

Universitatea POLITEHNICA din București

Facultatea de Automatică și Calculatoare,
Departamentul de Calculatoare



LUCRARE DE DIPLOMĂ

Procesarea și vizualizarea
electroencefalogramelor cu ajutorul
inteligenței artificiale

Autor:

Dochian Alexandru Adrian

Conducători Științifici:

sl. dr. ing. Alexandru Șorici
sl. dr. ing. Mihai Trăscău
sl. dr. ing. Irina Andra Tache

București, Iulie 2022

Această lucrare nu ar fi fost posibilă fără ajutorul Domnului Doctor Bogdan Pavel de la
Facultatea de Medicină, UMF "Carol Davila", București care a achiziționat
electroencefalograme intracraniene de calitate superioară pe regretați soareci și le-a împărtășit
cu mine și coordonatorii mei.

De asemenea adresez mulțumiri coordonatorilor mei, Doamnei sl. dr. ing. Irina Andra Tache,
Domnului sl. dr. ing. Mihai Trăscău și Domnului sl. dr. ing. Alexandru Șorici pentru timpul,
energia și cunoștințele dedicate acestei lucrări.

Compendiu

În lucrare de față este prezentat un sistem informatic care procesează electroencefalograme.

Din perspectiva sistemului, electroencefalogramele sunt semnale care poartă informații despre evoluția potențialului electric din punctul de eșantionare.

Sistemul procesează electroencefalogramele alese ca intrare și produce ca rezultat un tablou bidimensional în care punctele rezultate sunt semnalele inițiale. Putem astfel afirma că sistemul reușește să proiecteze relațiile dintre electroencefalograme într-un spațiu bidimensional.

În calea către rezultat, electroencefalogramele trec prin stagii configurabile de preprocesare și procesare.

Stagiile de preprocesare modificarea semnalelor în domeniul timp.

Stagiile de preprocesare extrag caracteristici din domeniile timp și frecvență. Ultimul pas al execuției îl constituie execuția algoritmului t-SNE pe datele procesate. t-Distributed Stochastic Neighbor Embedding (t-SNE) este un algoritm de învățare nesupervizată care reduce dimensionalitatea datelor de intrare la un spațiu bidimensional.

Cuprins

Mulțumiri	i
Compendiu	ii
1 Introducere	1
1.1 Electroencefalografia	1
1.1.1 Fundamente fizice	1
1.1.2 Istorie	1
1.1.3 Aplicații	2
1.2 Prezentarea sistemului	2
1.2.1 Datele folosite	2
1.2.2 Analiza clasică a electroencefalogramelor	3
1.2.3 Analiza electroencefalogramelor cu ajutorul sistemului	3
2 Arhitectură	4
2.1 Arhitectura codului	4
2.1.1 Pachetul engine	4
2.1.2 Pachetul stages	5
2.1.3 Pachetul pipelines	5
2.1.4 Pachetul exceptions	5
2.1.5 Pachetul configuration	5
2.1.6 Pachetul internal	5
2.2 Arhitectura execuției	6
3 Preprocesarea datelor	7
4 Procesarea datelor	8
5 Rezultate și Concluzii	9
6 Lucrări similare	10
7 Viitor	11
A Configurații de execuție	12
A.1 Configurația standard de execuție	12

Capitolul 1

Introducere

1.1 Electroencefalografia

Electroencefalografia este o modalitate prin care se relevă informații despre activitatea electrică a stratului de la suprafața creierului. Informația rezultată în urma procesului de achiziționare este evoluția în timp a potențialului electric în punctul de interes.

Electroencefalografia înregistrează date de pe scalpul subiecților și de aceea este predispusă la zgomot. Electroencefalografia sau electroencefalografia invazivă a fost inventată și rafinată. Ea presupune achiziționarea datelor direct de pe cortexul subiectului.

1.1.1 Fundamente fizice

Electroencefalogramul este un dispozitiv care folosește electrozi plasați pe scalpul subiectului pentru a înregistra fluctuațiile de voltaj din punctul de interes. Existența acestui voltaj ce poate fi măsurat se datorează existenței unui curent ionic între neuroni.

Electrozii sunt conductori electrici folosiți pentru a face contact cu părți nemetalice ale unui circuit electric. Pot fi construiți din mai multe materiale dintre care menționăm: Litiu (Li), Mangan (Mn), Cupru (Cu) și Zinc (Zn)

Un ion este un atom care prezintă un dezechilibru de sarcină electrică. Anionul este un ion format din mai mulți electroni decât protoni, astfel având o sarcină electrică negativă. Cationul este opusul anionului, fiind format din mai mulți protoni decât electroni și implicit are o sarcină electrică pozitivă. Încheind această prezentare a fundamentelor fizice, curentul ionic este un flux de de electroni produs de grupări atomice de ioni și întâlnit la diverse nivele ale materiei.

1.1.2 Istorie

În anul 1875 fizicianul englez Richard Canton a prezentat în jurnalul British Medical Journal descoperile sale despre fenomenele electrice relevate pe cortexuri de iepuri și maimuțe.

De-a lungul vremii oameni de știință ca fiziologul polonez Adolf Beck sau fiziologul ucrainean Vladimir Vladimirovich Pravdich-Neminsky au studiat aceste fenomene electrice ale creierului. În experimentele lor subiecții erau animale supuse la diverși stimuli exteriori.

Un eveniment de o importanță majoră în istoria electroencefalografiei este produs de fiziologul și psihiatrul german Hans Berger care reușește să achiziționeze primele date pe un subiect uman. Numește dispozitivul propus de el "Electroencefalogram" și astfel netezește și potențează această calea științifică ce și-a dovedit până în zilele noastre potențialul.

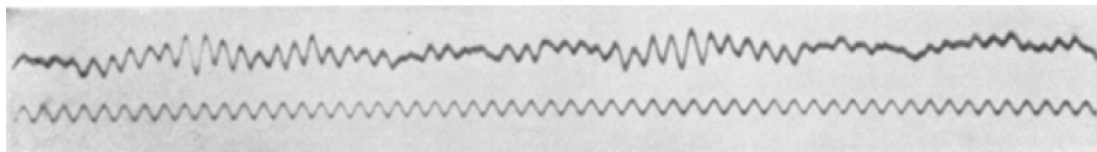


Figura 1.1: Primul EEG uman achiziționat de către Hans Berger în 1924. Semnalele din imagine sunt eșantionate la o frecvență de 10 Hz.

1.1.3 Aplicații

Activitatea creierului observată cu ajutorul acestei metode s-a dovedit prolifică în domeniul medical. Electroencefalogramele subiecților s-au dovedit utile în diagnosticarea și tratarea unor boli ca epilepsia, tumorile pe creier sau accidentele vasculare cerebrale. Unele boli ca apoplexia pot fi chiar și prezise uzitând activitatea cerebrală relevantă.

1.2 Prezentarea sistemului

1.2.1 Datele folosite

Domnul Doctor Bogdan Pavel de la Facultatea de Medicină, UMF "Carol Davila", București ne-a onorat cu împărtășirea unor electroencefalograme invazive obținute din experimente personale efectuate pe subiecți șobolani.

Părțile anatomice care prozeau cortexul subiecților a fost eliminate și anestezice au fost administrate direct în creier prin injecție.

Electrozi au fost plasați direct pe cortex în zonele corespunzătoare lobilor frontali și parietali.

Patru experimente au fost efectuate pe șobolani diferiți astfel obținându-se câte o electrocorticogramă pentru fiecare experiment.

De-a lungul acestei lucrări experimentele vor fi recunoscute prin fișierele rezultate corespunzătoare: "mouse1.acq", "mouse2.acq", "mouse3.acq" și "mouse4.acq".

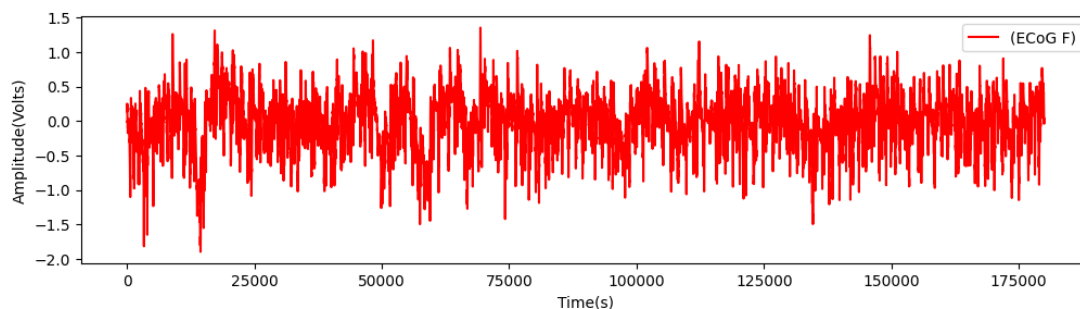


Figura 1.2: Elocorticograma lobului frontal al șobolanului 2 de-a lungul a aproximativ 180 secunde

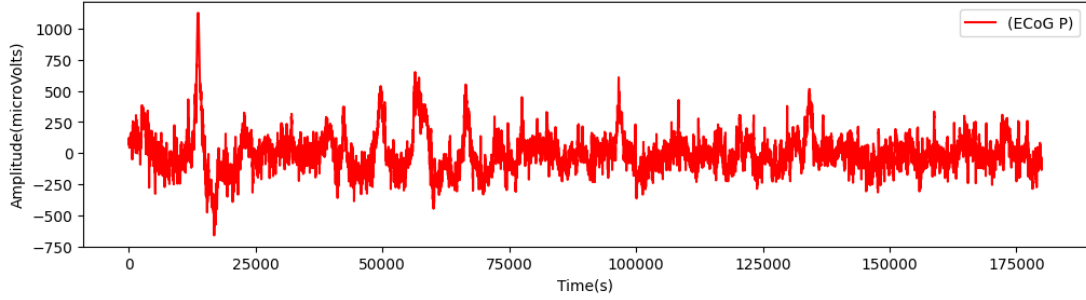


Figura 1.3: Electrocuticograma lobului parietal al șobolanului 2 de-a lungul a aproximativ 180 secunde

1.2.2 Analiza clasică a electroencefalogramelor

În mod normal astfel de informații se analizează ochiometric de către oameni specializați în acest sens. Un neurolog se antrenează timp îndelungat să dezvolte abilitățile necesare interpretării acestor semnale. Fără un sistem de calcul, analiza umană se poate face doar prin observarea unor lungi secvențe de semnale.

1.2.3 Analiza electroencefalogramelor cu ajutorul sistemului

Cu această lucrare, încercăm să oferim o nouă perspectivă pentru analiza semnalelor electroencefalografice ca Lobul Frontal sau Lobul Parietal ale șobolanului 2.

Sistemul informatic prezentat în această lucrare procesează electroencefalograme cu ajutorul învățării automate nesupervizate și le proiectează într-un spațiu bidimensional.

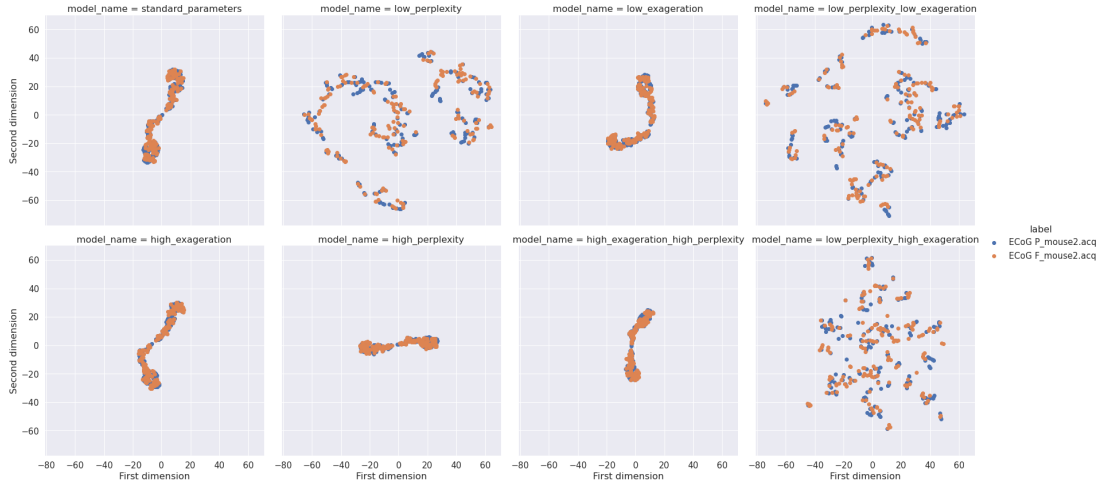


Figura 1.4: Execuția sistemului pe electroencefalogramele prezentate în Secțiunea 1.2.1

Modul în care sistemul produce aceste rezultate va fi explicitat în următoarele secțiuni ale lucrării.

Capitolul 2

Arhitectură

În acest capitol se va prezenta și explicita aplicația din diverse unghiuri și nivele de adâncime.

2.1 Arhitectura codului

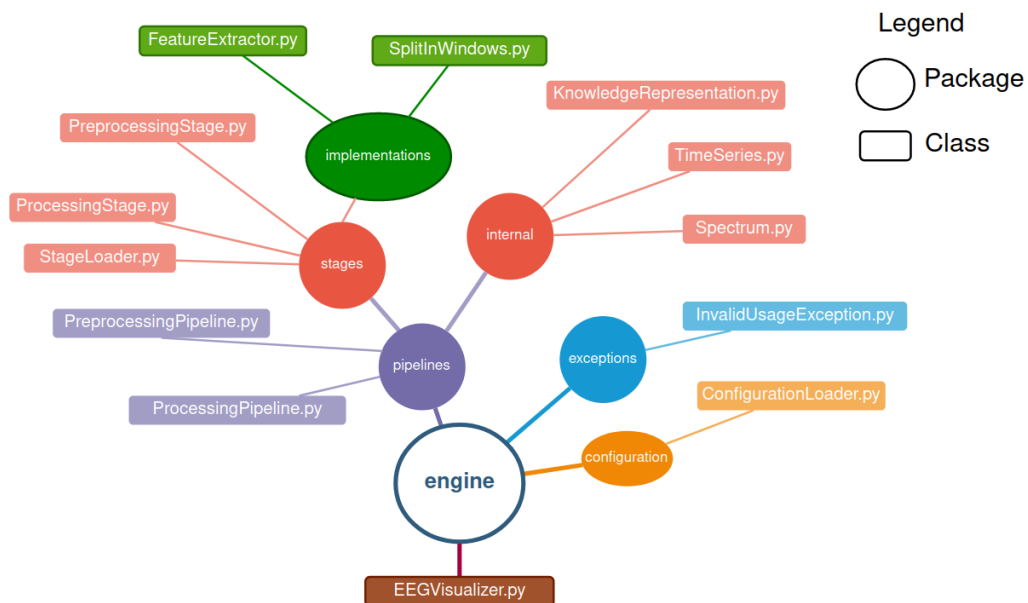


Figura 2.1: Structura codului

Sistemul este implementat folosind limbajul de programare Python.

Fiecare fișier din ierarhia prezentată în Structura codului conține doar o clasă de python cu același nume.

2.1.1 Pachetul engine

Pentru inițierea sistemului trebuie creat un obiect din clasa **EEGVisualizer**. Obiectul expune metodele necesare producerii de rezultate și creează la nevoie obiectele necesare din restul sistemului.

În mod implicit electroencefalogramele mouseX.acq prezentate în [Secțiunea 1.2.1](#) sunt încărcate în memorie la instanțierea clasei **EEGVisualizer**.

În continuare o să exemplific modul în care s-a folosit sistemul pentru generarea rezultatelor prezentate în [Figura 1.2.3](#).

```
1 eeg_visualizer = EEGVisualizer()
2 eeg_visualizer.build(files_name=["mouse2.acq"], eeg_channels=["ECoG_
   F", "ECoG_P"])
3 eeg_visualizer.execute()
4 eeg_visualizer.plot()
```

Listarea 2.1: Execution

2.1.2 Pachetul stages

TODO

2.1.3 Pachetul pipelines

TODO

2.1.4 Pachetul exceptions

TODO

2.1.5 Pachetul configuration

Pachetul respectiv oferă restului aplicației clasa ConfigurationLoader. La nevoie, alte părți ale sistemului cer din acest pachet configurația de execuție a sistemului. Un exemplu de configurație de execuție servită din acest pachet este prezentată în [Listarea A.1](#)

2.1.6 Pachetul internal

TODO

2.2 Arhitectura execuției

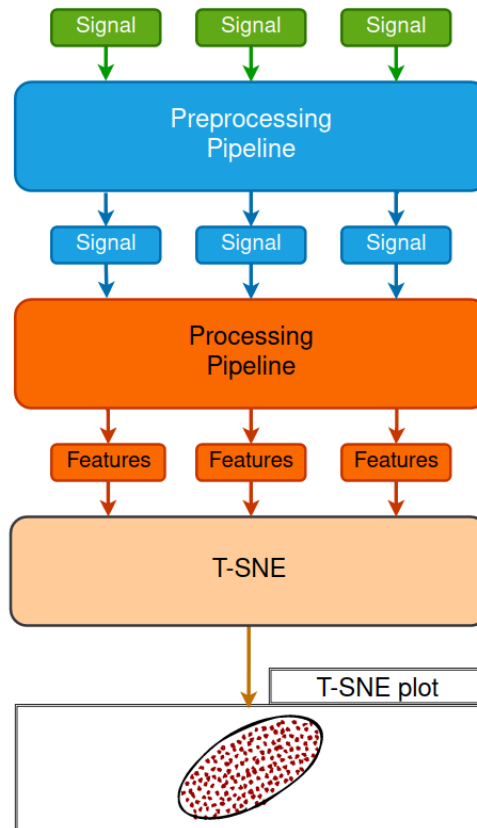


Figura 2.2: Structura execuției

Capitolul 3

Preprocesarea datelor

TODO:

Capitolul 4

Procesarea datelor

TODO: O să prezint informații despre T-SNE [1].

Capitolul 5

Rezultate și Concluzii

TODO:

Capitolul 6

Lucrări similare

TODO:

Capitolul 7

Viitor

TODO:

Appendix A

Configurații de execuție

A.1 Configurația standard de execuție

```
1 {
2   "preprocessing_pipeline_stages": [
3     {
4       "stage_name": "SplitInWindows",
5       "constructor_kwargs": {
6         "window_size": 1000
7       }
8     }
9   ],
10  "processing_pipeline_stages": [
11    {
12      "stage_name": "FeatureExtractor",
13      "constructor_kwargs": {
14        "features": [
15          "alpha_spectrum_coefficients_sum",
16          "beta_spectrum_coefficients_sum",
17          "low_gamma_spectrum_coefficients_sum",
18          "high_gamma_spectrum_coefficients_sum",
19          "delta_spectrum_coefficients_sum",
20          "theta_spectrum_coefficients_sum",
21          "mean",
22          "standard_deviation"
23        ]
24      }
25    }
26  ],
27  "t_sne_models": [
28    {
29      "model_name": "standard_parameters",
30      "parameters": {
31        "n_jobs": 6,
32        "learning_rate": "auto",
33        "init": "random",
34        "n_components": 2,
35        "early_exaggeration": 12,
```

```
36         "perplexity": 30
37     }
38 },
39 {
40     "model_name": "low_perplexity",
41     "parameters": {
42         "n_jobs": 6,
43         "learning_rate": "auto",
44         "init": "random",
45         "n_components": 2,
46         "early_exaggeration": 12,
47         "perplexity": 5
48     }
49 },
50 {
51     "model_name": "low_exaggeration",
52     "parameters": {
53         "n_jobs": 6,
54         "learning_rate": "auto",
55         "init": "random",
56         "n_components": 2,
57         "early_exaggeration": 1,
58         "perplexity": 30
59     }
60 },
61 {
62     "model_name": "low_perplexity_low_exaggeration",
63     "parameters": {
64         "n_jobs": 6,
65         "learning_rate": "auto",
66         "init": "random",
67         "n_components": 2,
68         "early_exaggeration": 1,
69         "perplexity": 5
70     }
71 },
72 {
73     "model_name": "high_exaggeration",
74     "parameters": {
75         "n_jobs": 6,
76         "learning_rate": "auto",
77         "init": "random",
78         "n_components": 2,
79         "early_exaggeration": 300,
80         "perplexity": 30
81     }
82 },
83 {
84     "model_name": "high_perplexity",
85     "parameters": {
86         "n_jobs": 6,
87         "learning_rate": "auto",
88         "init": "random",
```

```
89         "n_components": 2,  
90         "early_exaggeration": 12,  
91         "perplexity": 40  
92     }  
93 },  
94 {  
95     "model_name": "high_exaggeration_high_perplexity",  
96     "parameters": {  
97         "n_jobs": 6,  
98         "learning_rate": "auto",  
99         "init": "random",  
100        "n_components": 2,  
101        "early_exaggeration": 300,  
102        "perplexity": 40  
103    }  
104 },  
105 {  
106     "model_name": "low_perplexity_high_exaggeration",  
107     "parameters": {  
108         "n_jobs": 6,  
109         "learning_rate": "auto",  
110         "init": "random",  
111         "n_components": 2,  
112         "early_exaggeration": 300,  
113         "perplexity": 5  
114     }  
115 }  
116 ]  
117 }
```

Listarea A.1: Configurația standard de execuție

Bibliografie

- [1] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.