Midterm Project 1: Mental Attention States Classification Using EEG Data

NHÓM 21

December 2024

Group Members
Phạm Nguyễn Hoàng - 22280032
Bùi Phát Tài - 22280079
Phạm Minh Thái - 22280082
Mai Thị Hồng Trinh - 22280097

I. TỔNG QUAN PROJECT VÀ THÀNH VIÊN NHÓM

1. Đánh giá thành viên nhóm

MSSV	Họ & Tên	Nhiệm vụ & Hoàn thành (%)
22280032	Phạm Nguyễn Hoàng	100%
22280079	Bùi Phát Tài	100%
22280082	Phạm Minh Thái	100%
22280097	Mai Thị Hồng Trinh	100%

2. Tổng quan về project thực hiện

2.1. Mục Tiêu

Xây dựng và đánh giá các model học máy để phân loại trạng thái tâm lý của con người thông qua tín hiệu EEG (Electroencephalogram). Các trạng thái này bao gồm: *focused* (tập trung), *unfocused* (không tập trung), và *drowsy* (buồn ngủ). Dự án áp dụng các model phân loại khác nhau, bao gồm SVM (Support Vector Machine), XGBoost và Random Forest, để phân tích và dự đoán trạng thái tâm lý từ các đặc trưng EEG.

2.2. Dataset

Dữ liệu được sử dụng trong dự án này là bộ dữ liệu EEG của nhiều đối tượng (subjects), với mỗi đối tượng được ghi nhận trong các trạng thái tâm lý khác nhau: tập trung, không tập trung và buồn ngủ. Mỗi bản ghi trong bộ dữ liệu bao gồm các đặc trưng EEG cùng với nhãn lớp tương ứng mô tả trạng thái tâm lý của đối tượng tại thời điểm ghi nhận.

Mô Tả Bộ Dữ Liệu:

- **Số lượng thí nghiệm**: Dữ liệu bao gồm 34 thí nghiệm với 5 đối tượng tham gia.
- **Số lượng kênh EEG**: Dữ liệu EEG được thu thập từ các kênh số 4 đến số 17 của thiết bị EEG EMOTIV.
- **Tần suất mẫu**: Dữ liệu được ghi nhận với tần suất 128 Hz, nghĩa là mỗi giây có 128 mẫu được thu thập.
- **Dữ liệu vào**: Mỗi bản ghi dữ liệu chứa các tín hiệu EEG từ các kênh trên đầu người tham gia, trong đó mỗi kênh đại diện cho một vị trí điện cực cụ thể trên đầu.

II. DATA PROCESSING & FEATURE ENGINEERING

Quy trình preprocessing dữ liệu EEG thô, trích xuất các feature có liên quan bằng cách sử dụng Short-Time Fourier Transform (STFT) và gán nhãn dựa trên trạng thái chiếm ưu thế trong mỗi cửa sổ thời gian. Mục tiêu chính là chuyển đổi dữ liệu EEG thành dạng phù hợp cho các model tiến hành học và dự đoán.

1. Một số bước chuẩn bị

- **Trích xuất nhãn (Label)**: Cột `"Label"` được tách ra từ dữ liệu EEG và chuyển thành mảng numpy để dễ dàng xử lý sau này.
- **Chuyển đổi tín hiệu thành mảng numpy**
- **Window và step size**:
  - Cửa sổ Blackman được sử dụng để giảm nhiễu và lệch biên khi tính toán STFT.
  - Bước di chuyển giữa các Time Window được tính bằng cách chuyển đổi bước di chuyển (step size) từ giây thành mẫu.

- **Cắt dữ liệu**: Dữ liệu được cắt chỉ giữ lại 30 phút đầu của tín hiệu, tương đương với số mẫu `max_samples = 30 * 60 * sampling_freq`.

2. Feature Engineering

- **Tính toán Short-Time Fourier Transform (STFT)**:
  - STFT được sử dụng để chuyển tín hiệu EEG từ miền thời gian sang miền tần số. Mỗi kênh EEG sẽ được xử lý riêng biệt.
  - Một số tham số:
    - `fs=sampling_freq`: Tần số mẫu (128 Hz). Quyết định mức độ chi tiết của tín hiệu thu thập.
    - `window=window`: Cửa sổ Blackman, giúp giảm nhiễu và lệch biên.
    - `nperseg=1920`: window size là 1920 mẫu (15 giây). Cửa sổ dài giúp tăng độ phân giải tần số.
    - `noverlap=1920 - step_size_samples`: Mức độ overlapping (bước chồng lấn)
  - Kết quả là ma trận phổ tần số trong miền tần số (magnitude) cho mỗi kênh EEG.
- **Power Spectrum Calculation**: Phổ công suất được tính toán từ độ lớn của đầu ra STFT. Phổ này biểu thị cường độ của các thành phần tần số khác nhau trong tín hiệu. Công suất tỷ lệ thuận với bình phương độ lớn của các hệ số Fourier.
- **Frequency Binning**: Sau khi tính toán STFT, phổ tần số sẽ được phân chia thành các ô tần số (frequency bins), mỗi ô có độ rộng 0.5 Hz. Mỗi ô tần số sẽ chứa giá trị trung bình của phổ trong khoảng tần số đó.
- **Chuyển đổi về Decibel**: Phổ tần số được chuyển đổi từ đơn vị cường độ sang decibel (dB) để dễ dàng so sánh và phân tích. Công thức chuyển đổi: $\text{dB} = 10 \times \log_{10}(\text{power} + 1e^{-10})$
- **Combining Features**: Extracted features của tất cả các kênh EEG sẽ được kết hợp thành single feature vector chung cho mỗi time window. Điều này giúp dữ liệu có thể được sử dụng cho các mô hình phân loại.
- **Label Assignment (Dominant Label in Time Window)** Dựa trên các label của các mẫu trong time window, label chính được xác định là label xuất hiện nhiều nhất.

3. Kết Quả

Một `DataFrame` mới được tạo ra với các feature cho time window và label tương ứng. Các feature này sẽ được sử dụng cho các mô hình phân loại.

III. MODEL THEORY

1. Support Vector Machine (SVM)

SVM là thuật toán phân loại học máy giám sát, tìm kiếm một siêu phẳng (hyperplane) tối ưu để phân tách các lớp dữ liệu sao cho khoảng cách giữa các điểm gần nhất (support vectors) và siêu phẳng là lớn nhất.

Giải quyết bài toán phân loại: Dành cho dữ liệu tuyến tính, bài toán tối ưu của SVM là:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{với ràng buộc} \quad y_i(w^T x_i + b) \geq 1, \quad \forall i$$

Khi dữ liệu không tuyến tính, sử dụng kỹ thuật kernel để ánh xạ dữ liệu vào không gian cao hơn.

Phân loại đa lớp: SVM dùng phương pháp "one-vs-rest" (OvR) để xử lý phân loại đa lớp.

2. XGBoost (Extreme Gradient Boosting)

XGBoost là model ensemble sử dụng kỹ thuật gradient boosting. Nó xây dựng các cây quyết định tuần tự, mỗi cây cố gắng sửa lỗi của cây trước đó.
Cập nhật dự đoán: Dự đoán của model được cập nhật qua mỗi cây như sau:

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

Trong đó $f_t(x)$ là cây quyết định thứ t, và η là tốc độ học.
Phân loại đa lớp: XGBoost sử dụng hàm `softmax` để tính xác suất của các lớp.

3. Random Forest

Random Forest là model học máy ensemble sử dụng nhiều cây quyết định. Mỗi cây được huấn luyện trên một mẫu bootstrap ngẫu nhiên, và mỗi lần phân chia chỉ xem xét một tập con ngẫu nhiên các đặc trưng.
Dự đoán của model: Dự đoán của Random Forest được xác định bằng cách lấy lớp có số phiếu bầu cao nhất từ tất cả các cây:

$$\hat{y} = \text{mode}(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T)$$

Phân loại đa lớp: Random Forest chọn lớp có tỷ lệ phiếu bầu cao nhất trong số các cây.

IV. MODEL BUILDING

1. Một Số Thao Tác Xử Lý Dữ Liệu

Dữ liệu EEG đã được xử lý trước khi đưa vào huấn luyện model:

- **Trích xuất đặc trưng:** Các đặc trưng từ tín hiệu EEG được trích xuất để làm đầu vào cho các mô hình học máy. Các nhãn (labels) được mã hóa dưới dạng số: ví dụ, 0 cho trạng thái *focused*, 1 cho trạng thái *unfocused*, và 2 cho trạng thái *drowsy*.
- **Chuẩn hóa dữ liệu:** Các đặc trưng được chuẩn hóa bằng phương pháp MinMax Scaling để đảm bảo tất cả các đặc trưng có phạm vi giá trị giống nhau.
- **Chia dữ liệu:** Dữ liệu được chia thành hai phần: 80% dữ liệu dùng để huấn luyện và 20% còn lại dùng để kiểm tra model.

2. Huấn Luyện Model

Support Vector Machine (SVM)

SVM được xây dựng với nhân RBF (Radial Basis Function), một trong các kernel phổ biến giúp model có thể học các đặc trưng phi tuyến tính. Các tham số của model được thiết lập như sau:

- C (Tham số điều chỉnh độ phạt): Tham số này điều chỉnh mức độ phạt khi có lỗi trong phân lớp, được thiết lập là $C = 1$.
- Kernel: Kernel "rbf" (Radial Basis Function) được sử dụng để xử lý dữ liệu không phân lớp tuyến tính.

XGBoost (Extreme Gradient Boosting)

XGBoost là model học máy boosting sử dụng cây quyết định. Các tham số của model:

- Objective: Được chỉ định là `multi:softmax` cho phân loại đa lớp.
- Learning Rate: Tự động điều chỉnh trong quá trình huấn luyện.
- Số lớp (Num Class): Số lượng lớp được xác định tự động theo số lượng lớp trong dữ liệu.

Random Forest

Random Forest sử dụng tập hợp cây quyết định, mỗi cây được huấn luyện trên một mẫu ngẫu nhiên từ dữ liệu huấn luyện (bootstrap sampling). Các bước chính như sau:

- n_estimators: Số lượng cây quyết định trong rừng, được thiết lập là 100 cây.
- max_depth: Mức độ sâu tối đa của cây quyết định, không giới hạn để cho phép model phát hiện các mẫu phức tạp.
- class_weight: Được sử dụng để xử lý sự mất cân bằng lớp trong dữ liệu bằng cách đặt `class_weight='balanced'` để tự động điều chỉnh trọng số các lớp.

3. Đánh Giá Model

Các model được đánh giá bằng accuracy, ROC AUC và các chỉ số precision, recall và F1 score (Classification Report). Các model được đánh giá dựa trên hai chỉ số chính:

- **Độ chính xác (Accuracy):** Tỷ lệ các dự đoán chính xác so với tổng số dự đoán.
- **ROC AUC Score:** Diện tích dưới đường cong ROC, đánh giá khả năng phân biệt các lớp của model.

4. Tổng Quan Model

- **SVM:** Hoạt động tốt với dữ liệu nhỏ và ít chiều, nhưng có thể gặp khó khăn với dữ liệu lớn và phức tạp nếu không tối ưu hóa đúng cách.
- **XGBoost:** model mạnh mẽ và thường cho kết quả tốt nhất nhờ khả năng học các mẫu phức tạp và xử lý dữ liệu lớn.
- **Random Forest:** Cung cấp hiệu suất ổn định, đặc biệt khi dữ liệu có sự mất cân bằng giữa các lớp.

V. PERFORMANCE EVALUATION

1. Thực hiên train-test với từng Subject

- **Hiệu suất chung của các mô hình**: Nhìn chung, các mô hình đều đạt các chỉ số metric (*accuracy*, *AUC*, *precision*, *recall*) lớn hơn 0.9 đối với tất cả 5 subject, cho thấy khả năng phân loại chính xác và ổn định trên tập dữ liệu EEG. Các chỉ số này minh chứng cho khả năng xử lý tốt các mẫu dữ liệu phức tạp của các mô hình trong bài toán phân loại EEG. Trong báo cáo sẽ nhận xét chính về accuracy và AUC.
- **Mô hình XGBoost**:
  - **XGBoost** luôn đạt các chỉ số metric cao nhất trong ba mô hình, gồm *accuracy* và *AUC*. Mô hình này có khả năng xử lý dữ liệu phức tạp và đa dạng rất tốt nhờ vào việc sử dụng các kỹ thuật *boosting* để giảm thiểu *bias* và *variance*, đồng thời tự động điều chỉnh trọng số các mẫu mất cân bằng. Do đó mô hình này có thể đạt được độ chính xác ổn định và cao hơn so với các mô hình khác.
- **Mô hình SVM**:
  - **SVM** có hiệu suất thấp nhất trong ba mô hình, với độ chính xác thấp nhất đạt 0.92 ở subject 1. Điều này có thể giải thích bởi *SVM* hoạt động hiệu quả khi làm việc với dữ liệu nhỏ và ít chiều, nhưng gặp khó khăn khi xử lý dữ liệu phức tạp và nhiều chiều như dữ liệu EEG.
  - Khi sử dụng kernel RBF (Radial Basis Function), SVM gặp khó khăn trong việc học các *decision boundaries* chính xác khi không có đủ dữ liệu huấn luyện. Điều này dẫn đến việc mô hình không tận dụng được hết thông tin từ các đặc trưng của dữ liệu.
- **Mô hình Random Forest**:
  - **Random Forest** cho thấy một hiệu suất ổn định, với *accuracy* và *AUC* khá ổn định giữa các subject. Mô hình này sử dụng nhiều cây quyết định và các kỹ thuật

randomization để giảm thiểu overfitting, giúp duy trì mức độ chính xác nhất định khi xử lý các tình huống khác nhau.
  - Tuy nhiên, *Random Forest* không thể đạt được độ chính xác cao như *XGBoost* trong hầu hết các trường hợp, có thể do khả năng học các tương quan phức tạp và tính toán quyết định của *Random Forest* không tối ưu bằng *XGBoost*.

2. Thực hiện train-test trên toàn bộ dữ liệu

- **Mô hình SVM:**
  - **Accuracy:** 87.2%, **AUC:** 0.9659
  - Mô hình SVM đạt độ chính xác tốt nhưng thấp hơn so với XGBoost và Random Forest. AUC cao cho thấy khả năng phân biệt lớp tốt nhưng cũng tương tự như khi thực hiện trên tập các subject thì mô hình này vẫn là mô hình có các chỉ số metric thấp nhất.
- **Mô hình XGBoost:**
  - **Accuracy:** 92.5%, **AUC:** 0.9864
  - Mô hình XGBoost có hiệu suất vượt trội với độ chính xác và AUC cao. Các chỉ số precision, recall và f1-score đều rất tốt, đặc biệt là với state drowsy.
- **Mô hình RandomForest:**
  - **Accuracy:** 94.7%, **AUC:** 0.993
  - Mô hình RandomForest có hiệu suất cao với độ chính xác và AUC cao. Khác hẳn với việc thực hiện trên từng subject thì trên tập tất cả dữ liệu, mô hình RandomForest có các chỉ số metric cao nhất trong 3 model. Vì RandomForest hoạt động ổn định hơn nhờ tính chất ngẫu nhiên và khả năng tổng hợp kết quả từ nhiều cây quyết định, giúp chống lại sự đa dạng và phức tạp của dữ liệu tổng thể. Ngược lại, xử lý từng tập dữ liệu nhỏ (subject riêng lẻ), XGBoost có thể tập trung vào việc giảm lỗi một cách hiệu quả cho từng tập nhưng với tập dữ liệu lớn mô hình này rất có khả năng không tối ưu.

VI. USING NEURAL NETWORK

Mục tiêu: Xử lý dữ liệu EEG, huấn luyện, kiểm tra và đánh giá mô hình mạng nơ-ron để phân loại trạng thái chú ý (e.g., Focused, Unfocused, Drowsy) bằng cách sử dụng một quy trình học sâu có cấu trúc.

1. Preprocessing Pipeline

Mục tiêu: Chuẩn bị tín hiệu EEG thô để huấn luyện bằng cách phân đoạn (segmenting), chuẩn hóa (normalizing) và gán nhãn (labeling).

Các bước

- **Data Loading**: Trích xuất tín hiệu EEG và các kênh liên quan từ các tệp `.mat`. Giữ lại các cột cần thiết (các kênh EEG cụ thể).
- **Segmentation**: Chia tín hiệu EEG thành các overlapping windows dựa trên `window_size` và `step_size` được xác định trước. Đảm bảo trích xuất đặc trưng hiệu quả trong khi tối đa hóa số lượng mẫu huấn luyện.
- **Label Assignment**: Gán label cho mỗi đoạn tín hiệu dựa trên label phổ biến nhất hoặc label có giá trị cao nhất trong cửa sổ.
- **Normalization**: Chuẩn hóa từng đoạn tín hiệu về giá trị trung bình bằng 0 và phương sai đơn vị để đảm bảo tính nhất quán về thang đo giữa các đặc trưng.

Kết quả

Dữ liệu EEG đã được phân đoạn và chuẩn hóa sẵn sàng làm input cho mô hình. Label tương ứng cho mỗi đoạn tín hiệu.

Key Features

- Các overlapping window tăng cường sự đa dạng mẫu.
- Chuẩn hóa từng đoạn giúp giảm ảnh hưởng từ biên độ tín hiệu khác nhau.
- Labeling đảm bảo sự liên kết giữa các đoạn EEG theo trạng thái chú ý.

2. Training and Cross-Validation

Mục tiêu: Đảm bảo đánh giá mô hình toàn diện và tối ưu hóa hiệu suất thông qua cross-validation và kiểm tra trên tập validation.

Các bước

- **Cross-Validation Setup**: Thực hiện đánh giá chéo k-fold (e.g., 5-fold) để chia dữ liệu thành tập train và tập test. Tập train được chia nhỏ thành các tập train và tập validation.
- **Weighted Sampling**: Xử lý class imbalance bằng cách sử dụng `WeightedRandomSampler` để gán trọng số cao hơn cho các lớp ít xuất hiện.
- **Training**: Sử dụng tập train để cập nhật tham số mô hình thông qua backpropagation. Tối ưu hàm mất mát bằng optimizer **Adam** và **CrossEntropy** loss.
- **Validation**: Đánh giá mô hình trên tập validation sau mỗi epoch để theo dõi hiện tượng overfitting và khả năng khái quát hóa của mô hình.
- **Early Stopping**: Ngừng huấn luyện nếu hàm mất mát trên tập validation không cải thiện trong một số lượng epoch được xác định trước. Đảm bảo quá trình huấn luyện hiệu quả và tránh overfitting.

Kết quả

Mô hình được huấn luyện trên dữ liệu đã cân bằng với early stopping. Kết quả đánh giá trên tập validation giúp theo dõi hiệu suất.

Key Features

- **Cross-Validation**: Đảm bảo đánh giá toàn diện trên nhiều tập dữ liệu.
- **Weighted Sampling**: Cân bằng phân phối lớp để cải thiện khả năng khái quát.
- **Validation**: Ngăn chặn overfitting bằng cách đánh giá trên dữ liệu không thấy trong quá trình huấn luyện.
- **Early Stopping**: Tiết kiệm tài nguyên tính toán trong khi đảm bảo hiệu suất tối ưu.

3. Model Architecture: EEG 1D CNN

Input Layer

Nhận tín hiệu EEG với dạng `(batch_size, input_channels, input_timepoints)`.

Convolutional and Pooling Layers

Ba convolutional layers liên tiếp với batch normalization và max-pooling. Trích xuất các đặc trưng không gian và thời gian từ tín hiệu EEG.

Fully Connected Layers

- Một lớp dense với activation **ReLU**.
- Lớp dropout để giảm hiện tượng overfitting.
- Lớp dense cuối cùng với activation **softmax** để xuất xác suất phân lớp.

Output Layer

Xuất các xác suất cho từng lớp tương ứng với trạng thái chú ý của EEG.

Key Features

- **Convolution Layers**: Trích xuất local temporal dependencies giữa các kênh.
- **Batch Normalization**: Cải thiện độ ổn định và tốc độ huấn luyện.
- **Dropout**: Giảm hiện tượng overfitting.
- **Softmax Activation**: Chuyển đổi giá trị logits thành xác suất.

4. Performance Evaluation

- Các chỉ số metric Accuracy và AUC đều khá tốt. Subject 1 có mức accuracy thấp nhất với 0.74 và Subject 3 và Subject 5 có mức accuracy cao nhất với 0.93. AUC cũng tương tự khi Subject 1 có mức thấp nhất ở 3 class lần lượt là 0.94, 0.86, 0.89 và Subject 5 có mức cao nhất với lần lượt là 0.99, 0.98, 0.99. Điều thấy cho thấy khả năng tích cực của mô hình.
- Mức chênh lệch giá trị metric giữa các Subject lớn hơn nhiều so với việc train-test bằng các model machine learning.

VII. Discuss challenges faced and potential ways to improve accuracy

- Khi train model theo cách thông thường sử dụng toàn bộ data train thì gặp vấn đề "Mất cân bằng", làm cho độ chính xác của 2 label Focused và Unfocused không dự đoán được gần như là 0, sau đó chỉ lấy tập dữ liệu của mỗi file là 30 phút đầu của mỗi file và lấy 10 phút đầu tiền gán là focused, 10 tiếp theo unfocused và 10 phút tiếp là Drowsy. thì độ chính xác của mô hình có cải thiện và dữ liệu không còn mất cân bằng.
- Sử dụng phương pháp STFT thử nghiệm với nhiều window size, khi sử dụng với window size lớn thì cho thấy độ chính xác của dữ liệu tăng hơn. Vì khi sử dụng window size nhỏ và BlackMan sẽ khó khăn để phân tách các tần số gần nhau. và việc gán label vô từng window thì một số label bị ngán chưa đúng do việc overlaping cách xử lý là kiểm tra phần label trên window nào lớn thì gán theo label đó, và khi làm điều đó kết quả tốt hơn.
- Thử nghiệm và xử lý với CNN thì có thử thách tìm ra kiến trúc mạng phù hợp để cho kết quả độ chính xác tốt hơn
- Thử Nghiệm với việc chia tập dữ liệu với việc train cho từng người hoặc train trên toàn bộ kết quả cho thấy khi chia ra từng tập train thì kết quả không tốt hơn khi train trên tất cả.
- Thử nghiệm với việc huấn luyện trên toàn bộ channel và lọc ra các channel có độ hoạt động tốt hơn.