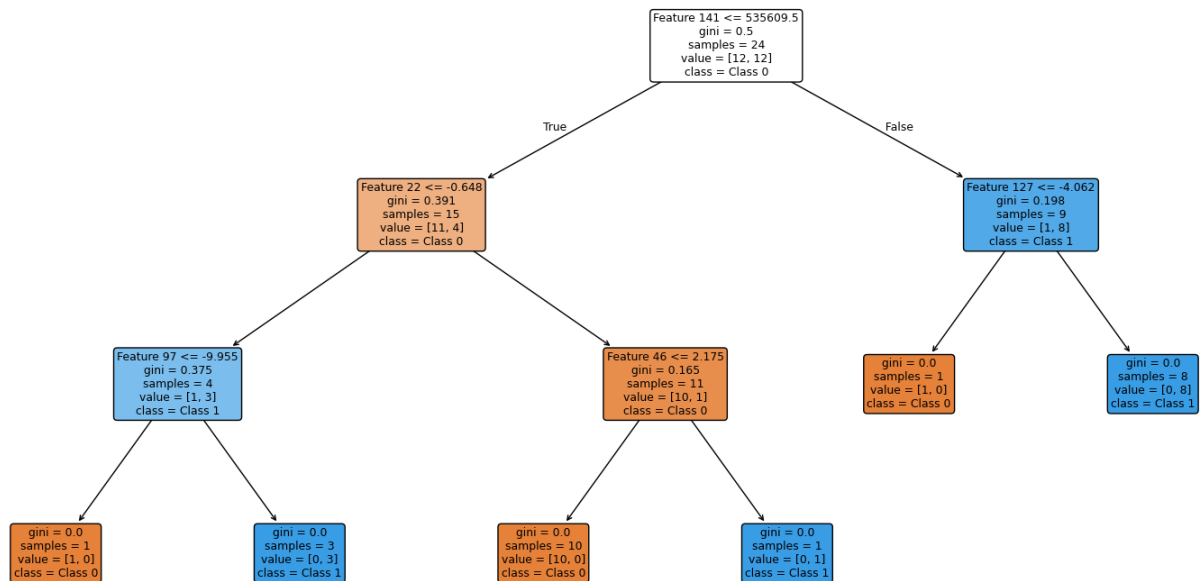
Slika 15. Krivulja učenja - 33% točnost

- Trening gubitak: Stabilan, ne raste, i vjerojatno vrlo nizak ili ravan.
- Validacijski gubitak: Raste kroz epohe, što sugerira prenapučenosti. Model pamti podatke za treniranje, ali ne uspijeva generalizirati na neviđene validacijske podatke.

### 7.6.7. Stablo odlučivanja (DT)

Analiza stabla odlučivanja predstavlja binarni klasifikacijski problem. Ključni elementi stabla su čvorovi gdje je korijenski čvor značajka 141. Pravilo odluke u svakom čvoru prikazano

je kao *Značajka*  $x \leq \text{vrijednost}$  gdje se odluka temelji na tome je li vrijednost manja ili jednaka pragu. Gini indeks mjeri čistoću podjele (bliže nuli znači veća čistoća).



Slika 16. Decision tree - 83 % točnost

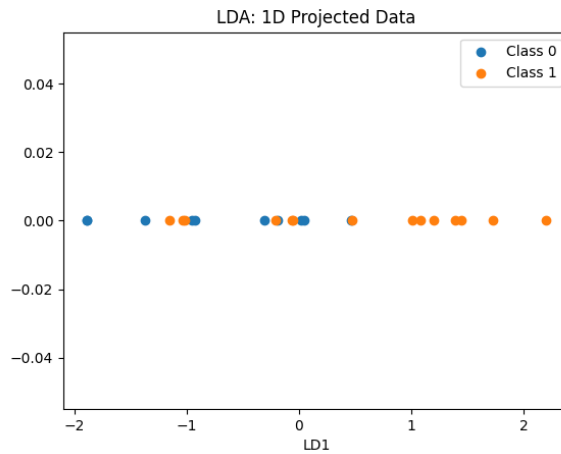
Korijenski čvor stabla dijeli se na *Značajka*  $141 \leq 535609.5$ , gdje je gini indeks 0.5, što pokazuje na savršeno miješanu skupinu od 12 uzoraka u svakoj klasi.

Na lijevoj strani stabla podjela se vrši na *Značajka*  $22 \leq -0.648$ , što smanjuje gini indeks na 0.391. Ovaj čvor sadrži 15 uzoraka, s time da je 11 u prvoj klasi a 4 u drugoj. Zatim se stablo dijeli na dva čvora gdje je u svakome gini indeks 0.0 što ukazuje na čistu klasifikaciju bez nejasnoća.

Na desnoj strani stabla podjela se vrši na *Značajka*  $127 \leq -4.062$ , što smanjuje gini indeks. Isto kao i na lijevoj strani stablo se dijeli na dva lista na kojima je gini indeks 0.0.

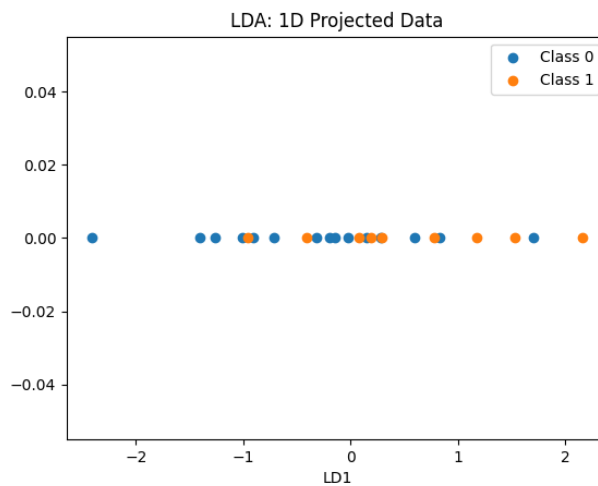
### 7.6.8. Linearna diskriminantna analiza (LDA)

Za analizu putem linearne diskriminantne analize (LDA) koristit će se 1D scatter plot koji kao metrike uspješnosti ima jasno razdvajanje klasa i preklapanja. Što su skupovi jasno razdvojeni LDA je uspješno maksimizirala odvajanje klasa.



*Slika 17. 1D Scatter plot - 83.33 % točnost*

Na 1D scatter plot grafu prikazanom iznad, stopa predviđanja točnosti je 83.33 % i to se vidi u stvari po poziciji plavih i narančastih točkica raspodijeljenih na grafu. Plave točkice su na lijevoj strani tj. manje od nule dok su narančaste više na desnoj ali ne i sve pa se iz tog razloga točnost modela smanjuje.



*Slika 18. 1D Scatter plot - 33.33 % točnost*

U slučaju ovog 1D scatter plot grafa, može se vidjeti da je dosta i plavih i narančastih točkica na sredini grafikona tj. postoji dosta preklapanja između klasa i to sugerira da model ima poteškoća u razlikovanju tih klasa.

## 8. Zaključak

U ovom radu istražio sam različite tehnike analize podataka i strojnog učenja kako bismo poboljšali točnost i učinkovitost sustava za interakciju mozak-računalo (BCI). Cilj je bio odgovoriti na ključne izazove u obradi i interpretaciji EEG signala, koji su često složeni i šumni. Korištenjem naprednih algoritama strojnog učenja, kao što su Linearna diskriminantna analiza (LDA), strojevi potpornih vektora (SVM), Neuronske mreže (NN), Nasumične šume (RF), stablo odlučivanja (DT) pokazao sam potencijal za izdvajanje korisnih obrazaca iz moždanih signala i njihovu pretvorbu u naredbe za vanjske uređaje.

Ova analiza naglašava važnost pretprocesiranja podataka, ekstrakcije značajki i tehnika smanjenja šuma u BCI domeni. Pažljivim odabirom i primjenom tehnika poput Nezavisne komponente analize (ICA) i filtriranja na temelju frekvencije, uspio sam značajno poboljšati kvalitetu EEG podataka korištenih za klasifikaciju. Ovi koraci pretprocesiranja igrali su ključnu ulogu u poboljšanju performansi modela strojnog učenja.

Rezultati ove analize sugeriraju da kombiniranje različitih pristupa strojnog učenja, ovisno o specifičnom zadatku, može optimizirati ukupne performanse BCI sustava. Osim toga, kako tehnologija nastavlja napredovati, integracija tih sustava s učinkovitijim algoritmima i hardverom dodatno će proširiti mogućnosti BCI sustava.

## Popis literature

- [1] „What is brain-computer interface (BCI), TechTarget Network“ [Online] Dostupno: <https://www.techtarget.com/whatis/definition/brain-computer-interface-BCI>.
- [2] "The history of Brain-Computer Interfaces (BCIs) - Timeline," RoboticsBiz. [Online]. Dostupno: <https://roboticsbiz.com/the-history-of-brain-computer-interfaces-bcis-timeline/>.
- [3] "The History Of Brain-Computer Interface Technology - When, Where, And How It Started?," The Brainy Bits. [Online]. Dostupno: <https://thebrainybits.com/the-history-of-brain-computer-interface-technology/>.
- [4] R. Aladjem and J. Platt, "2024 Is the Year for Brain-Computer Interfaces," Technology Networks, 11. Srpanj 2024. [Online]. Dostupno: [www.technologynetworks.com](http://www.technologynetworks.com).
- [5] A. McFarland and D. McCane, "Brain-computer interfaces for augmented control of prosthetic limbs", IEEE Xplore, 2023. [Online] Dostupno: <https://ieeexplore.ieee.org/document/9761043>.
- [6] M. Krauledat, K. R. Müller, and G. Curio, "Advances in EEG-based brain-computer interfaces," Journal of Neural Engineering, vol. 4, no. 3, pp. 96-104, 2023. [Online]. Dostupno: <https://iopscience.iop.org/article/10.1088/1741-2560/4/3/003>.
- [7] „What are Neural Implants“, DeltecBank. [Online]. Dostupno: <https://www.deltecbank.com/news-and-insights/what-are-neural-implants/>.
- [8] „Summary of Over Fifty Years with Brain-Computer Interface – A Review“, National Library of Medicine. [Online] Dostupno: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7824107/>.
- [9] „Types of Machine Learning“, Geeks for Geeks. [Online] Dostupno: <https://www.geeksforgeeks.org/types-of-machine-learning/>.
- [10] „A guide to the types of machine learning algorithms and their applications“, SAS – Katrina Wakefield. [Online] Dostupno: [https://www.sas.com/en\\_gb/insights/articles/analytics/machine-learning-algorithms.html](https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html).
- [11] „Advantages and Disadvantages of Machine Learning“, Geeks for Geeks. [Online] Dostupno: <https://www.geeksforgeeks.org/what-is-machine-learning/?ref=shm>.
- [12] Christoph Guger, Günter Edinger, Gunther Krausz, „Hardware/Software Components and Applications of BCIs“, [Online] Dostupno: [https://www.researchgate.net/publication/221910598\\_HardwareSoftware\\_Components\\_and\\_Applications\\_of\\_BCI](https://www.researchgate.net/publication/221910598_HardwareSoftware_Components_and_Applications_of_BCI).

- [13] Yosuke Fujitora, Junichi Ushiba, „Deep Residual Convolutional Neural Networks for Brain–Computer Interface to Visualize Neural Processing of Hand Movements in the Human Brain“, *Frontiers*, Volume 16, 20. Svibanj 2022. [Online] Dostupno: <https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2022.882290/full>
- [14] Thorsten Wuest i Co., „Machine learning in manufacturing: advantages, challenges, and applications“, Volume 4 2016. [Online] Dostupno: <https://www.tandfonline.com/doi/full/10.1080/21693277.2016.1192517#d1e204>.
- [15] Suvankas Das, „Advantages and disadvantages of Machine Learning“, Ello 2024. [Online] Dostupno: <https://ello.io/advantages-and-disadvantages-of-machine-learning/>.
- [16] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767-791, 2002. [Online]. Dostupno: <https://www.sciencedirect.com/science/article/pii/S1388245702000573>.
- [17] P. D. Ganesh Kumar, D. Selvakumar, and S. Siva Prasanna, "Neuroprosthetics: Trends, challenges, and future directions," *Journal of Neural Engineering*, vol. 15, no. 4, pp. 1-14, 2018.
- [18] N. Birbaumer and L. G. Cohen, "Brain-computer interfaces: Communication and restoration of movement in paralysis," *Journal of Physiology*, vol. 579, no. 3, pp. 621-636, 2007. [Online]. Dostupno: <https://www.physiology.org/>.
- [19] A. Kübler, B. Kotchoubey, J. Kaiser, J. R. Wolpaw, and N. Birbaumer, "Brain-computer communication: Unlocking the locked in," *Psychological Bulletin*, vol. 127, no. 3, pp. 358-375, 2001. [Online]. Dostupno: <https://www.apa.org/pubs/journals/>.
- [20] S. B. Kotikalapudi, "Cognitive rehabilitation through brain-computer interfaces: Challenges and opportunities," *Journal of Neural Engineering*, vol. 10, no. 3, pp. 1-10, 2013. [Online]. Dostupno: <https://iopscience.iop.org/>.
- [21] J. D. Bayliss and D. H. Ballard, "A virtual reality testbed for brain-computer interface research," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 188-190, 2000. [Online]. Dostupno: <https://ieeexplore.ieee.org/document/847426>.
- [22] J. van de Laar, S. Gürkök, M. Poel, D. Heylen, and A. Nijholt, "Experiencing BCI control in a popular computer game," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 5, no. 2, pp. 176-184, Lipanj 2013. [Online]. Dostupno: <https://ieeexplore.ieee.org/document/6375137>.

- [23] G. Pfurtscheller, G. R. Müller-Putz, R. Scherer, and C. Neuper, "Rehabilitation with brain-computer interface systems," *IEEE Computer*, vol. 41, no. 10, pp. 58-65, Listopad 2008. [Online]. Dostupno: <https://ieeexplore.ieee.org/document/4634060>.
- [24] J. del R. Millán, F. Renkens, J. Mouriño, and W. Gerstner, "Noninvasive brain-actuated control of a mobile robot by human EEG," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1026-1033, Lipanj 2004. [Online]. Dostupno: <https://ieeexplore.ieee.org/document/1300802>.
- [25] J. Wolpaw and E. W. Wolpaw, „Brain-Computer Interfaces: Principles and Practice“, Oxford University Press, 2012.
- [26] F. Lotte, L. Bougrain, and A. Cichocki, "A review of classification algorithms for EEG-based brain-computer interfaces: a 10-year update," *Journal of Neural Engineering*, vol. 15, no. 3, Travanj 2018. [Online]. Dostupno: <https://iopscience.iop.org/article/10.1088/1741-2552/aab2f2>.
- [27] A. Nijholt, „Brain-Computer Interfaces for HCI and Games“, Springer, 2015. [Online]. Dostupno: <https://link.springer.com/book/10.1007/978-3-319-10978-7>.
- [28] Schalk, G.McFarland, D.J Hinterberger, T. Birbaumer, N. Wolpaw, J.R, BCI2000, „A General-Purpose Brain-Computer Interface (BCI) System. *IEEE Transactions on Biomedical Engineering* 51(6): 1034-1043, 2004. [Online] Dostupno: <https://physionet.org/content/eegmidb/1.0.0/#files-panel>.
- [29] Python, „pyEDFlib 0.1.38“. [Online] Dostupno: <https://pypi.org/project/pyEDFlib/>.
- [30] Neural Network Science, „Filtering EEG Data“, [Online] Dostupno: [https://neuralsciences.io/7-eeq/erp\\_filtering.html](https://neuralsciences.io/7-eeq/erp_filtering.html).
- [31] Neural Network Science, „Artifacts in EEG Data“, [Online] Dostupno: [https://neuralsciences.io/7-eeq/erp\\_artifacts.html](https://neuralsciences.io/7-eeq/erp_artifacts.html).
- [32] Neural Network Science, „Averaging and Re-referencing ERP-s“, [Online] Dostupno: [https://neuralsciences.io/7-eeq/erp\\_avg\\_reref.html](https://neuralsciences.io/7-eeq/erp_avg_reref.html).
- [33] Neural Network Science, „Segmentation into ERP epochs“, [Online] Dostupno: [https://neuralsciences.io/7-eeq/erp\\_segmentation.html](https://neuralsciences.io/7-eeq/erp_segmentation.html).
- [34] Kalaivaani Natarajan, „Feature extraction in EEG signals“, Medium. [Online] Dostupno: <https://medium.com/@kalaivaaninatarajan/feature-extraction-in-eeq-signals-f8e5dfb94cdf>.
- [35] Scikit learn, „Support Vector Machines“. [Online] Dostupno: <https://scikit-learn.org/stable/modules/svm.html>.



- [36] Adam Shafi „Random Forest Classification with Scikit-Learn“, Datacamp. [Online] Dostupno: <https://www.datacamp.com/tutorial/random-forests-classifier-python>.
- [37] Deborah Mesquita, „Python AI: How to Build a Neural Network & Make Predictions“, Real Python. [Online] Dostupno: <https://realpython.com/python-ai-neural-network/>.
- [38] Scikit learn, „Decision Trees“. [Online] Dostupno: <https://scikit-learn.org/stable/modules/tree.html>
- [39] Eda Kavlakoglu „Implementing linear discriminant analysis (LDA) in Python“, IBM. [Online] Dostupno: <https://developer.ibm.com/tutorials/awb-implementing-linear-discriminant-analysis-python/>.
- [40] Nabil Ajali-Hernández, Carlos M. Travieso-Gonzalez „Analysis of Brain Computer Interfaces Using Deep and Machine Learning“, IntechOpen. [Online] Dostupno: <https://www.intechopen.com/chapters/83501>.
- [41] Science Direct „Biosignal Processing and Classification Using Computational Learning and Intelligence“, 2022. [Online] Dostupno: <https://www.sciencedirect.com/topics/engineering/confusion-matrix>

# Popis slika

Slika 1. Komponente BCI sustava i njegove komunikacijske metode.....	5
Slika 2. EEG uređaj i njegova očitavanja.....	8
Slika 3. Vrste strojnog učenja (Vlastita izrada).....	12
Slika 4. Čitanje EDF datoteke.....	31
Slika 5. Prikaz nefiltriranih podataka kanala Fc5.....	31
Slika 6. Prikaz kanala elektroda.....	33
Slika 7. Prikaz filtriranih podataka kanala Fc5. (0.5 - 40 Hz).....	34
Slika 8. Razdvajanje signala u niske i visoke frekvencije.....	37
Slika 9. Klasifikacijska matrica za SVM.....	39
Slika 10. Prikaz provedbe algoritma neuronskih mreža i njegove točnosti.....	42
Slika 11. Klasifikacijski izvještaj (DT).....	43
Slika 12. Scatter plot za SVM - 50.0 % točnost.....	48
Slika 13. Konfuzijska matrica - 33.3 % točnost.....	48
Slika 14. Krivulja učenja – 67 % točnost.....	50
Slika 15. Krivulja učenja - 33% točnost.....	50
Slika 16. Decision tree - 83 % točnost.....	51
Slika 17. 1D Scatter plot - 83.33 % točnost.....	52
Slika 18. 1D Scatter plot - 33.33 % točnost.....	52

## Popis tablica

Tablica 1. Klasifikacija bez uzimanja u obzir IPMS hipotezu .....	46
Tablica 2. Klasifikacija uzimajući u obzir IPMS hipotezu .....	47