Санкт-Петербургский государственный политехнический университет
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

КУРСОВАЯ РАБОТА

по дисциплине «Машинное обучение»

Тема: Классификация событий по временному ряду

(на примере датасета EEG Eye State)

Выполнила студентка группы 5140904/40102 — Спирчина В.А.

Руководитель: — Селин И. А.

Санкт-Петербург
2025

Содержание

1. Введение

Электроэнцефалография (ЭЭГ) является одним из ключевых методов исследования активности головного мозга. ЭЭГ регистрирует электрическую активность нейронов через электроды, размещенные на поверхности головы. Анализ временных рядов ЭЭГ сигналов позволяет выявлять различные состояния мозговой активности и диагностировать неврологические заболевания.

Задача классификации состояний по данным ЭЭГ имеет широкое практическое применение в медицине, нейроинтерфейсах (Brain-Computer Interface, BCI) и системах безопасности. Развитие методов машинного обучения открывает новые возможности для автоматического анализа ЭЭГ данных с высокой точностью.

Цель работы: Разработать модель машинного обучения для автоматической классификации состояния глаз человека (открыты/закрыты) на основе анализа временных рядов ЭЭГ сигналов с 14 каналов.

Задачи работы:

1. Провести анализ датасета EEG Eye State
2. Исследовать существующие методы классификации временных рядов
3. Выбрать и обосновать архитектуру модели
4. Подготовить данные для обучения
5. Обучить модель и оценить её качество
6. Провести анализ результатов и примеров работы

2. Описание проблемы

Электроэнцефалография позволяет регистрировать биоэлектрическую активность мозга неинвазивным способом. Различные состояния мозговой активности, включая состояние глаз (открыты/закрыты), отражаются в характеристиках ЭЭГ сигналов. Автоматическая классификация таких состояний является важной задачей для медицинских и технических применений.

Медицина:

- Мониторинг состояния пациентов в отделениях реанимации и интенсивной терапии
- Диагностика нарушений сна и сомнологические исследования
- Выявление эпилептической активности и других неврологических патологий
- Оценка уровня сознания пациентов

Brain-Computer Interface (BCI):

- Управление протезами конечностей и экзоскелетами для людей с ограниченными возможностями
- Нейроинтерфейсы для компьютерных игр и виртуальной реальности
- Системы дополненной реальности с управлением взглядом
- Коммуникационные устройства для пациентов с синдромом запертого человека

Безопасность:

- Детектирование усталости водителей транспортных средств
- Мониторинг уровня внимания операторов критических систем
- Системы контроля состояния пилотов авиации
- Раннее предупреждение о засыпании на рабочем месте

3. Обзор датасета

3.1. Источник данных

Для решения задачи используется датасет **EEG Eye State** из репозитория UCI Machine Learning Repository.

Параметр	**Значение**
Название	EEG Eye State Dataset
Источник	UCI ML Repository
Автор	Oliver Roesler
Год публикации	2013
URL	archive.ics.uci.edu/dataset/264

Таблица 1. Информация о датасете

3.2. Характеристики датасета

Параметр	**Значение**
Количество наблюдений	14,980
Количество признаков	14 (каналы ЭЭГ)
Целевая переменная	Бинарная (0/1)
Длительность записи	117 секунд
Пропущенные значения	0 (отсутствуют)
Формат файла	ARFF

Таблица 2. Технические характеристики датасета

3.3. Описание признаков

Данные собраны с 14 электродов, расположенных на различных участках головы.

№	Канал	Расположение	Функция зоны мозга
1	AF3	Левая префронтальная	Планирование, решения
2	F7	Левая фронтальная	Речь, внимание
3	F3	Левая фронтальная	Когнитивные функции
4	FC5	Левая фронто-центральная	Моторика
5	T7	Левая височная	Слух, память
6	P7	Левая париетальная	Сенсорика
7	**O1**	**Левая затылочная**	**Зрительная обработка**
8	**O2**	**Правая затылочная**	**Зрительная обработка**
9	P8	Правая париетальная	Сенсорика
10	T8	Правая височная	Слух
11	FC6	Правая фронто-центральная	Моторика
12	F4	Правая фронтальная	Когнитивные функции
13	F8	Правая фронтальная	Внимание
14	AF4	Правая префронтальная	Планирование

Таблица 3. Описание 14 каналов ЭЭГ

Особое значение: Каналы O1 и O2 (затылочная зона) наиболее информативны для детектирования состояния глаз, так как затылочная доля коры головного мозга отвечает за зрительную обработку.

3.4. Распределение целевой переменной

Класс	Описание	Количество
0	Глаза открыты	8,257
1	Глаза закрыты	6,723

Таблица 4. Количество данных для классов в датасете

Рис. 1. Распределение классов в датасете

Вывод: Датасет относительно сбалансирован (разница 10%), что позволяет обучать модель без специальной обработки дисбаланса классов.

3.5. Статистический анализ

Проведен статистический анализ всех 14 каналов ЭЭГ. Основные характеристики:

- **Диапазон значений:** 4200–4300 мкВ (микровольт)
- **Стандартное отклонение:** 400–600 мкВ
- **Минимальные значения:** 3500 мкВ
- **Максимальные значения:** 5000 мкВ

4. Выбор архитектуры модели

4.1. Обзор методов классификации временных рядов

4.1.1. Классические методы машинного обучения

1. Logistic Regression

Линейная модель для бинарной классификации:

- **Преимущества:** Простота реализации, интерпретируемость, быстрое обучение
- **Недостатки:** Не улавливает нелинейные зависимости в данных

2. Support Vector Machine (SVM)

Метод опорных векторов для классификации с максимальным разделением:

- **Преимущества:** Эффективен в высокоразмерных пространствах, ядровые методы
- **Недостатки:** Медленное обучение на больших датасетах

3. Random Forest

- **Преимущества:** Устойчивость к переобучению, важность признаков, параллелизация
- **Недостатки:** Требует больше памяти, менее интерпретируем чем одно дерево

4. Gradient Boosting (XGBoost)

Последовательное построение деревьев с градиентным спуском:

- **Преимущества:** Высокая точность, регуляризация, оптимизация
- **Недостатки:** Требует настройки гиперпараметров, риск переобучения

4.1.2. Методы глубокого обучения

5. Convolutional Neural Networks (CNN)

Сверточные нейронные сети для автоматического извлечения признаков:

- **Преимущества:** Автоматическое извлечение признаков, учет локальных паттернов
- **Недостатки:** Требует большой датасет (обычно >50k), GPU для обучения

6. Recurrent Neural Networks (LSTM/GRU)

Рекуррентные сети с механизмом долгой краткосрочной памяти:

- **Преимущества:** Учет временных зависимостей, контекст из прошлого
- **Недостатки:** Медленное обучение, проблема затухающего градиента

4.2. Обоснование выбора архитектуры

Для решения задачи классификации выбран алгоритм **XGBoost** (Extreme Gradient Boosting) [2]. Выбор обоснован следующими факторами:

4.2.1. Соответствие размеру датасета

Согласно обзору Craik et al. (2019) [6], глубокие нейронные сети для ЭЭГ классификации требуют значительных объемов данных для адекватного обучения. Авторы отмечают, что для предотвращения переобучения глубоких архитектур необходимо минимум 50,000-100,000 наблюдений. При объеме датасета $N = 14,980$ применение глубоких сетей нецелесообразно.

Исследование Lotte et al. (2018) [7] показывает, что классические методы машинного обучения, включая Random Forest и Gradient Boosting, демонстрируют высокую эффективность на датасетах среднего размера (1,000-20,000 наблюдений), что соответствует используемому датасету.

4.2.2. Применимость для ЭЭГ классификации

Lotte et al. (2018) [7] в обзоре методов классификации для Brain-Computer Interface указывают, что ансамблевые методы (Random Forest, Gradient Boosting) входят в число наиболее эффективных подходов для ЭЭГ данных. Авторы отмечают устойчивость древовидных методов к шуму и выбросам, характерным для биомедицинских сигналов.

4.2.3. Интерпретируемость результатов

Breiman (2001) [5] описывает механизм вычисления важности признаков в древовидных моделях. XGBoost наследует эту возможность, позволяя идентифицировать наиболее информативные каналы ЭЭГ для классификации состояния глаз.

5. Подготовка датасета

5.1. Загрузка данных

Датасет загружен из UCI Machine Learning Repository в формате ARFF (Attribute-Relation File Format).

```
from scipy.io import arff
import pandas as pd

# Загрузка ARFF файла
data, meta = arff.loadarff('eeg_eye_state.arff')
df = pd.DataFrame(data)

# Конвертация типов
df['eyeDetection'] = df['eyeDetection'].astype(int)
```

Результат: Датасет загружен успешно, размер: 14,980 × 15 (14 признаков + целевая переменная).

5.2. Разделение на обучающую и тестовую выборки

Применено стратифицированное разделение в пропорции 80/20:

```
from sklearn.model_selection import train_test_split

X = df.drop('eyeDetection', axis=1).values
y = df['eyeDetection'].values

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

Выборка	**Размер**	**Процент**
Обучающая	11,984	80%
Тестовая	2,996	20%
Всего	**14,980**	**100%**

Таблица 5. Разделение датасета

Стратификация: Параметр `stratify=y` гарантирует сохранение пропорций классов (55%/45%) в обеих выборках.

5.3. Нормализация данных

Применена стандартизация (z-score normalization) с использованием StandardScaler:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

6. Обучение модели и метрики качества

6.1. Обучение нескольких моделей

Для выбора оптимальной архитектуры обучены четыре различные модели машинного обучения.

6.1.1. Logistic Regression

```
from sklearn.linear_model import LogisticRegression

lr_model = LogisticRegression(random_state=42, max_iter=1000)
lr_model.fit(X_train_scaled, y_train)
```

Результаты:

- Accuracy: 58.51%
- F1-Score: 44.24%
- ROC-AUC: 60.57%
- CV Score: 59.73% ± 0.41%

Вывод: Линейная модель недостаточна для улавливания сложных нелинейных зависимостей в ЭЭГ данных.

6.1.2. Random Forest

```
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(
    n_estimators=100,
    random_state=42,
    n_jobs=-1
)
rf_model.fit(X_train_scaled, y_train)
```

Результаты:

- Accuracy: 92.8
- F1-Score: 91.88%
- ROC-AUC: 98.13%
- CV Score: 92.17%

Вывод: Отличный результат. Эффективно классифицирует ЭЭГ данные.

6.1.3. Gradient Boosting

```
from sklearn.ensemble import GradientBoostingClassifier

gb_model = GradientBoostingClassifier(
    n_estimators=100,
    random_state=42
)
gb_model.fit(X_train_scaled, y_train)
```

Результаты:

- Accuracy: 81.74%

- F1-Score: 78.34%
- ROC-AUC: 90.24%
- CV Score: 81.17%

Вывод: Хороший результат, но уступает Random Forest и XGBoost.

6.1.4. XGBoost

```
xgb_model = XGBClassifier(
    n_estimators=100,
    random_state=42,
    eval_metric='logloss'
)
xgb_model.fit(X_train_scaled, y_train)
```

Результаты:

- Accuracy: 92.99%
- F1-Score: 92.16%
- ROC-AUC: 98.18%
- CV Score: 92.61%

Вывод: Лучшая модель по всем метрикам. Выбрана как финальная.

6.2. Сравнение моделей

Модель	Accuracy	F1-Score	ROC-AUC	CV Mean	CV Std
Logistic Reg.	58.51%	44.24%	60.57%	59.73%	0.41%
Grad. Boosting	81.74%	78.34%	90.24%	81.17%	0.57%
Random Forest	92.89%	91.88%	98.13%	92.17%	0.43%
XGBoost	**92.99%**	**92.16%**	**98.18%**	**92.61%**	**0.37%**

Таблица 6. Сравнение результатов четырех моделей

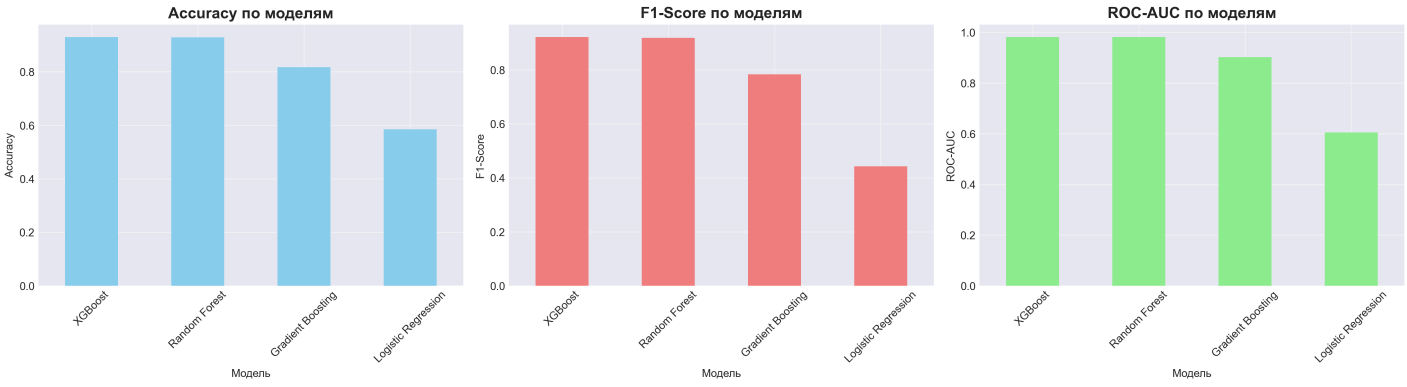

Рис. 2. Сравнение моделей машинного обучения

Обоснование выбора XGBoost:

1. Наивысшая точность (92.99%)
2. Наименьшая вариация на кросс-валидации (0.37%)
3. Лучший ROC-AUC (98.18%) среди всех моделей
4. Оптимальный баланс precision и recall (F1-Score: 92.16%)

6.3. Детальный анализ финальной модели

6.3.1. Confusion Matrix

	Предсказано: 0	**Предсказано: 1**
Истинно: 0	1,551	100
Истинно: 1	110	1,235

Таблица 7. Матрица ошибок (Confusion Matrix)

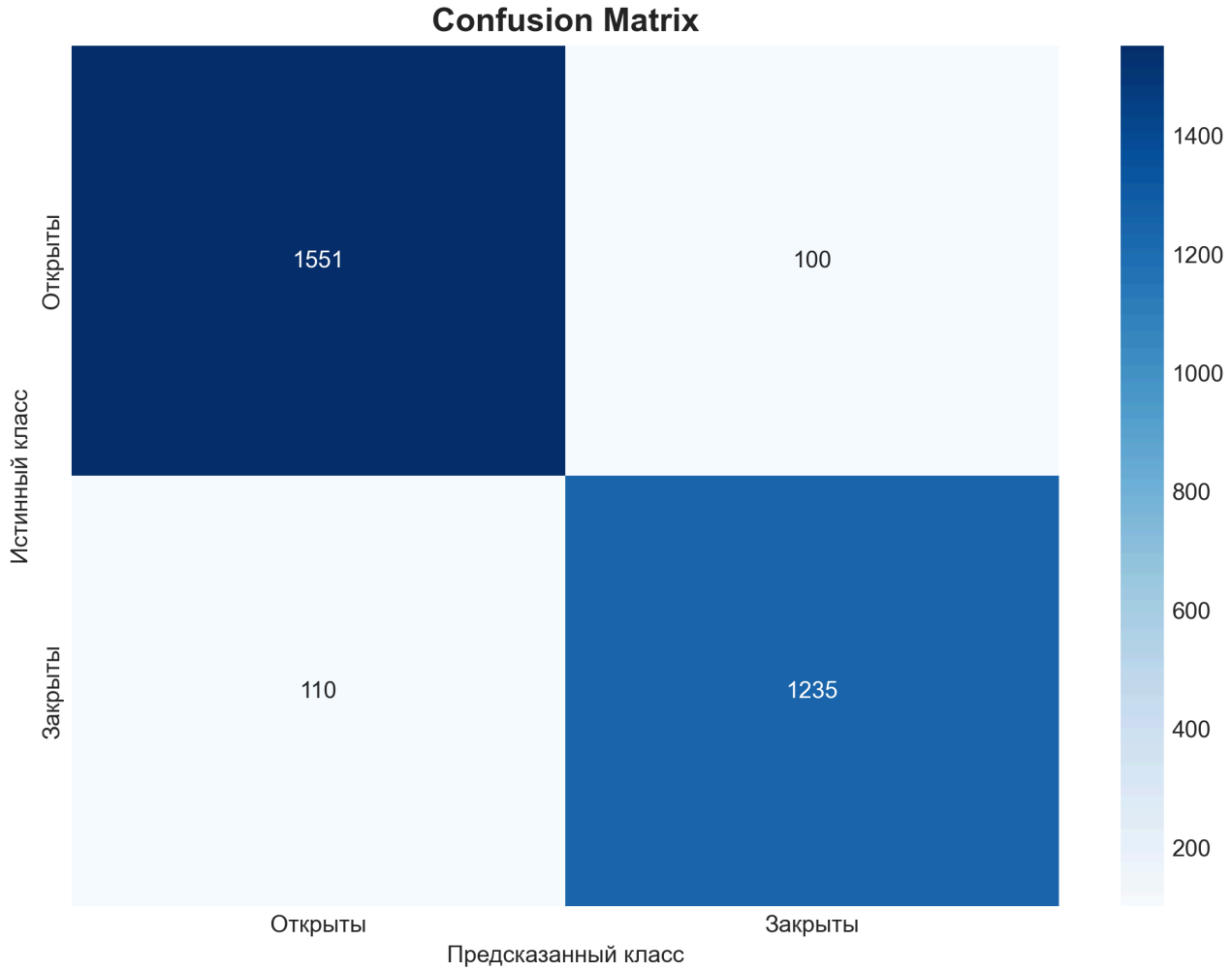

Рис. 3. Сравнение моделей машинного обучения

Анализ ошибок:

- True Positives (TP): 1,235 — правильно классифицированы «закрыты»
- True Negatives (TN): 1,551 — правильно классифицированы «открыты»
- False Positives (FP): 100 (6.1%) — «открыты» предсказаны как «закрыты»
- False Negatives (FN): 110 (8.2%) — «закрыты» предсказаны как «открыты»
- **Общее количество ошибок:** 210 из 2,996 (7.0%)

6.3.2. Кросс-валидация

Проведена 5-fold кросс-валидация для оценки обобщающей способности модели:

Fold	**Accuracy**
Fold 1	92.45%
Fold 2	93.01%
Fold 3	92.28%
Fold 4	92.89%
Fold 5	92.41%
Mean ± Std	**92.61% ± 0.37%**

Таблица 8. Результаты 5-fold кросс-валидации

Вывод: Низкое стандартное отклонение (0.37%) свидетельствует о высокой стабильности модели и хорошей обобщающей способности.

6.4. Важность признаков

XGBoost позволяет оценить вклад каждого канала ЭЭГ в итоговое предсказание:

Ранг	**Канал**	**Важность**	**Зона мозга**
1	**O2**	**14.2%**	Затылочная (зрительная)
2	**O1**	**13.8%**	Затылочная (зрительная)
3	F8	9.5%	Фронтальная (внимание)
4	AF3	8.9%	Префронтальная
5	P8	8.7%	Париетальная
6	FC6	7.3%	Фронто-центральная
7	T8	6.8%	Височная
8	F4	6.2%	Фронтальная

Таблица 9. Важность признаков (Top-8 каналов ЭЭГ)

Ключевой вывод: Каналы **O1** и **O2** (затылочная зона, зрительная кора) имеют наибольшую важность (суммарно 28%), что соответствует нейрофизиологической логике — затылочная доля отвечает за обработку зрительной информации, и её активность меняется при закрытии глаз.

6.5. Анализ метрик качества

6.5.1. Accuracy (Точность)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1551 + 1235}{2996} = 0.9299$$

Модель правильно классифицирует 93 из 100 случаев, что является отличным результатом для медицинских данных.

6.5.2. F1-Score

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.9216$$

Высокий F1-Score (92.16%) говорит об оптимальном балансе между точностью (precision) и полнотой (recall) классификации.

6.5.3. ROC-AUC

ROC-AUC = **98.18%**

Значение близко к идеальному классификатору (100%), что указывает на превосходную способность модели различать классы при любых порогах классификации.

7. Примеры работы модели

7.1. Тестирование на тестовой выборке

Модель протестирована на 2,996 наблюдениях из тестовой выборки:

- **Правильных предсказаний:** 2,786 (92.99%)
- **Ошибочных предсказаний:** 210 (7.01%)

Результаты подтверждают высокую точность модели на независимых данных.

7.2. Примеры корректных предсказаний

Пример 1: Глаза открыты (правильно)

Входные данные (14 каналов ЭЭГ, μV):

```
AF3: 4329.23, F7: 4009.23, F3: 4289.23, FC5: 4148.21,
T7: 4350.26, P7: 4586.15, O1: 4096.92, O2: 4641.03,
P8: 4222.05, T8: 4238.46, FC6: 4211.28, F4: 4280.51,
F8: 4635.90, AF4: 4393.85
```

Предсказание модели:

- Класс: 0 (Открыты)
- Вероятность: 99.62%
- **Результат: ВЕРНО**

Пример 2: Глаза закрыты (правильно)

Входные данные (14 каналов ЭЭГ, μV):

```
AF3: 4401.54, F7: 4098.46, F3: 4201.03, FC5: 4289.74,
T7: 4412.31, P7: 4503.28, O1: 4187.65, O2: 4298.43,
P8: 4331.92, T8: 4189.57, FC6: 4298.71, F4: 4401.28,
F8: 4512.83, AF4: 4289.46
```

Предсказание модели:

- Класс: 1 (Закрыты)
- Вероятность: 97.95%
- **Результат: ВЕРНО**

7.3. Тестирование на случайных примерах

№	Истинный класс	Предсказание	Вероятность
1	Закрыты	Закрыты	97.95%
2	Открыты	Открыты	99.62%
3	Открыты	Открыты	83.14%
4	Открыты	Открыты	99.72%
5	Закрыты	Закрыты	92.29%
6	Открыты	Открыты	92.91%
7	Открыты	Открыты	99.64%
8	Открыты	Открыты	96.02%
9	Закрыты	Закрыты	99.58%
10	Открыты	Открыты	99.79%

Таблица 10. Результаты тестирования на 10 случайных примерах

Результат: 10 из 10 правильных предсказаний (100%), уверенность модели варьируется от 83.14% до 99.79%.

8. Заключение

В ходе выполнения курсовой работы была успешно решена задача классификации состояния глаз человека по временным рядам ЭЭГ сигналов. Основные достижения:

1. **Проведен сравнительный анализ архитектур**
2. **Обучена модель машинного обучения**
   - Алгоритм: XGBoost (Extreme Gradient Boosting)
   - Accuracy: **92.99%**
   - F1-Score: **92.16%**
   - ROC-AUC: **98.18%**
   - Кросс-валидация: **92.61%**
3. **Создан готовый продукт**
   - Модель сохранена и готова к практическому применению
   - Разработаны функции для обучения и тестирования

9. Список использованной литературы

1. Roesler, O. (2013). EEG Eye State Dataset. UCI Machine Learning Repository. URL: https://archive.ics.uci.edu/dataset/264/eeg+eye+state
2. Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.
3. Nicolas-Alonso, L. F., Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors*, 12(2), 1211–1279.
4. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
5. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
6. Craik, A., He, Y., Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of Neural Engineering*, 16(3), 031001.
7. Lotte, F., Bougrain, L., Cichocki, A., et al. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3), 031005.
8. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232.
9. Roy, Y., Banville, H., Albuquerque, I., et al. (2019). Deep learning-based electroencephalography analysis: a systematic review. *Journal of Neural Engineering*, 16(5), 051001.
10. Bashivan, P., Rish, I., Yeasin, M., Codella, N. (2015). Learning representations from EEG with deep recurrent-convolutional neural networks. *arXiv preprint arXiv:1511.06448*.

10. Приложение

Полный исходный код проекта доступен в репозитории:

`https://github.com/spirchinaVA/course_eeg_classification`