

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (государственный университет)  
ФАКУЛЬТЕТ УПРАВЛЕНИЯ И ПРИКЛАДНОЙ МАТЕМАТИКИ  
КАФЕДРА «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ»

Самохина Алина Максимовна

**Непрерывное представление времени  
в задачах декодирования сигналов**

010990 — Интеллектуальный анализ данных

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА

**Научный руководитель:**  
д. ф.-м. н. Стрижов Вадим Викторович

Москва  
2021

# Содержание

Введение	4
Обозначения	8
<b>1 Постановка задачи</b>	<b>9</b>
1.1 Построение непрерывного представления сигнала . . . . .	9
1.2 Задача классификации сигнала . . . . .	9
1.2.1 Выборка P300 . . . . .	9
1.2.2 Выборка UCI . . . . .	10
<b>2 Обзор существующих алгоритмов</b>	<b>12</b>
2.1 Стандартные подходы к классификации ЭЭЭГ, использующие дискретное время . . . . .	12
2.1.1 ERPCov TS LR . . . . .	12
2.1.2 EEGNet . . . . .	13
2.2 Модели, использующие непрерывное время . . . . .	15
2.2.1 Neural CDE . . . . .	15
2.2.2 ODE LSTM . . . . .	15
2.2.3 LMU . . . . .	16
<b>3 Предлагаемый метод</b>	<b>20</b>
<b>4 Вычислительный эксперимент</b>	<b>21</b>
4.1 Постановка эксперимента . . . . .	21
4.2 Экспериментальные данные . . . . .	21
4.2.1 Набор данных потенциалов DemonsP300 . . . . .	21
4.2.2 Набор данных по видам активности UCI . . . . .	23
4.3 Результаты . . . . .	25
4.3.1 Эксперимент на данных P300 . . . . .	25
4.3.2 Эксперимент на данных UCI . . . . .	25
4.3.3 Непрерывное представление времени . . . . .	26
4.3.4 Выводы . . . . .	26
4.3.5 Воспроизводимость . . . . .	27
<b>Заключение</b>	<b>28</b>
<b>Список литературы</b>	<b>29</b>

## Аннотация

В задачах декодирования сигнала данные представляются как многомерные временные ряды. При решении задач используется дискретное представление времени. Однако недавние работы по нейронным обыкновенным дифференциальным уравнениям иллюстрируют возможность работать со скрытым состоянием рекуррентных нейронных сетей, как с решениями дифференциальных уравнений. Это позволяет рассматривать временные ряды как непрерывные по времени.

В данной работе рассматривается возможность представления сигнала как непрерывной по времени функции. Исследуются возможности практического применения данного подхода: для изменения частоты семплирования сигнала, обработки нерегулярных сигналов или сигналов с пропущенными значениями. Данные применения являются актуальными для сигналов с различных носимых устройств: акселерометров, пульсометров, устройств для снятия сигналов головного мозга.

Основными результатами работы являются исследование алгоритмов, позволяющих работать с сигналом как с непрерывным по времени и получение непрерывного по времени представления сигнала для последующей работы с ним в рамках построения фазовых пространств.

**Ключевые слова:** ЭЭГ, нейронные ОДУ, управляемые ДУ, полиномы Лежандра, временные ряды, классификация.

# Введение

**Актуальность темы.** Магистерская работа посвящена задаче непрерывного представления времени в задачах декодирования сигналов.

Фокусом работы является исследование того, какие возможности дает непрерывное представление сигнала по времени, и способы получения таких представлений.

Примерами актуальных задач декодирования сигнала являются задачи декодирования сигнала носимых устройств. Под носимыми устройствами подразумеваются акселерометры, пульсометры и устройства для снятия ЭЭГ. Данные устройства должны быть доступны для широкого круга пользователей, в связи с чем должны иметь невысокую себестоимость. Результатом низкой стоимости устройств являются: низкая частота дискретизации сигнала, малое количество датчиков, периодическая потеря контакта с устройством во время передачи данных. Таким образом, на выходе с устройства мы получаем зашумленный сигнал с нерегулярным шагом по временной сетке.

Предполагается, что непрерывность сигнала по времени позволит решить выше-названные проблемы с получаемыми сигналами.

В последние несколько лет появилось большое количество работ по теме нейронных обыкновенных дифференциальных уравнений (нейронных ОДУ) [1–9].

Большинство подходов опираются на статью [1], в которой авторы предлагают рассматривать блок ResNet [10] как обыкновенное дифференциальное уравнение, исходя из синтаксической схожести скрытого слоя с формулой Эйлера для численного интегрирования ОДУ. Таким образом, авторы получают непрерывный в глубину аналог ResNet. Однако входные данные для этих сетей не являются временными рядами и непрерывность рассматривается исключительно в контексте непрерывности скрытого состояния рекуррентных нейронных сетей (RNN). Общий класс таких подходов называется нейронными дифференциальными уравнениями (Neural ODE).

Так как синтаксической схожести недостаточно для утверждения о том, что скрытое состояние нейронной сети может представлять собой решение дифференциального уравнения, авторы работы [6] показали, что несмотря на то, что ResidualBlock не является аналогом метода Эйлера, конструкция Neural ODE, рассматривающая схемы интегрирования методом Рунге-Кутты четвертого порядка, может считаться численным интегратором по всем свойствам сходимости.

Следующим шагом в подходах, использующих нейронные ОДУ стали работы [7, 11], которые, в свою очередь, рассматривают различные варианты работы с временными рядами в контексте нейронных обыкновенных дифференциальных уравнений и байесовских моделей. Однако ни один из этих подходов не позволяет получить непрерывное представление исходного сигнала.

В основу данного исследования легли следующие работы: исследование [8], рассматривающее возможность применения нейронных ОДУ к временным рядам с нерегулярной сеткой по времени с использованием управляемых дифференциальных уравнений, работа [12], представляющая непрерывное по времени обобщение LSTM [13] и работа [14], позволяющая получить непрерывное представление сигнала в виде разложения на полиномы Лежандра [15], используя в качестве коэффициентов состояние памяти LSTM.

Большинство рассмотренных работ рассматривают в качестве данных для тестирования моделей простые смоделированные наборы данных и MNIST [16]. В данной

работе внимание направлено на решение реальных задач для временных рядов. Основная задача данного исследования — рассмотреть применимость данных методов к декодированию сигналов с носимых устройств на реальных данных, сравнить их качество со стандартными подходами и оценить получаемые непрерывные представления сигналов.

В качестве основной задачи декодирования сигнала в данной работе рассматривается классификация электроэнцефалограмм [17]. В последнее время большое количество работ посвящено методам считывания мозговой активности и декодирования информации [18–25]. Основным приложением данных методов являются нейрокомпьютерные интерфейсы.

Нейрокомпьютерный интерфейс (НКИ) — технология, позволяющая человеку взаимодействовать с компьютером с помощью анализа данных о мозговой активности. В первую очередь НКИ разрабатывались для пациентов с ограниченными возможностями и использовались для коммуникации пациентов с различными формами паралича и для реабилитации моторной функции после инсультов и спинальных травм [26]. Однако в последнее время большой интерес вызывают не только нейроинтерфейсы для медицинских целей [18–24]. На сегодняшний день существует множество подходов к использованию НКИ, основанных на неинвазивных электродах, для здоровых людей [27]. Кроме того, нейроинтерфейсы могут быть использованы и в рекреационных целях, например, в играх [28, 29].

Для построения НКИ используются различные виды биосигналов как требующих операции для имплантации электродов [30], так и неинвазивных: fNIRS [31], фМРТ [32], ЭЭГ. Нейроинтерфейсы на основе электроэнцефалограмм (ЭЭГ) на данный момент являются наиболее популярными за счет лёгкой настройки, низкой стоимости и высокого временного разрешения метода.

Существует несколько разновидностей нейрокомпьютерных интерфейсов: пассивные, активные и реактивные.

Пассивные НКИ — вид нейроинтерфейсов, в которых в фоновом режиме считываются реакции мозга человека не в контексте осознанного выполнения задачи нейроуправления. Полученные данные, которые могут говорить об уровне утомления и когнитивной нагрузке на фоне выполнения задачи, могут использоваться для обратной связи и облегчения работы пользователя [33].

Так называемые "активные" НКИ распознают изменение активности мозга при выполнении различных ментальных действий [34]. Примером является нейроинтерфейс, распознающий изменения ритмической активности ЭЭГ при расслаблении и ментальной концентрации [35].

В активных нейроинтерфейсах на основе анализа ритмической активности ЭЭГ пользователи могут достигать точности выбора двух или трех команд около 70%. Достижение более высокой точности осложнено сложностью ментальной задачи для неподготовленного человека и необходимостью обучения пользователя ее правильному выполнению.

Реактивные нейроинтерфейсы работают за счет выявления реакций мозга на специфическую стимуляцию. Пользователь привлекает свое внимание к одному из нескольких стимулов, которые вызывают реакции, регистрируемые в ЭЭГ. Стимулы могут иметь различную модальность (к примеру, слуховую и тактильную), но самые надежные НКИ создаются на основе зрительных стимулов разных типов, которые вызывают различные ответы в активности мозга.

Самый популярный вид реактивных НКИ — НКИ на основе вызванного потенциала P300. В данном типе нейроинтерфейса управление совершается за счет привлечения внимания к разнесенным во времени стимулам. При активации стимула, на который пользователь хочет отреагировать, в окружении стимулов, которые ему безразличны, в центральных отделах мозга регистрируется потенциал связанный с событием с латентностью (задержкой во времени) пика около 300 мс после стимула [36]. Нейроинтерфейсы на основе P300 имеют высокую скорость передачи информации за счет высокой точности классификации команд и потенциально большое их количество. На основе этой технологии также можно создание систем набора текста. Преимущество НКИ на P300 заключается в меньшей чувствительности к модальности стимула: один и тот же интерфейс можно использовать со стимулами любых форм и размеров, при условии, что они достаточно легко различимы. Это позволяет создать интерфейс не только полезный пациентам, но и интересный для здоровых пользователей.

В данной работе рассматриваются данные электроэнцефалограмм, полученные на здоровых пользователях при игре в виртуальной реальности с нейроуправлением.

**Цель работы.** Целью работы является построение нейроинтерфейса, работающего с многомерными временными рядами как с непрерывными по времени. Предполагается, что рассматриваемые подходы позволят решить следующие задачи для сигналов с носимых устройств:

- передискретизация сигнала при низких частотах семплирования,
- обработка нерегулярных по времени данных при непостоянном контакте со считывающим устройством.

Также предполагается, что алгоритмы, используемые при построении такого нейроинтерфейса позволяют получить непрерывное представление сигнала.

**Методы исследования.** В работе рассматривается применение алгоритмов на основе нейронных дифференциальных уравнений к задаче определения объекта внимания пользователя по данным с неинвазивных электродов на основе парадигмы P300 [37]. Также, с целью подтвердить результаты экспериментов на данных с других носимых устройств, было проведено исследование алгоритмов на наборе данных для определения вида активности человека по данным акселерометров [38].

**Основные положения, выносимые на защиту.**

1. Исследована возможность декодирования сигналов как непрерывных по времени;
2. Доказана теорема о градиентах для LMU LSTM;
3. Для реальной задачи проведено сравнение качества моделей, использующих нейронные ОДУ со стандартными подходами к декодированию сигналов ЭЭГ;
4. Получены непрерывные по времени представления сигнала.

**Теоретическая значимость.** Применение рассматриваемых методов позволяет работать с электроэнцефалограммами и данными с других носимых устройств как с непрерывными по времени. Это предоставляет возможности дальнейшего исследования непрерывных сигналов, например, в контексте построения фазовых пространств для непрерывного времени и сравнения их с фазовыми пространствами для дискретного времени.

**Практическая значимость.** Методы, использующие нейронные обыкновенные дифференциальные уравнения, улучшают качество декодирования сигналов и позволяют работать с задачами передискретизации сигнала и обработки нерегулярных или пропущенных значений.

**Степень достоверности и апробация работы.** Достоверность результатов подтверждена экспериментальной проверкой полученных методов на реальных задачах. Некоторые результаты, связанные с темой работы, докладывались и обсуждались на следующих научных конференциях:

- 13-я Международная конференция «Интеллектуализация обработки информации», 2020 г.;
- 63-я Всероссийская научная конференция МФТИ, 2020 г..

## Обозначения и сокращения

- ОДУ(ODE) — обыкновенное дифференциальное уравнение (ordinary differential equation)
- УДУ(CDE) — управляемое дифференциальное уравнение (controlled differential equation)
- ЭЭГ — электроэнцефалограмма
- НКИ — нейрокомпьютерный интерфейс
- LSTM — нейронная сеть с долгой кратковременной памятью
- ODE LSTM — LSTM со скрытым состоянием, представленным дифференциальным уравнением
- ODE RNN — рекуррентная нейронная сеть со скрытым состоянием, представленным дифференциальным уравнением
- LMU LSTM — LSTM со скрытым состоянием, являющимся коэффициентами разложения сигнала по полиномам Лежандра



# 1 Постановка задачи

## 1.1 Построение непрерывного представления сигнала

Одной из основных задач данной работы является получение непрерывного представления сигнала. Пусть, имеется непрерывный процесс (активность мозга, движение):

$$V(t), t \in \mathbb{R}$$

Тогда данные выборки, регистрируемый сигнал — реализация процесса  $V(t)$ :

$$\mathbf{X} = \{\mathbf{x}_t\}_{t \in T}, \mathbf{x}_t \in \mathbb{R}^E, T = \{t_i\}_{i=1}^N, t_i \in \mathbb{R}$$

$$\mathbf{x}_{t_i} \approx V(t_i).$$

Предполагается, что можно получить:

$$f_{\mathbf{X}}(t) : \mathbb{R} \rightarrow \mathbb{R}^E, f_{\mathbf{X}}(t) = \mathbf{x}_t \approx V(t).$$

В данной работе построение непрерывного представления рассматривается в рамках решения задачи классификации сигналов. Данная задача не ставится отдельным экспериментом. Непрерывное представление извлекается из скрытых состояний рассматриваемых далее алгоритмов классификации.

## 1.2 Задача классификации сигнала

Данная задача рассматривается для двух наборов данных в двух различных постановках. Для каждой постановки задачи рассматриваются два варианта входных данных: исходные данные, являющиеся регулярными по сетке времени, и изменённые данные, содержащие 80% отсчётов времени, выбранных случайным образом для каждого элемента сигнала.

Для каждой задачи выборка делится на две части: обучающую и валидационную в соотношении 80:20.

### 1.2.1 Выборка P300

**Регулярные данные.** Выборка изначально является регулярной по времени. Пусть, в выборке присутствует  $M$  наблюдений. Тогда сигнал и целевая переменная определяются следующим образом:

$$\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^M,$$

$$\mathbf{X}_i = \{\mathbf{x}_t\}_{t \in T}, \mathbf{x}_t \in \mathbb{R}^E, T = \{t_i\}_{i=1}^N, t_i \in \mathbb{R}$$

$$\mathbf{Y} = \{y_i\}_{i=1}^M, y_i \in \{0, 1\}$$

$$\Delta t = t_i - t_{i-1} = \text{const}$$

$E = 8$  — количество электродов

$N = 40$  — количество наблюдений в одном отрезке ЭЭГ

**Нерегулярные данные.** Определение сигнала и целевой переменной для нерегулярных данных совпадает с определением для регулярных, за исключением длины сигнала и шага по времени:

$$\Delta t = t_i - t_{i-1} \neq \text{const}$$

$$E = 8 \text{ — количество электродов}$$

$$N = 32 \text{ — количество наблюдений в одном отрезке ЭЭГ}$$

**Постановка задачи классификации.** В парадигме P300 решается задача бинарной классификации отрезков ЭЭГ. Задача — определить наличие в отрезке ЭЭГ потенциала P300.

Для данных, рассмотренных в двух предыдущих разделах, требуется получить целевую функцию:

$$g_\theta : \mathbf{X} \rightarrow \mathbf{Y}.$$

Критерием качества в данной задаче является бинарная кросс-энтропия:

$$L = -\frac{1}{M} \sum_{m=1}^M \left[ y_m \log p_m + (1 - y_m) \log(1 - p_m) \right]$$

$$p_m = g_\theta(\mathbf{X}_m) \text{ — вероятность 1 класса для } \mathbf{X}_m$$

Решается оптимизационная задача:

$$\hat{\theta} = \arg \max_{\theta} L(\theta, \mathbf{X}).$$

Внешними критериями качества для задачи классификации являются точность и F1-score.

## 1.2.2 Выборка UCI

**Регулярные данные.** Данная выборка также изначально является регулярной по времени. Пусть, в выборке присутствует  $M$  наблюдений. Тогда сигнал и целевая переменная определяются следующим образом:

$$\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^M,$$

$$\mathbf{X}_i = \{\mathbf{x}_t\}_{t \in T}, \mathbf{x}_t \in \mathbb{R}^E, T = \{t_i\}_{i=1}^N, t_i \in \mathbb{R}$$

$$\mathbf{Y} = \{\mathbf{Y}_i\}_{i=1}^M, \mathbf{Y}_i = \{y_t\}_{t \in T}, y_t \in \{0, \dots, C\}$$

$$\Delta t = t_i - t_{i-1} = \text{const}$$

$$C = 7 \text{ — количество классов активности}$$

$$E = 7 \text{ — количество датчиков}$$

$$N = 32 \text{ — количество наблюдений в одной записи активности}$$

**Нерегулярные данные.** Определение сигнала и целевой переменной для нерегулярных данных также совпадает с определением для регулярных, за исключением длины записи и шага по времени:

$$\Delta t = t_i - t_{i-1} \neq \text{const}$$

$$E = 7 \text{ — количество датчиков}$$

$$N = 25 \text{ — количество наблюдений в одной записи активности}$$

**Постановка задачи классификации.** В задаче классификации активности предлагается получить классификацию положения для каждого отсчёта времени наблюдении.

Для данных, рассмотренных в двух предыдущих разделах, требуется получить целевую функцию:

$$g_\theta : \mathbf{X} \rightarrow \mathbf{Y}.$$

Критерием качества в данной задаче является кросс-энтропия:

$$L = -\frac{1}{M} \sum_{m=1}^M \left[ \sum_{c=0}^C y_m^c \log(p_m^c) \right]$$

$$p_m^c = g_\theta(\mathbf{X}_m) \text{ — вероятность класса } c \text{ для } \mathbf{X}_m$$

Решается оптимизационная задача:

$$\hat{\theta} = \arg \max_{\theta} L(\theta, \mathbf{X}).$$

Внешними критериями качества для данной задачи также являются точность и F1-score.

## 2 Обзор существующих алгоритмов

Рассмотрим два типа подходов к классификации сигналов ЭЭГ: основанные на дискретном и на непрерывном представлении времени. Первый подход является классическим и подразумевает равные промежутки между отсчётами времени и фиксированную длину последовательности. В качестве базовой модели рассматривается стандартная логистическая регрессия с преобразованием признакового пространства в терминах римановой геометрии [39]. Основная модель, использующая дискретное время, с которой будет проводиться сравнение методов, использующих непрерывное время — свёрточная нейронная сеть EEGNet [40]. Данные модели выбраны, так как для используемого набора данных DemonsP300 они показывают наилучшие результаты среди классических и нейросетевых подходов соответственно [25]. Второй подход к классификации ЭЭГ использует время как непрерывную переменную. В рамках этого подхода рассматриваются модели NeuralCDE [8], ODE LSTM [12] и LMU LSTM [14]. Данные модели были выбраны, так как они позволяют работать с нерегулярными данными и, предполагается, могут быть использованы для построения непрерывного представления сигнала.

### 2.1 Стандартные подходы к классификации ЭЭГ, использующие дискретное время

#### 2.1.1 ERPCov TS LR

Одним из наиболее успешных классических подходов к классификации ЭЭГ является алгоритм ERPCov TS LR, основанный на Римановой геометрии [25, 39]. Первым шагом данного алгоритма является построение признакового пространства.

Если одна запись ЭЭГ представляется как

$$\mathbf{X}_i = \begin{bmatrix} x_{1,1}^i & x_{1,2}^i & \dots & x_{1,N}^i \\ \dots & \dots & \dots & \dots \\ x_{E,1}^i & x_{E,2}^i & \dots & x_{E,N}^i \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1 \\ \dots \\ \mathbf{t}_E \end{bmatrix}$$

Тогда стандартная матрица ковариации:

$$\mathbf{C}_i = \frac{1}{T-1} \mathbf{X}_i \mathbf{X}_i^T, \mathbf{C}_i \in \mathbb{R}^{E \times E} \forall i \in \{1, \dots, N\}$$

$$\text{Cov}(\mathbf{X}_i) = \begin{bmatrix} \text{Var}(\mathbf{t}_1) & \text{Cov}(\mathbf{t}_1, \mathbf{t}_2) & \dots & \text{Cov}(\mathbf{t}_1, \mathbf{t}_E) \\ \text{Cov}(\mathbf{t}_1, \mathbf{t}_2) & \text{Var}(\mathbf{t}_2) & \dots & \text{Cov}(\mathbf{t}_1, \mathbf{t}_E) \\ \dots & \dots & \dots & \dots \\ \text{Cov}(\mathbf{t}_E, \mathbf{t}_1) & \text{Cov}(\mathbf{t}_E, \mathbf{t}_2) & \dots & \text{Var}(\mathbf{t}_E) \end{bmatrix}$$

Для распознавания P300 в данном алгоритме используется расширенная матрица ковариации: матрица ковариации потенциалов связанных с событием (Event-related potential covariance, ERPCov)

$$\mathbf{C}_i = \frac{1}{T-1} \mathbf{P}_i \mathbf{P}_i^T, \mathbf{C}_i \in \mathbb{R}^{E(c+1) \times E(c+1)} \forall i \in \{1, \dots, N\}$$

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{X}_i \end{bmatrix}$$

где  $\mathbf{P}_0$  и  $\mathbf{P}_1$  — средние геометрические классов 0 и 1 соответственно.

$$\mathbf{P}_i = \arg \min_{\mathbf{P}_i \in \mathbb{R}^{E \times T}} \sum_{m=1}^M \mathcal{I}[\mathbf{Y}_m == i] \delta_R^2(\mathbf{P}_i, \mathbf{X}_m)$$

Пространство положительно определенных симметричных матриц является римановым многообразием. Матрицы ковариации являются положительно определенными и симметричными, поэтому дальнейшая работа с ними производится как с элементами Риманова многообразия.

В каждой точке  $\mathbf{C}$  Риманова многообразия имеется касательная плоскость (tangent space, TS) с определенным на ней скалярным произведением. Построив касательную плоскость в точке среднего геометрического известных ковариационных матриц по Римановой метрике, получим общую касательную плоскость для отображения всех матриц ковариации нашей выборки.

Геометрическое среднее симметричных положительно определённых матриц задаётся

$$\mathbf{C} = \mathfrak{G}(\mathbf{C}_1, \dots, \mathbf{C}_I) = \arg \min_{\mathbf{C} \in \mathcal{C}(n)} \sum_{i=1}^I \delta_R^2(\mathbf{C}, \mathbf{C}_i)$$

Метрикой является риманово геодезическое расстояние

$$\delta_R(\mathbf{C}_1, \mathbf{C}_2) = \|\log(\mathbf{C}_1^{-1} \mathbf{C}_2)\|_F = \left[ \sum_{i=1}^n \log^2 \lambda_i \right]^{1/2}$$

$\lambda_i$  — собственные числа  $\mathbf{C}_1^{-1} \mathbf{C}_2$

Для каждой ковариационной матрицы  $\mathbf{C}_i$  существует касательный вектор  $\mathbf{S}$ . Отображения  $\mathbf{C}_i \rightarrow \mathbf{S}_i$  и  $\mathbf{S}_i \rightarrow \mathbf{C}_i$ :

$$\text{Exp}_{\mathbf{C}}(\mathbf{S}_i) = \mathbf{C}_i = \mathbf{C}^{\frac{1}{2}} \exp\left(\mathbf{C}^{-\frac{1}{2}} \mathbf{S}_i \mathbf{C}^{-\frac{1}{2}}\right) \mathbf{C}^{\frac{1}{2}}$$

$$\log_{\mathbf{C}}(\mathbf{C}_i) = \mathbf{S}_i = \mathbf{C}^{\frac{1}{2}} \log\left(\mathbf{C}^{-\frac{1}{2}} \mathbf{C}_i \mathbf{C}^{-\frac{1}{2}}\right) \mathbf{C}^{\frac{1}{2}}$$

Операции построения ковариационных матриц и их отображения в касательное пространство выполняются при использовании библиотеки PyRiemann [41]. Полученные касательные векторы используются как новое признаковое пространство и классифицируются при помощи алгоритма логистической регрессии.

### 2.1.2 EEGNet

В работе [40] представлена свёрточная нейросеть, осуществляющая классификацию данных ЭЭГ в различных парадигмах, в том числе в рассматриваемой парадигме P300.

Нейронная сеть EEGNet состоит из двух основных блоков:

1. Основа блока 1 состоит из двух свёрток:

- (a)  $F_1$  двумерных свёрток размера  $(\frac{1}{\text{rate}})$  позволяют получить представление сигнала на различных частотах.
- (b)  $D$  свёрток в глубину (depthwise convolution) размера  $(E, 1)$  производит пространственную фильтрацию

Данный блок вдохновлен алгоритмом FBCSP [42].

2. Основная часть блока 2 состоит из двух свёрток:

- (a)  $D \cdot F_1$  разделимых свёрток (separable convolution)
- (b)  $F_2$  поточечных свёрток

Данная часть отвечает за представление входных данных в разных временных шкалах

Полная структура данной модели приведена в таблице 1. В работе использованы следующие параметры для данной архитектуры:

$$F_1 = 4, D = 4, p_1 = 2, p_2 = 4$$

Гиперпараметры подбирались для решения данной задачи на данных DemonsP300 с помощью поиска по сетке и кросс-валидации. Параметры  $C = 8, N = 40, \text{rate} = 50$  извлекаются из используемых данных.

Блок	Слой	Количество фильтров	Размер фильтров	Размер данных на выходе
1	Input			$(E, N)$
	Reshape			$(1, E, N)$
	Conv2D	$F_1$	$(1, \frac{1}{2}\text{rate})$	$(F_1 \cdot E, N)$
	BatchNorm			$(F_1 \cdot E, N)$
	ReLU			$(F_1 \cdot E, N)$
	DepthwiseConv2D	$D * F_1$	$(E, 1)$	$(D * F_1, 1, N)$
	BatchNorm			$(D * F_1, 1, N)$
	ReLU			$(D * F_1, 1, N)$
	AveragePool2D		$(1, p_1)$	$(D * F_1, 1, \frac{N}{p_1})$
	Dropout*			$(D * F_1, 1, \frac{N}{p_1})$
2	SeparableConv2D	$F_2$	$(1, \frac{1}{4}\text{rate})$	$(F_2, 1, \frac{N}{p_1})$
	BatchNorm			$(F_2, 1, \frac{N}{p_1})$
	ReLU			$(F_2, 1, \frac{N}{p_1})$
	AveragePool2D		$(1, p_2)$	$(F_2, 1, \frac{N}{p_1 * p_2})$
	Dropout*			$(F_2, 1, \frac{N}{p_1 * p_2})$
	Flatten			$(F_2 * \frac{N}{p_1 * p_2})$
класс	Dense	$C * \frac{F_2 * N}{p_1 * p_2}$		$C$

Таблица 1: Структура EEGNet

## 2.2 Модели, использующие непрерывное время

### 2.2.1 Neural CDE

Управляемые нейронные дифференциальные уравнения позволяют работать с данными, имеющими нерегулярный шаг по сетке по времени. Как показано в статье [12] данный подход, как и другие ODE-RNN, имеет проблемы с затухающими и взрывающимися градиентами для больших длин последовательностей.

Модель NeuralCDE состоит из нескольких шагов:

1. аппроксимация входных данных натуральным кубическим сплайном
2. решение дифференциального уравнения относительно скрытого состояния модели
3. классификация временного ряда по информации о скрытых состояниях

Распишем данные шаги формально.

$S$  — аппроксимация процесса  $V(t)$ , реализации которого представлены в  $\mathbf{X}$ .

$S_{\mathbf{X}}(t) : [t_0, t_n] \rightarrow \mathbb{R}^{E+1}$  — натуральный кубический сплайн с узлами в  $t_0, \dots, t_n$ ,  $S_{t_i} = (\mathbf{x}_i, t_i)$ .

Пусть  $f_{\theta} : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times (E+1)}$  — нейронная сеть, зависящая от параметров  $\theta$ ,  $d$  — размерность скрытого пространства

Скрытое состояние Neural CDE — решение управляемого дифференциального уравнения

$$\mathbf{z}_t = \mathbf{z}_{t_0} + \int_{t_0}^t f_{\theta}(\mathbf{z}_{\tau}) dS_{\tau} \quad \text{for } t \in (t_0, t_n]$$

Получение значения скрытого состояния: `ODESolve()`

Функция, определяющая начальное скрытое состояние:  $i_{\theta_i} : \mathbb{R}^{E+1} \rightarrow \mathbb{R}^d$

Функция классификации:  $r_{\theta_r} : \mathbb{R}^d \rightarrow \{0, \dots, c\}$ ,  $c$  — количество классов в выборке

Результирующая модель:

$$r_{\theta_r}(\text{ODESolve}(S, i_{\theta_i}(S)))$$

В результирующей модели можно увидеть функцию `ODESolve()`. Метод обратного распространения ошибки в данном блоке модели реализован с помощью алгоритма сопряжённого распространения ошибки [43].

### 2.2.2 ODE LSTM

Как показывают авторы статьи [12], данная модель решает проблему затухающих и взрывающихся градиентов, возникающую для ODE-RNN. Архитектура ODE LSTM основывается на стандартной архитектуре LSTM, добавляя в неё непрерывность скрытого состояния.

Скрытое состояние LSTM — пара  $(\mathbf{c}_t, \mathbf{h}_t)$ , где  $\mathbf{c}_t$  — состояние долгосрочной памяти,  $\mathbf{h}_t$  — обычное скрытое состояние модели. Функция обновления этих состояний  $f_{\theta}(\mathbf{x}_{t+1}, (\mathbf{c}_t, \mathbf{h}_t), \mathbf{1}) \mapsto (\mathbf{c}_{t+1}, \mathbf{h}_{t+1})$  записывается следующим набором уравнений:

$$\begin{aligned}
\mathbf{z}_{t+1} &= \tanh(\mathbf{W}_z \mathbf{x}_{t+1} + \mathbf{R}_z \mathbf{h}_t + \mathbf{b}_z) \\
\mathbf{i}_{t+1} &= \sigma(\mathbf{W}_i \mathbf{x}_{t+1} + \mathbf{R}_i \mathbf{h}_t + \mathbf{b}_i) \\
\mathbf{f}_{t+1} &= \sigma(\mathbf{W}_f \mathbf{x}_{t+1} + \mathbf{R}_f \mathbf{h}_t + \mathbf{b}_f + \mathbf{1}) \\
\mathbf{o}_{t+1} &= \sigma(\mathbf{W}_o \mathbf{x}_{t+1} + \mathbf{R}_o \mathbf{h}_t + \mathbf{b}_o) \\
\mathbf{c}_{t+1} &= \mathbf{z}_{t+1} \odot \mathbf{i}_{t+1} + \mathbf{c}_t \odot \mathbf{f}_{t+1} \\
\mathbf{h}_{t+1} &= \tanh(\mathbf{c}_{t+1}) \odot \mathbf{o}_{t+1}
\end{aligned}$$

В уравнениях выше приняты следующие обозначения:  $\sigma: x \mapsto \frac{1}{(1+\exp(-x))}$ , матрицы  $\mathbf{W}_x, \mathbf{R}_x$ , и векторы  $\mathbf{b}_x$  для  $x \in \{z, i, f, o\}$  — веса LSTM.

При переходе от LSTM к ODE-LSTM в уравнения выше вносятся следующие корректировки:

$$\begin{aligned}
(\mathbf{c}_i, \mathbf{h}'_i) &= \text{LSTM}(\theta_i, (\mathbf{c}_{i-1}, \mathbf{h}_{i-1}), x_i) \\
\mathbf{h}_i &= \text{ODESolve}(f_\theta, \mathbf{h}_{i-1}, \mathbf{h}'_i, t_i - t_{i-1}) \\
\mathbf{o}_i &= \mathbf{h}_i \mathbf{W}_{\text{output}} + b_{\text{output}}
\end{aligned}$$

Решение дифференциального уравнения получается численно, с использованием методов Рунге-Кутты четвёртого порядка. Методы реализованы с использованием обыкновенных линейных преобразований, используется стандартное обратное распространение ошибки.

### 2.2.3 LMU

Основными проблемами ODE-RNN подходов являются затухание и взрыв градиентов и, как следствие, невозможность обрабатывать длинные последовательности входных данных. В статье [12] доказаны теоремы о затухании градиентов в ODE-RNN и об отсутствии аналогичных проблем в ODE-LSTM. Определим модель LMU LSTM и покажем, что она также устойчива к затуханиям и взрывам градиентов.

Основной компонент LMU — ячейка памяти, ортогонализирующая непрерывную по времени историю изменения сигнала  $u(t) \in \mathbb{R}$ , по скользящему окну  $\theta \in \mathbb{R}_{>0}$ . Ячейка построена по аналогии с линейной функцией переноса для непрерывной задержки,  $F(s) = e^{-\theta s}$ , которую наилучшим образом аппроксимируют  $d$  связанных ОДУ:

$$\theta \dot{\mathbf{c}}(t) = \mathbf{A} \mathbf{c}(t) + \mathbf{b} u(t),$$

где  $\mathbf{c}(t) \in \mathbb{R}^d$  вектор состояния долгосрочной памяти размерности  $d$ . Наилучшие матрицы  $(\mathbf{A}, \mathbf{b})$ , получаются при использовании аппроксимантов Паде [44]:

$$\begin{aligned}
\mathbf{A} &= [a]_{ij} \in \mathbb{R}^{d \times d}, \quad a_{ij} = (2i+1) \begin{cases} -1 & i < j \\ (-1)^{i-j+1} & i \geq j \end{cases} \\
\mathbf{b} &= [b]_i \in \mathbb{R}^{d \times 1}, \quad b_i = (2i+1)(-1)^i, \quad i, j \in [0, d-1]
\end{aligned}$$

Основное свойство данной динамической системы заключается в том, что  $\mathbf{c}$  представляет собой скользящее окно  $u$  по полиномам Лежандра степени до  $d-1$ :

$$u(t - \theta') \approx \sum_{i=0}^{d-1} \mathcal{P}_i\left(\frac{\theta'}{\theta}\right) c_i(t), \quad 0 \leq \theta' \leq \theta, \quad \mathcal{P}_i(r) = (-1)^i \sum_{j=0}^i \binom{i}{j} \binom{i+j}{j} (-r)^j$$



где  $\mathcal{P}_i(r)$  — сдвинутые полиномы Лежандра. Таким образом может быть получена единственная и оптимальная декомпозиция, в которой  $\mathbf{c}$  отвечает вычислениям в окнах размера  $\theta$ , спроецированным на  $d$  ортогональных базисных функций.

$$\mathbf{h}_t = \tanh(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_c \mathbf{c}_t)$$

$$u_t = \mathbf{e}_x^T \mathbf{x}_t + \mathbf{e}_h^T \mathbf{h}_{t-1} + \mathbf{e}_c^T \mathbf{c}_{t-1}$$

В данных уравнениях  $\mathbf{W}_x, \mathbf{W}_h, \mathbf{W}_c$  — матрицы весов линейных слоёв LSTM,  $\mathbf{e}, \mathbf{e}_h, \mathbf{e}_m$  — векторы, кодирующие соответствующие входы LSTM.

Определив модель LMU, обратим внимание на градиенты. Так как одной из основных проблем подходов ODE-RNN является затухание градиентов, покажем, что модель LMU устойчива к затуханиям и взрывам градиентов.

**Теорема 1** При нормировке градиента на  $\frac{4}{3}\|A\|_\infty$ , сеть LMU LSTM не подвержена затуханию или взрыву градиентов.

**Доказательство.**

Рассмотрим градиент состояния долгой кратковременной памяти (состояние памяти):

$$\begin{aligned} \frac{\partial \mathbf{c}_{t+1}}{\partial \mathbf{c}_t} &= \frac{\partial (\mathbf{A} \mathbf{c}_t + \mathbf{b} u_{t+1})}{\partial \mathbf{c}_t} \\ &= \frac{\partial \mathbf{A}}{\partial \mathbf{c}_t} \mathbf{c}_t + \mathbf{A} \frac{\partial \mathbf{c}_t}{\partial \mathbf{c}_t} + \frac{\partial \mathbf{b}}{\partial \mathbf{c}_t} u_{t+1} + \mathbf{b} \frac{\partial u_{t+1}}{\partial \mathbf{c}_t} \end{aligned} \quad (1)$$

Так как матрица  $\mathbf{A}$  и вектор  $\mathbf{b}$  задаются однозначно аппроксимантами Паде, то в уравнении 1 выполняется

$$\frac{\partial \mathbf{A}}{\partial \mathbf{c}_t} = 0, \quad \frac{\partial \mathbf{b}}{\partial \mathbf{c}_t} = 0.$$

Таким образом, получаем следующее выражение:

$$\frac{\partial (\mathbf{A} \mathbf{c}_t + \mathbf{b} u_{t+1})}{\partial \mathbf{c}_t} = \mathbf{A} \mathbf{e} + \mathbf{b} \frac{\partial u_{t+1}}{\partial \mathbf{c}_t} \quad (2)$$

Рассмотрим второй член уравнения 2:

$$\begin{aligned} \frac{\partial u_{t+1}}{\partial \mathbf{c}_t} &= \frac{\partial (\mathbf{e}_x^T \mathbf{x}_{t+1} + \mathbf{e}_h^T \mathbf{h}_t + \mathbf{e}_c^T \mathbf{c}_t)}{\partial \mathbf{c}_t} \\ &= \frac{\partial \mathbf{e}_x^T}{\partial \mathbf{c}_t} \mathbf{x}_{t+1} + \mathbf{e}_x^T \frac{\partial \mathbf{x}_{t+1}}{\partial \mathbf{c}_t} + \frac{\partial \mathbf{e}_h^T}{\partial \mathbf{c}_t} \mathbf{h}_t + \mathbf{e}_h^T \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} + \frac{\partial \mathbf{e}_c^T}{\partial \mathbf{c}_t} \mathbf{c}_t + \mathbf{e}_c^T \frac{\partial \mathbf{c}_t}{\partial \mathbf{c}_t} \end{aligned} \quad (3)$$

Векторы  $\mathbf{e}_x^T, \mathbf{e}_h^T, \mathbf{e}_c^T$  задаются независимо от  $\mathbf{c}_t$ , а  $\mathbf{x}_t$ , как входные данные также не зависят от состояния памяти, поэтому в уравнении 3:

$$\frac{\partial \mathbf{e}_x^T}{\partial \mathbf{c}_t} = \frac{\partial \mathbf{e}_h^T}{\partial \mathbf{c}_t} = \frac{\partial \mathbf{e}_c^T}{\partial \mathbf{c}_t} = \frac{\partial \mathbf{x}_{t+1}}{\partial \mathbf{c}_t} = 0$$

И само уравнение относительно  $u_{t+1}$  принимает следующий вид:

$$\frac{\partial u_{t+1}}{\partial \mathbf{c}_t} = \mathbf{e}_h^T \frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} + \mathbf{e}_c^T \frac{\partial \mathbf{c}_t}{\partial \mathbf{c}_t} \quad (4)$$

Рассмотрим градиент скрытого состояния  $\mathbf{h}_t$ :

$$\frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} = \frac{\partial \tanh(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_c \mathbf{c}_t)}{\partial \mathbf{c}_t}$$

Распишем скрытое состояние  $\mathbf{h}_{t-1}$ . Оно так же не зависит от  $\mathbf{c}_t$ :

$$\mathbf{h}_{t-1} = \tanh(\mathbf{W}_x \mathbf{x}_{t-1} + \mathbf{W}_h \mathbf{h}_{t-2} + \mathbf{W}_c \mathbf{c}_{t-1}) \Rightarrow \frac{\partial \mathbf{h}_{t-1}}{\partial \mathbf{c}_t} = 0$$

Матрицы линейных отображений задаются независимо от  $\mathbf{c}_t$ :

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{c}_t} = \frac{\partial \mathbf{W}_x}{\partial \mathbf{c}_t} = \frac{\partial \mathbf{W}_h}{\partial \mathbf{c}_t} = \frac{\partial \mathbf{W}_c}{\partial \mathbf{c}_t} = 0$$

Тогда производная скрытого состояния принимает вид:

$$\frac{\partial \mathbf{h}_t}{\partial \mathbf{c}_t} = (1 - \tanh^2(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_c \mathbf{c}_t)) \cdot \mathbf{W}_c \frac{\partial \mathbf{c}_t}{\partial \mathbf{c}_t} \quad (5)$$

Подставим полученное в 5 значение в равенство 4, а затем в 2:

$$\frac{\partial u_{t+1}}{\partial \mathbf{c}_t} = \mathbf{e}_h^T (1 - \tanh^2(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_c \mathbf{c}_t)) \mathbf{W}_c \mathbf{e} + \mathbf{e}_c^T \mathbf{e} \quad (6)$$

$$\frac{\partial \mathbf{c}_{t+1}}{\partial \mathbf{c}_t} = \mathbf{A} \mathbf{e} + \mathbf{b} [\mathbf{e}_h^T (1 - \tanh^2(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_c \mathbf{c}_t)) \mathbf{W}_c \mathbf{e} + \mathbf{e}_c^T \mathbf{e}] \quad (7)$$

Далее примем во внимание следующие факты:

$$\tanh^2(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_c \mathbf{c}_t) \in [0, 1]$$

$\mathbf{W}_c$ ,  $\mathbf{e}_c$ ,  $\mathbf{e}_h$  инициализируются числами, близкими к нулю.

Тогда можно пренебречь вторым слагаемым и сказать, что

$$\frac{\partial \mathbf{c}_{t+1}}{\partial \mathbf{c}_t} \approx \mathbf{A} \mathbf{e}$$

Для того, чтобы можно было сказать об отсутствии затухания и взрыва градиентов,  $\|\frac{\partial \mathbf{c}_{t+1}}{\partial \mathbf{c}_t}\|$  должна быть меньше 1, но много больше 0.

Рассмотрим  $\|\frac{\partial \mathbf{c}_{t+1}}{\partial \mathbf{c}_t}\| \approx 0.75$

$$\mathbf{Ae} = \begin{bmatrix} \sum_{j=0}^d a_{1,j} \\ \sum_{j=0}^d a_{2,j} \\ \vdots \\ \sum_{j=0}^d a_{d,j} \end{bmatrix}$$

Для вектора производной состояния долгой краткосрочной памяти рассмотрим векторную норму  $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$ . Для вектора  $\mathbf{Ae}$  она совпадает с  $\|\mathbf{A}\|_\infty$ .

Таким образом, норма градиента  $\|\frac{\partial \mathbf{c}_{t+1}}{\partial \mathbf{c}_t}\|_\infty \approx \|\mathbf{Ae}\|_\infty = \|\mathbf{A}\|_\infty$ .

При нормировке градиента на  $\frac{4}{3}\|\mathbf{A}\|_\infty$  мы получим результат

$$\frac{1}{\|\mathbf{A}\|_\infty} \|\frac{\partial \mathbf{c}_{t+1}}{\partial \mathbf{c}_t}\|_\infty \approx 0.75$$

■

### 3 Предлагаемый метод

В рамках данной работы предлагаются следующие части исследования:

1. Исследование возможных практических приложений непрерывности времени в задачах декодирования сигналов.
2. Оценка качества моделей на реальных данных.
3. Извлечение непрерывных представлений из предложенных моделей, оценка их применимости для дальнейшего использования.

Для рассмотрения первых двух пунктов предлагается провести вычислительный эксперимент на двух задачах классификации сигналов. Результаты работы моделей NeuralCDE, ODE LSTM, LMU LSTM необходимо сравнить с существующими методами классификации сигналов. Также предлагается рассмотреть возможность использования данных моделей на данных с привнесённой в записи нерегулярностью. Провести сравнение качества с результатами, полученными для регулярных данных. Сделать выводы об области применимости подхода к классификации, использующего нейронные обыкновенные дифференциальные уравнения. Сравнения предлагается проводить по среднему качеству моделей за 5 запусков обучения, используя внешние критерии качества моделей: точность и F1-score.

Основная исследовательская задача — получение непрерывного представления сигнала. Для LMU LSTM непрерывное представление сигнала — разложение на полиномы Лежандра с коэффициентами, получаемыми из состояния долгой краткосрочной памяти. Для ODE LSTM и Neural CDE предлагается рассмотреть скрытое состояние как возможное непрерывное представление сигнала, так как по построению алгоритмов именно скрытое состояние предполагается как непрерывная функция. Необходимо оценить возможность использования полученных представлений как функции непрерывного представления сигнала. Предлагаются следующие параметры оценки полученных представлений:

1. скрытое состояние обученной модели можно извлечь и использовать самостоятельно как представление сигнала,
2. полученное представление сохраняет форму приближаемого процесса,
3. функция представления позволяет получить значения в произвольный момент времени.

Скрытые состояния, отвечающие всем указанным требованиям можно считать непрерывным представлением сигнала и использовать для дальнейшей работы в различных задачах декодирования сигналов для построения новых признаков или фазовых пространств.

По результатам данных вычислительных экспериментов предлагается сделать выводы об оптимальной архитектуре модели для решения задач классификации сигналов и построения непрерывного представления сигнала.

## 4 Вычислительный эксперимент

### 4.1 Постановка эксперимента

#### Цель эксперимента.

1. Сравнить качество работы алгоритмов на основе NeuralODE с распространёнными на практике подходами к классификации сигналов.
2. Солучить непрерывное по времени представление сигнала.

**Постановка эксперимента.** Для запуска вычислительного эксперимента использовались два набора данных, описанных в следующей секции. Для реализации алгоритмов была использована библиотека PyTorch [45], для автоматизации процесса обучения — Lightning [46].

Каждый набор данных делился на две части: обучающую и валидационную выборки в соотношении 80:20. Для запусков экспериментов на нерегулярных данных производилась модификация исходных данных. Для каждой последовательности случайным образом удалялось 20% точек. Таким образом, шаг по времени между точками становился нерегулярным. Для удобства вычислений длины всех последовательностей сохранялись одинаковыми, что также позволило получить качество классических моделей на нерегулярных данных.

Таким образом для каждого набора данных было обучено по три модели с использованием Neural ODE. Для набора данных потенциалов P300 также был обучен классификатор на римановской геометрии и свёрточная нейросеть для сравнения качества.

**Условия эксперимента.** Все модели обучались с помощью библиотеки Lightning. Для обучения моделей использовался алгоритм оптимизации Adam. Оптимизационная задача ставилась относительно критерия перекрёстной энтропии. Параметры оптимизации и количество эпох подбирались экспериментально для каждой из моделей. Для каждой модели в процессе обучения сохранялись три лучших набора параметров, отобранные по F1-score на валидационной выборке. Все запуски обучения производились на CPU.

### 4.2 Экспериментальные данные

Вычислительные эксперименты проводились на следующих наборах данных:

1. Датасет потенциалов P300.
2. Датасет UC1 активности людей.

#### 4.2.1 Набор данных потенциалов DemonsP300

**Участники.** Набор данных записан на 60 здоровых людях (23 мужчины), никогда ранее не встречавшихся с нейроинтерфейсами. Возраст участников от 19 до 45, средний — 28 лет. Все респонденты соответствовали необходимым условиям по здоровью и состоянию и подписали информированное согласие на эксперимент.

**Стимулы и запись ЭЭГ.** ЭЭГ записывались с помощью энцефалографа NVX-52 (МКС, Зеленоград) с частотой 500 Гц. Для записи использовались 8 губчатых электродов (Cz, P3, P4, PO3, POz, PO4, O1, O2), см. рис. 1. Стимулы предъявлялись с помощью шлема HTC Vive Pro VR.

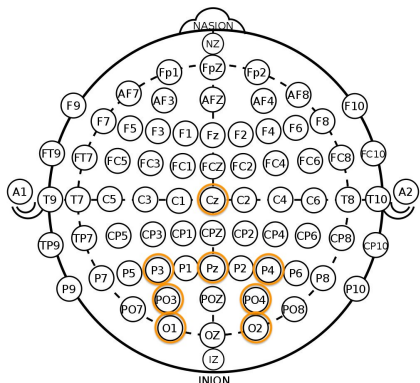


Рис. 1: Расположение электродов

Рис. 2: Сцена из VR игры с енотами и демонами

Участникам эксперимента была предложена игра в виртуальной реальности с нейроуправлением, основанном на классификации потенциалов P300. Во время игры на этапе обучения производилась тренировка логистической регрессии, во время игры на этапе обратной связи НКИ отрезки ЭЭГ классифицировались с использованием обученной на участнике логистической регрессии. Нейрокомпьютерный интерфейс был встроен в игровой сюжет, в котором игрок должен был кормить енотов и защищать их от демонов.

На этапе обучения участнику предъявлялось пять упорядоченных визуальных стимулов — анимированных енотов (см. рис.2). Активации стимулов — прыжки енотов в случайном порядке. Случайный порядок введен для того, чтобы предотвратить ожидание активации. Обучение составляло пять игровых актов (подробная структура приведена в следующем параграфе). В каждом акте участникам указывался конкретный енот, на котором они должны были концентрироваться: к примеру, считать количество прыжков. Каждый акт состоял из 10 активаций каждой цели, после чего указанный в начале акта енот получал еду. По итогу этапа обучения получалось 50 целевых (содержащих потенциал P300) и 450 нецелевых отрезков ЭЭГ для обучения бинарного классификатора.

Во время этапа обратной связи НКИ, стимулы предъявлялись как анимированные демоны. Выбранный демон, при достаточной фокусировке игрока, уничтожался в конце акта, оставшиеся в живых — подходили ближе. Как только демон подходил достаточно близко к игроку, он хватал ближайшего к нему енота и убегал. В любом случае после исчезновения с экрана одного демона, на его месте, в начальной позиции, появлялся новый демон, чтобы количество стимулов оставалось постоянным.

Все данные электроэнцефалограмм, полученных во время игры были записаны для дальнейшего анализа.

**Термины и структура данных.** Введём термины, используемые в дальнейшем:

- Эпоха — секундный отрезок ЭЭГ, отсчитывающийся после активации стимула.

- Раунд (блок) — последовательность эпох с общим целевым стимулом, где каждый стимул активируется один раз, порождая одну эпоху. Активация стимулов происходит в случайном порядке.
- Акт — последовательность раундов с общим целевым стимулом, после которого принимается решение о целевом стимуле. Количество актов в экспериментах — 5 при обучении и 10 при игре.
- Запись (игра) — последовательность актов, полученная от одного человека за одну сессию записи (одну игру).
- Датасет — коллекция записей, сделанных на одном и том же оборудовании с одинаковыми визуальными активациями.

Пусть  $E$  — количество каналов записи ЭЭГ,  $N$  — число отсчётов времени в одной эпохе.

Обычно в парадигме P300 представлено несколько стимулов (в рассматриваемом датасете — 5) и основной задачей является определение стимула, выбранного пользователем — задача мультиклассовой классификации.

На данный момент мультиклассовая задача решается как агрегация бинарных задач, т.е.:

$$P_j^{mult} = \frac{\sum_{i=1}^M P_{j,i}^{bin}}{M}$$

$j$  — рассматриваемый класс мультиклассовой классификации,

$M$  — количество предъявлений одного стимула в акте игры,

$s$  — число визуальных стимулов, количество классов мультиклассовой классификации,

$P_{j,i}^{bin}$  — вероятность того, что стимул  $j$  целевой при предъявлении  $i$ .

Итоговый стимул выбирается как стимул с наибольшей вероятностью  $P_j^{mult}$ .

В данной работе мы рассматриваем решение бинарной задачи, описанной в разделе «Постановка задачи». В данном наборе данных как обучающая, так и тестовая выборки являются несбалансированными, так как каждый стимул активируется одинаковое количество раз, но только один из них является целевым. Таким образом, получается соотношение классов 1 к  $(s - 1)$ , где  $s$  — количество стимулов.

Итоговое количество записей в датасете: 56540.

Данные можно получить по ссылке или воспользоваться датасетом DemonsP300 библиотеки moabb [47].

#### 4.2.2 Набор данных по видам активности UCI

Рассматривается набор данных «Данные о положении для восстановления позы» [38]. Данные использовались авторами для наблюдения за пожилыми людьми и детектирования их падений. Каждый участник эксперимента надевал четыре акселерометра: на обе лодыжки, на пояс и на грудь. Каждому акселерометру был присвоен уникальный идентификатор и каждый передавал данные о положении как три координаты. Данные можно получить по ссылке.

Количество последовательностей в датасете: 9711.

Для каждого отсчёта времени известен тип активности из следующих вариантов:

1. ходьба (walking)
2. падение (falling)
3. принятие положения лёжа (lying down)
4. положение лёжа (lying)
5. принятие положения сидя (sitting down)
6. положение сидя (sitting)
7. подъем из положения лёжа (standing up from lying)
8. на четвереньках (on all fours)
9. положение сидя на полу (sitting on the ground)
10. подъём из положения сидя (standing up from sitting)
11. подъем из положения сидя на земле (standing up from sitting on the ground)

Каждый элемент представлен набором следующих данных:

Содержание атрибута	Пример значения
идентификатор записи "номер человека, номер записи"	A05
идентификатор акселерометра	010-000-024-033
отметка времени (уникальные значения)	633790226052361498
дата	25.10.2010 09:54:33:013
координата акселерометра по x	4.364077091217041
координата акселерометра по y	1.7676914930343628
координата акселерометра по z	1.2999963760375977
тип активности	falling

Распределение классов:

Класс	Количество записей
ходьба	32710
падение	2973
принятие лежачего положения	6168
положение лёжа	54480
принятие сидячего положения	1706
сидение	27244
подъем из положения лёжа	18361
на четвереньках	5210
сидение на полу	11779
подъём из положения сидя	1381
подъем из положения сидя на земле	2848



## 4.3 Результаты

### 4.3.1 Эксперимент на данных P300

В таблицах 2, 3 представлены результаты эксперимента на данных потенциалов P300 для регулярного и нерегулярного шага по времени соответственно. По данным таблиц можно сказать, что модели с непрерывным временем показывают лучшее качество, нежели стандартные подходы даже на регулярных данных, следовательно, использование непрерывного времени дает преимущества не только при применении на данных с нерегулярным шагом по времени. Однако можно заметить, что скорость обучения и предсказания для подходов с дискретным и непрерывным временем существенно отличается. Тем не менее, на нерегулярных данных различие данных подходов становится заметнее: качество стандартных подходов заметно ухудшается, в то время как точность и F1-score для Neural ODE моделей остаётся на том же уровне.

Модель	Точность на обучении	Точность на тесте	F1-score на обучении	F1-score на тесте
CDE	0.83	<b>0.81</b>	0.82	<b>0.73</b>
ODE-LSTM	0.76	0.72	0.70	0.67
LMU	0.77	0.76	0.71	0.70
EEGNet	0.81	0.71	0.80	0.69
PyRiemann	0.74	0.51	0.70	0.53

Таблица 2: Результаты вычислительного эксперимента на данных потенциалов P300

Модель	Точность на обучении	Точность на тесте	F1-score на обучении	F1-score на тесте
CDE	0.82	<b>0.81</b>	0.80	<b>0.72</b>
ODE-LSTM	0.74	0.72	0.71	0.67
LMU	0.76	0.74	0.71	0.69
EEGNet	0.67	0.59	0.63	0.58
PyRiemann	0.61	0.48	0.59	0.46

Таблица 3: Результаты вычислительного эксперимента на нерегулярных данных потенциалов P300

### 4.3.2 Эксперимент на данных UCI

Эксперимент на данных об активности проводился с целью подтвердить выводы, сделанные на данных о потенциалах P300, а также оценить возможность рассматриваемых моделей производить классификацию в каждый отсчёт времени, а не только всего временного ряда в целом. В таблицах 4, 5 представлены результаты эксперимента на данных об активности для регулярного и нерегулярного шага по времени соответственно. Модели показывают достаточно высокое качество классификации на рассматриваемом наборе данных. Модель Neural CDE при определенных модификациях также позволяет получать предсказания класса активности с большей частотой, нежели частота получения реализаций сигнала.

Модель	Точность на обучении	Точность на тесте	F1-score на обучении	F1-score на тесте
CDE	0.77	0.71	0.80	0.69
ODE-LSTM	0.76	0.69	0.75	0.70
LMU	0.81	<b>0.73</b>	0.80	<b>0.72</b>

Таблица 4: Результаты вычислительного эксперимента на данных об активности

Модель	Точность на обучении	Точность на тесте	F1-score на обучении	F1-score на тесте
CDE	0.79	0.71	0.77	0.68
ODE-LSTM	0.77	0.70	0.75	0.71
LMU	0.82	<b>0.75</b>	0.80	<b>0.71</b>

Таблица 5: Результаты вычислительного эксперимента на нерегулярных данных об активности

### 4.3.3 Непрерывное представление времени

В ходе эксперимента было установлено, что единственная модель, скрытое состояние которой удовлетворяет всем требованиям, описанным в разделе «Предлагаемый метод» — модель NeuralCDE. Скрытые состояния моделей ODE LSTM и LMU LSTM не удовлетворяют третьему условию: функции данных состояний не позволяют получить значение представления сигнала в произвольный момент времени.

Для скрытого состояния  $\mathbf{z}(t)$  Neural CDE были получены следующие результаты.

На графиках 3, 4 приведены визуализации исходного сигнала и скрытого состояния NeuralCDE. Можно увидеть, что форма скрытого состояния хорошо описывает пик P300 как на регулярных, так и на нерегулярных данных. Скрытое состояние не является точным описанием процесса мозговой активности, но сохраняет его ожидаемую форму. Следовательно, данное состояние можно использовать в дальнейшем для получения непрерывного отображения сигнала.

### 4.3.4 Выводы

Наиболее обоснованным применением методов, использующих непрерывное время, являются случаи данных, имеющих пропуски или переменный шаг по сетке времени. Также данные методы разумно применять в случаях, когда требуется получить прогноз целевой переменной в те моменты времени, для которых нет сигнала: к примеру, прогнозировать класс движения один раз в две секунды, когда сигнал приходит один раз в четыре. Однако методы, использующие дискретное время, требуют меньше вычислительных ресурсов, так как не требуют решать дифференциальные уравнения и могут быть параллелизованы. В случаях задач, требующих быстрых вычислений в заданные регулярные моменты времени, дискретные методы остаются оптимальным выбором несмотря на небольшие потери в качестве декодирования сигнала, показанные в таблице 2.

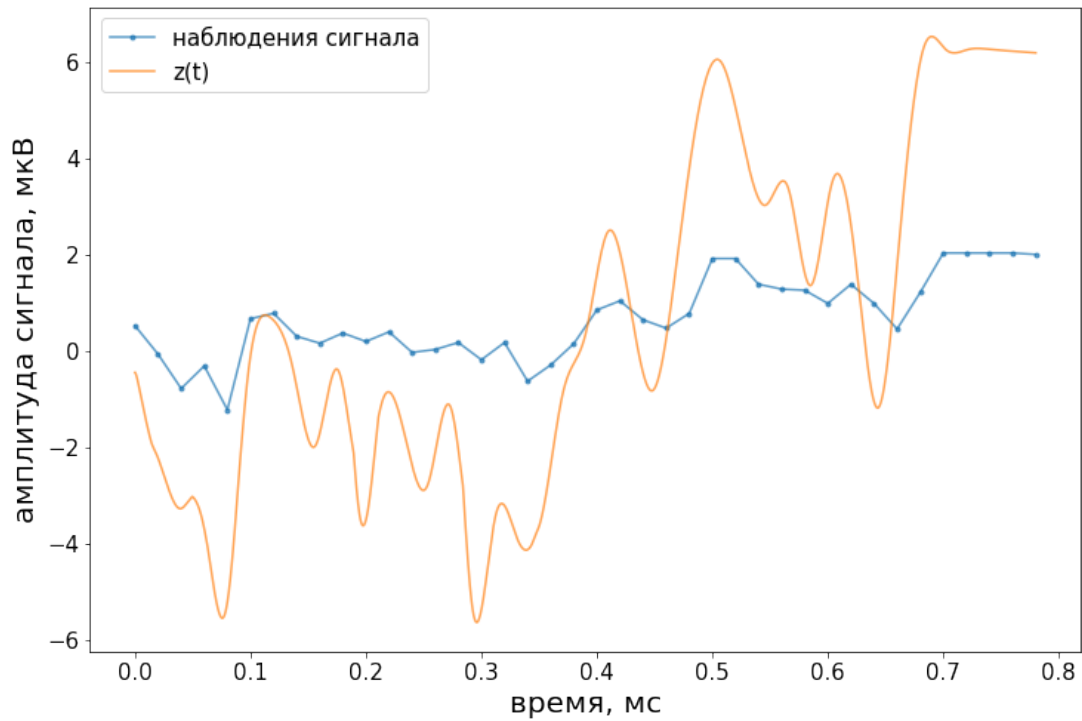


Рис. 3: Скрытое состояние NeuralCDE и исходный сигнал для регулярных данных

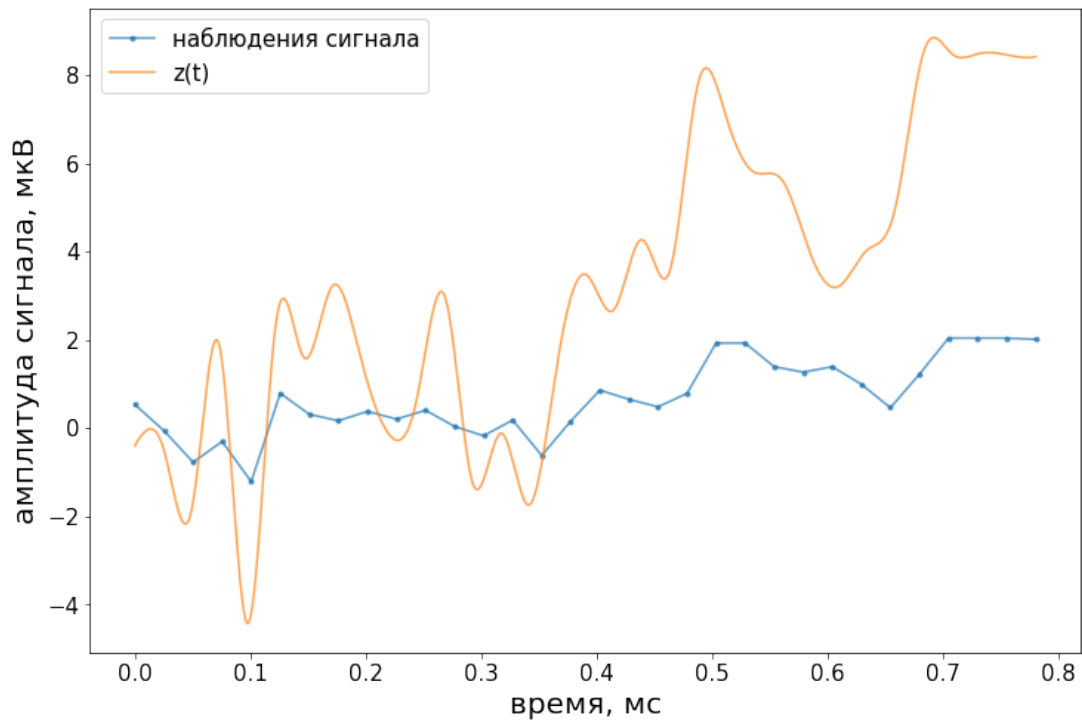


Рис. 4: Скрытое состояние NeuralCDE и исходный сигнал для нерегулярных данных

#### 4.3.5 Воспроизводимость

Результаты работы можно увидеть на [github](#).

## Заключение

В рамках работы над магистерской диссертацией были получены как теоретические, так и практические результаты. Было рассмотрено несколько моделей с непрерывным представлением времени. На реальных задачах показано, что данные модели превосходят по качеству стандартные подходы, применимы к временным рядам переменной длины и переменного шага по времени. Из использованных моделей было извлечено непрерывное представление сигнала, соответствующее ожиданиям по форме полученной функции. Данное непрерывное представление предоставляет возможность передискретизации сигнала и дальнейшего исследования признаков и фазовых пространств сигналов.

Основными результатами работы являются:

1. теорема о градиентах для LMU LSTM,
2. непрерывное представление сигнала по времени как скрытое состояние Neural CDE,
3. вычислительный эксперимент, демонстрирующий преимущества моделей, использующих непрерывное представление времени.

## Список литературы

- [1] Neural Ordinary Differential Equations / Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David K Duvenaud // Advances in Neural Information Processing Systems / Ed. by S. Bengio, H. Wallach, H. Larochelle et al. — Vol. 31. — Curran Associates, Inc., 2018. — Access mode: <https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf>.
- [2] GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series / Edward De Brouwer, J. Simm, A. Arany, Y. Moreau // NeurIPS. — 2019.
- [3] Vid-ODE: Continuous-Time Video Generation with Neural Ordinary Differential Equation / Sunghyun Park, Kangyeol Kim, Junsoo Lee et al. — 2021. — 2010.08188.
- [4] Go with the Flow: Adaptive Control for Neural ODEs / Mathieu Chalvidal, Matthew Ricci, Rufin VanRullen, Thomas Serre. — 2021. — 2006.09545.
- [5] Chen Ricky T. Q., Amos Brandon, Nickel Maximilian. Learning Neural Event Functions for Ordinary Differential Equations. — 2021. — 2011.03902.
- [6] Continuous-in-Depth Neural Networks / Alejandro Queiruga, N. Erichson, Dane Taylor, Michael Mahoney. — 2020. — 08.
- [7] Rubanova Yulia, Chen Ricky T. Q., Duvenaud David K. Latent Ordinary Differential Equations for Irregularly-Sampled Time Series // Advances in Neural Information Processing Systems / Ed. by H. Wallach, H. Larochelle, A. Beygelzimer et al. — Vol. 32. — Curran Associates, Inc., 2019. — Access mode: <https://proceedings.neurips.cc/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf>.
- [8] Neural Controlled Differential Equations for Irregular Time Series / Patrick Kidger, James Morrill, James Foster, Terry J. Lyons // NeurIPS. — 2020. — Access mode: <https://proceedings.neurips.cc/paper/2020/hash/4a5876b450b45371f6cfe5047ac8cd45-Abstract.html>.
- [9] Morrill James, Salvi Christopher, Kidger Patrick et al. Neural Rough Differential Equations for Long Time Series. — 2021. — 2009.08295.
- [10] Deep Residual Learning for Image Recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — P. 770–778.
- [11] Yildiz Cagatay, Heinonen Markus, Lahdesmaki Harri. ODE2VAE: Deep generative second order ODEs with Bayesian neural networks // Advances in Neural Information Processing Systems / Ed. by H. Wallach, H. Larochelle, A. Beygelzimer et al. — Vol. 32. — Curran Associates, Inc., 2019. — Access mode: <https://proceedings.neurips.cc/paper/2019/file/99a401435dcb65c4008d3ad22c8cdad0-Paper.pdf>.
- [12] Lechner Mathias, Hasani Ramin. Learning Long-Term Dependencies in Irregularly-Sampled Time Series. — 2020. — 2006.04418.

- [13] Hochreiter Sepp, Schmidhuber Jürgen. Long Short-term Memory // Neural computation. — 1997. — 12. — Vol. 9. — P. 1735–80.
- [14] Voelker Aaron, Kajić Ivana, Eliasmith Chris. Legendre Memory Units: Continuous-Time Representation in Recurrent Neural Networks // Advances in Neural Information Processing Systems / Ed. by H. Wallach, H. Larochelle, A. Beygelzimer et al. — Vol. 32. — Curran Associates, Inc., 2019. — Access mode: <https://proceedings.neurips.cc/paper/2019/file/952285b9b7e7a1be5aa7849f32ffff05-Paper.pdf>.
- [15] Legendre Adrien Marie. Recherches sur l’attraction des spheroides homogenes // Memoires de mathematique et de physique : pres. à l’Academie Royale des Sciences, par divers savans, et lûs dans ses assemblees. — 2007. — Vol. 1785. — P. 411 – 434.
- [16] LeCun Yann, Cortes Corinna. MNIST handwritten digit database. — 2010. — Access mode: <http://yann.lecun.com/exdb/mnist/>.
- [17] Teplan Michal. Fundamental of EEG Measurement // MEASUREMENT SCIENCE REVIEW. — 2002. — 01. — Vol. 2.
- [18] Decoding unconstrained arm movements in primates using high-density electrocorticography signals for brain-machine interface use / Kejia Hu, Mohsen Jamali, Ziev B. Moses et al. // Scientific Reports. — 2018. — jul. — Vol. 8, no. 1.
- [19] Song Yilin, Wang Yao, Viventi Jonathan. Unsupervised Learning of Spike Patterns for Seizure Detection and Wavefront Estimation of High Resolution Micro Electrocorticographic Data // IEEE Transactions on NanoBioscience. — 2017. — sep. — Vol. 16, no. 6. — P. 418–427.
- [20] Loza Carlos A., Principe Jose C. Unsupervised robust detection of behavioral correlates in ECoG // 2017 8th International IEEE/EMBS Conference on Neural Engineering (NER). — IEEE, 2017. — may.
- [21] Eliseyev Andrey, Aksenova Tetiana. Penalized Multi-Way Partial Least Squares for Smooth Trajectory Decoding from Electrocorticographic (ECoG) Recording // PLOS ONE. — 2016. — may. — Vol. 11, no. 5. — P. e0154878.
- [22] Separate spatial and temporal frequency tuning to visual motion in human MT measured with ECoG / Anna Gaglianese, Ben M. Harvey, Mariska J. Vansteensel et al. // Human Brain Mapping. — 2016. — sep. — Vol. 38, no. 1. — P. 293–307.
- [23] David T Bundy Mrinal Pahwa Nicholas Szrama Eric C Leuthardt. Decoding three-dimensional reaching movements using electrocorticographic signals in humans // Journal of Neural Engineering. — 2016. — feb. — Vol. 13, no. 2. — P. 026021.
- [24] Brain-machine interface to control a prosthetic arm with monkey ECoGs during periodic movements / Soichiro Morishita, Keita Sato, Hidenori Watanabe et al. // Frontiers in Neuroscience. — 2014. — dec. — Vol. 8.
- [25] Goncharenko V., Grigoryan R., Samokhina A. Raccoons vs Demons: multiclass labeled P300 dataset. — 2020. — 2005.02251.

- [26] Chaudhary Ujwal, Mrachacz-Kersting Natalie, Birbaumer Niels. Neuropsychological and neurophysiological aspects of brain-computer-interface (BCI) control in paralysis // *The Journal of Physiology*. — 2020.
- [27] Objective and Subjective Evaluation of Online Error Correction during P300-Based Spelling / Perrin Margaux, Maby Emmanuel, Daligault Sébastien et al. // *Adv. in Hum.-Comp. Int.* — 2012. — Jan. — Vol. 2012. — Access mode: <https://doi.org/10.1155/2012/578295>.
- [28] Adapting the P300-Based Brain-Computer Interface for Gaming: A Review / A. Y. Kaplan, S. L. Shishkin, I. P. Ganin et al. // *IEEE Transactions on Computational Intelligence and AI in Games*. — 2013. — Vol. 5, no. 2. — P. 141–149.
- [29] Recreational Applications of OpenViBE: Brain Invaders and Use-the-Force / Anton Andreev, Alexandre Barachant, Fabien Lotte, Marco Congedo // *Brain-Computer Interfaces 2: Technology and Applications* / Ed. by Maureen Clerc, Laurent Bougrai, Fabien Lotte. — John Wiley, 2016. — Aug. — Vol. chap. 14. — P. 241–257. — Access mode: <https://hal.archives-ouvertes.fr/hal-01366873>.
- [30] High-performance neuroprosthetic control by an individual with tetraplegia / Jennifer L Collinger, Brian Wodlinger, John E Downey et al. // *The Lancet*. — 2013. — Vol. 381, no. 9866. — P. 557–564.
- [31] Enhancing classification accuracy of fNIRS-BCI using features acquired from vector-based phase analysis / Hammad Nazeer, Noman Naseer, Rayyan Azam Khan et al. // *Journal of Neural Engineering*. — 2020. — Vol. 17, no. 5. — P. 056025.
- [32] Brain-computer interface using fMRI: spatial navigation by thoughts / Seung-Schik Yoo, Ty Fairney, Nan-Kuei Chen et al. // *Neuroreport*. — 2004. — Vol. 15, no. 10. — P. 1591–1595.
- [33] Toward neuroadaptive support technologies for improving digital reading: a passive BCI-based assessment of mental workload imposed by text difficulty and presentation speed during reading / Lena M Andreessen, Peter Gerjets, Detmar Meurers, Thorsten O Zander // *User Modeling and User-Adapted Interaction*. — 2020. — P. 1–30.
- [34] Enhancing human-computer interaction with input from active and passive brain-computer interfaces / Thorsten O Zander, Christian Kothe, Sabine Jatzev, Matti Gaertner // *Brain-computer interfaces*. — Springer, 2010. — P. 181–199.
- [35] Model analyses of visual biofeedback training for EEG-based brain-computer interface / Chih-Wei Chen, Ming-Shaung Ju, Yun-Nien Sun, Chou-Ching K Lin // *Journal of computational neuroscience*. — 2009. — Vol. 27, no. 3. — P. 357.
- [36] Polich John. Updating P300: an integrative theory of P3a and P3b // *Clinical neurophysiology*. — 2007. — Vol. 118, no. 10. — P. 2128–2148.
- [37] Patel Salil H., Azzam Pierre N. Characterization of N200 and P300: Selected Studies of the Event-Related Potential // *Int J Med Sci*. — 2005. — Vol. 2. — P. 147–154. — Access mode: <https://www.medsci.org/v02p0147.htm>.

- [38] An Agent-Based Approach to Care in Independent Living / Bostjan Kaluza, Violeta Mirchevska, Erik Dovgan et al. — 2010. — 10. — P. 177–186.
- [39] Riemannian Geometry Applied to BCI Classification / Alexandre Barachant, Stéphane Bonnet, Marco Congedo, Christian Jutten. — 2010. — 09.
- [40] EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces / Vernon Lawhern, Amelia Solon, Nicholas Waytowich et al. // Journal of Neural Engineering. — 2016. — 11. — Vol. 15.
- [41] Congedo Marco, Barachant Alexandre, Andreev Anton. A New Generation of Brain-Computer Interface Based on Riemannian Geometry // CoRR. — 2013. — Vol. abs/1310.8115. — 1310.8115.
- [42] Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b / Kai Keng Ang, Zheng Yang Chin, Chuanchu Wang et al. // Frontiers in Neuroscience. — 2012. — Vol. 6. — P. 39. — Access mode: <https://www.frontiersin.org/article/10.3389/fnins.2012.00039>.
- [43] Optimal Control of Unsteady Flows Using a Discrete and a Continuous Adjoint Approach / Angelo Carnarius, Frank Thiele, Emre Özkaya et al. // 25th System Modeling and Optimization (CSMO) / Ed. by Dietmar Hömberg, Fredi Tröltzsch. — Vol. AICT-391 of System Modeling and Optimization. — Berlin, Germany : Springer, 2011. — Sep. — P. 318–327. — Part 5: Flow Control. Access mode: <https://hal.inria.fr/hal-01347552>.
- [44] Voelker Aaron. Dynamical Systems in Spiking Neuromorphic Hardware : Ph.D. thesis / Aaron Voelker. — 2019. — 05.
- [45] PyTorch: An Imperative Style, High-Performance Deep Learning Library / Adam Paszke, Sam Gross, Francisco Massa et al. // Advances in Neural Information Processing Systems 32 / Ed. by H. Wallach, H. Larochelle, A. Beygelzimer et al. — Curran Associates, Inc., 2019. — P. 8024–8035. — Access mode: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [46] Falcon WA et al. PyTorch Lightning // GitHub. Note: <https://github.com/PyTorchLightning/pytorch-lightning>. — 2019. — Vol. 3.
- [47] Jayaram Vinay, Barachant Alexandre. MOABB: trustworthy algorithm benchmarking for BCIs // Journal of Neural Engineering. — 2018. — sep. — Vol. 15, no. 6. — P. 066011. — Access mode: <https://doi.org/10.1088/1741-2552/aadea0>.