

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA TP.HCM



**ĐỒ ÁN GIỮA KÌ: MENTAL ATTENTION STATES
CLASSIFICATION USING EEG DATA**

Học phần: **NHẬN DẠNG MẪU**
Giảng viên hướng dẫn: **Thầy Mẫn**
Trợ Giảng: **Thầy Đức**

Thành viên nhóm :

- **Trảo An Huy – 22280041 (Nhóm trưởng)**
- Nguyễn Xuân Việt Đức – 22280012
- Mai Phong Đăng – 22280008
- Lê Đức Hòa – 22280027

TPHCM – 2024

Nội dung

I. Data Preprocessing:	1
1. Trích xuất và tải các dữ liệu kênh EEG (4-17) từ các file matlab.	1
2. Thực hiện các bước tiền xử lý cần thiết cho data.	2
II. EDA và ICA (Dùng file đầu tiên để visualize)	3
1. Một số đồ thị	3
2. Giải thích về ICA.	7
III. Feature Engineering:	9
1. Trích xuất các đặc điểm có ý nghĩa từ tín hiệu EEG	9
2. So sánh các đặc điểm giữa các trạng thái chú ý để xác định các mẫu	10
3. Trích xuất Label 3 trạng thái tinh thần và tạo ảnh Spectrogram từ các label để chuẩn bị input data cho CNN.	11
IV. Model Development:	14
1. Triển khai ít nhất hai mô hình phân loại	14
2. Support Vector Machine (SVM):	15
3. KNN (K-Nearest Neighbors)	18
4. Convolutional Neural Networks (CNN):	20
5. XGboost.	22
V. So sánh mô hình	25
VI. Tài liệu tham khảo.	26

NHẬN DẠNG MẪU

BÁO CÁO NHÓM 5- BÀI TẬP LỚN GIỮA KÌ

Thành viên nhóm :

- **Trảo An Huy – 22280041 (Nhóm trưởng)**
- Nguyễn Xuân Việt Đức – 22280012
- Mai Phong Đăng – 22280008
- Lê Đức Hòa – 22280027

I. Data Preprocessing:

1. Trích xuất và tải các dữ liệu kênh EEG (4-17) từ các file matlab.

- EEG (Electro Encephalo Gram) data là gì?
- Electrocepalogram machine ghi nhận giá trị hoạt động điện của não. các điện cực đặt trên da đầu sẽ thu các điện tích, chúng được khuếch đại và hiển thị dưới dạng biểu đồ để giải thích.
- Được sử dụng để kiểm tra các dạng sóng não, đặc biệt là để phát hiện phản ứng với các kích thích.
- Hoạt động điện não (EEG) liên quan đến việc ghi lại các tín hiệu điện do não bộ tạo ra. Những tín hiệu này có thể được biểu diễn dưới dạng hàm toán học, chủ yếu là các dao động (sóng) có dạng hình sin hoặc tổ hợp của chúng.
- Dữ liệu thí nghiệm bao gồm tín hiệu từ 25 kênh được thiết bị EMOTIV ghi lại trong 34 thí nghiệm. Các tín hiệu này được sử dụng để xây dựng mô hình nhận diện trạng thái tinh thần của con người, cụ thể là trạng thái tập trung, không tập trung và buồn ngủ.
- Mặc dù toàn bộ 25 kênh được thu thập, chỉ các kênh từ 4 đến 17, liên quan đến tín hiệu EEG, là cần thiết cho phân tích. Điều này giúp loại bỏ các tín hiệu không liên quan như dữ liệu từ gia tốc kế hoặc dấu thời gian.
- Kích thước: Mỗi file Matlab chứa một đối tượng dữ liệu (o.data) với kích thước {số mẫu} x 25.
- Tần số lấy mẫu: 128 Hz.- Cấu trúc: Các cột tương ứng với 25 kênh tín hiệu như sau:

ID	Tên kênh	Loại dữ liệu	Mô tả
1	ED_COUNTER	Bộ đếm	Đếm số lần xảy ra sự kiện
2	ED_INTERPOLATED	Trạng thái nội suy	Cho biết dữ liệu có bị nội suy không
3	ED_RAW_CQ	Chất lượng tín hiệu	Đánh giá chất lượng tín hiệu thu được
4-17	EEG Channels	Tín hiệu điện não EEG	Hoạt động điện của não
18-19	ED_GYROX, ED_GYROY	Dữ liệu gia tốc	Gia tốc theo trục X và Y
20-25	Các tín hiệu khác	Dấu thời gian, marker	Thông tin bổ sung về dữ liệu

Kết quả extract data:

```
{ '__header__': b'MATLAB 5.0 MAT-file, Platform: PCWIN, Created on: Thu Oct 09 21:03:01 2014', '__version__': '1.0', '__globals__': [], 'o': array([[[array(['201410092013.0091B844'], dtype='<f8'),
[0],
[0],
...,
[0],
[0],
dtype='uint8'], array([[2.0140e+03, 1.0000e+01, 9.0000e+00, 2.0000e+01, 1.4000e+01,
1.5360e+00],
[2.0140e+03, 1.0000e+01, 9.0000e+00, 2.0000e+01, 1.4000e+01,
1.5360e+00],
[2.0140e+03, 1.0000e+01, 9.0000e+00, 2.0000e+01, 1.4000e+01,
1.5360e+00],
...,
[2.0140e+03, 1.0000e+01, 9.0000e+00, 2.0000e+01, 5.4000e+01,
1.6494e+01],
[2.0140e+03, 1.0000e+01, 9.0000e+00, 2.0000e+01, 5.4000e+01,
1.6494e+01],
[2.0140e+03, 1.0000e+01, 9.0000e+00, 2.0000e+01, 5.4000e+01,
1.6494e+01]]], array([[ 3., 0., 463., ..., 0., 0., 0.],
[ 4., 0., 0., ..., 0., 0., 0.],
[ 5., 0., 447., ..., 0., 0., 0.],
...,
[25., 0., 0., ..., 0., 0., 0.],
[26., 0., 0., ..., 0., 0., 0.],
[27., 0., 0., ..., 0., 0., 0.]])], array([[[[4441.02564103, 4441.02564103, 4440.
...,
4333.33333333, 4333.84615385, 4334.87179487],
[4147.69238769, 4144.1025641 , 4146.15384615, ...,
4122.05128205, 4130.25641026, 4135.8974359 ]]]]))
dtype=[('id', 'o'), ('tag', 'o'), ('ns', 'o'), ('sampFreq', 'o'), ('marker', 'o'), ('timestamp', 'o'), ('data', 'o'), ('trials', 'o')])]
```

2. Thực hiện các bước tiền xử lý cần thiết cho data.

- B1: Preprocessing, dựa vào paper chúng em có tham khảo một số cách

- Thứ nhất, dữ liệu cần được chuyển sang miền tần số và lọc nhiễu, lọc ồn và em sử dụng FFT để lọc miền tần số vì FFT được dùng để loại bỏ nhiễu không mong muốn bằng cách chỉ giữ lại các thành phần tín hiệu trong một dải tần số hữu ích (ví dụ: 0–30 Hz).
- Thứ hai, dữ liệu cần được chuyển sang miền normalize, tại em sử dụng min-max scaler để xử lý phần này
- Thứ ba, tại em sử dụng STFT để chuyển đổi các đoạn dữ liệu thành các phân mảnh nhỏ hơn với tần số bin là 0.5hz, tạo ra được một spectrum (dải quang phổ), trong spectrum thì mỗi spectrum được tạo ra chứa một lượng dữ liệu tương ứng với label theo thử nghiệm.

II. EDA và ICA (Dùng file đầu tiên để visualize)

1. Một số đồ thị

```
[ ] column_3_to_17_mat1 = matfile1.data.iloc[:,3:17]
```

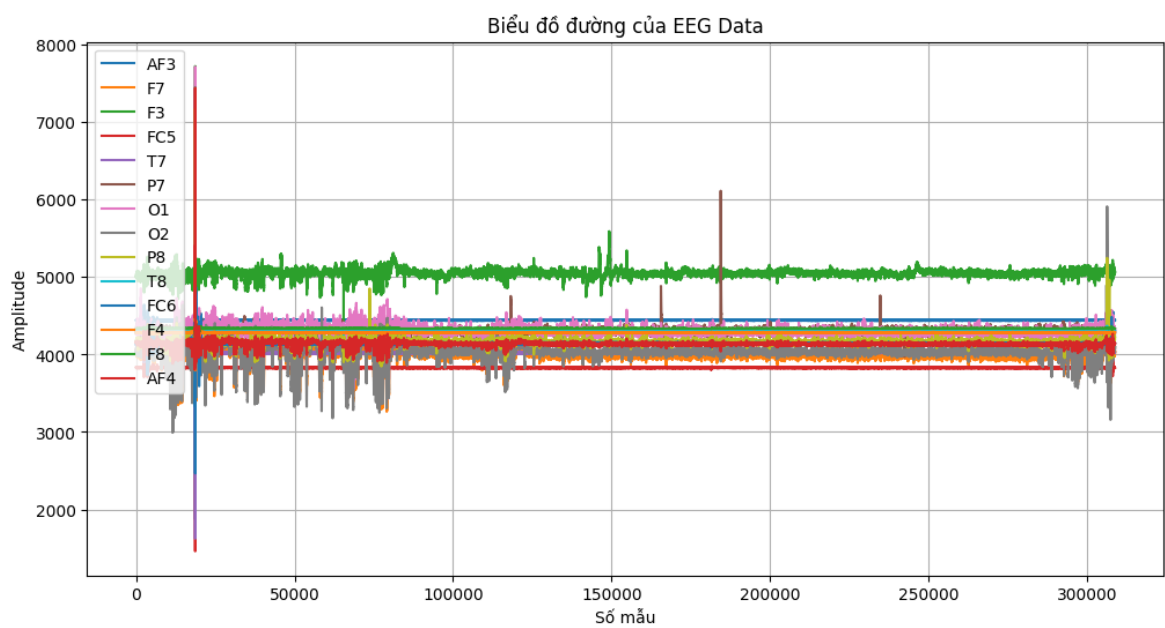
```
column_3_to_17_mat1.head() # in ra thử data frame
```

	AF3	F7	F3	FC5	T7	P7	O1	O2	P8	T8	F
0	4440.000000	4417.948718	5390.769231	3833.846154	4019.487179	4656.410256	4742.051282	4415.384615	4538.461538	4308.205128	4125.6410
1	4439.487179	4417.948718	5389.230769	3830.256410	4020.000000	4658.974359	4738.974359	4414.871795	4535.897436	4308.205128	4136.4102
2	4438.974359	4414.871795	5385.641026	3829.743590	4017.948718	4655.897436	4734.358974	4414.871795	4537.435897	4308.205128	4152.8201
3	4438.974359	4410.256410	5381.025641	3831.794872	4016.410256	4649.230769	4730.256410	4409.743590	4535.384615	4308.717949	4157.9487
4	4439.487179	4407.179487	5378.461538	3831.282051	4019.487179	4647.692308	4729.230769	4408.205128	4526.666667	4307.179487	4157.9487

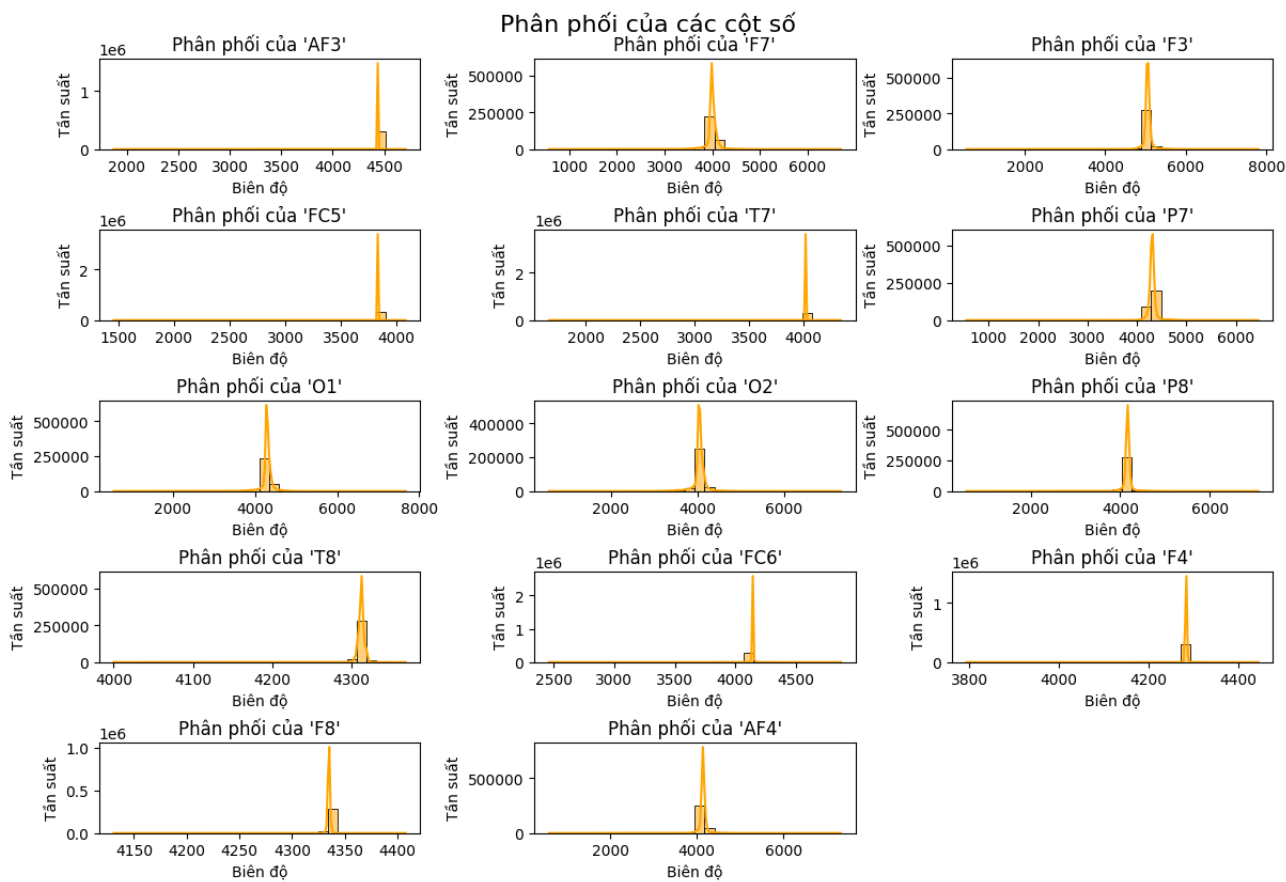
Ma trận tương quan:



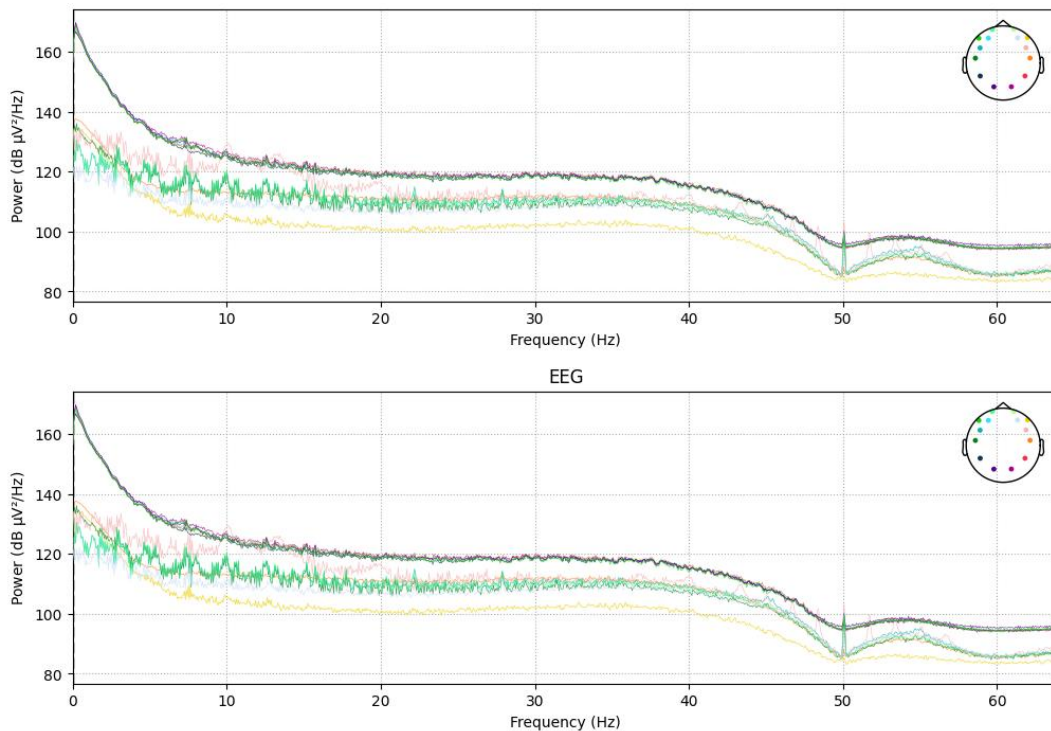
Biểu đồ đường theo thời gian:



Biểu đồ phân phối các kênh của dữ liệu



Đồ thị power spectral density (PSD)



Nhận Xét Năng lượng có dây công suất mạnh nhất là ở khoảng từ 0 -> 10 Hz và có xu hướng giảm dần tại điểm có dây là 50 thì xuất hiện một số đột biến nhỏ trong tập dữ liệu trên tại thời điểm power = 100 thì giá trị tập dữ liệu giảm đi còn 1/3 -Biểu đồ PSD cho thấy tín hiệu EEG chủ yếu tập trung ở dải tần số thấp (0–10 Hz), với hoạt động Alpha (8–13 Hz) nổi bật ở vùng chẩm (O1, O2). Điều này phù hợp với trạng thái thư giãn của người tham gia thí nghiệm. Tuy nhiên, nhiễu điện lưới xuất hiện ở 50–60 Hz cần được lọc bỏ để cải thiện chất lượng tín hiệu..

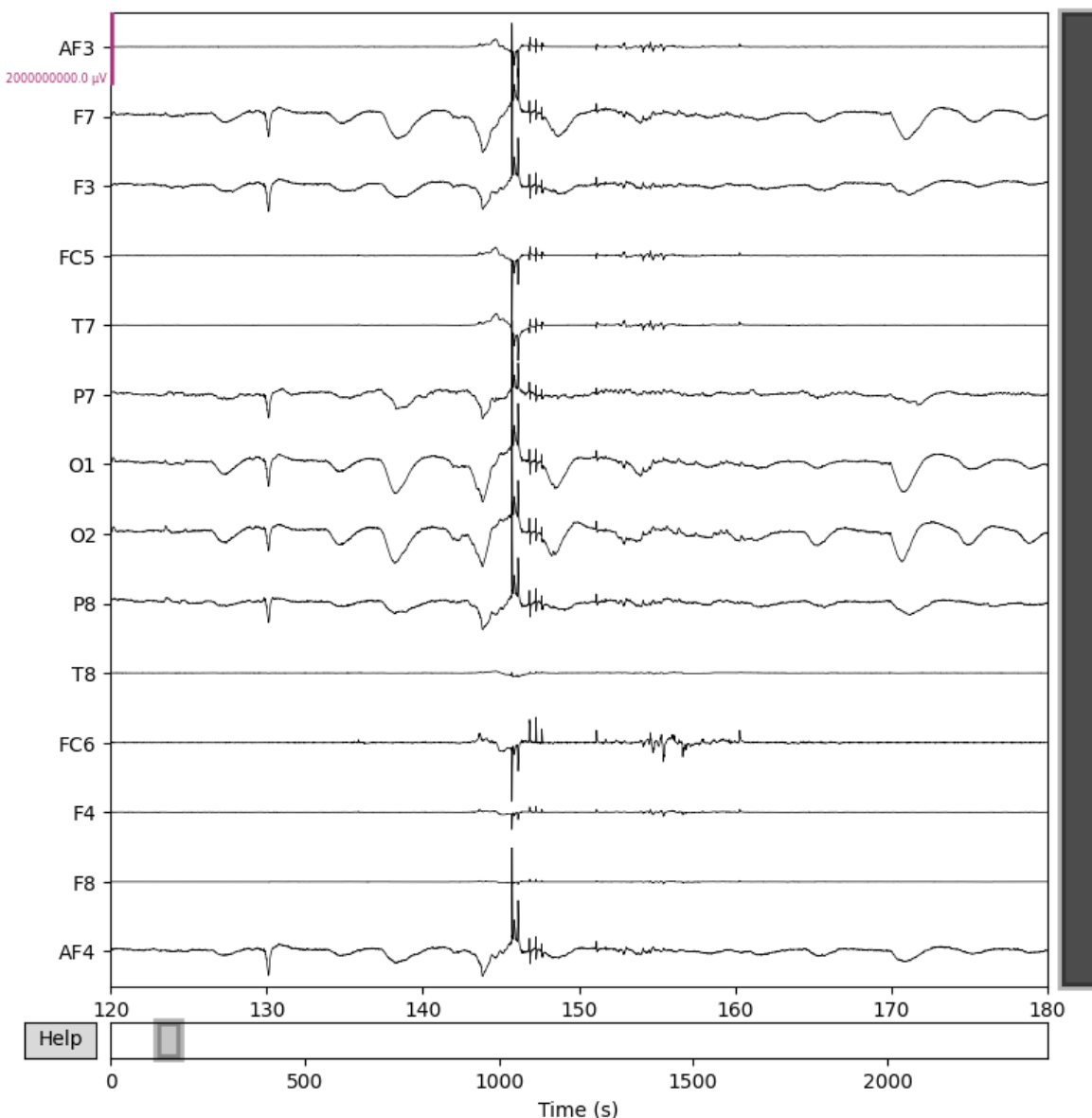
Trục hoành (Frequency): Biểu thị tần số của sóng não, thường được chia thành các băng tần như Delta, Theta, Alpha, Beta và Gamma. Trục Tung (Power): Thể hiện công suất của tín hiệu EEG tại từng tần số. Đơn vị dB $\mu V/Hz$ cho biết mức năng lượng trong một băng tần cụ thể. Đường Đồ Thị: Mỗi đường màu có thể đại diện cho các kênh EEG khác nhau hoặc các điều kiện khác nhau trong quá trình ghi nhận. Sự thay đổi của các đường này cho thấy sự khác biệt trong hoạt động điện não giữa các kênh hoặc điều kiện. Biểu Đồ Phụ (Inset Plot): Hình tròn nhỏ ở góc phải có thể biểu thị vị trí của các kênh EEG trên đầu, giúp người xem hiểu rõ hơn về nguồn gốc của tín hiệu.

Vẽ đồ thị time-series plot.

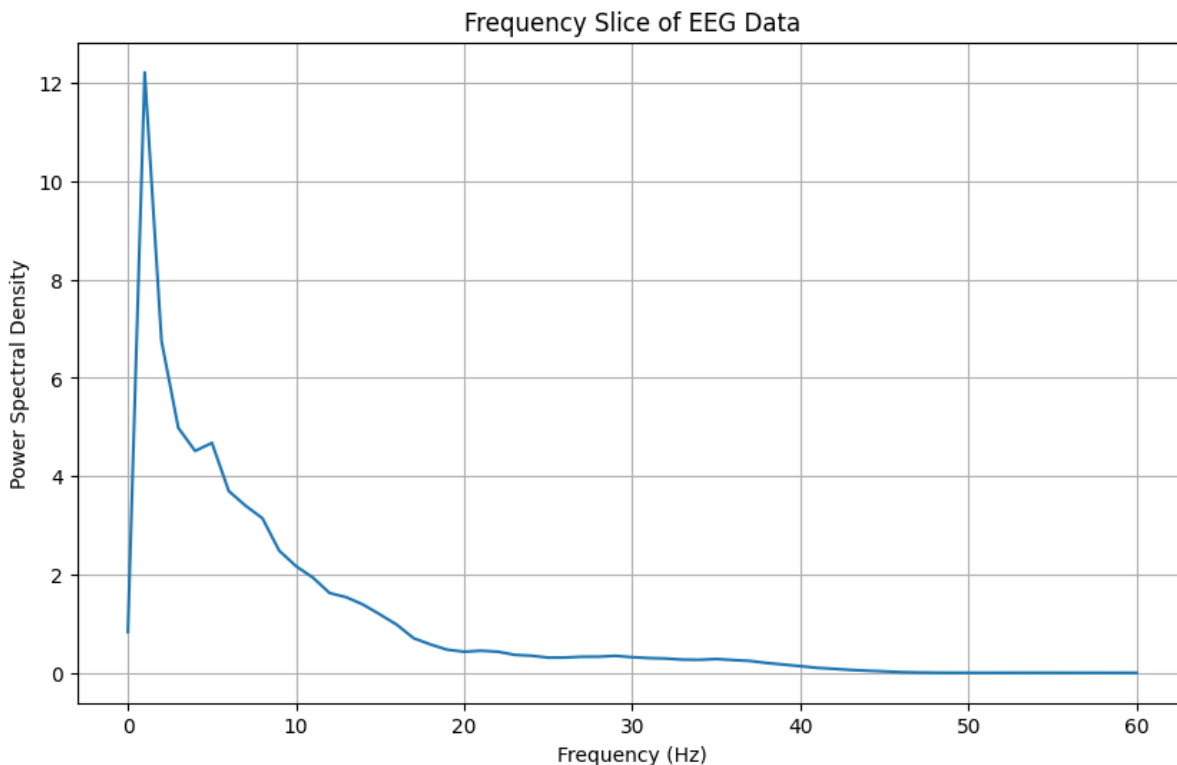
Nhận Xét Năng lượng có dây công suất mạnh nhất là ở khoảng từ 0 -> 10 Hz và có xu hướng giảm dần tại điểm có dây là 50 thì xuất hiện một số đột biến nhỏ trong tập dữ liệu trên tại thời điểm power = 100 thì giá trị tập dữ liệu giảm đi còn 1/3 -Biểu đồ PSD cho thấy tín hiệu EEG chủ yếu tập trung ở dải tần số thấp (0–10 Hz), với hoạt động Alpha (8–13 Hz) nổi bật ở vùng chẩm (O1, O2). Điều này phù hợp với trạng thái thư giãn của người tham gia thí nghiệm. Tuy nhiên, nhiễu điện lưới xuất hiện ở 50–60 Hz cần được lọc bỏ để cải thiện chất lượng tín hiệu..

Trục Hoành (Frequency): Biểu thị tần số của sóng não, thường được chia thành các băng tần như Delta, Theta, Alpha, Beta và Gamma. Trục Tung (Power): Thể hiện công suất của tín hiệu EEG tại từng tần số. Đơn vị dB $\mu\text{V}/\text{Hz}$ cho biết mức năng lượng trong một băng tần cụ thể. Đường Đồ Thị: Mỗi đường màu có thể đại diện cho các kênh EEG khác nhau hoặc các điều kiện khác nhau trong quá trình ghi nhận. Sự thay đổi của các đường này cho thấy sự khác biệt trong hoạt động điện não giữa các kênh hoặc điều kiện. Biểu Đồ Phụ (Inset Plot): Hình tròn nhỏ ở góc phải có thể biểu thị vị trí của các kênh EEG trên đầu, giúp người xem hiểu rõ hơn về nguồn gốc của tín hiệu.

Vẽ đồ thị time-series plot.



Vẽ đồ thị Frequency Slice tập dữ liệu.



2. Giải thích về ICA

Là một kỹ thuật phân tích dữ liệu dùng để tách các tín hiệu độc lập từ một tập hợp các tín hiệu hỗn hợp. Mục đích của ICA là tìm ra các nguồn gốc tín hiệu độc lập, không liên quan đến nhau, từ một hoặc nhiều tín hiệu đã bị trộn lẫn.

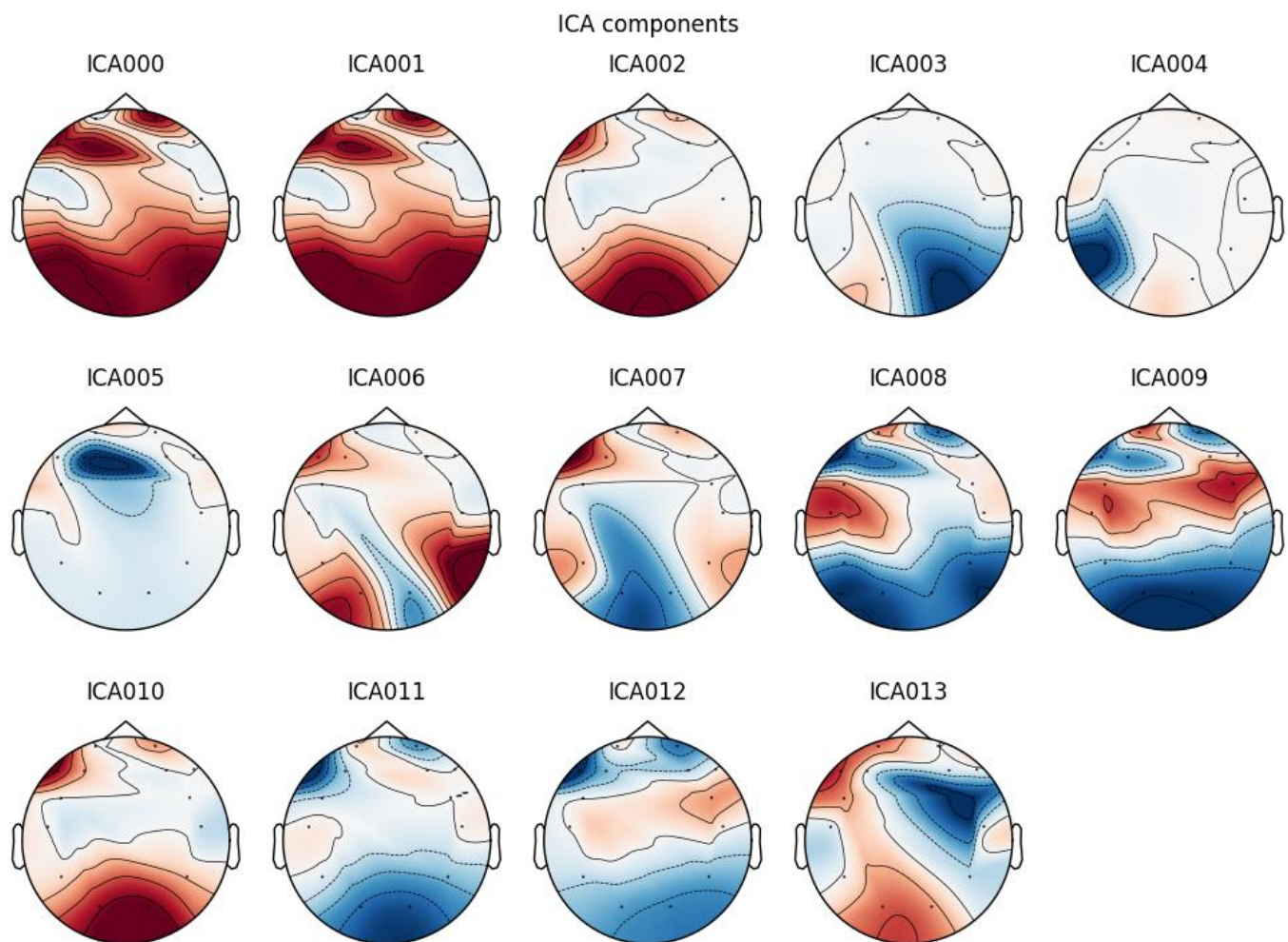
Hãy tưởng tượng bạn đang ngồi trong một phòng và nghe 3 người nói cùng lúc. Nếu bạn chỉ có một microphone ghi âm, tất cả âm thanh từ 3 người sẽ trộn lẫn với nhau và bạn sẽ không thể nghe rõ từng người nói. ICA sẽ giúp bạn "tách" âm thanh của từng người ra từ tín hiệu hỗn hợp mà microphone ghi được, giống như việc bạn có thể nghe rõ ràng từng người nói dù chỉ có một microphone.

Ví dụ với EEG:

EEG (Electroencephalogram) là một phương pháp ghi lại hoạt động điện não. Các điện cực trên da đầu sẽ ghi nhận tín hiệu điện từ não, nhưng đôi khi tín hiệu này bị "trộn" với những tín hiệu khác, như tín hiệu cơ thể hoặc nhiễu từ môi trường xung quanh.

Giả sử bạn đang ghi nhận tín hiệu EEG của một người trong khi họ đang nghe nhạc. Tín hiệu EEG sẽ chứa các thông tin từ não của họ khi họ nghe nhạc, nhưng cũng có thể có thêm tín hiệu nhiễu từ các cơ bắp (như nhịp đập của tim hoặc các chuyển động cơ thể).

Nếu dùng ICA, nó sẽ giúp "tách" các tín hiệu này ra. Ví dụ, ICA có thể tách riêng tín hiệu từ não (hoạt động não khi nghe nhạc) và tín hiệu từ các cơ bắp (như tiếng chuyển động hoặc đập tim). Điều này rất hữu ích để phân tích và hiểu rõ hơn về các hoạt động của não mà không bị nhiễu.



Mỗi thành phần ICA: Mỗi hình trong hình ảnh đại diện cho một thành phần được tách ra từ tín hiệu EEG bằng phương pháp ICA. Các thành phần này có thể là các nguồn gốc tín hiệu độc lập, ví dụ như tín hiệu từ não, tín hiệu cơ thể, hoặc nhiễu.

Màu sắc: Màu sắc trên bản đồ không gian cho thấy cường độ của tín hiệu ở các vị trí khác nhau trên đầu. Màu đỏ và màu xanh tương phản thể hiện các giá trị cực đại và cực tiểu của tín hiệu tại các vị trí khác nhau.

Màu đỏ: Cường độ tín hiệu dương (tín hiệu mạnh ở những vùng này). Màu xanh: Cường độ tín hiệu âm (tín hiệu yếu hoặc không có tín hiệu ở các khu vực này). Các đường viền (contours) cũng thể hiện sự phân bố của tín hiệu trên các khu vực khác nhau. Vị trí kênh EEG: Mỗi hình ảnh thể hiện sự phân bố tín hiệu của thành phần ICA tại các vị trí trên đầu, như thể hiện trong hệ thống điện cực EEG chuẩn (ví dụ: 10-20 system).

Số thành phần ICA: Các thành phần ICA được đánh số từ ICA000 đến ICA013, mỗi thành phần có thể tương ứng với một nguồn tín hiệu độc lập được tách ra từ các tín hiệu hỗn hợp ban đầu.

chọn useful_channels là như trên vì F7 và F3: Thường liên quan đến các vùng trán trước (frontal), có thể hữu ích trong nghiên cứu về các tình trạng nhận thức hoặc hoạt động thần kinh trong quá trình ra quyết định, tập trung, hay các trạng thái tâm lý khác. P7, P8: Liên quan đến vùng chẩm (parietal) và có thể ghi lại tín hiệu liên quan đến quá trình nhận thức và hành động. O1, O2: Liên

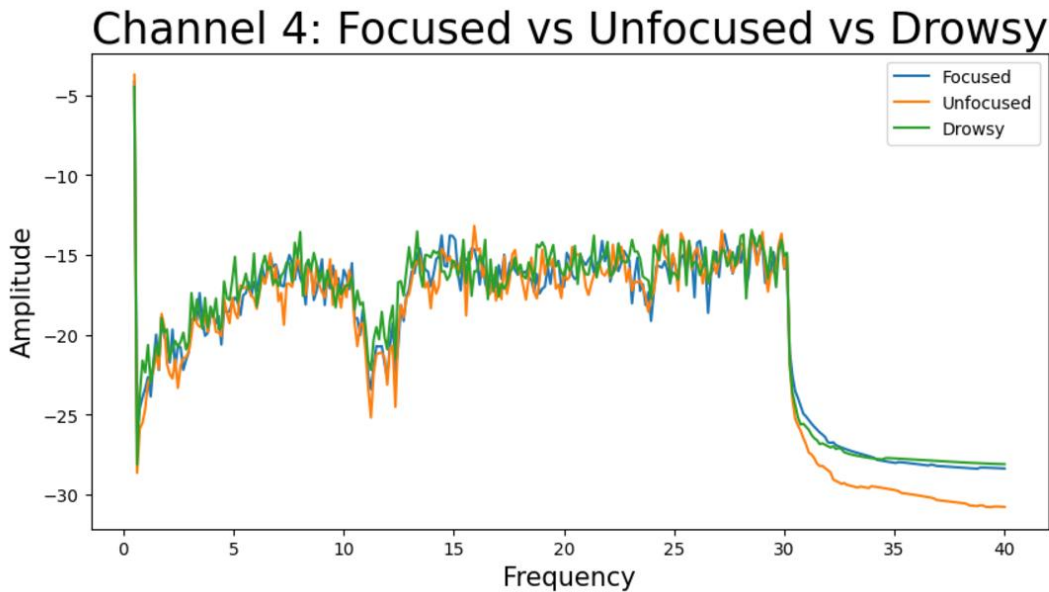
quan đến vùng chẩm (occipital), có thể ghi lại hoạt động liên quan đến thị giác hoặc trạng thái nghỉ ngơi. AF4: Vị trí gần vùng trán (frontal) phía trước, có thể hữu ích trong việc phát hiện các thay đổi trong hoạt động não.

III. Feature Engineering:

1. Trích xuất các đặc điểm có ý nghĩa từ tín hiệu EEG

- Việc chọn các kênh EEG hợp lý là rất quan trọng trong nghiên cứu não bộ. Nghiên cứu này sử dụng hệ thống 10-20 tiêu chuẩn để định vị các điện cực, tập trung vào các vùng não quan trọng như trán (frontal), đỉnh (central), và chẩm (occipital).
- Ban đầu, dữ liệu được thu thập từ 7 kênh: F3, F4, Fz, C3, C4, Cz, Pz. Tuy nhiên, không phải tất cả các kênh đều có giá trị như nhau. Nghiên cứu đã tinh chỉnh để chọn ra một tập hợp các kênh hữu ích hơn: F7, F3, F4, AF4, P7, P8, O1, O2.
- Các kênh được lựa chọn với mục đích cụ thể:
 - Kênh vùng trán (F7, F3, F4, AF4): Giúp phân tích các sóng não liên quan đến trạng thái nhận thức như sự tập trung và căng thẳng.
 - Kênh vùng đỉnh (P7, P8): Hỗ trợ nghiên cứu các hoạt động cảm giác và xử lý thông tin không gian.
 - Kênh vùng chẩm (O1, O2): Cung cấp thông tin về các hoạt động thị giác và sự chú ý.
- Quá trình xử lý tín hiệu bao gồm các bước chính:
 - Chia tín hiệu EEG thành các cửa sổ nhỏ (ví dụ 8 giây) Sử dụng Biến đổi Fourier nhanh (FFT) để chuyển đổi tín hiệu sang miền tần số.
 - Tính toán năng lượng tín hiệu trong các dải tần số dưới 40 Hz.
 - Chuẩn hóa năng lượng để dễ dàng so sánh.
- Lợi ích của phương pháp này:
 - Giảm thiểu khối lượng dữ liệu
 - Tăng hiệu quả xử lý
 - Tập trung vào các kênh mang thông tin quan trọng nhất
 - Cung cấp cái nhìn sâu sắc về các trạng thái não bộ khác nhau
- Việc tính toán năng lượng tín hiệu giúp đo lường cường độ hoạt động não ở các dải tần số cụ thể, từ đó hỗ trợ việc phân biệt và phân tích các trạng thái não bộ một cách chính xác hơn.

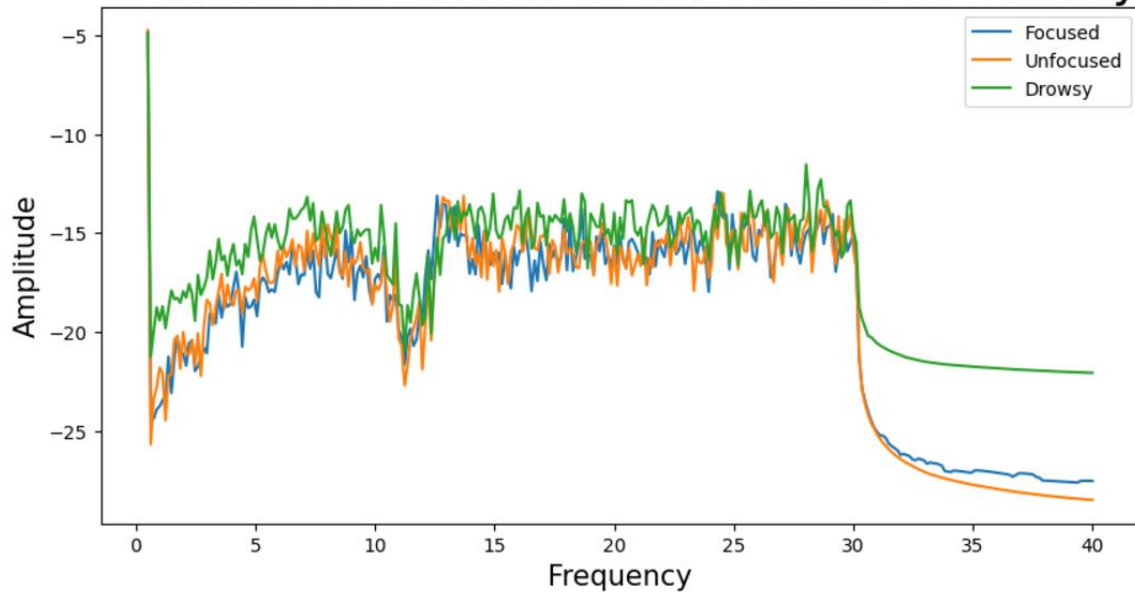
2. So sánh các đặc điểm giữa các trạng thái chú ý để xác định các mẫu.



- Đây là biểu đồ so sánh biên độ theo tần số của ba trạng thái (Focused, Unfocused, Drowsy) ở kênh O1.

- **Khu vực tần số thấp (0-5 Hz):** Ở trạng thái Drowsy, biên độ cao hơn hẳn so với hai trạng thái còn lại, đặc biệt là ở tần số rất thấp. Điều này có thể cho thấy sự xuất hiện mạnh mẽ của sóng Delta, thường gắn liền với trạng thái buồn ngủ hoặc trạng thái thư giãn sâu.
- **Khu vực tần số trung bình (5-30 Hz):** Biên độ ở cả ba trạng thái có xu hướng giao thoa và dao động xung quanh mức tương đương. Trạng thái Unfocused có sự biến động mạnh hơn, thể hiện qua biên độ dao động lớn hơn ở nhiều tần số. Điều này có thể cho thấy sự thiếu ổn định trong hoạt động não khi không tập trung.
 - Trạng thái Drowsy có sự khác biệt rõ ràng ở cả hai đầu phổ tần số, đặc biệt là sóng Delta mạnh và biên độ giảm nhanh ở tần số cao.
 - Trạng thái Focused có sự ổn định hơn so với Unfocused, với biên độ ít biến động ở vùng trung bình.
 - Unfocused cho thấy dấu hiệu hoạt động không ổn định, có thể liên quan đến việc mất tập trung.

Channel 7: Focused vs Unfocused vs Drowsy

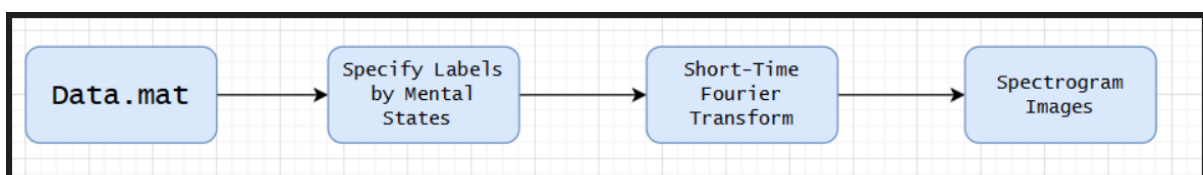


- **Tập trung (Focused):** Mô hình khá tương tự kênh O1 với biên độ giảm nhanh ở tần số thấp và dao động ổn định ở khoảng tần số trung bình.
- **Không tập trung (Unfocused):** Biên độ biến đổi nhiều hơn trong dải tần số từ 10 đến 30 Hz, nhưng giảm dần ở mức tần số cao.
- **Buồn ngủ (Drowsy):** Biên độ rất thấp ở dải tần số cao (trên 30 Hz), phản ánh trạng thái não bị ức chế trong tình trạng buồn ngủ.

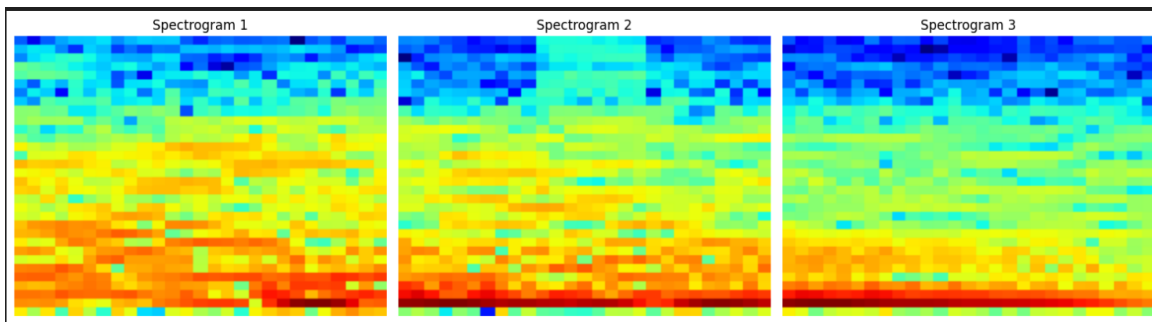
Kênh O1 (vùng chẩm): Chủ yếu liên quan đến xử lý thị giác, cho thấy sự khác biệt rõ rệt giữa ba trạng thái (tập trung, không tập trung, buồn ngủ) qua các tần số thấp và trung bình.

Kênh AF4 (vùng trán): Liên quan đến chức năng điều khiển nhận thức, cảm xúc. Biểu đồ AF4 phản ánh rõ hơn sự thay đổi trạng thái từ không tập trung sang buồn ngủ, đặc biệt ở tần số cao (trên 30 Hz).

3. Trích xuất Label 3 trạng thái tinh thần và tạo ảnh Spectrogram từ các label để chuẩn bị input data cho CNN.



- Dữ liệu đầu vào: Bộ dữ liệu gồm các tệp .mat chứa tín hiệu EEG đa kênh, được ghi với tần số lấy mẫu 128 Hz.
- Mỗi tệp chứa tín hiệu EEG tương ứng với ba trạng thái tinh thần:
 - Tập trung: 0–76800 mẫu.
 - Mất tập trung: 76801–153600 mẫu.
 - Buồn ngủ: 153601–230400 mẫu.
- Dữ liệu đầu ra: Tín hiệu được xuất ra dưới dạng tệp .csv, phân đoạn theo trạng thái tinh thần.
- Phổ đồ của từng kênh và đoạn tín hiệu được lưu dưới dạng hình ảnh để phân tích trực quan hoặc làm đầu vào cho mô hình học máy.



- Feature engineering XGBOOST:

✓ Phân tích toán học của Short Time Fourier Transform (STFT)

1. Định nghĩa cơ bản của STFT

STFT của một tín hiệu ($x(t)$) được định nghĩa bằng cách tính Fourier Transform trên các đoạn nhỏ của tín hiệu, sử dụng một hàm cửa sổ ($w(t)$). Biểu thức:

$$X(t, f) = \int_{-\infty}^{\infty} x(\tau) \cdot w(\tau - t) \cdot e^{-j2\pi f\tau} d\tau$$

- ($x(t)$): Tín hiệu gốc trong miền thời gian.
- ($w(t)$): Hàm cửa sổ trượt qua tín hiệu (như Hamming, Hanning,...).
- (t): Thời gian tại tâm của cửa sổ ($w(t)$).
- (f): Tần số trong miền Fourier.
- ($e^{-j2\pi f\tau}$): Phần chuyển đổi Fourier (dạng sóng phức).

2. Các bước phân tích

Bước 1: Cửa sổ hóa tín hiệu

Tín hiệu ($x(t)$) được nhân với một hàm cửa sổ ($w(\tau - t)$) để chọn một đoạn nhỏ quanh thời điểm (t). Kết quả là một tín hiệu cục bộ ($x_{\text{local}}(\tau, t)$):

$$x_{\text{local}}(\tau, t) = x(\tau) \cdot w(\tau - t)$$

Bước 2: Biến đổi Fourier trên đoạn cục bộ

Lấy Fourier Transform của tín hiệu ($x_{\text{local}}(\tau, t)$):

$$X(t, f) = \int_{-\infty}^{\infty} x(\tau) \cdot w(\tau - t) \cdot e^{-j2\pi f\tau} d\tau$$

Ý nghĩa:

- ($X(t, f)$) là một hàm phức trong miền thời gian (t) và miền tần số (f).
- Độ lớn ($|X(t, f)|$) biểu diễn năng lượng của tín hiệu ở tần số (f) tại thời điểm (t).
- Pha ($\angle X(t, f)$) cung cấp thông tin về pha của tín hiệu.

Đầu tiên, khởi tạo với các tham số như tần số lấy mẫu danh sách các kênh tín hiệu cần thiết và độ dài tín hiệu.

Sau đó, định vị kênh cần thiết trong danh sách tất cả các kênh. Để phân tích tín hiệu, phương thức sử dụng phương thức STFT chuyển đổi tín hiệu từ miền thời gian sang miền tần số thông qua phép biến đổi Fourier ngắn hạn (stft), trả về các giá trị tần số, thời gian, và phổ biên độ của tín hiệu.

Tiếp theo, phương thức chia tín hiệu thành các phân đoạn nhỏ theo khoảng thời gian và bước nhảy, sau đó tính phổ thời gian thông qua stft. Phương pháp này chỉ giữ các tần số hữu ích (≤ 30 Hz) để loại bỏ nhiễu, đồng thời chuẩn hóa năng lượng tín hiệu qua thang decibel (db). Phương thức làm sạch dữ liệu bằng cách giới hạn tín hiệu trong khoảng thời gian nhất định, sau đó trích xuất các đặc trưng năng lượng từ các phân đoạn tín hiệu sau đó sử dụng min max scaler để chuẩn hóa dữ liệu về dạng 0-1.

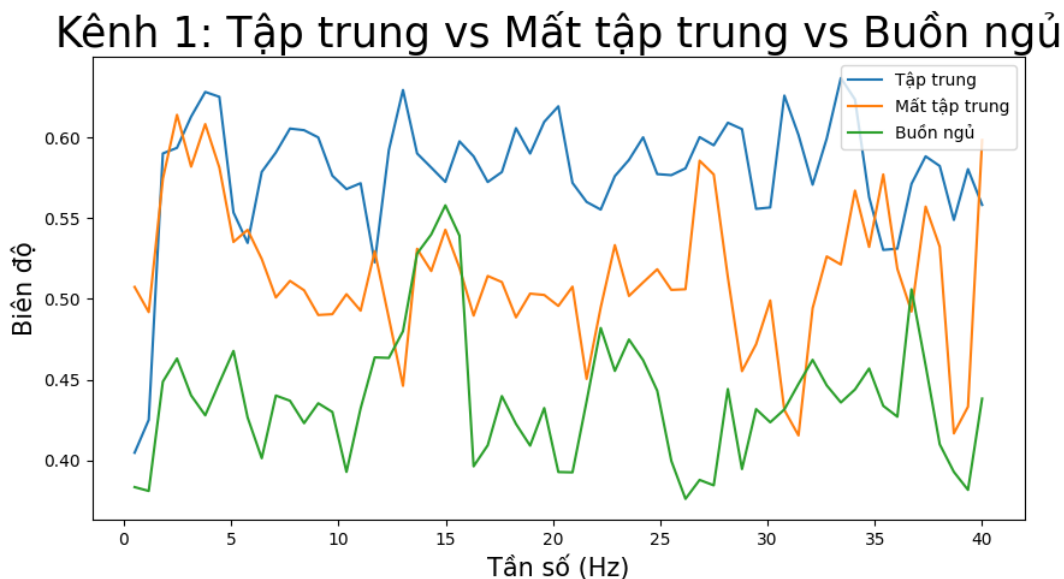
Cuối cùng, pipeline chính xử lý toàn bộ dữ liệu EEG từ các tệp mat. Dữ liệu được chia thành ba trạng thái: focussed (10 phút đầu), unfocussed (10 phút tiếp theo), và drowsed (10 phút cuối).

Mỗi trạng thái được làm sạch và trích xuất đặc trưng riêng biệt. Kết quả xử lý được lưu vào các tệp .pkl, chứa dữ liệu tín hiệu đã được xử lý, danh sách các kênh hữu ích, và tần số lấy mẫu, để phục vụ cho các bước phân tích hoặc huấn luyện mô hình tiếp theo.

Đặc biệt khác với phương pháp xử lý SVM hay KNN thì DATA được đưa về dạng thang Db để dễ thấy sự phân cách của dữ liệu hơn..

Data thu được:

```
10000_data
{
  'channels': ['F7', 'F3', 'P7', 'O1', 'O2', 'P8', 'AF4'],
  'fs': 128,
  'trial': {'focussed': array([[0.42311116, 0.463779, 0.35992482, ..., 0.35054127,
0.3425254, 0.34435661],
[0.11966197, 0.16948072, 0.70762136, ..., 0.61598107,
0.6982161, 0.68995816],
[0.58990151, 0.61118734, 0.72118062, ..., 0.75793578,
0.6839754, 0.64709417],
...,
[0.60436894, 0.63275195, 0.7808134, ..., 0.80523292,
0.90992298, 0.93563889],
[0.09663466, 0.096354, 0.12099875, ..., 0.33411629,
0.36856368, 0.42439465],
[0.38883896, 0.41422283, 0.51353668, ..., 0.635744,
0.6716564, 0.6395221 ]],
[[0., 0., 0., ..., 0.,
0., 0., ],
[0.35208304, 0.4244318, 0.70989057, ..., 0.46915516,
0.63059672, 0.74897714],
[0.69444661, 0.73014449, 0.95052312, ..., 0.87473391,
0.78629803, 0.71955814],
...,
[0.35310497, 0.38508908, 0.41741943, ..., 0.52949204,
0.607195, 0.71419771],
[0.26646297, 0.23715412, 0.22194527, ..., 0.65433778,
0.85069873, 0.76233468],
[0.36338517, 0.39143805, 0.82496287, ..., 0.83972164,
0.89828741, 0.91907795]]],
[[0.64764502, 0.67463621, 0.78744288, ..., 0.57921711,
```



Biểu đồ hiển thị các state của của XGboost sau khi feature extraction.

IV. Model Development:

1. Triển khai ít nhất hai mô hình phân loại

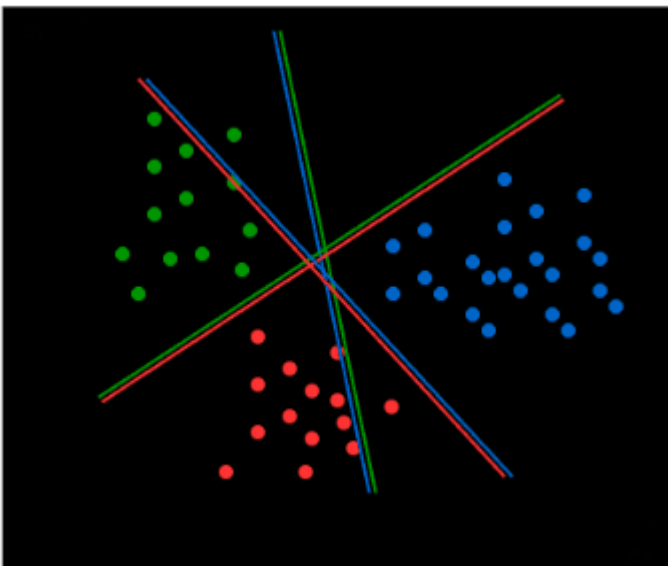
- Sử dụng các model học máy để huấn luyện model với nguồn data là một tập dữ liệu được gộp từ 34 file .mat, được tạo thành từ bước feature engineering thu được một tập dữ trong notebook với các cấu trúc như sau:

- Feature engineering.
- model theory

- Hyper parameter.
- Performance evaluation.
- Xây dựng class và pipeline.
- Visualize

2. Support Vector Machine (SVM):

- Support Vector Machine (SVM) là một thuật toán học có giám sát, được sử dụng để phân loại và hồi quy. Trong phân loại, SVM hoạt động bằng cách tìm kiếm một siêu phẳng (hyperplane) tối ưu để phân chia dữ liệu thành các lớp khác nhau. Mục tiêu của SVM là tối đa hóa khoảng cách giữa các điểm dữ liệu gần nhất của hai lớp và siêu phẳng phân chia chúng. Khoảng cách này được gọi là lề (margin), và một siêu phẳng với lề lớn hơn thường có khả năng tổng quát hóa tốt hơn khi phân loại các điểm dữ liệu mới.
- Phân loại trạng thái nhận thức của người dùng (tập trung, buồn ngủ, hoặc không tập trung).
- Xử lý dữ liệu phức tạp và phi tuyến tính: EEG là tín hiệu phức tạp, có thể chứa các mẫu phi tuyến. SVM với hạt nhân (kernel) như Gaussian RBF hoặc Polynomial có khả năng mô hình hóa tốt các ranh giới phi tuyến trong không gian đặc trưng. Chồng nhiễu tốt: Tín hiệu EEG thường bị ảnh hưởng bởi nhiễu từ các nguồn bên ngoài hoặc thiết bị. SVM có khả năng xử lý tốt dữ liệu không hoàn hảo nhờ tối ưu hóa biên cách ly.



○ Ưu điểm:

- Tốt với dữ liệu có số chiều lớn và ít nhiễu.
- Mô hình đơn giản, dễ sử dụng, và yêu cầu ít dữ liệu hơn so với các phương pháp học sâu.
- Đạt hiệu quả cao trong việc tìm ranh giới phân chia tối ưu giữa các lớp.

○ Nhược điểm:

- Yêu cầu chi phí tính toán lớn khi số lượng mẫu hoặc chiều dữ liệu rất lớn.
- Việc lựa chọn kernel phù hợp và tinh chỉnh tham số có thể phức tạp.
- Không dễ mở rộng cho bài toán đa lớp mà cần các chiến lược như one-vs-one hoặc one-vs-rest.
- Do dữ liệu quá lớn nên nếu như train model SVM thông thường thì sẽ rất lâu nên sử dụng framework pytorch để train model

Kết quả train model

```

--- Độ chính xác ---
Accuracy: 0.8578

--- Ma trận nhầm lẫn ---
[[1700  145   96]
 [ 147 1652  172]
 [ 107  167 1678]]

--- Báo cáo chi tiết ---

```

	precision	recall	f1-score	support
0	0.87	0.88	0.87	1941
1	0.84	0.84	0.84	1971
2	0.86	0.86	0.86	1952
accuracy			0.86	5864
macro avg	0.86	0.86	0.86	5864
weighted avg	0.86	0.86	0.86	5864

Độ chính xác tổng thể

- Accuracy: 85.78%: Đây là kết quả tương đối tốt, đặc biệt trong bối cảnh dữ liệu EEG thường phức tạp và có nhiễu.

Phân tích ma trận nhầm lẫn

- Lớp Drowsy:
 - Dự đoán đúng: $1700/1941 = 87.5\%$.
 - Sai lệch chủ yếu: Dự đoán nhầm sang lớp Unfocused (145 mẫu).
- Lớp Unfocused:
 - Dự đoán đúng: $1652/1971 = 83.8\%$.
 - Sai lệch chính: Dự đoán nhầm sang lớp Focused (172 mẫu).
- Lớp Focused:
 - Dự đoán đúng: $1678/1952 = 86\%$.
 - Sai lệch chủ yếu: Dự đoán nhầm sang lớp Unfocused (167 mẫu).
- Lớp Drowsy có hiệu suất tốt nhất với tỷ lệ dự đoán đúng cao hơn một chút so với các lớp khác.

- Sự nhầm lẫn giữa các lớp liên kề (Drowsy-Unfocused, Unfocused-Focused) cho thấy mô hình gặp khó khăn trong việc phân tách các lớp có tín hiệu đặc điểm tương đồng. Báo cáo chi tiết (Precision, Recall, F1-score)
- Lớp Drowsy:
 - Precision = 87%: Trong số các dự đoán là lớp Drowsy, 87% là đúng.
 - Recall = 88%: Trong số thực tế là lớp Drowsy, mô hình nhận diện đúng 88%.
 - F1-score = 87%: Trung bình hài hòa giữa Precision và Recall, thể hiện mô hình cân bằng tốt.
- Lớp Unfocused:
 - Precision = 84%, Recall = 84%, F1-score = 84%: Hiệu suất thấp hơn một chút so với lớp Drowsy.
- Lớp Focused:
 - Precision = 86%, Recall = 86%, F1-score = 86%: Cân bằng và tương đối tốt.
- Macro Avg (Trung bình trên các lớp): Precision, Recall, và F1-score đều đạt 86%, cho thấy mô hình duy trì hiệu suất đồng đều trên tất cả các lớp.
- Weighted Avg (Trung bình có trọng số theo số lượng mẫu): Cũng đạt 86%, khẳng định không có lớp nào bị lệch trọng số quá mức.

- Quá trình hyper parameters tuning cho ra các tham số: 'dropout': 0.3, 'epochs': 50, 'hidden_dim': 256, 'lr': 0.001, 'weight_decay': 0.0001
 - Tỷ lệ học (Learning Rate - lr): Điều chỉnh bước nhảy khi cập nhật trọng số trong quá trình huấn luyện.
 - Số lượng nơ-ron ở lớp ẩn (hidden_dim): Điều chỉnh kích thước của lớp ẩn để phù hợp với độ phức tạp của dữ liệu.
 - Số epoch (epochs): Xác định số vòng lặp qua toàn bộ dữ liệu huấn luyện.
 - Trọng số phạt cho regularization (weight_decay): Kiểm soát mức độ phạt cho các trọng số lớn để tránh overfitting.
 - Tỷ lệ dropout (dropout): Giảm overfitting bằng cách ngẫu nhiên bỏ qua một số nơ-ron trong quá trình huấn luyện.

--- Độ chính xác ---
Accuracy: 0.8678

--- Ma trận nhầm lẫn ---
[[1763 110 68]
[151 1670 150]
[117 179 1656]]

--- Báo cáo chi tiết ---

	precision	recall	f1-score	support
Drowsed	0.87	0.91	0.89	1941
Unfocussed	0.85	0.85	0.85	1971
Focussed	0.88	0.85	0.87	1952
accuracy			0.87	5864
macro avg	0.87	0.87	0.87	5864
weighted avg	0.87	0.87	0.87	5864

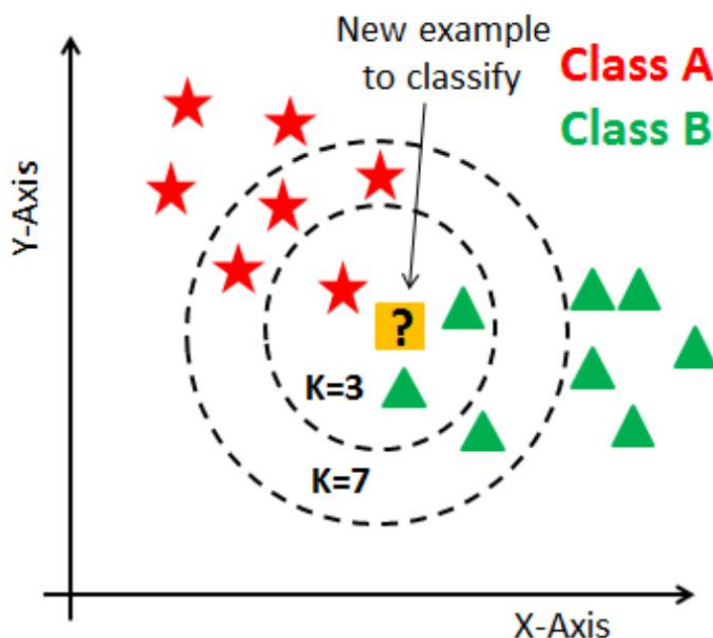
Hiệu suất tổng thể:

- Accuracy tăng nhẹ: Từ 85.78% (model cũ) lên 86.78% (model mới), cho thấy hiệu suất đã được cải thiện.
- Loss thấp hơn: Model mới đạt mức loss cuối cùng là 0.1017, trong khi model cũ có mức Test Loss chững lại quanh 0.4. Điều này cho thấy mô hình mới đã tối ưu hóa tốt hơn.

Ma trận nhầm lẫn: So sánh sự phân loại nhầm lẫn giữa hai mô hình:

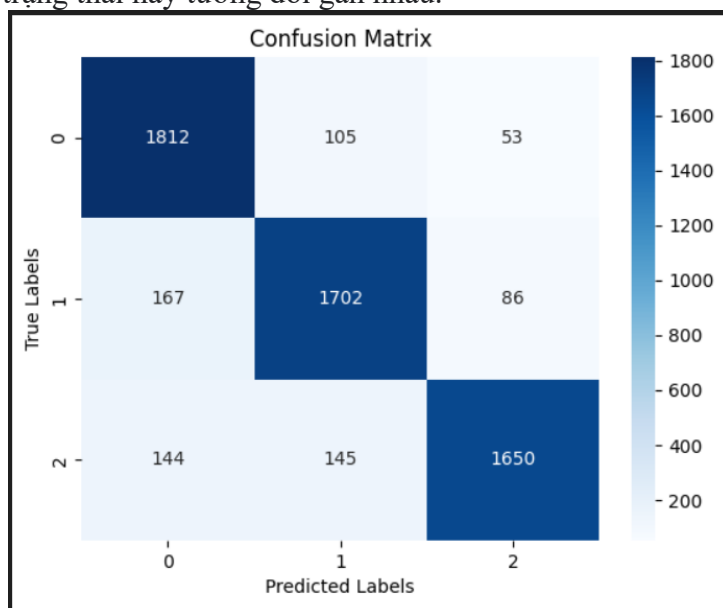
- Lớp "Drowsed" (Lớp 0):
 - Số dự đoán đúng tăng từ 1700 (model cũ) lên 1763 (model mới).
 - Sai số với lớp khác giảm rõ rệt (nhầm lẫn với lớp 1 giảm từ 145 xuống 110, nhầm lẫn với lớp 2 giảm từ 96 xuống 68).
- Lớp "Unfocussed" (Lớp 1):
 - Dự đoán đúng tăng từ 1652 lên 1670.
 - Sai số với lớp 0 giảm nhẹ (từ 147 xuống 151), nhưng sai số với lớp 2 giảm từ 172 xuống 150.
- Lớp "Focussed" (Lớp 2):
 - Số dự đoán đúng giảm từ 1678 xuống 1656, nhưng tổng thể mô hình vẫn ổn định. Nhận xét ma trận nhầm lẫn:
- Model mới giảm nhầm lẫn giữa các lớp 0-1 và 0-2, cho thấy khả năng phân biệt giữa các trạng thái đã cải thiện.
- Tuy nhiên, nhầm lẫn giữa lớp 1-2 vẫn còn tương đối cao, cần chú ý hơn trong việc xử lý các tín hiệu tương đồng.

3. KNN (K-Nearest Neighbors)



- **K-Nearest Neighbors (KNN)** là một thuật toán học máy thuộc nhóm **học có giám sát**, được sử dụng phổ biến cho các bài toán phân loại và hồi quy. KNN hoạt động dựa trên nguyên lý rằng các điểm dữ liệu có nhãn giống nhau thường nằm gần nhau trong không gian đặc trưng.
- **Accuracy** đạt ~88% trên tập kiểm tra.

- Báo cáo phân loại cho thấy mô hình hoạt động tốt nhất trên nhãn 0 (buồn ngủ) và kém hơn trên nhãn 2 (tập trung).
- Một số trường hợp nhãn 1 (mất tập trung) bị nhầm lẫn với nhãn 2 (tập trung), điều này có thể do tín hiệu của hai trạng thái này tương đối gần nhau.



Từ ma trận, ta có thể thấy rằng mô hình đã có hiệu suất phân loại tương đối tốt:

- Mô hình đã chính xác phân loại 1812 mẫu dữ liệu với nhãn 0.
- Mô hình đã chính xác phân loại 1702 mẫu với nhãn 1.
- Mô hình đã chính xác phân loại 1650 mẫu với nhãn 2.

Training completed with default KNN parameters.

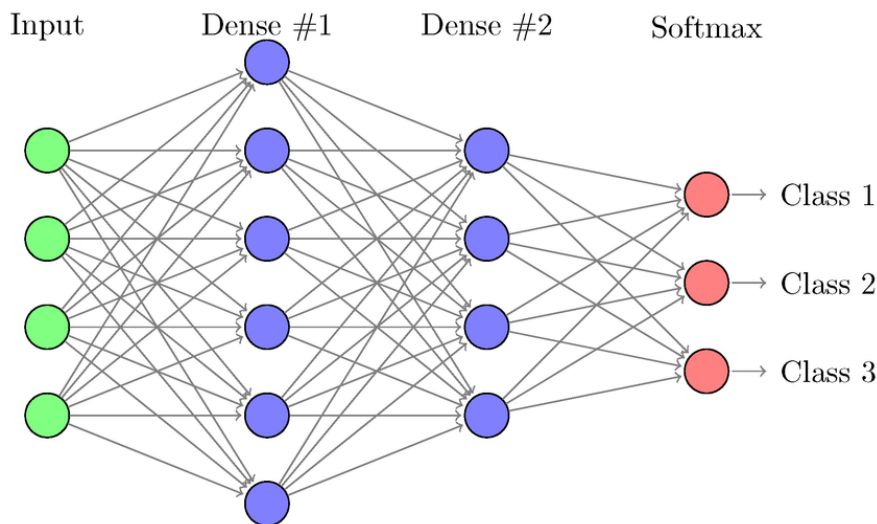
Test Accuracy: 0.8806275579809004

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.92	0.89	1970
1	0.87	0.87	0.87	1955
2	0.92	0.85	0.89	1939
accuracy			0.88	5864
macro avg	0.88	0.88	0.88	5864
weighted avg	0.88	0.88	0.88	5864

- Nhìn chung, mô hình KNN đã đạt được hiệu suất phân loại rất tốt, với độ chính xác cao cho cả ba lớp. Các chỉ số hiệu suất như độ chính xác, độ thu hồi và F1-score đều ở mức cao, trên 0.85.
- Điều này cho thấy mô hình đã học được các đặc trưng phân biệt rõ ràng giữa các lớp dữ liệu và có thể đưa ra dự đoán chính xác. Việc đạt được độ chính xác kiểm tra trên 88% cũng là một kết quả đáng ghi nhận.

4. Convolutional Neural Networks (CNN):

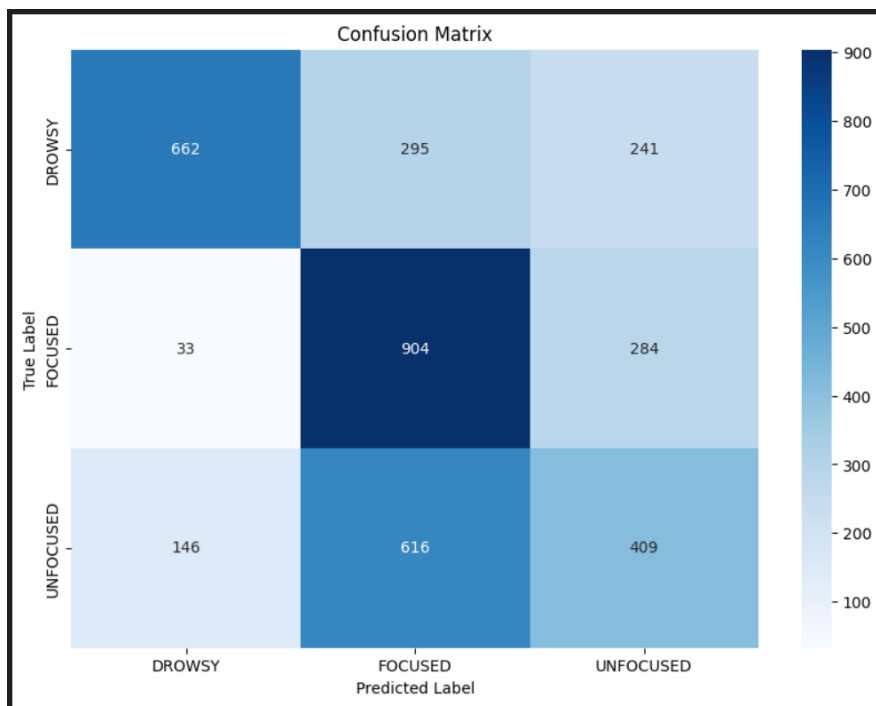


- CNN là một Mạng Nơ-ron Tích chập là một loại mạng nơ-ron nhân tạo sâu, chủ yếu được sử dụng để phân tích hình ảnh. CNN rất thích hợp cho các tác vụ như phân loại ảnh, phát hiện đối tượng và phân đoạn ảnh. Đặc điểm nổi bật của CNN là việc sử dụng các lớp tích chập, áp dụng một tập các bộ lọc có thể học được (hoặc nhân tử) lên ảnh đầu vào. Các bộ lọc này được thiết kế để phát hiện các đặc trưng cấp thấp như cạnh, hình dạng và kết cấu ở các lớp đầu tiên, sau đó dần học các đặc trưng cấp cao hơn ở các lớp sâu hơn.
- Kiến trúc và Thiết kế Mô hình CNN
- Mạng CNN rất hiệu quả cho dữ liệu hình ảnh nhờ khả năng tự động học các đặc trưng không gian. Kiến trúc của mô hình gồm:
 - **Lớp đầu vào:**
 - Nhận hình ảnh phổ kích thước (image_size, image_size, 3) (định dạng RGB):
 - **Các lớp tích chập:**
 - Lớp Conv2D:
 - Trích xuất các mẫu không gian từ phổ, như dải tần số và sự biến đổi theo thời gian.
 - Hai lớp tích chập:
 - Học các đặc trưng từ mức độ thấp đến cao với bộ lọc kích thước (3, 3), có 32 bộ lọc kích thước, trong khi lớp thứ hai có 64 bộ lọc cùng kích thước
 - Activation ReLU:
 - Tạo tính phi tuyến, giúp mô hình nắm bắt các mẫu phức tạp.
 - **Các lớp pooling:**
 - Lớp MaxPooling2D:
 - Giúp làm phẳng các bản đồ đặc trưng thành một vector biểu diễn.
 - Giảm kích thước đặc trưng để giảm khối lượng tính toán nhưng vẫn giữ lại các đặc trưng quan trọng.
 - **Làm phẳng (Flatten):**

- Chuyển đổi ma trận đặc trưng 2D thành vector 1D để đưa vào các lớp kết nối đầy đủ.
- **Các lớp Dense:**
 - Hai lớp Dense:
 - Với hàm kích hoạt ReLU, học các mối quan hệ phức tạp giữa các đặc trưng đã trích xuất.
- **Các lớp Dropout:**
 - với (tỷ lệ dropout 0,5) giúp giảm hiện tượng overfitting bằng cách loại ngẫu nhiên một số nơ-ron trong quá trình huấn luyện.
- **Lớp đầu ra:**
 - Lớp Dense:
 - Với activation Softmax, đưa ra xác suất của ba lớp:
 - Tập trung
 - Không tập trung
 - Buồn ngủ

- Mô Hình đã được áp dụng các phương pháp hyperparameter tuning quen thuộc như thay đổi tham số learning rate (giảm trong trường hợp này vì so với kết quả ban đầu dường như model đang overfitting và test loss không hội tụ), tăng thêm số lượng các lớp tích chập, thêm lớp dropout để góp phần giảm việc overfitting cho mô hình, thay đổi batch-size tùy thuộc vào số lượng dữ liệu đầu vào, thay đổi epoch

CNN: đạt ~55% trên tập kiểm tra với 50 epochs, tại 7 kênh rời rạc và số record là 10 với ảnh được resize về kích cỡ 128x128, kết quả trên là kết quả cuối cùng sau khi thực hiện các bước hiệu chỉnh hyperparameters.



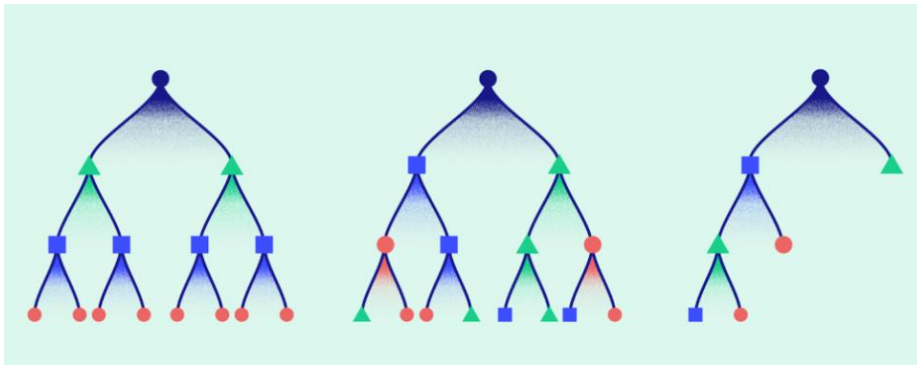
- Mô hình đã nhận diện chính xác phần lớn các nhãn FOCUSED (904/1200), nhưng hiệu suất cho các nhãn DROWSY và UNFOCUSED có vẻ kém hơn, đặc biệt là với UNFOCUSED, khi nhiều dự đoán bị nhầm thành FOCUSED.

- Ma trận này cho thấy sự phân phối không đồng đều của mô hình giữa các lớp, với tỷ lệ nhầm lẫn cao nhất xảy ra giữa FOCUSED và UNFOCUSED.

1	1.1307	0.3443	1.0861	0.3819
2	1.0880	0.3681	1.0792	0.4708
3	1.0776	0.3763	1.0765	0.5125
4	1.0557	0.4214	1.0015	0.5253
5	1.0244	0.4361	0.9416	0.5295
6	0.9851	0.4771	0.9461	0.5387
7	0.9675	0.4859	0.9180	0.5435
8	0.9587	0.4889	0.9278	0.5446
9	0.9455	0.5022	0.9139	0.5454
10	0.9352	0.5041	0.9198	0.5379
11	0.9273	0.5109	0.9053	0.5479
12	0.9125	0.5198	0.9042	0.5479
13	0.9079	0.5259	0.9012	0.5501
14	0.8991	0.5296	0.9038	0.5448
15	0.8871	0.5335	0.9071	0.5448
16	0.8777	0.5405	0.9226	0.5373
17	0.8674	0.5468	0.9132	0.5404
18	0.8614	0.5478	0.9168	0.5418
19	0.8491	0.5592	0.9351	0.5148
20	0.8346	0.5615	0.9337	0.5192
21	0.8320	0.5658	0.9573	0.5373
22	0.8185	0.5772	0.9395	0.5279
23	0.8125	0.5774	0.9567	0.5189
Test Loss: 0.9011748433113098				
Test Accuracy: 0.5501392483711243				

Mô hình đã đạt được kết quả tốt nhất là Test Accuracy ~ 0.5501 tại epoch thứ 13 và đã tự ngừng việc train tại epoch thứ 23 do hàm early stopping đã được gọi để ngừng train khi hàm lost không trở nên tốt hơn sau 10 epochs

5. XGboost.



XGBoost là một thuật toán học máy mạnh mẽ, thường được sử dụng trong các bài toán dự đoán. Ý tưởng chính là nó kết hợp nhiều "cây quyết định nhỏ" (weak learners) để tạo ra một mô hình dự đoán tốt hơn. Hãy tưởng tượng việc giải một bài toán khó, bạn nhờ nhiều bạn giúp đỡ, mỗi bạn chỉ giỏi một phần, nhưng kết hợp lại, bạn có thể tìm ra đáp án hoàn chỉnh.

- Dữ liệu huấn luyện

Đầu tiên, ta chuẩn bị một bộ dữ liệu, trong đó:

- Có nhiều yếu tố (cột) ảnh hưởng đến kết quả.
- Một cột chính là kết quả mà ta muốn dự đoán (gọi là "biến mục tiêu").

Có một "hàm mất mát" để đo lường xem dự đoán của mô hình sai lệch bao nhiêu so với kết quả thật. Ví dụ: Trong bài toán dự đoán điểm thi, các yếu tố có thể là thời gian học, điểm kiểm tra giữa kỳ, v.v., và kết quả là điểm cuối kỳ.

- Học dần

Mô hình bắt đầu với một giá trị dự đoán đơn giản, ví dụ, điểm trung bình của tất cả học sinh. Từ đó, thuật toán tìm cách cải thiện dự đoán bằng cách thêm dần những "cây quyết định nhỏ".

- Thêm cây mới

Dựa trên gradient và hessian, thuật toán tạo thêm một cây mới để sửa lỗi từ cây trước. Quá trình này lặp lại nhiều lần, và mỗi cây giúp mô hình dự đoán ngày càng chính xác.

Hàm mất mát có thể được biểu diễn bằng:

$$L(y, \hat{y}) = \sum_{i=1}^n l(y_i, \hat{y}_i)$$

Trong đó:

- (y) là giá trị thực tế.
- (y_{μ}) là giá trị dự đoán.
- (l) là hàm mất mát.

- Tốc độ học (Learning rate)

- Có một "nút điều chỉnh tốc độ" gọi là learning rate, giúp mô hình học từ từ. Nếu học quá nhanh, mô hình dễ bị "quá đà" (overfitting). Nếu học quá chậm, mô hình sẽ mất nhiều thời gian.

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta f_t(x)$$

Trong đó:

- $(\{y_{\mu}\}^{(t)})$ là giá trị dự đoán tại bước (t) .
- (f_t) là cây quyết định mới được thêm vào.

- Kết quả cuối cùng

- Sau khi thêm tất cả các cây, mô hình sẽ gộp kết quả từ từng cây lại để đưa ra dự đoán cuối cùng. Điều này giống như tổng hợp ý kiến của tất cả các bạn để đưa ra đáp án.
- Kết quả cuối cùng có thể được biểu diễn như sau:

$$\hat{y} = \sum_{m=1}^M f_m(x)$$

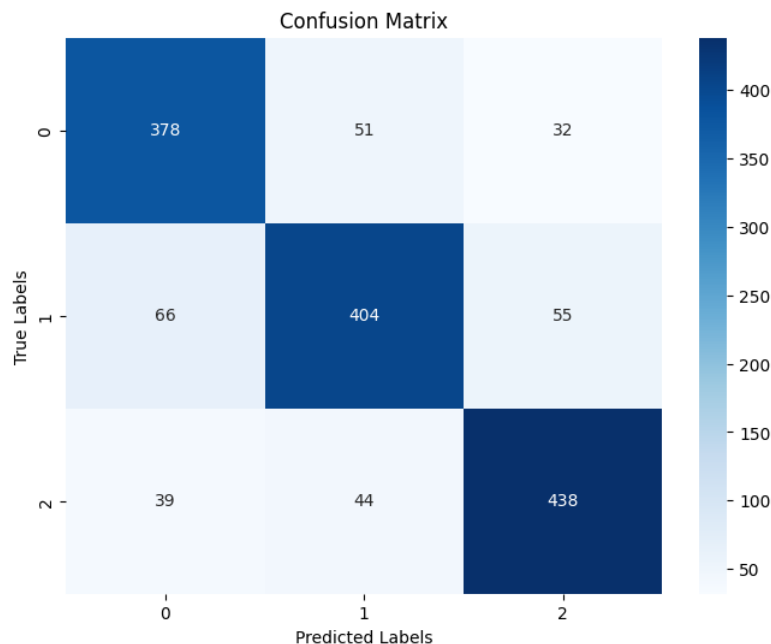
Trong đó:

- $\{\hat{y}_m\}$ là giá trị dự đoán cuối cùng.
- f_m là cây quyết định thứ m .
- M là tổng số cây.

```
Test Accuracy: 0.8096
Classification Report:

```

	precision	recall	f1-score	support
0	0.78	0.82	0.80	461
1	0.81	0.77	0.79	525
2	0.83	0.84	0.84	521
accuracy			0.81	1507
macro avg	0.81	0.81	0.81	1507
weighted avg	0.81	0.81	0.81	1507



Chính xác Kiểm tra: Độ chính xác của mô hình là 0,8096, một kết quả tương đối tốt.

Báo cáo Phân loại:

- Độ chính xác (precision): Cho thấy mức độ chính xác của các dự đoán, dao động từ 0,78 đến 0,83 cho các lớp.
- Thu hồi (recall): Cho thấy khả năng thu lại các mẫu đúng, cũng dao động trong khoảng 0,77 đến 0,84.
- F1-score: Chỉ số kết hợp giữa độ chính xác và thu hồi, dao động từ 0,79 đến 0,84.
- Hỗ trợ (support): Cho biết số lượng mẫu trong mỗi lớp, từ 461 đến 525 mẫu.

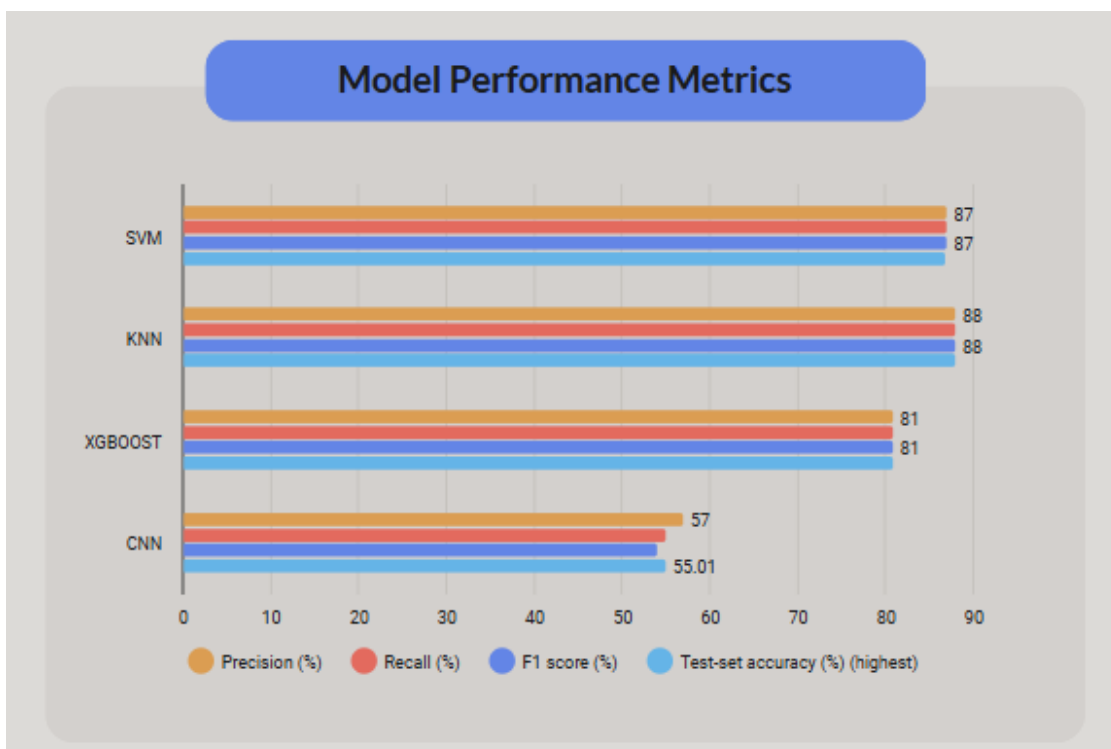
Ma trận Nhầm lẫn:

- Hình ảnh minh họa rõ ràng các sai lầm phân loại giữa các lớp.
- Có 378 mẫu được phân loại chính xác vào lớp 0, 51 mẫu nhầm vào lớp 1, và 32 mẫu nhầm vào lớp 2.
- Ngược lại, có 66 mẫu lớp 1 được phân loại sai vào lớp 0, 404 mẫu được phân loại đúng, và 44 mẫu nhầm vào lớp 2.
- Đối với lớp 2, có 39 mẫu được phân loại sai vào lớp 0, 55 mẫu được phân loại đúng, và 438 mẫu nhầm vào lớp 1.

Nhận xét tổng quát:

- Mô hình có độ chính xác tốt, đạt 80,96%, cho thấy khả năng phân loại các trạng thái tâm lý (tập trung, không tập trung, buồn ngủ) khá hiệu quả.
- Các chỉ số như độ chính xác, thu hồi và F1-score cũng ở mức tương đối cao, trên 0,75, cho thấy mô hình hoạt động ổn định.
- Ma trận nhầm lẫn cung cấp thông tin chi tiết về các sai lầm phân loại giữa các lớp, giúp định hướng cải thiện mô hình trong tương lai.
- Nhìn chung, kết quả này cho thấy mô hình có hiệu suất tốt trong việc phân loại các trạng thái tâm lý dựa trên dữ liệu EEG.

V. So sánh mô hình



So sánh hiệu suất các mô hình:

Dựa trên các chỉ số hiệu suất được cung cấp, có thể thấy rằng các mô hình KNN và SVM đang thể hiện những kết quả tốt nhất. Đây là những mô hình cân bằng và toàn diện, với độ chính xác và độ nhạy đáng kể.

Cụ thể hơn, các mô hình KNN và SVM có độ chính xác dự đoán (precision) ấn tượng, đạt tới 88%. Điều này cho thấy chúng có khả năng hạn chế các dự đoán dương tính giả một cách hiệu quả. Về độ nhạy (recall), KNN và SVM cũng thể hiện mức độ tương đối tốt, gần 80%.

Khi cân bằng cả độ chính xác và độ nhạy, các mô hình này đạt được điểm F1 cao nhất, ở mức 0,81-0,84. Chỉ số này phản ánh sự cân bằng tối ưu giữa hai tiêu chí này. Điều này đồng thời được khẳng định thông qua độ chính xác trên tập kiểm tra, lên tới 88% - mức cao nhất trong số các mô hình so sánh.

Mô hình XGBoost cũng thể hiện hiệu suất ấn tượng, đặc biệt về độ nhạy ở mức 84%. Tuy nhiên, độ chính xác của nó hơi thấp hơn so với KNN và SVM. Điểm F1 và độ chính xác trên tập kiểm tra cũng đạt mức tốt, nhưng chưa bằng hai mô hình hàng đầu.

Mô hình CNN lại tỏ ra kém hơn so với các đối thủ khác trên hầu hết các chỉ số. Độ chính xác thấp, chỉ 55%, cho thấy nó có khả năng cao trong việc dự đoán sai lệch. Các chỉ số khác như độ nhạy, điểm F1 và độ chính xác trên tập kiểm tra cũng đều thấp nhất trong số các mô hình.

Nhìn chung, KNN và SVM thực sự nổi bật với hiệu suất toàn diện và cân bằng. XGBoost cũng là một mô hình đáng chú ý, đặc biệt về độ nhạy. Trong khi đó, CNN dường như cần được cải thiện nhiều hơn để có thể cạnh tranh với các mô hình hàng đầu khác.

VI. Tài liệu tham khảo

1. [Distinguishing mental attention states of humans via an EEG based passive BCI using Machine Learning Methods](#)
2. [Guide to Electroencephalography: From EEG Data to Classification](#)
3. [CNN documentation](#)
4. [KNN documentation](#)
5. [Support Vector Machine \(SVM\) Algorithm](#)
6. [MNE documentation](#)
7. [ML | XGBoost \(eXtreme Gradient Boosting\)](#)

