



PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student	Roman Razvan Marian	GRUPA:	30131	Nota	

EEG Signal Processing and Machine Learning

Autor: Roman Razvan

Grupa: **30131**

AN UNIVERSITAR: 2022-2023

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

Cuprins

1. Scopul Proiectului	3
a. Obiective	3
b. Specificații	3
2. Studiu bibliografic	5
3. Analiză, proiectare, implementare	12
4. Concluzii	19
a. Rezultate obținute	19
b. Direcții de dezvoltare	25

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

1. Scopul Proiectului

Scopul scriptului este de a realiza decodarea imaginilor motorii utilizând datele EEG prin aplicarea metodei CSP (Common Spatial Patterns) pentru a decompune semnalele. Apoi, scriptul aplică un algoritm de clasificare folosind analiza discriminantă liniară (Linear Discriminant Analysis) pentru a clasifica caracteristicile extrase din semnalele filtrate cu ajutorul metodei CSP. De asemenea, am încercat să antrenez un CNN (Convolutional Neural Network) pentru a realiza clasificarea pe 4 clase diferite. Scopul proiectului este de a explora diferite tehnici de clasificare, utilizand algoritmi de machine learning, pentru a obține rezultate precise și robuste în clasificarea semnalelor EEG asociate cu imaginile motorii.

a. Obiective

1. Studiul și implementarea algoritmului de decompunere a semnalelor EEG utilizând metoda CSP (Common Spatial Patterns) pentru a extrage caracteristicile relevante asociate cu imaginile motorii.
2. Aplicarea analizei discriminante liniare (LDA) ca algoritm de clasificare pentru a realiza clasificarea semnalelor EEG în funcție de imaginile motorii analizate.
3. Evaluarea performanței algoritmilor de clasificare prin generarea unui graf care reprezintă evoluția acurateței în timp pentru subiecții selectați.
4. Demonstrarea că algoritmul implementat poate fi utilizat în aplicații de asistență medicală pentru îmbunătățirea calității vieții pacienților cu tulburări senzomotorii prin decodarea semnalelor EEG asociate cu imaginile motorii.

b. Specificații

1. Decodarea datelor EEG: Sistemul meu este proiectat să fie capabil de a procesa datele EEG brute pentru a extrage caracteristicile relevante, cu accent pe ritmurile senzomotorii asociate cu diferite sarcini motorii.
2. Clasificarea Imaginii Motorii: Am implementat un sistem care clasifică datele EEG pe baza a doua sarcini motorii diferite, in 4 cazuri: imaginarea mișcării mâinilor versus picioarelor, mișcarea reală a mâinilor versus picioarelor, imaginarea mișcării mâinii drepte versus stânga și mișcarea reală a mâinii drepte versus stânga.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

3. Implementarea algoritmului CSP (Common Spatial Pattern): Pentru extragerea caracteristicilor din semnalele EEG, am utilizat algoritmul CSP. Acesta este un algoritm popular în analiza EEG pentru identificarea modelelor spațiale care maximizează varianța între două condiții.
4. Implementarea LDA (Linear Discriminant Analysis): În procesul de clasificare, am folosit LDA. Acesta este un algoritm supervised machine learning care folosește etichetele de clasă pentru a modela diferența dintre clase.
5. Generarea unui Grafic de Performanță: Sistemul meu generează un grafic care indică acuratețea pe axa Oy și timpul pe axa Ox, doar pentru subiecții selectați în funcție de stabilitatea clasificării.
6. Calibrare și Performanță: Scopul meu este de a atinge o acuratețe medie de clasificare de peste 80%. Am obținut și acuratețe de 92,5% pentru subiecții selectați la clasificarea miscării imaginare a mainilor versus picioare. De asemenea, sistemul este capabil să detecteze și să ignore subiecții care au o acuratețe în timp instabilă și sub 70%, având în vedere că semnalele EEG pot varia semnificativ de la o persoană la alta.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

2. Studiu bibliografic

- Ioannis Xygonakis , 1,2 Alkinoos Athanasiou , 1 Niki Pandria , 1 Dimitris Kugiumtzis , 2 and Panagiotis D. Bamidis ,
Decoding Motor Imagery through Common Spatial Pattern Filters at the EEG Source Space
<https://doi.org/10.1155/2018/7957408>

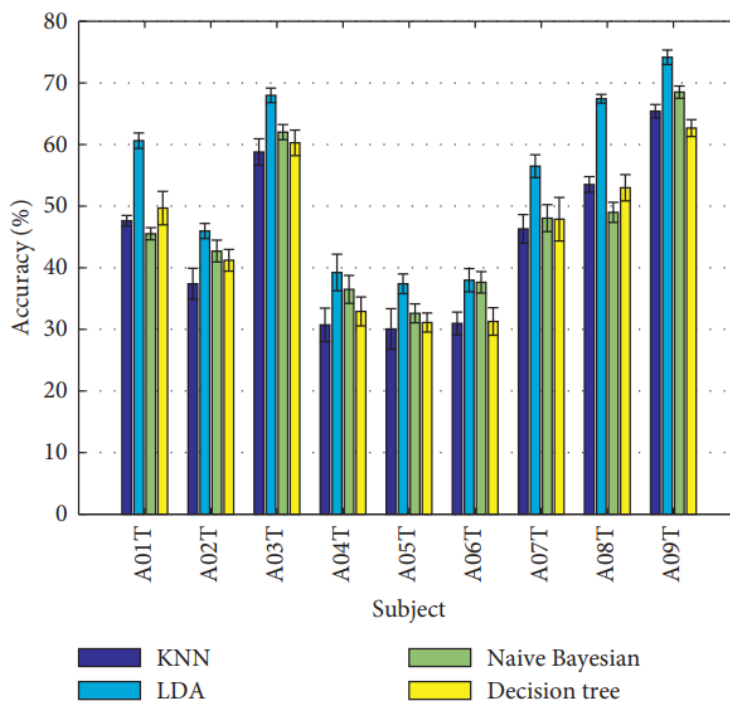
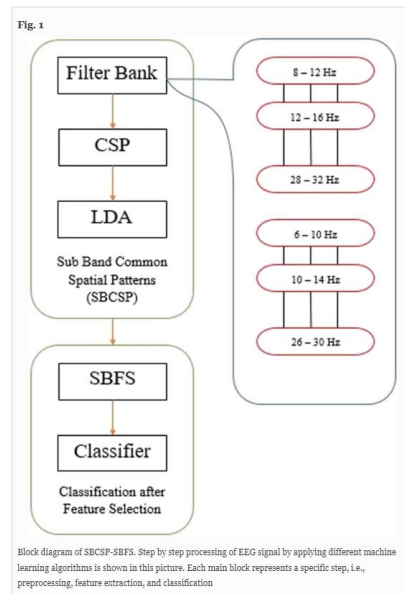


FIGURE 4: Classification accuracy of kNN, LDA, Naïve Bayesian, and Decision Tree classifiers across all subject data.

Acest studiu se concentrează pe utilizarea interfețelor creier-computer (BCIs), folosind metodele CSP și LDA pentru a îmbunătăți clasificarea semnalelor EEG în scopuri de reabilitare. Datele au fost preluate din setul de date BCI Competition IV 2a, care cuprinde înregistrări EEG de la pacienți ce își imaginău mișcări ale mâinilor sau picioarelor. Tehnica CSP a fost folosită pentru a extrage caracteristici spațiale relevante din datele EEG, iar LDA s-a dovedit a fi cel mai eficient clasificator în comparație cu kNN, Naive Bayesian și Decision Tree, cu o acuratețe medie de 54.1%. Rezultatele indică faptul că metoda dezvoltată a îmbunătățit acuratețea clasificării cu 5.6% față de metoda bazată pe senzori, sugerând potențialul CSP și LDA în îmbunătățirea acurateței BCIs non-invazive.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

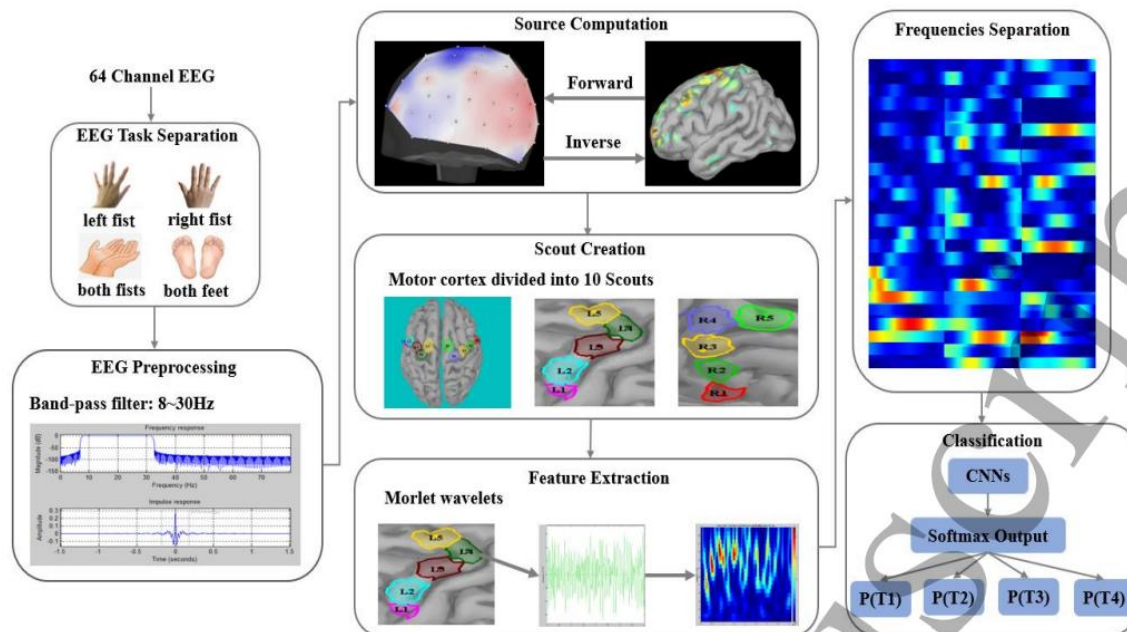
- Javeria Khan, Muhammad Hamza Bhatti, Usman Ghani Khan & Razi Iqbal, Multiclass EEG motor-imagery classification with sub-band common spatial patterns



Studiul se concentrează pe îmbunătățirea acurateței clasificării multiclase pentru semnalele EEG bazate pe imagistica motorie în interfețele creier-computer (BCIs). Metoda propusă, numită "sub-band common spatial patterns with sequential backward floating selection" (SBCSP-SBFS), folosește o bancă de filtre pentru a îndepărta zgomotul din semnalele EEG brute, după care sunt aplicate CSP și LDA pentru extracția caracteristicilor. Pentru a optimiza setul de caracteristici, este folosită selecția secvențială inversă flotantă (SBFS). Metoda este evaluată pe două seturi de date cu trei clase (imagistica motorie a mâinii drepte, imagistica motorie a mâinii stângi și starea de repaus) și a obținut o acuratețe maximă de 60,61% pentru Emotiv Epoc și 86,50% pentru electrozii cu gel umed, reprezentând o creștere de 7% a acurateței comparativ cu metodele anterioare de clasificare multiclase EEG. Acuratețea sistemului este comparată folosind diferite clasificatoare (SVM, KNN și NBPW) cu un număr variabil de caracteristici selectate. Studiul concluzionează că abordarea propusă are potențialul de a îmbunătăți aplicațiile BCI și poate fi aplicată în controlul BCI din lumea reală.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

- Yimin Hou¹, Lu Zhou¹, Shuyue Jia², Xiangmin Lun³, A Novel Approach of Decoding EEG Four-Class Motor Imagery Tasks via Scout ESI and CNN, <https://doi.org/10.1088/1741-2552/ab4af6>



Acest articol prezintă o abordare inovatoare care combină Imagistica Surselor EEG (ESI) cu Rețele Neuronale Convoluționale (CNN) pentru clasificarea sarcinilor de imaginație motorie (MI). Metoda ESI utilizează metoda elementelor de graniță (BEM) și estimarea normei minime ponderate (WMNE) pentru a rezolva problemele EEG directe și inverse. Sunt create "scout"-uri în cadrul cortexului motor ca regiuni de interes (ROI) și caracteristicile sunt extrase din seriile temporale ale scout-urilor folosind o abordare a undei Morlet. Apoi, CNN este folosit pentru a clasifica sarcinile MI.

Rezultatele arată că abordarea propusă atinge o acuratețe medie generală de 94.5% pe baza de date Physionet, cu acurateți individuale de 95.3%, 93.3%, 93.6% și 96% pentru sarcinile de închidere a pumnului stâng, închidere a pumnului drept, închiderea ambelor pumni și mișcarea ambilor picioare, respectiv, validate folosind validarea încrucișată 10-fold.

Arhitectura CNN propusă, bazată pe arhitectura Deep ConvNet, constă din șase straturi convoluționale, două straturi de max-pooling și trei straturi complet conectate. Pentru a îmbunătăți performanța rețelei și a preveni supraînvățarea, au fost implementate tehnici de dropout, normalizare în loturi și concatenare.

PROIECT INGINERIA REGLARII AUTOMATE II				
NUME student		GRUPA:		Nota

Combinatia dintre ESI de tip scout si CNN imbunatateste performanta decodarii sarcinilor MI cu patru clase in interfețele creier-calculator (BCIs). Metoda propusa depaseste abordările anterioare, atingând o acuratețe mare în clasificarea sarcinilor MI. Utilizarea ESI de tip scout imbogătește caracteristicile extrase din cortexul motor și oferă informații spațiale, temporale și de frecvență. CNN îmbunătățește generalizarea modelelor, ducând la o acuratețe de clasificare îmbunătățită.

INCERCARE PROPRIE A SOLUTIEI ARTICOLULUI:

Pe baza acestui articol am incercat sa rulez si antrenez modelul CNN propus gasit pe link-ul <https://github.com/SuperBruceJia/EEG-Motor-Imagery-Classification-CNNs-TensorFlow>, insa folosind dataset-ul de 109 subiecti a rezultat in OOM (out of memory error) . Folosind laptopul personal , doar pe 10 subiecti am reusit sa rulez si antrenez modelul, insa testing accuracy rezultat era sub 60% , iar training accuracy 100%, ceea ce denota ca modelul facea overfitting pe setul de date. Am incercat sa fac hyperparameter tuning, modificand variabilele : learning rate, dropout rate, batch size, and regularization coefficients (L2 , also named Loss Function) , insa acuratetea la testare nu s-a modificat , in schimb modelul nu mai facea overfitting. My dataset (.csv):

Training_data_1: 67,243*640 cells with values, Training_labels: 134488*4 cells with binary values

Test_data: 14943*640 cells with values, Test_labels: 14943*4 cells with binary values

Before hypertuning:

Iter 22, Testing Accuracy: 0.49285713, Training Accuracy: 1.0

Iter 22, Testing Loss: 0.41219229, Training Loss: 0.24417464

Learning rate is 1e-04

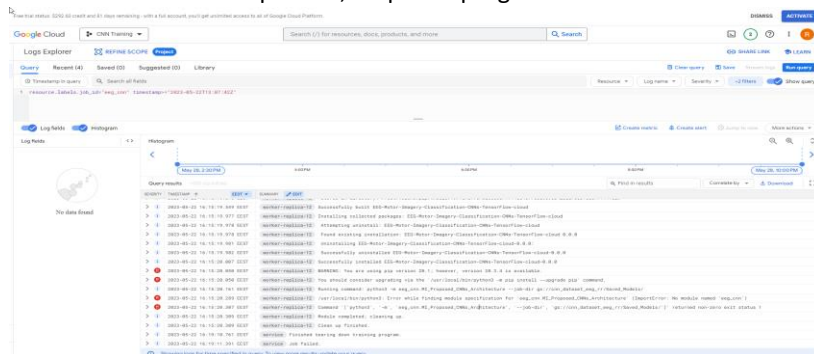
After hypertuning:

Iter 50, Testing Accuracy: 0.5011905, Training Accuracy: 0.6

Iter 50, Testing Loss: 0.19612616, Training Loss: 0.19333936

Learning rate is 1e-04

Am ajuns la concluzia ca setul de date EEG era prea mic pentru complexitatea modelului, si am incercat o implementare a modelului in Google Cloud, insa pentru AI Platform Training & Prediction API , limbajul Python versiunea 3.6 nu era suportata, iar pe 3.5 programul nu rula .



PROIECT INGINERIA REGLARII AUTOMATE II				
NUME student		GRUPA:		Nota

```

D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\ai-platform jobs submit training eeg_cnn --job-dir gs://cnn_dataset_eeg_rr/Saved_Models/ --package-path ./EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud --module-name eeg_cnn.MI_Proposed_CNN_Architecture --region europe-west1 --runtime-version 1.13 --python-version 3.6 --scale-tier PREMIUM_1 --staging-bucket gs://cnn_dataset_eeg_rr
ERROR: (gcloud.ai-platform.jobs.submit.training) [D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud] is not a valid Python package because it does not contain a __init__.py file. Please create one and try again. Also, please ensure that --package-path refers to a local directory.

D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\ai-platform jobs submit training eeg_cnn --job-dir gs://cnn_dataset_eeg_rr/Saved_Models/ --package-path EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud --module-name eeg_cnn.MI_Proposed_CNN_Architecture --region europe-west1 --runtime-version 1.13 --python-version 3.6 --scale-tier PREMIUM_1 --staging-bucket gs://cnn_dataset_eeg_rr
ERROR: (gcloud.ai-platform.jobs.submit.training) [D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud] is not a valid Python package because it does not contain a __init__.py file. Please create one and try again. Also, please ensure that --package-path refers to a local directory.

D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\ai-platform jobs submit training eeg_cnn --job-dir gs://cnn_dataset_eeg_rr/Saved_Models/ --package-path EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud --module-name eeg_cnn.MI_Proposed_CNN_Architecture --region europe-west1 --runtime-version 1.13 --python-version 3.6 --scale-tier PREMIUM_1 --staging-bucket gs://cnn_dataset_eeg_rr
ERROR: (gcloud.ai-platform.jobs.submit.training) [D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud] is not a valid Python package because it does not contain a __init__.py file. Please create one and try again. Also, please ensure that --package-path refers to a local directory.

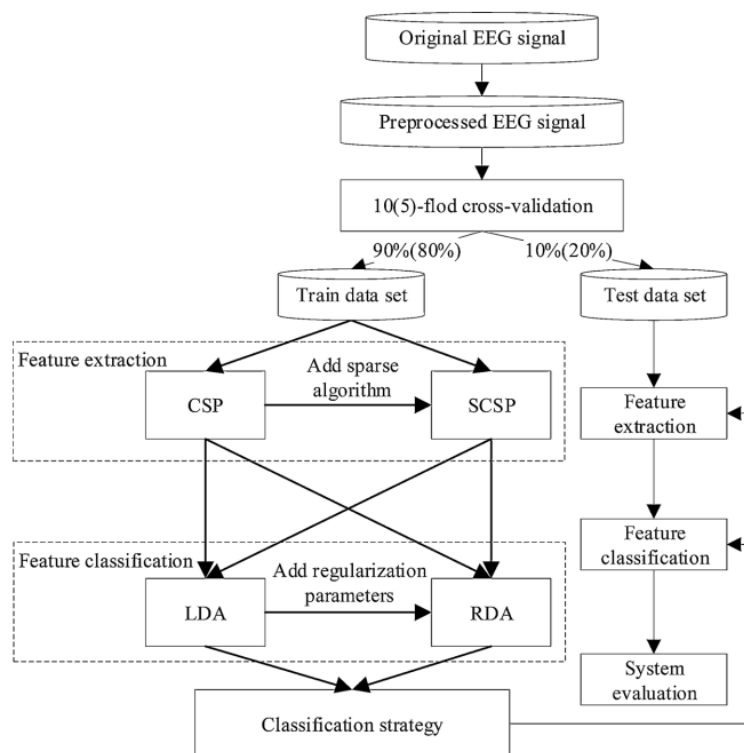
D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\ai-platform jobs submit training eeg_cnn --job-dir gs://cnn_dataset_eeg_rr/Saved_Models/ --package-path D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud --module-name eeg_cnn.MI_Proposed_CNN_Architecture --region europe-west1 --runtime-version 1.13 --python-version 3.6 --scale-tier PREMIUM_1 --staging-bucket gs://cnn_dataset_eeg_rr
ERROR: (gcloud.ai-platform.jobs.submit.training) Source directory [D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud] is not a valid directory.

D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\ai-platform jobs submit training eeg_cnn --job-dir gs://cnn_dataset_eeg_rr/Saved_Models/ --package-path EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud --module-name eeg_cnn.MI_Proposed_CNN_Architecture --region europe-west1 --runtime-version 1.13 --python-version 3.6 --scale-tier PREMIUM_1 --staging-bucket gs://cnn_dataset_eeg_rr
ERROR: (gcloud.ai-platform.jobs.submit.training) [D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud] is not a valid Python package because it does not contain a __init__.py file. Please create one and try again. Also, please ensure that --package-path refers to a local directory.

D:\Faculta\IRAI\EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud\ai-platform jobs submit training eeg_cnn --job-dir gs://cnn_dataset_eeg_rr/Saved_Models/ --package-path EEG-Motor-Imagery-Classification-CNNs-TensorFlow-cloud --module-name eeg_cnn.MI_Proposed_CNN_Architecture --region europe-west1 --runtime-version 1.13 --python-version 3.6 --scale-tier PREMIUM_1 --staging-bucket gs://cnn_dataset_eeg_rr
ERROR: (gcloud.ai-platform.jobs.submit.training) INVALID_ARGUMENT: field: python_version Error: The specified Python version '3.6' is not supported.
fieldViolations:
- description: The specified Python version '3.6' is not supported.
  field: python_version

```

- Rongrong Fu, Mengmeng Han, Yongsheng Tian*, Peiming Shi, Improvement motor imagery EEG classification based on sparse common spatial pattern and regularized discriminant analysis <https://doi.org/10.1016/j.jneumeth.2020.108833>



Articolul se concentrează pe îmbunătățirea procesului de extragere a caracteristicilor pentru creșterea acurateței clasificării semnalelor EEG bazate pe imagini motorii în sistemele de interfață creier-calculator (BCI).

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

Autori propun un algoritm CSP care folosește tehnici sparse și căutare iterativă pentru a extrage caracteristici discriminative. Mai exact, aceștia încorporează metoda sparse în CSP tradițional și utilizează o căutare greedy iterativă pentru a identifica cea mai bună combinație de canale pentru o extragere îmbunătățită a caracteristicilor și o acuratețe mai mare a clasificării. În plus, îmbunătățesc analiza discriminantă liniară (LDA) introducând doi parametri de regularizare.

Rezultatele studiului indică faptul că algoritmul Sparse CSP selectează eficient canalele EEG cu caracteristicile cele mai relevante. Analiza discriminantă regularizată îmbunătățită ajută la depășirea problemei singularității și îmbunătățește acuratețea clasificării caracteristicilor. Algoritmul propus este evaluat utilizând setul de date de la competiția BCI IV și propriul set de date al autorilor. Rezultatele experimentale demonstrează că algoritmul îmbunătățit obține o acuratețe mai mare cu 10,75% comparativ cu algoritmul tradițional și se comportă favorabil în comparație cu metodele existente pentru aceleași date.

În concluzie, studiul oferă dovezi suficiente care susțin afirmația că algoritmul propus îmbunătățește semnificativ performanța clasificării recunoașterii intenției motorii în sistemele BCI bazate pe EEG.

Alexander Craik*, Yongtian He, Jose L. Contreras-Vidal , A Deep learning for electroencephalogram (EEG) classification: <https://doi.org/10.1088/1741-2552/ab0ab5>

Acest articol este o revizuire sistematică a literaturii privind aplicarea învățării profunde în analiza electroencefalografică (EEG). Scopul recenziei este să răspundă la trei întrebări esențiale legate de sarcinile de clasificare EEG: 1) Care sarcini de clasificare EEG au fost explorate cu învățarea profundă? 2) Ce formule de intrare au fost folosite pentru instruirea rețelelor adânci? 3) Există anumite structuri de rețea de învățare profundă potrivite pentru anumite tipuri de sarcini?

Recenzia a identificat cinci grupe generale de sarcini care au folosit învățarea profundă în clasificarea EEG: recunoașterea emoțiilor, imagistica motorie, încărcătura mentală, detectarea crizelor, detecția potențialului legat de evenimente și evaluarea somnului.

Recenzia a analizat de asemenea tendințele în alegerile de arhitectură de învățare profundă pentru sarcinile de clasificare EEG. Arhitectura cea mai frecvent utilizată a fost cea a rețelelor neuronale convoluționale (CNN), prezentă în 43% dintre studii, urmată de Deep Belief Network (DBN) cu 18%.

Formularea intrărilor a fost, de asemenea, examinată, cu caracteristicile calculate, cum ar fi densitatea spectrală de putere și descompunerea în wavelet, fiind cea mai frecvent utilizată formulare de intrare, reprezentând 41% dintre studii. Valorile semnalelor în domeniul timpului sunt, de asemenea, utilizate direct ca intrări, în special cu ideea de învățare de la început la capăt .

În cele din urmă, recenzia prezintă studii de caz pe un set de date EEG partajat pentru recunoașterea emoțiilor, evidențiind că o simplă comparație a acurateței între studii care analizează seturi de date diferite nu este validă. În schimb, studiile care au examinat seturi de date identice folosind abordări diferite oferă o comparație mai semnificativă. Recenzia oferă și recomandări pentru proiectarea de arhitecturi de învățare profundă pentru aceste sarcini.

Concluzia articolului evidențiază aplicarea cu succes a clasificării de învățare profundă pentru diverse sarcini EEG, inclusiv imagistica motorie, detectarea crizelor, încărcătura mentală, evaluarea etapelor somnului, potențialul legat de evenimente și recunoașterea emoțiilor. De asemenea, menționează nevoia de cercetări suplimentare pentru optimizarea numărului și aranjamentului diferitelor straturi în arhitecturile hibride și pentru compararea modului în care rețelele profunde interpretează datele EEG brute față de cele denumite.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

- Meysam Golmohammadi, Amir Hossein Harati Nejad Torbati, Silvia Lopez de Diego, Iyad Obeid and Joseph Picone*, Automatic Analysis of EEGs Using Big Data and Hybrid Deep Learning Architectures: doi: 10.3389/fnhum.2019.00076

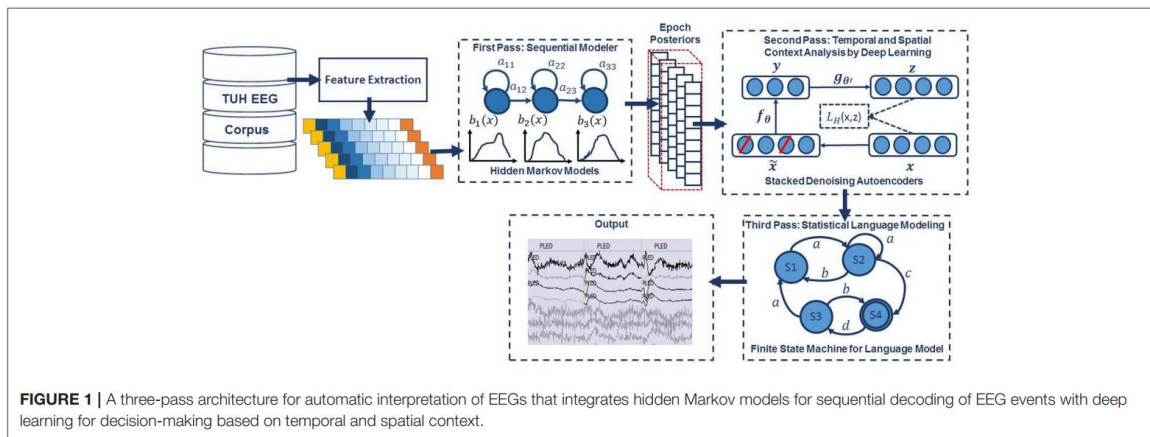


Figura arată un sistem propus pentru interpretarea automată a EEG-urilor, care constă în trei etape: extragerea caracteristicilor, modelarea secvențială și analiza contextului temporal și spațial, urmată de modelarea lingvistică statistică. Scopul este de a clasifica datele și de a produce un rezultat util pentru diagnostic și monitorizare în timp real.

Prima etapă implică extragerea caracteristicilor din semnalul EEG multicanal, iar aceste caracteristici sunt apoi analizate cu ajutorul Modelurilor Markov Ascunse (HMMs) pentru a produce ipoteze despre evenimente. În a doua etapă, analiza contextului temporal și spațial este realizată prin învățare profundă, folosind Autoencodere Stivuite de Eliminarea Zgomotului. În a treia etapă, modelarea lingvistică statistică se face folosind o Mașină cu Stare Fină (FSM) pentru a genera rezultatul final.

Sistemul este antrenat și evaluat folosind corpusul de EEG-uri al Spitalului Universitar Temple, fiind capabil să obțină o acuratețe ridicată și putând fi utilizat ca un instrument de suport pentru deciziile clinice în monitorizarea și diagnosticul în timp real al tulburărilor cerebrale.

Pre-procesarea constă în extragerea caracteristicilor din semnalele EEG. Metoda folosită se bazează pe coeficienții cepstrali de frecvență liniară (LFCCs) și alte caracteristici suplimentare.

Primul pas folosește modelele HMM pentru analiza semnalelor EEG. În al doilea pas, se folosesc Autoencodere Stivuite de Eliminarea Zgomotului pentru a învăța tiparele temporale și spațiale ale semnalelor EEG. Al treilea pas îmbunătățește performanța de detectare folosind un model de limbaj probabilistic.

În concluzie, sistemul hibrid HMM/deep learning oferă o acuratețe peste 90%, făcând din analiza automată o opțiune viabilă pentru medici. Acest cadru poate fi aplicat în alte sarcini de clasificare, cum ar fi detecția convulsiilor sau a anormalităților.

Alte link-uri utile:

https://mne.tools/stable/auto_examples/decoding/decoding_csp_eeg.html - scriptul original CSP și LDA

<https://physionet.org/content/eegmmidb/1.0.0/> - EEG dataset

<https://github.com/SuperBruceJia/EEG-DL#Notice> - Deep Learning models library

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

<https://github.com/SuperBruceJia/EEG-Motor-Imagery-Classification-CNNs-TensorFlow> - CNN model code

<https://cloud.google.com/ai-platform/training/docs> , <https://cloud.google.com/products/ai>

<https://docs.conda.io/projects/conda/en/latest/commands/info.html> - Conda enviornment

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html> - python

https://en.wikipedia.org/wiki/Common_spatial_pattern - CSP

https://en.wikipedia.org/wiki/Linear_discriminant_analysis - LDA

Zoltan J. Koles. The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. *Electroencephalography and Clinical Neurophysiology*, 79(6):440–447, 1991. doi:10.1016/0013-4694(91)90163-X.

Gerwin Schalk, Dennis J. McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R. Wolpaw. BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043, 2004. doi:10.1109/TBME.2004.827072.

Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 2000. doi:10.1161/01.CIR.101.23.e215.

3. Analiză, proiectare, implementare

- **Metode aplicate:**

Am utilizat o serie de tehnici și algoritmi pentru a rezolva problema clasificării imaginilor motorii din semnalele EEG. În primul rând, am aplicat un filtru trece-bandă pentru a selecta frecvențele relevante pentru analiza imaginilor motorii. Acest filtru trece-bandă selectează frecvențele cuprinse între 7.0 Hz și 30.0 Hz, eliminând componentele nedorite. Apoi, în continuare, am folosit funcțiile *'events_from_annotations'* și *'pick_types'* pentru a extrage evenimentele și canalele relevante pentru analiză.

În al doilea rând, am aplicat algoritmul CSP (Common Spatial Patterns) pentru decompunerea semnalelor EEG și extragerea caracteristicilor relevante. Acest algoritm mi-a permis să identific pattern-urile comune asociate cu imaginile motorii și să le utilizez în procesul de clasificare.

Apoi, am aplicat analiza discriminantă liniară (LDA) ca algoritm de clasificare. Acest algoritm m-a ajutat să realizez clasificarea efectivă a semnalelor EEG pe baza caracteristicilor extrase cu ajutorul algoritmului CSP.

- Ce este CSP:

Common Spatial Patterns (CSP) este o tehnică utilizată în prelucrarea semnalelor EEG pentru a extrage caracteristici relevante și discriminative asociate cu imaginile motorii. Scopul principal al algoritmului CSP este să

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

identifice combinații lineare de electrozi care maximizează diferența de varianță între două clase de semnale EEG. Aceste combinații lineare sunt numite filtre spațiale.

- Cum funcționează CSP:

1. Se obțin semnalele EEG brute de la electrozi selectați.
2. Se formează matricea covarianței pentru fiecare clasă de semnale EEG. Aceasta este o măsură a corelației dintre semnalele EEG de la diferiți electrozi.
3. Se calculează media matricelor de covarianță pentru fiecare clasă. Aceasta reprezintă media distribuției semnalelor EEG în cadrul fiecărei clase.
4. Se calculează matricea combinată de covarianță prin adăugarea matricilor de covarianță pentru cele două clase și se calculează inversa acesteia.
5. Se calculează valorile și vectorii proprii ai matricei inverse de covarianță combinată.
6. Se sortează valorile proprii în ordine descrescătoare și se selectează primele N valori proprii. Aceste valori proprii corespund celor mai semnificative caracteristici ale semnalelor EEG.
7. Se calculează filtrele spațiale prin înmulțirea matricei inverse de covarianță combinată cu vectorii proprii selectați.
8. Se aplică filtrele spațiale asupra semnalelor EEG pentru a obține semnalele filtrate CSP.

Astfel, prin utilizarea algoritmului CSP, obținem filtre spațiale care maximizează diferența de varianță între două clase de semnale EEG. Aceste filtre sunt ulterior utilizate pentru extragerea caracteristicilor relevante în procesul de clasificare.

- Exemplu pentru algoritmul CSP:

Presupunem că avem două clase de semnale EEG, "Clasa A" și "Clasa B". Fie X_A și X_B matricele de semnale EEG corespunzătoare acestor două clase, unde fiecare coloană reprezintă un vector de semnal. Avem $X_A = [x_{a1}, x_{a2}, \dots, x_{aN}]$ și $X_B = [x_{b1}, x_{b2}, \dots, x_{bN}]$, unde N reprezintă numărul de semnale din fiecare clasă.

1. Calculăm matricele de covarianță pentru fiecare clasă:

$$\text{CovA} = \text{cov}(X_A)$$

$$\text{CovB} = \text{cov}(X_B)$$

2. Calculăm media matricilor de covarianță pentru fiecare clasă:

$$\text{MeanA} = \text{mean}(\text{CovA})$$

$$\text{MeanB} = \text{mean}(\text{CovB})$$

3. Calculăm matricea combinată de covarianță prin adăugarea matricilor de covarianță pentru cele două clase și inversarea acesteia:

$$\text{CovCombined} = \text{CovA} + \text{CovB}$$

$$\text{CovInverse} = \text{inv}(\text{CovCombined})$$

4. Calculăm valorile și vectorii proprii ai matricei inverse de covarianță combinată:

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

EigVals, EigVecs = eig(CovInverse)

- Sortăm valorile proprii în ordine descrescătoare și selectăm primele N valori proprii.

EigValsSorted = sort(EigVals, 'descend')

EigVecsSelected = EigVecs[:, :N]

- Calculăm filtrele spațiale prin înmulțirea matricei inverse de covarianță combinată cu vectorii proprii selectați:

Filters = CovInverse * EigVecsSelected

- Aplicăm filtrele spațiale asupra semnalelor EEG pentru a obține semnalele filtrate CSP:

XA_CSP = Filters.T * XA

XB_CSP = Filters.T * XB

Astfel, obținem semnalele filtrate CSP, care conțin caracteristici relevante și discriminative pentru clasificare.

- Ce este LDA:

Linear Discriminant Analysis (LDA) este o tehnică de clasificare utilizată pentru a separa și clasifica datele în două sau mai multe clase. Scopul principal al LDA este de a găsi o proiecție liniară a datelor într-un nou spațiu astfel încât să se maximizeze separabilitatea între clase și să se minimizeze variabilitatea internă a claselor.

- Cum funcționează LDA:

- Se obțin caracteristicile extrase (de exemplu, semnalele filtrate CSP) și etichetele corespunzătoare ale claselor.
- Se calculează vectorii medie pentru fiecare clasă. Acești vectori reprezintă media caracteristicilor pentru fiecare clasă.
- Se calculează matricea de dispersie între clase, care măsoară diferența între medii ale claselor.
- Se calculează matricea de dispersie internă pentru fiecare clasă, care măsoară variabilitatea caracteristicilor în cadrul fiecărei clase.
- Se calculează matricea de dispersie totală ca sumă a matricilor de dispersie între clase și matricilor de dispersie interne.
- Se calculează vectorii proprii și valorile proprii ale matricei inverse a matricei de dispersie totală.
- Se sortează valorile proprii în ordine descrescătoare și se selectează primele K valori proprii. Acestea reprezintă cele mai semnificative caracteristici pentru clasificare.
- Se construiește un model liniar folosind vectorii proprii selectați ca bază pentru proiecția datelor într-un nou spațiu.
- Se clasifică noile instanțe de date proiectându-le în noul spațiu și aplicând un criteriu de decizie (de exemplu, dreapta de decizie).

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

Prin utilizarea algoritmului LDA, se realizează o clasificare liniară eficientă și robustă prin proiecția datelor într-un nou spațiu cu caracteristici semnificative. Acest algoritm se bazează pe maximizarea separabilității între clase și minimizarea variabilității interne a claselor, permițând astfel o bună separare și clasificare a datelor.

- Exemplu pentru algoritmul LDA:

Presupunem că avem două clase de caracteristici, "Clasa A" și "Clasa B". Fie X_A și X_B matricele de caracteristici corespunzătoare acestor două clase, unde fiecare coloană reprezintă un vector de caracteristici. Avem $X_A = [x_{a1}, x_{a2}, \dots, x_{aM}]$ și $X_B = [x_{b1}, x_{b2}, \dots, x_{bN}]$, unde M și N reprezintă numărul de caracteristici din fiecare clasă.

1. Calculăm vectorii medie pentru fiecare clasă:

$\text{MeanA} = \text{mean}(X_A)$

$\text{MeanB} = \text{mean}(X_B)$

2. Calculăm matricea de dispersie între clase:

$\text{ScatterBetween} = (\text{MeanA} - \text{MeanB}) * (\text{MeanA} - \text{MeanB}).T$

3. Calculăm matricea de dispersie internă pentru fiecare clasă:

$\text{ScatterA} = \text{cov}(X_A)$

$\text{ScatterB} = \text{cov}(X_B)$

$\text{ScatterWithin} = \text{ScatterA} + \text{ScatterB}$

4. Calculăm matricea de dispersie totală:

$\text{ScatterTotal} = \text{ScatterWithin} + \text{ScatterBetween}$

5. Calculăm vectorii proprii și valorile proprii ale matricei inverse a matricei de dispersie totală:

$\text{EigVals}, \text{EigVecs} = \text{eig}(\text{inv}(\text{ScatterTotal}))$

6. Sortăm valorile proprii în ordine descrescătoare și selectăm primele K valori proprii.

$\text{EigValsSorted} = \text{sort}(\text{EigVals}, 'descend')$

$\text{EigVecsSelected} = \text{EigVecs}[:, :K]$

7. Construim un model liniar folosind vectorii proprii selectați ca bază pentru proiecția datelor într-un nou spațiu:

$\text{NewDataA} = \text{EigVecsSelected}.T * X_A$

$\text{NewDataB} = \text{EigVecsSelected}.T * X_B$

8. Clasificăm noile instanțe de date proiectându-le în noul spațiu și aplicând un criteriu de decizie (de exemplu, dreapta de decizie).

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

Astfel, obținem o clasificare liniară folosind algoritmul LDA pe baza caracteristicilor relevante selectate într-un nou spațiu de dimensiune redusă.

- **Materiale și proceduri:**

Pentru realizarea studiului, am folosit un set de date EEG care conținea peste 1500 de înregistrări EEG obținute de la 109 voluntari. Aceste înregistrări au fost efectuate în timp ce subiecții realizau diferite sarcini motorii sau își imaginau aceste sarcini. Setul de date a fost pre-etichetat, ceea ce mi-a permis să asociez fiecare înregistrare cu una dintre cele patru cazuri de clasificare a câte 2 clase: Imaginat (picioare vs. mâini), Miscare reala (picioare vs. mâini), Imaginat (mâna dreaptă vs. mâna stângă) și Miscare reala (mâna dreaptă vs. mâna stângă).

Am prelucrat și filtrat semnalele EEG folosind metode specifice și am aplicat algoritmul CSP și LDA pentru extragerea caracteristicilor și clasificarea semnalelor.

Mai multe informații despre dataset se pot găsi aici:

<https://physionet.org/content/eegmmidb/1.0.0/>

- **Metodologia de proiectare:**

Am urmat o metodologie riguroasă pentru proiectarea și implementarea soluțiilor propuse. Am început prin definirea parametrilor necesari și a obiectivelor de clasificare. Am studiat literatura de specialitate și am selectat algoritmul CSP și LDA ca fiind cele mai potrivite pentru rezolvarea proiectului, ulterior după încercarea modelului CNN. Apoi, am dezvoltat un flux de lucru care a inclus etapele de prelucrare a datelor, decompunerea semnalelor EEG și extragerea caracteristicilor cu ajutorul algoritmului CSP, și clasificarea utilizând LDA, iterativ de la o singură persoană la 109 persoane, clasificarea pe cele 4 cazuri de clasificare a câte 2 clase și selectarea doar a subiecților compatibili cu sistemul meu de clasificare. Am optimizat și ajustat parametrii algoritmilor pentru a obține cele mai bune rezultate de clasificare posibile, impunând condiția ca 'if np.mean(mean_performance[1:]) > 0.70' care verifică dacă media performanței, calculată în timp, pentru un anumit subiect îndeplinește o anumită valoare de prag de 0.70 după prima secundă, astfel verificând stabilitatea algoritmului pe o anumită persoană.

- **Informații pentru replicabilitate:**

Pentru a asigura replicabilitatea lucrării, am folosit biblioteci și pachete de software populare precum MNE și scikit-learn pentru a implementa algoritmi și metodele necesare. De asemenea, am inclus și descrieri detaliate ale setului de date utilizat și a parametrilor utilizați în analiză și clasificare. Scriptul a fost scris în visual studio code și a fost rulat în command prompt folosind un conda environment.

Un conda environment (mediu Conda) este un mediu izolat în care puteți instala și gestiona versiuni specifice de Python și pachete asociate. Acesta vă permite să aveți mai multe medii separate pentru diferite proiecte sau configurații, evitând conflictele între dependențe. Este util pentru dezvoltarea software și gestionarea eficientă a pachetelor și a dependențelor într-un mod reproducibil.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

MNE (MNE-Python) este o bibliotecă Python pentru analiza semnalelor EEG/MEG (electroencefalografie/magnetoencefalografie). Aceasta oferă un set de instrumente puternice pentru prelucrarea, vizualizarea și analiza datelor EEG/MEG, inclusiv citirea, filtrarea, segmentarea, transformarea și vizualizarea semnalelor. MNE este frecvent utilizat în cercetarea neuroștiinței și în domeniul analizei și interpretării semnalelor cerebrale.

TensorFlow este o bibliotecă de calcul numeric cu sursă deschisă dezvoltată de Google. Ea este proiectată pentru a permite construirea și antrenarea de modele de învățare automată și rețele neurale profunde. TensorFlow oferă un mediu flexibil și extensibil pentru a construi și rula algoritmi de învățare automată pe diferite platforme, cum ar fi computerele personale, serverele, dispozitivele mobile și sistemele de operare pe plăci de dezvoltare. TensorFlow este utilizat într-o gamă largă de domenii, inclusiv învățarea automată, viziunea artificială, prelucrarea limbajului natural și multe altele.

Pentru a rula script-ul va trebui sa urmariti urmatoorii pasi:

Deschide-ti cmd si rulati: (fara \$)

- 1) \$ conda create -n eeg-decoding python=3.10
- 2) \$ conda activate eeg-decoding
- 3) \$ conda install mne
- 4) \$ conda install tensorflow

Dupa care folositi comanda `\$ cd` pentru a naviga catre fisierul care contine script.py si folositi comanda:
- python script.py

Pentru a modifica cazul de clasificare documentati-va despre dataset-ul folosit si decommentati o singura instructiune dintre:

```
runs = [6, 10, 14] # motor imagery: hands vs feet
# runs = [5, 9, 13] # actual movement: both fists or both feet
# runs = [4, 8, 12] # motor imagery: left vs right hand
# runs = [3, 7, 11] # actual movement: left vs right hand
```

Pentru a ajusta parametrul de selectare a subiectilor in functie de performanta, modificati instructiunea:

```
if np.mean(mean_performance[1:]) > 0.70:
```

Pentru a salva si vizualiza modelele CSP estimate cu scopul de a obține o înțelegere vizuală a caracteristicilor relevante extrase din semnalele EEG, asigurați-va ca aveți 2 foldere cu denumirile respective(CSP_Patterns_Selected_Imaginary), sau modificati denumirea in cod:

```
if np.mean(mean_performance[1:]) > 0.70:
    fig = csp.plot_patterns(epochs.info, ch_type='eeg', units='Patterns (AU)', size=1.5)
```

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

```

fig.savefig(f'CSP_Patterns_Selected_Imaginary/subject_{subject}_CSP_patterns.png')
plt.close(fig)
else:
    fig = csp.plot_patterns(epochs.info, ch_type='eeg', units='Patterns (AU)', size=1.5)
    fig.savefig(f'CSP_Patterns_Unselected_Imaginary/subject_{subject}_CSP_patterns.png')
    plt.close(fig)

```

Totodata, asigurati-va ca aveti un document text denumit "classification_info.txt" pentru a stoca acuratetea pentru fiecare subiect in parte, respectiv acuratetea medie pentru subiectii selectati si vectorul subiectilor alesi.

- **Testare si Validare:**

Metodologia de testare a aplicației a implicat rularea experimentelor pentru fiecare caz de clasificare în parte (imaginat: picioare vs. maini, mana stanga vs. mana dreapta, mișcare reală: picioare vs. maini, mișcare reală: mana stanga vs. mana dreapta). Fiecare experiment a fost realizat de trei ori, aplicând o condiție de performanță specifică pentru selectarea subiecților (0.65, 0.70, 0.75).

În fiecare experiment, au fost înregistrate următoarele informații:

- Subiecții selectați care au îndeplinit condiția de performanță specificată.
- Acuratețea medie obținută pentru subiecții selectați.
- Modelele CSP estimate pentru fiecare subiect selectat.

Analiza rezultatelor a constat în evaluarea performanței clasificării pentru fiecare experiment. S-au analizat acuratețea medie obținută pentru subiecții selectați în funcție de condiția de performanță și clasificarea realizată. De asemenea, s-au examinat modelele CSP estimate pentru a identifica caracteristicile spațiale relevante asociate cu imaginile motorii.

Rezultatele obținute în urma experimentelor au furnizat informații despre performanța aplicației de clasificare a imaginilor motorii folosind algoritmul CSP și analiza discriminantă liniară. S-a evaluat acuratețea clasificării și s-a identificat subiecții care au obținut o performanță înaltă în funcție de criteriul de selecție.

- Rezolvarea erorii : "ValueError: setting an array element with a sequence. The requested array has an inhomogeneous shape after 1 dimensions. The detected shape was (22,) + inhomogeneous part."

Pentru a putea trasa graficul pentru performanța medie în funcție de timp, este necesar ca toate elementele din mean_performance_over_time_selected să aibă aceeași lungime. Deoarece fiecare element din listă corespunde unei perioade de timp diferite, lungimea acestora poate varia.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

Pentru a rezolva această problemă, s-a calculat lungimea minimă dintre toate elementele din vectorul respectiv, apoi, fiecare element din listă a fost redus la lungimea minimă utilizând sintaxa de segmentare a listei.

4. Concluzii

a. Rezultate obținute

- **Imaginary hands vs feet:**

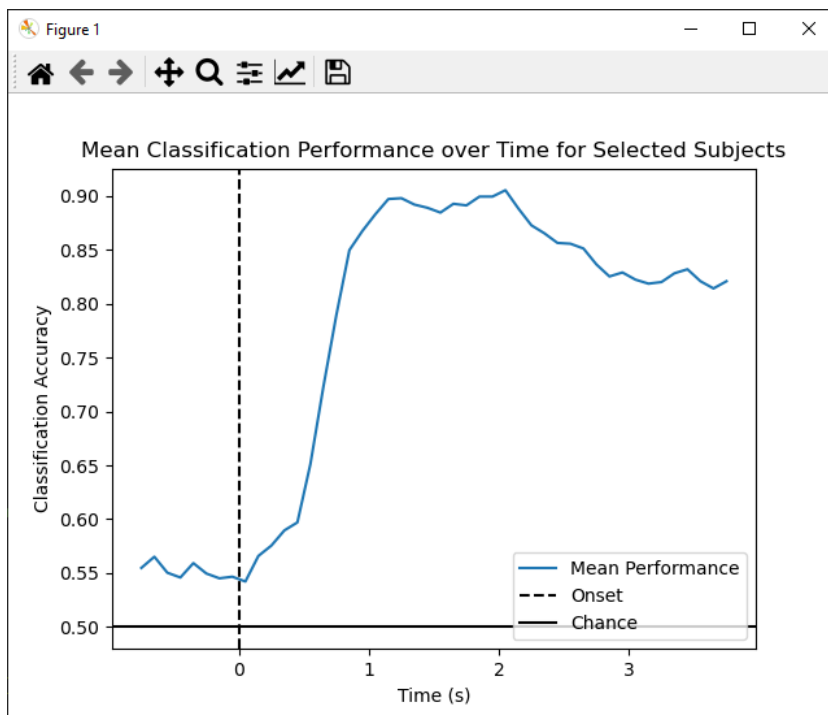
Performanța 0.75: Numărul de subiecți neselectați: 99, Numărul de subiecți selectați: 10, Acuratețe: 0.924

Performanța 0.70: Numărul de subiecți neselectați: 94, Numărul de subiecți selectați: 15, Acuratețe: 0.924

Performanța 0.65: Numărul de subiecți neselectați: 87, Numărul de subiecți selectați: 22, Acuratețe: 0.891

Performanța 0.60: Numărul de subiecți neselectați: 79, Numărul de subiecți selectați: 30, Acuratețe: 0.849

Average Classification Accuracy for All Subjects: 0.6299311926605505



(Performanta 0.7)

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

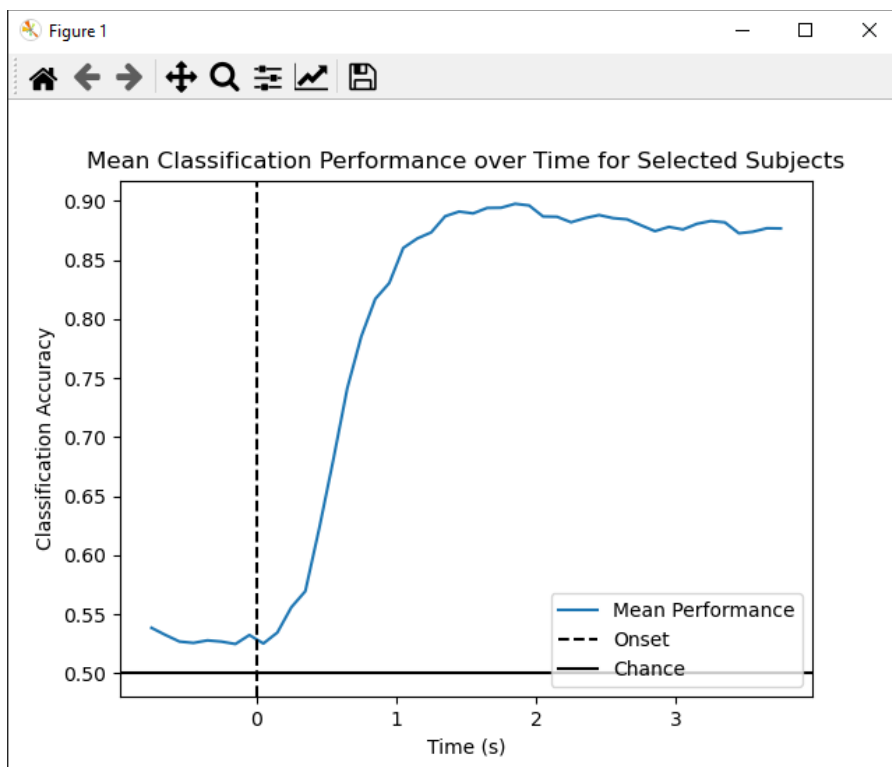
- **Actual Movement hands vs feet:**

Performanța 0.80: Numărul de subiecți neselectați: 97, Numărul de subiecți selectați: 12, Acuratețe: 0.976 (O performanță foarte bună)

Performanța 0.70: Numărul de subiecți neselectați: 77, Numărul de subiecți selectați: 32, Acuratețe: 0.924 (O performanță foarte bună)

Performanța 0.60: Numărul de subiecți neselectați: 49, Numărul de subiecți selectați: 60, Acuratețe: 0.862

Average Classification Accuracy for All Subjects: 0.728980996068152



(Performanta 0.7)

- **Imaginary right vs left hand:**

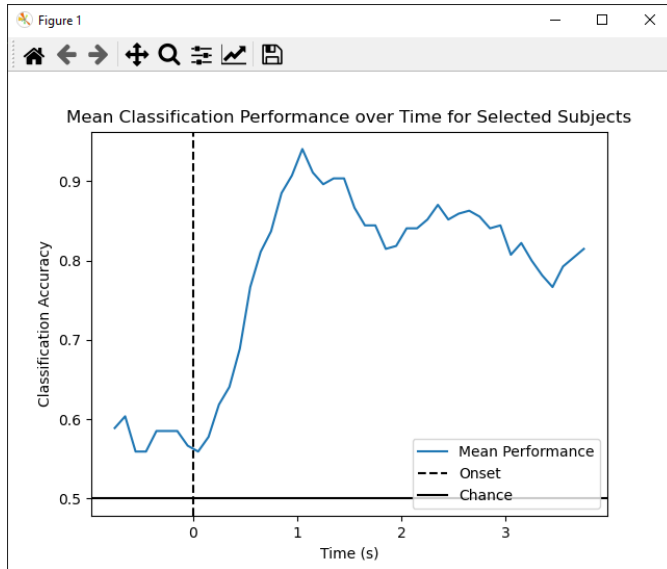
Performanța 0.70: Numărul de subiecți neselectați: 106, Numărul de subiecți selectați: 3, Acuratețe: 0.911

Performanța 0.65: Numărul de subiecți neselectați: 102, Numărul de subiecți selectați: 7, Acuratețe: 0.802

Performanța 0.60: Numărul de subiecți neselectați: 96, Numărul de subiecți selectați: 13, Acuratețe: 0.776

Average Classification Accuracy for All Subjects: 0.5320463812436288

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	



(Performanta 0.7)

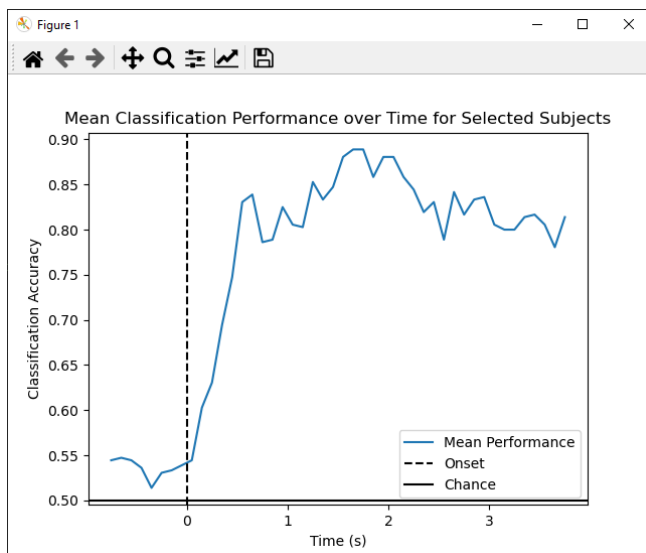
- Actual movement right vs left hand:**

Performanța 0.70: Numărul de subiecți neselectați: 105, Numărul de subiecți selectați: 4, Acuratețe: 0.911

Performanța 0.65: Numărul de subiecți neselectați: 100, Numărul de subiecți selectați: 9, Acuratețe: 0.828

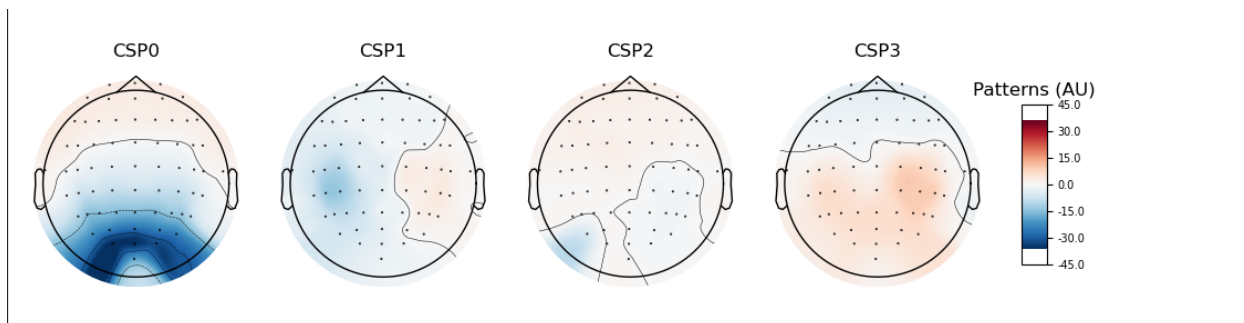
Performanța 0.60: Numărul de subiecți neselectați: 93, Numărul de subiecți selectați: 16, Acuratețe: 0.769

Average Classification Accuracy for All Subjects: 0.5501216291353908



(Performanta 0.7)

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	



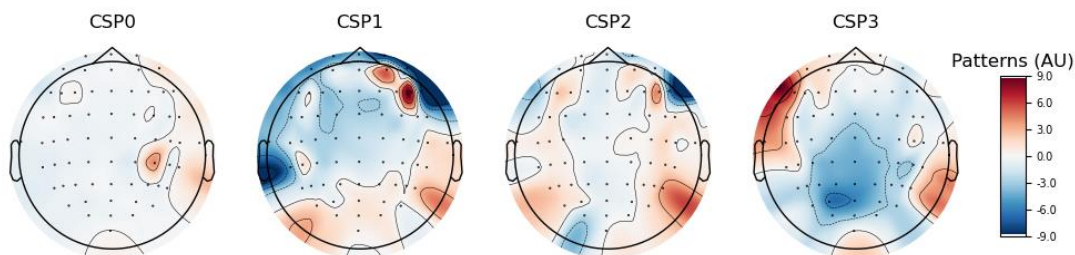
(Subject 7 – Imaginary Hands vs Feet) – Subiectul care a fost selectat in toate performantele
Fiecare caz de clasificare are un model CSP diferit per subiect.

Se pot observa 4 imagini reprezentând distributia electrozilor, denumite CSP0, CSP1, CSP2 și CSP3.
Acele imagini reprezintă distribuția componentelor spațiale estimate de algoritmul CSP și reprezintă regiuni diferite ale capului uman.

Regiunile colorate de la albastru la roșu indică amplitudinile relative ale componentelor spațiale.
Albastrul reprezintă valori negative, iar roșul reprezintă valori pozitive. Culoarea reflectă astfel polaritatea și magnitudinea componentelor respective.

Unitățile de măsură ale acestor modele CSP sunt Patterns (AU), care indică intensitatea sau amplitudinea relativă a caracteristicilor extrase. Aceste modele oferă o reprezentare vizuală a componentelor spațiale relevante identificate prin algoritmul CSP pe baza semnalelor EEG analizate.

În ceea ce privește distincția între imaginarea mișcării mâinilor și picioarelor, subiectul 7 prezintă pattern-uri CSP caracteristice și semnificative, evidențiate prin un număr maxim de 35 de patterns (AU). Acesta poate fi considerat un subiect cu o activitate cerebrală semnificativă și specifică pentru această distincție. Ploturile CSP asociate subiectului 7 arată o separare clară între cele două stări, indicând o procesare diferită a semnalului cerebral în funcție de mișcarea imaginată.



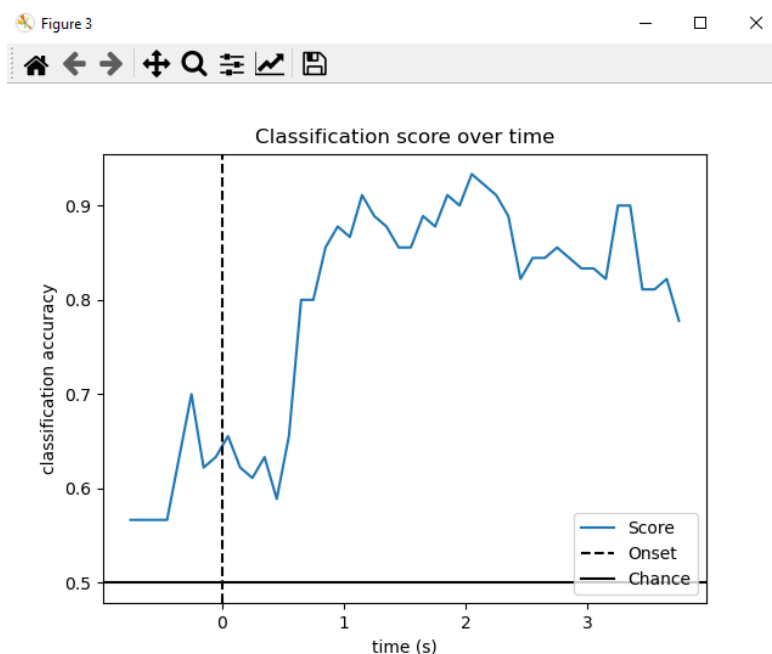
(Subject 5 – Imaginary Hands vs Feet) – Subiectul care nu a fost selectat in nicio performanta.

În schimb, subiectul 5 nu prezintă un număr semnificativ de patterns (AU), având un număr maxim de 9. Acest lucru sugerează că subiectul 12 nu manifestă un model distinctiv al activărilor în contextul

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

distincției între imaginarea mișcării mâinilor și picioarelor. Ploturile CSP asociate acestui subiect prezintă o suprapunere mai mare între cele două stări, făcând mai dificilă distingerea lor.

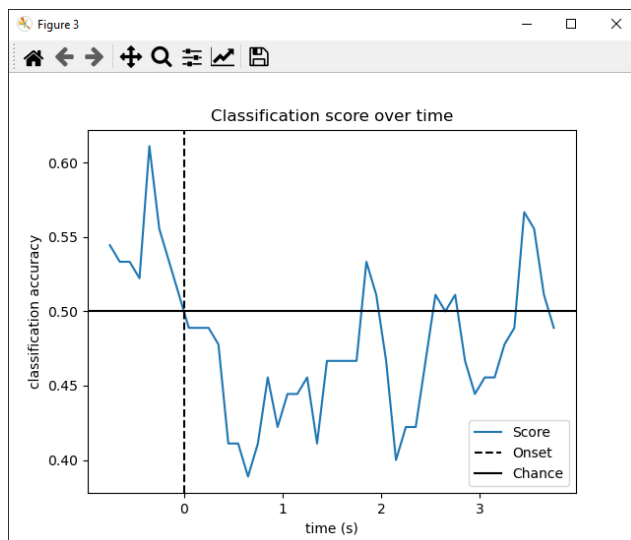
Astfel, diferența principală între cei doi subiecți rezidă în prezența și intensitatea pattern-urilor CSP relevante pentru distincția între imaginarea mișcării mâinilor și picioarelor. Subiectul 7 prezintă un profil distinctiv al activității cerebrale în această discriminare, în timp ce subiectul 5 nu prezintă aceleași caracteristici semnificative în ceea ce privește procesarea semnalului cerebral asociat imaginării mișcării mâinilor vs. picioarelor.



(Graf Acuratete-Timp Subiect 7 Imaginary Hands vs Feet)

Classification accuracy: 0.900000

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	



(Graf Acuratete-Timp Subiect 5 Imaginary Hands vs Feet)

Classification accuracy: 0.433333

- Concluziile:**

În ceea ce privește detectarea imaginii mâinilor versus picioarelor, s-a obținut o acuratețe relativ ridicată în toate cazurile, cu o ușoară scădere a acurateței pe măsură ce s-a mărit numărul de subiecți selectați. Este important să menționăm că performanța generală a fost bună, iar calitatea rezultatelor a fost în general satisfăcătoare.

În ceea ce privește detectarea mișcărilor reale ale mâinilor versus picioarelor, s-au obținut performanțe excelente în toate cazurile, cu o acuratețe foarte ridicată. Acest lucru indică faptul că algoritmul utilizat este eficient în distingerea mișcărilor reale ale mâinilor și picioarelor.

În cazul detectării imaginare a mâinii drepte față de mâna stângă, s-au obținut acuratețe ridicate, dar cu un număr redus de subiecți selectați. Acest lucru poate indica o dificultate în distingerea imaginilor imaginare ale mâinii drepte față de mâna stângă și necesită investigații suplimentare.

În cazul detectării mișcării reale a mâinii drepte față de mâna stângă, s-au obținut, de asemenea, acuratețe relativ ridicate, dar cu un număr redus de subiecți selectați. Totuși, performanța nu este mai mare decât în cazul imaginilor imaginare, sugerând o dificultate mai mare în distingerea mișcărilor ale mâinii drepte față de mâna stângă decât în distingerea mișcărilor picioare versus maini.

În general, calitatea rezultatelor este bună, cu performanțe excelente în cazul detecției mișcărilor reale

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

ale mâinilor versus picioarelor. Cu toate acestea, se pot observa anumite dificultăți în distingerea imaginilor imaginare ale mâinii drepte față de mâna stângă și mișcărilor reale ale mâinii drepte față de mâna stângă, care pot necesita investigații și îmbunătățiri ulterioare. Totodată, este important de menționat că selecția adecvată a subiecților a fost crucială pentru obținerea rezultatelor precise.

- Contribuțiile mele în domeniul temei abordate includ următoarele:
 1. Extinderea programului inițial pentru a permite clasificarea EEG pentru 109 subiecți în loc de un singur subiect.
 2. Implementarea automatizată a selecției subiecților cu performanță mai mare și generarea graficului acurateței în timp și a acurateței medii pentru subiecții selectați.
 3. Analizarea și adunarea datelor pentru cele patru cazuri de clasificare diferite: imaginat (picioare vs. mâini), mișcare reală (picioare vs. mâini), imaginat (mâna dreaptă vs. mâna stângă), mișcare reală (mâna dreaptă vs. mâna stângă).
 4. Modificarea codului pentru a salva modelele CSP în foldere diferite în funcție de subiecții selectați și neselectați.
 5. Gestionarea și rezolvarea erorilor asociate cu modificările aduse codului.
 6. Rularea și analizarea datelor în funcție de parametrul de performanță pentru selecția subiecților de minim trei ori pentru fiecare dintre cele patru cazuri.

Aceste contribuții au permis extinderea și îmbunătățirea funcționalității programului inițial, realizând o analiză aprofundată a datelor EEG și obținând rezultate relevante pentru clasificarea imaginilor motorii.

b. Direcții de dezvoltare

Direcțiile posibile de dezvoltare pentru această temă includ:

Adaptarea codului pentru a permite și selecția subiecților cu acuratețe mică odată cu subiecții cu acuratețe mare. Acest lucru ar putea ajuta la identificarea și analiza diferențelor între subiecții care au dificultăți în clasificarea imaginilor motorii din semnalele EEG.

Reîncercarea soluției folosind un model CNN (Convolutional Neural Network) pentru clasificarea imaginilor motorii din semnalele EEG. Această abordare poate aduce o perspectivă diferită asupra clasificării și poate îmbunătăți performanța sistemului în identificarea și discriminarea imaginilor motorii.

PROIECT INGINERIA REGLARII AUTOMATE II					
NUME student		GRUPA:		Nota	

Utilizarea metodei CSP și LDA în alte domenii de aplicații ale interfețelor creier-computer (BCIs), cum ar fi reabilitarea. Aceasta ar putea implica adaptarea algoritmului și a metodelor de analiză pentru a se potrivi nevoilor specifice ale altor domenii, precum controlul robotic al protezelor sau al scaunelor cu roțile sau asistența medicală.

Investigarea și dezvoltarea altor tehnici de prelucrare a semnalelor EEG pentru îmbunătățirea clasificării, cum ar fi utilizarea filtrelor sub-band spectral pentru a captura informații specifice în anumite frecvențe ale semnalelor EEG.