

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Образовательная программа бакалавриата «Программная инженерия»

СОГЛАСОВАНО

УТВЕРЖДАЮ

Руководитель направления,
Лаборатория Искусственного Интеллекта,
Сбербанк


_____ К. С. Егоров
« ____ » _____ 2021 г.

Академический руководитель
образовательной программы «Программная
инженерия» профессор департамента
программной инженерии, канд. техн. наук
_____ В. В. Шилов
« ____ » _____ 2021 г.

АВТОМАТИЧЕСКАЯ РАСШИФРОВКА ЭКГ ПО ХОЛТЕРУ

Пояснительная записка

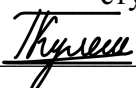
ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.01-01 81 01-1 ЛУ

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Исполнитель

студент группы БПИ197

 / П. О. Кулешова /
« ____ » _____ 2021 г.

Москва 2021

УТВЕРЖДЕН
RU.17701729.04.01-01 81 01-1

АВТОМАТИЧЕСКАЯ РАСШИФРОВКА ЭКГ ПО ХОЛТЕРУ

Пояснительная записка

RU.17701729.04.01-01 81 01-1

Листов 31

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

1.	ВВЕДЕНИЕ	3
1.1.	НАИМЕНОВАНИЕ ПРОГРАММЫ	3
1.2.	ДОКУМЕНТ, НА ОСНОВЕ КОТОРОГО ВЕДЕТСЯ РАЗРАБОТКА	3
2.	НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	4
2.1.	НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	4
2.2.	КРАТКАЯ ХАРАКТЕРИСТИКА ОБЛАСТИ ПРИМЕНЕНИЯ	4
3.	ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	5
3.1.	ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ ПРОГРАММЫ	5
3.2.	ОПИСАНИЕ АЛГОРИТМА ФУНКЦИОНИРОВАНИЯ ПРОГРАММЫ С ОБОСНОВАНИЕМ ВЫБОРА СХЕМЫ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ	5
3.2.1.	Preprocessing	5
3.2.2.	Создание выборки	9
3.2.3.	Инициализация Автоэнкодера.....	9
3.2.4.	Обучение	13
3.2.5.	Кластеризация	17
3.2.6.	Возможные взаимодействия программы с другими программами.....	19
3.3.	ОПИСАНИЕ И ОБОСНОВАНИЕ ВЫБОРА ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ	20
3.3.1.	Описание метода организации входных и выходных данных	20
3.3.2.	Обоснования выбора метода организации входных и выходных данных	20
3.4.	ОПИСАНИЕ И ОБОСНОВАНИЕ ВЫБОРА СОСТАВА ТЕХНИЧЕСКИХ СРЕДСТВ	20
3.4.1.	Состав технических и программных средств.....	20
3.4.2.	Обоснование выбора технических и программных средств	20
4.	ОЖИДАЕМЫЕ ТЕХНИКО ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ.....	21
5.	ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ	22
ПРИЛОЖЕНИЕ 1 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ БИБЛИОТЕК И КЛАССОВ.....		24
ПРИЛОЖЕНИЕ 2 ТАБЛИЦЫ С ОПИСАНИЕМ МЕТОДОВ		25
ПРИЛОЖЕНИЕ 3 ДИАГРАММА UML		31

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы: «Автоматическая расшифровка ЭКГ по Холтеру»

Наименование темы разработки на английском языке: «Automatic Interpretation of Holter ECG»

1.2. Документ, на основе которого ведется разработка

Приказ декана факультета компьютерных наук И.В. Аржанцева "Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук" № *** от ***.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение разработки

Назначение разработки – решение задачи оптимальной кластеризации ЭКГ пациента. Программа использует предобученную нейронную сеть для более эффективной кластеризации данных по каждому пациенту индивидуально.

2.2. Краткая характеристика области применения

Программа предназначена для кластеризации данных по ЭКГ. На данный момент имеющиеся аналоги недостаточно качественно производят анализ. Программа предназначена для врачей функциональной диагностики и кардиологов для сокращения времени и процента ошибок при анализе данных ЭКГ снятых с Холтера.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Цель проекта – создание консольного приложения для кластеризации данных ЭКГ, которое в дальнейшем может быть использовано в рамках более крупных проектов.

3.2. Описание алгоритма функционирования программы с обоснованием выбора схемы алгоритма решения задачи

Кластеризация данных ЭКГ состоит из нескольких основных пунктов:

На этапе разработки:

- Предобработка данных
- Обучение Автоэнкодера
- Тестирование

На этапе использования программы:

- Предобработка данных
- Кластеризация

3.2.1. Preprocessing

Входной сигнал имеет два канала. На рисунках изображён только один из них для простоты восприятия. С обоими каналами производятся идентичные операции.

Предобработка сигнала необходима для дальнейшего обучения, так как позволяет повысить качество обучающей выборки.

Шаг 1. Обрезание первых и последних 100 отсчётов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

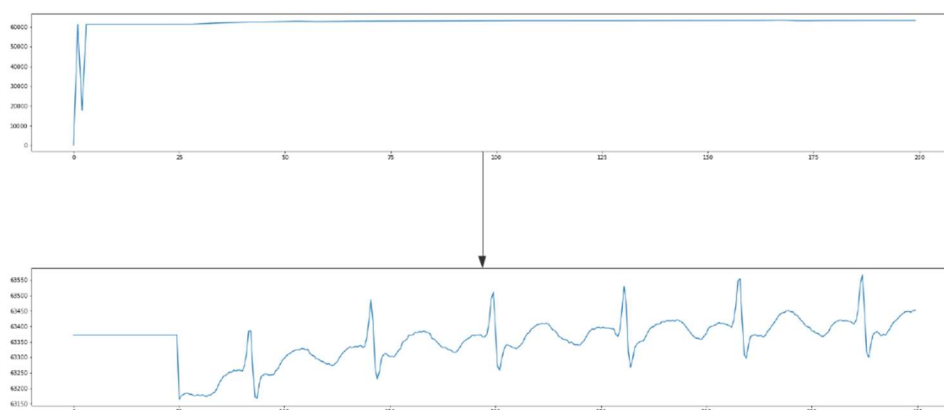


Рисунок 1 Обрезание первых и последних 100 отсчётов

Необходимо обрезание первых и последних 100 отсчётов, так как в них высокое содержание шума.

Шаг 2. Заполнение мест отхождения электрода

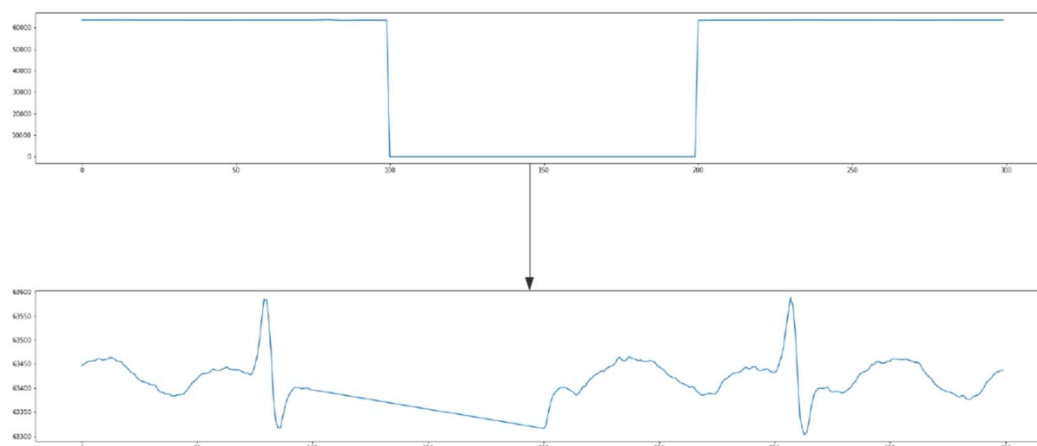


Рисунок 2 Заполнение мест отхождения электрода

В местах отхождения электрода сигнал становится нулевым, что сильно повышает ошибку на всей обучающей выборке. Поэтому «края» временных участков, между которыми отошёл электрод, было принято соединить отрезком.

Шаг 3. Центрирование данных относительно нуля

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

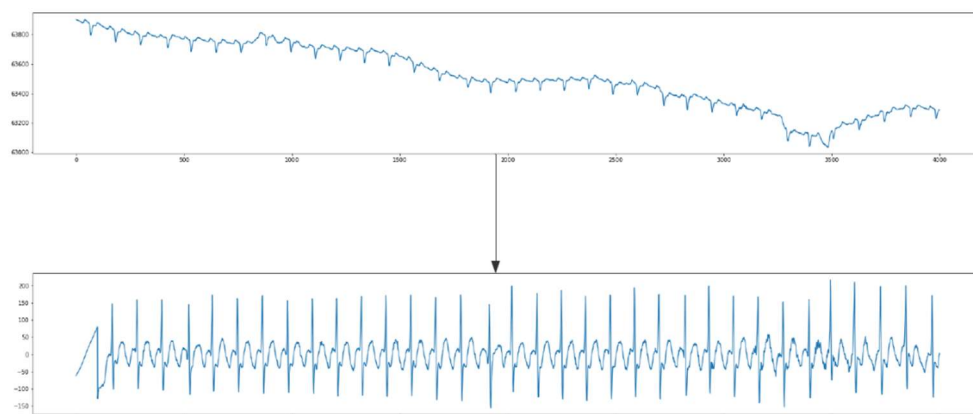


Рисунок 3 Центрирование данных относительно нуля

Для обучения необходима нормализация данных. В данном случае представлено центрирование относительно нуля путём вычитания медианы.

Шаг 4. Удаление низкочастотного шума

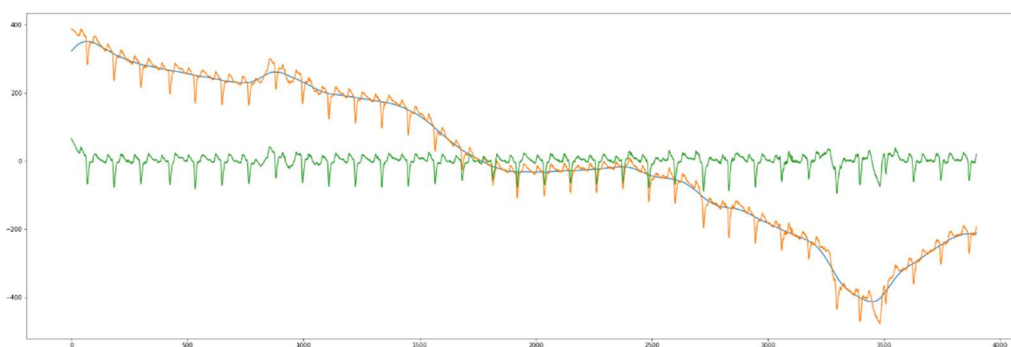


Рисунок 4 Низкочастотный шум

Низкочастотный шум был убран методом скользящего среднего.

Шаг 5. Сглаживание высокочастотного шума

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

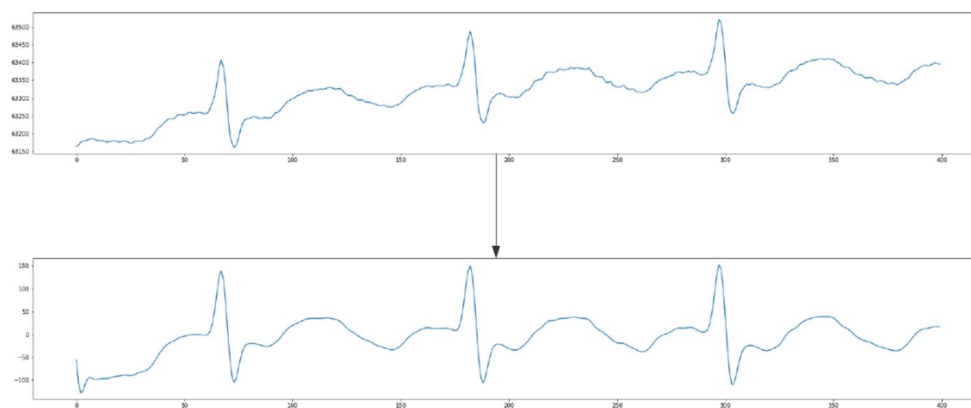


Рисунок 5 Высокочастотный шум

Сглаживание высокочастотного шума позволяет повысить соотношение сигнал/шум и снизить ошибку на выходе.

Шаг 6. Уменьшение амплитуды сигнала

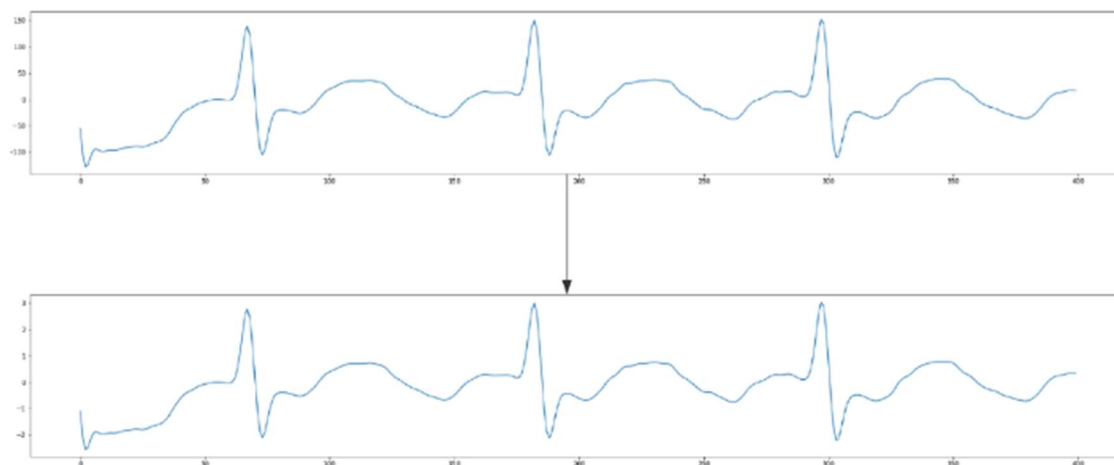


Рисунок 6 Уменьшение размерности сигнала

Амплитуда сигнала была уменьшена в 50 раз.

Шаг 7. Обрезание пиков

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

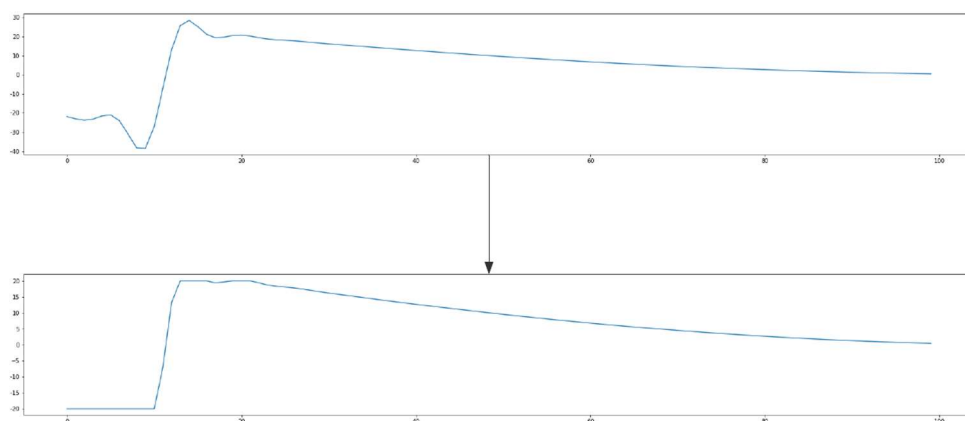


Рисунок 7 Обрезание пиков

Так как высокие пики говорят о наличии шума, то было принято решение обрезать их для улучшения качества обучения.

Шаг 8. Уменьшение частоты дискретизации

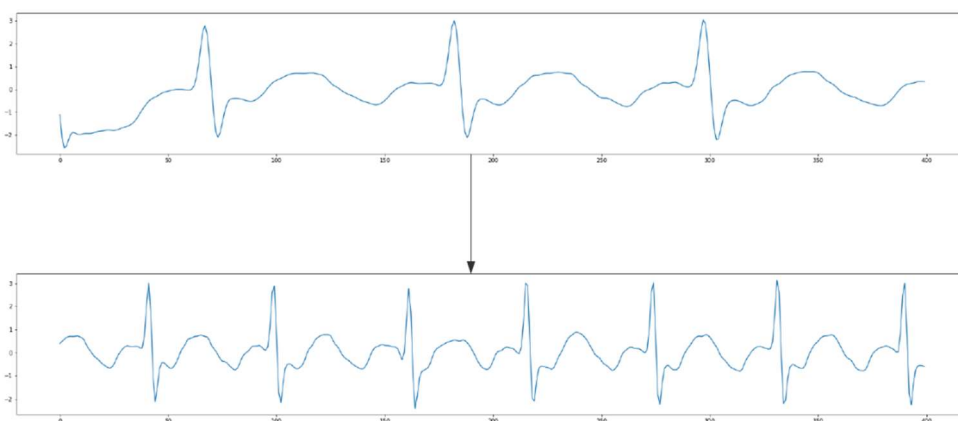


Рисунок 8 Уменьшение частоты дискретизации

Так как текущая дискретизация 250Гц, а информационная 80Гц, то, по теореме Котельникова [3], уменьшение частоты не повлияет на возможность восстановления сигнала. Это необходимо для повышения скорости обработки сигнала.

3.2.2. Создание выборки

Был взят dataset 5000 записей длительностью 24 часа каждый от ранних пациентов, размеченных автоматически и проверенных и исправленных врачом. В каждой записи порядка 100000 QRS комплексов. Перед началом обучения создали равномерную выборку из узких и широких QRS комплексов (до 50 примеров каждого класса от каждого пациентов).

3.2.3. Инициализация Автоэнкодера

Рассмотрим построение класса и инициализацию случайными весами нейронной сети.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

Веса инициализируем случайными величинами по правилу Хе [8]. Для реализации алгоритма обучения без учителя за основу возьмём автоэнкодер [4]. Автокодировщики – это тип нейронных сетей, которые используются для обучения без учителя. Основная идея заключается в попытке нейронной сети воспроизвести входные данные, но, для исключения прямого копирования, размерность скрытых слоев ниже размерности входного слоя. Строение автоэнкодера можно представить следующим образом:

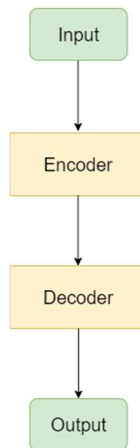


Рисунок 9 Автоэнкодер

Рассмотрим каждый блок.

Шаг 1. Encoder

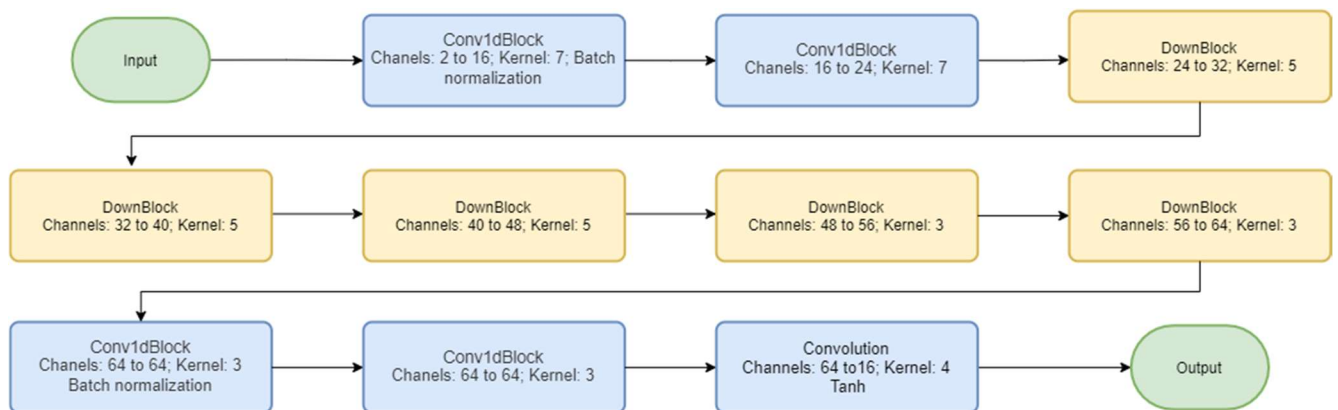


Рисунок 10 Строение Encoder

Encoder для нелинейного преобразования входного сигнала в пространство более низкой размерности. Так, чтобы декодер мог раскодировать данный сигнал.

Рассмотрим строение Conv1dBlock:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

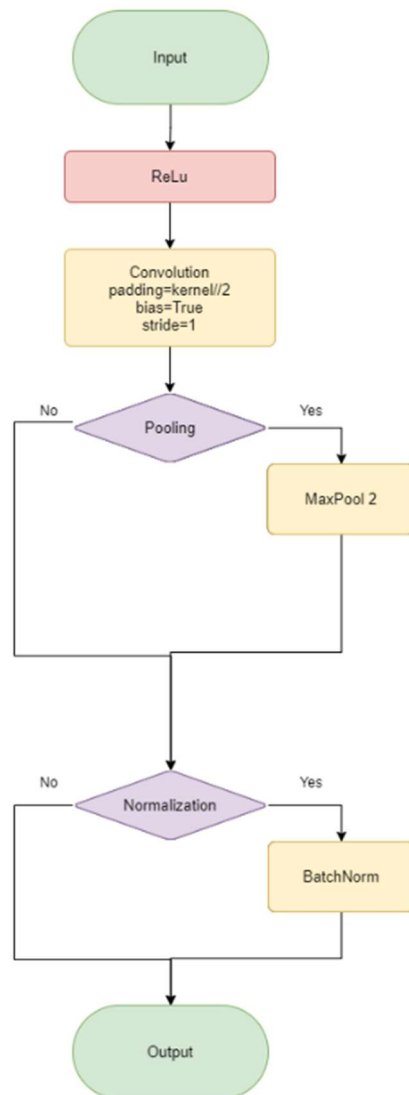


Рисунок 11 Conv1dBlock

В данном случае используется одномерная свёртка со смещением (stride) в один отсчёт, padding равным целой части от деления ядра свёртки на 2, чтобы избежать изменения размерности сигнала. Также используется свободный член (bias).

Рассмотрим DownBlock, который состоит из двух последовательных Conv1dBlock:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

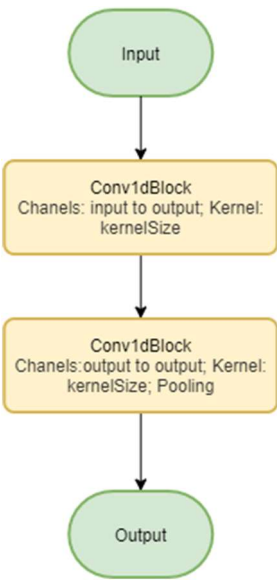


Рисунок 12 DownBlock

Шаг 2. Decoder

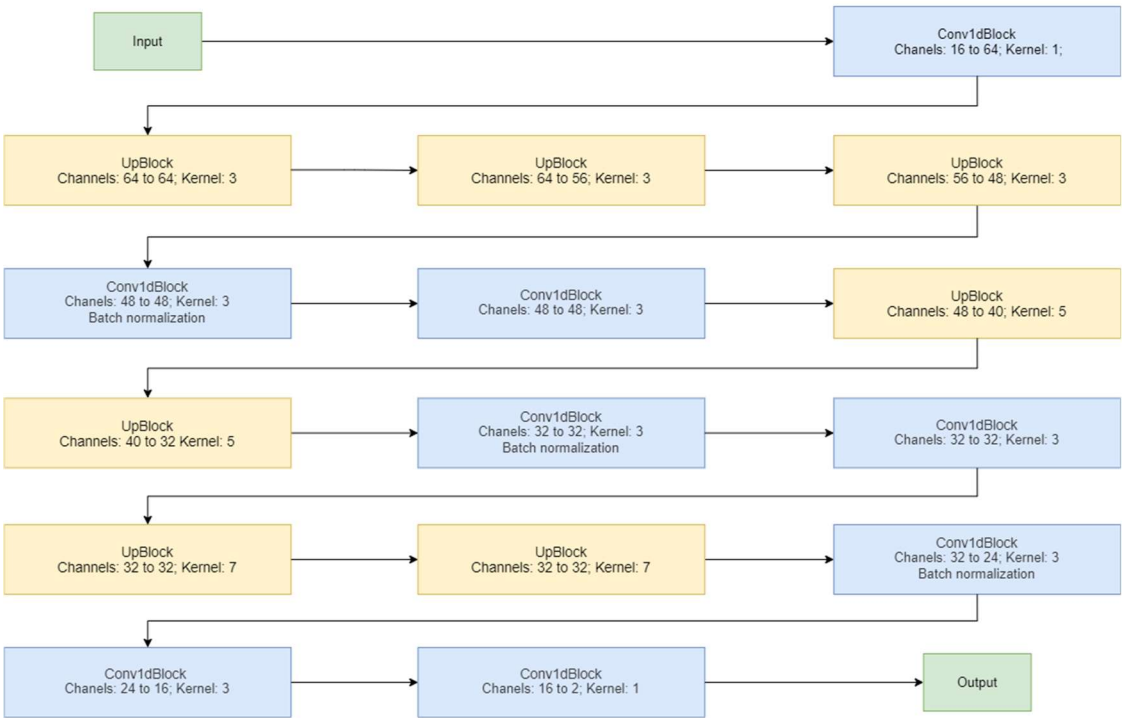


Рисунок 13 Decoder

И переводит данные из пространство низкой размерности в исходное.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

Рассмотрим UpBlock:

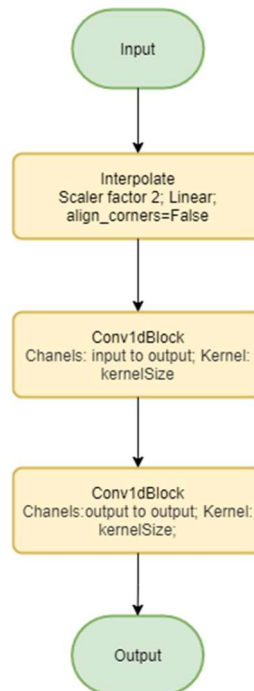


Рисунок 14 UpBlock

В данном случае используем линейную интерполяцию с `scaler_factor=2` для увеличения размерности по координате времени сигнала. То есть это «обратная» операция `max_pool`, который в размерности времени уменьшает сигнал.

3.2.4. Обучение

Схему обучения можно представить следующим образом:

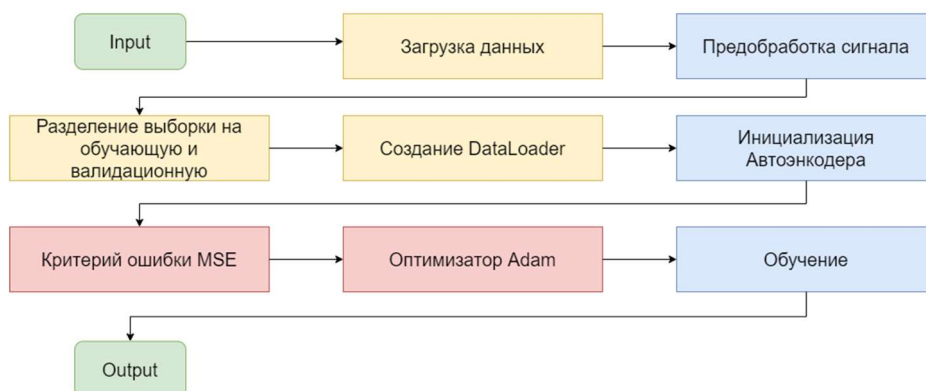


Рисунок 15 Подготовка к обучению

Мы разобрали предобработку сигнала. Для обучения необходимо загрузить предобработанный сигнал. Создать `dataloader` (класс библиотеки `torch.utils.data.DataLoader`), для сокращения обращений к базе данных. Теперь приступим к рассмотрению схемы обучения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

В качестве оптимизатора был выбран адаптационный метод Adam, как наиболее популярный и эффективный [11].

Обучение проходит по эпохам. А в каждой эпохе – по пакетам(batches). Рассмотрим одну эпоху.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

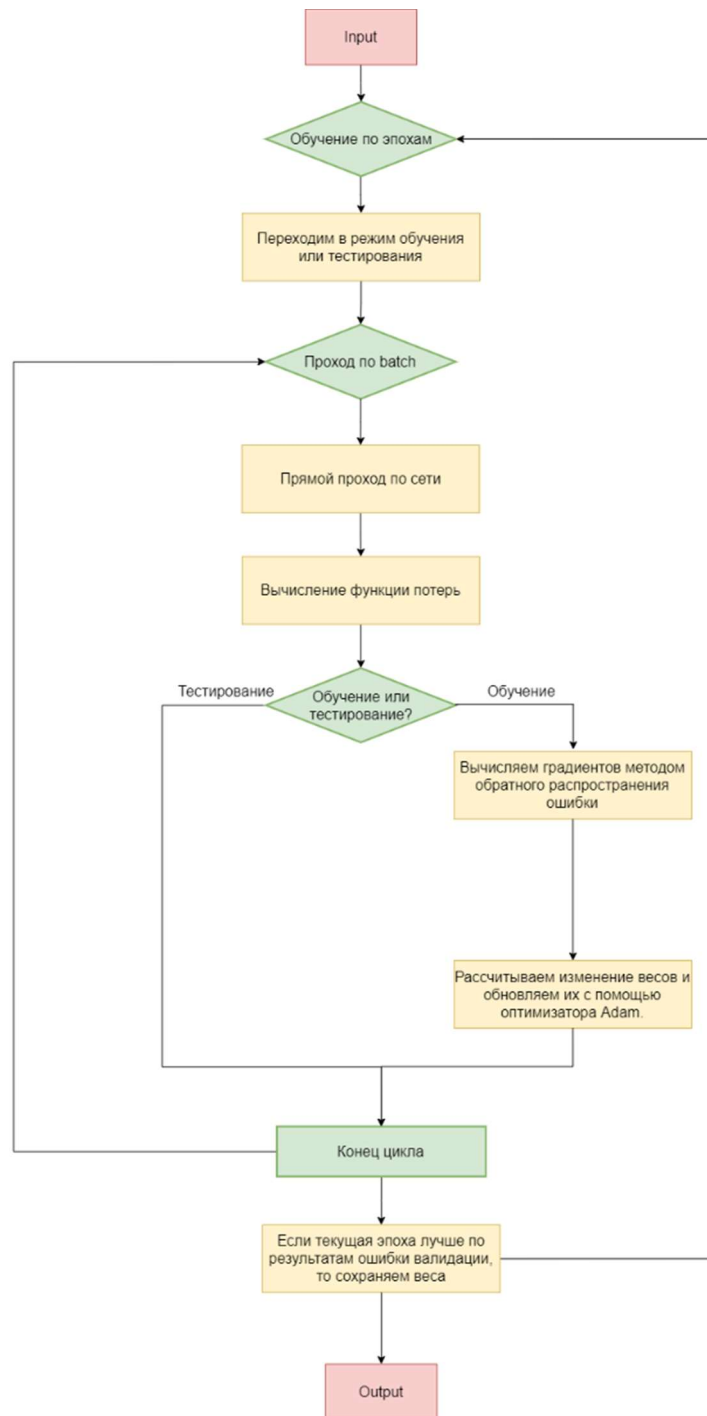


Рисунок 16 Обучение внутри batch

Рассмотрим, как выглядит процесс обучения:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

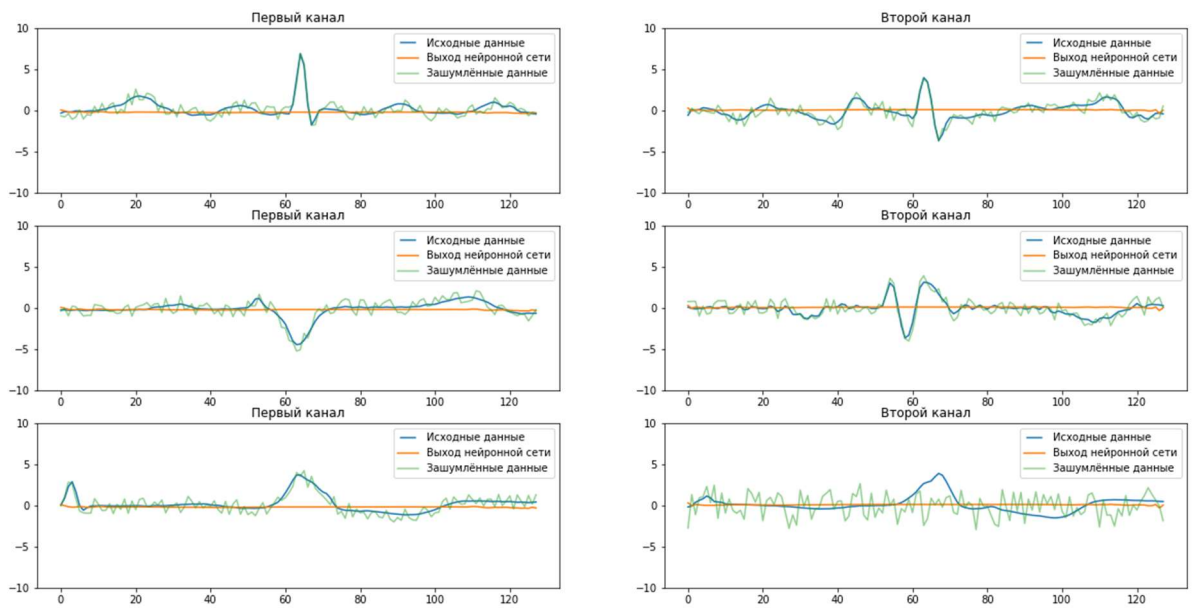


Рисунок 17 Нулевая эпоха

И рассмотрим 7-ю эпоху:

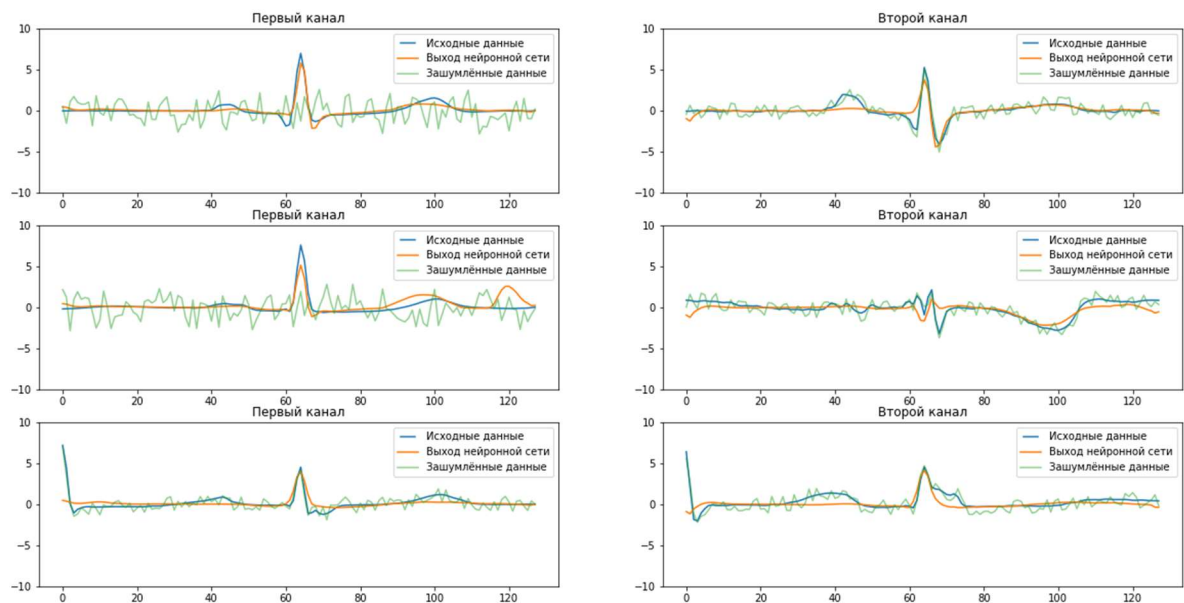


Рисунок 18 Седьмая эпоха

Видно, что нейросеть активно обучается.

И процесс валидации:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

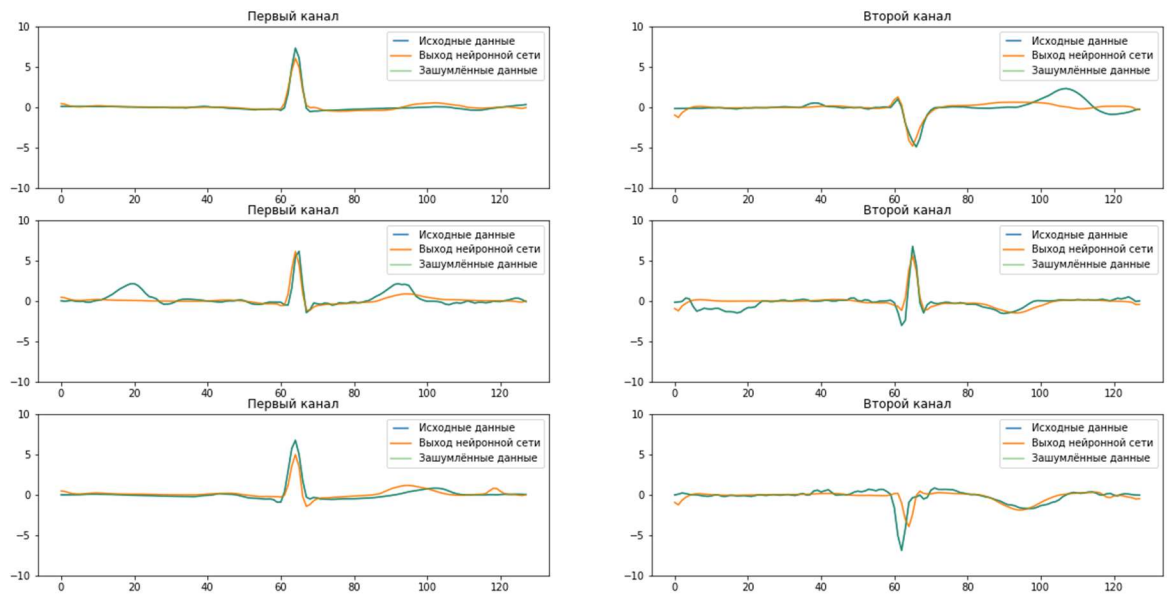


Рисунок 19 Процесс валидации

3.2.5. Кластеризация

Для данной задачи подходили только кластеризаторы без заданного заранее числа кластеров. В статье [5] указаны графики скорости разных кластеризаторов на больших данных:

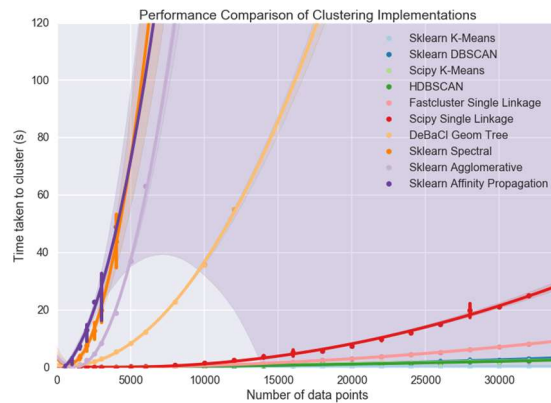


Рисунок 20 Скорость работы кластеризаторов

Для кластеризации используется HDBSCAN [6] с алгоритмом построения минимального остовного дерева методом Борувка. Также необходима кластеризация, которая может выделять шум из данных.

Рассмотрим результат работы алгоритма и пример кластеров. Пример одного из кластеров

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

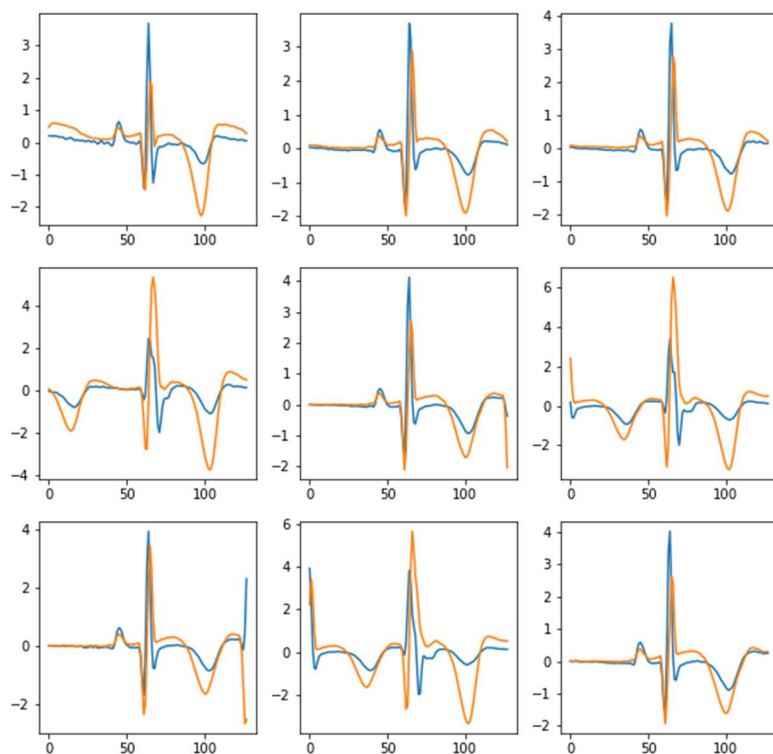


Рисунок 21 Пример одного из кластеров

И пример другого кластера:

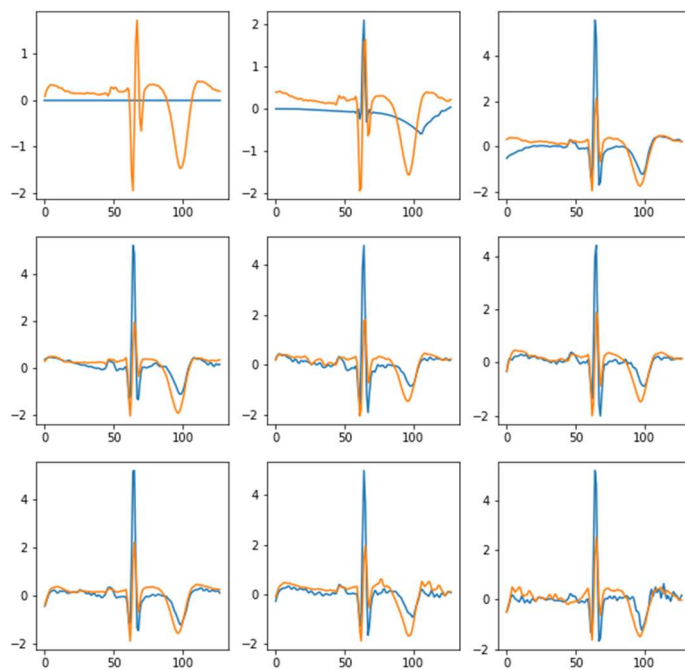


Рисунок 22 Пример другого кластера

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

С помощью метода анализа главных компонент (PCA) изобразим эти кластеры (без шума):

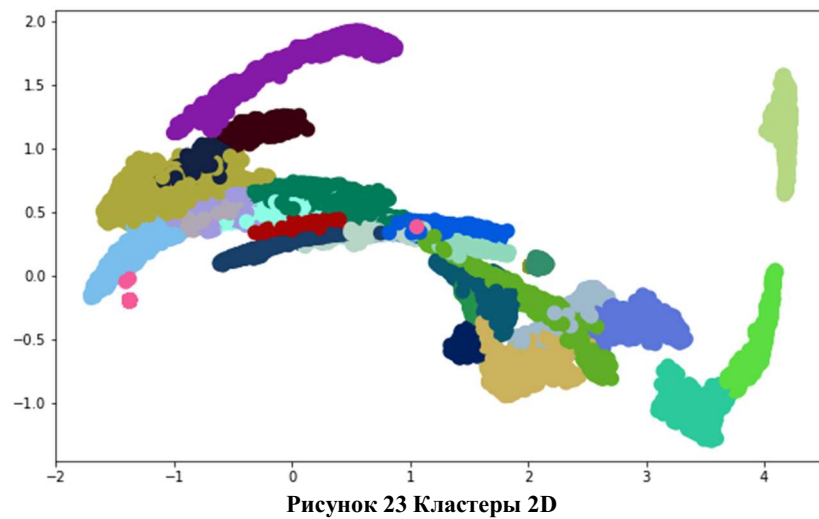


Рисунок 23 Кластеры 2D

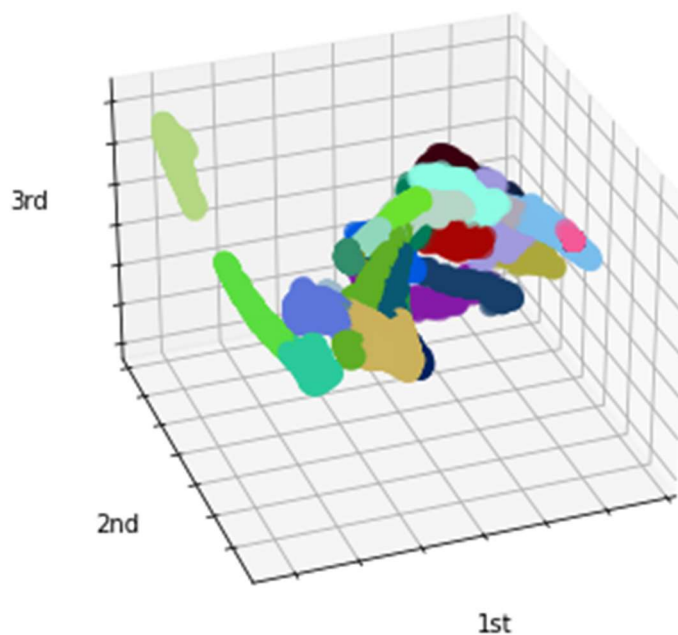


Рисунок 24 Кластеры 3D

Для визуализации кластеров использовался метод, описанный в статье [13], с использованием библиотек [14]-[17].

Во время подготовки также использовались материалы [2]-[12].

3.2.6. Возможные взаимодействия программы с другими программами

Программа предназначена для совместной работы с другими программными обеспечениями для анализа ЭКГ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.3. Описание и обоснование выбора входных и выходных данных

3.3.1. Описание метода организации входных и выходных данных

Входными данными является файл сигнала, снятого с ЭКГ, формата .ecg. И файл с координатами R-пиков в формате .rr2. В качестве выходных данных предоставляется .csv с номерами кластеров, где «-1» класс шума.

3.3.2. Обоснования выбора метода организации входных и выходных данных

Данный метод ввода входных данных был выбран так как исходные данные предоставляются в данном формате. Формат .csv - наиболее удобный для хранения таблиц.

3.4. Описание и обоснование выбора состава технических средств

3.4.1. Состав технических и программных средств

Для работы программы требуется компьютер, имеющий следующие технические характеристики:

1. Мышь и клавиатура,
2. Экран с разрешением 1920*1080
3. Свободное место на диске (не менее 512 МБ)
4. 4 GB RAM
5. CPU Intel Pentium 2,40 GHz

Состав программных средств, необходимых для работы программы:

Операционная система Windows 10, Linux или Mac.

3.4.2. Обоснование выбора технических и программных средств

Программа разработана с использованием кроссплатформенности к ОС. Исходным языком программирования для данной разработки является Python. Среда разработки – Anaconda, PyCharm.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ОЖИДАЕМЫЕ ТЕХНИКО ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Программа разработана для улучшения системы кластеризации. Так как на данный момент нет эффективного способа произвести разделение на кластеры, требует участия врачей. Однако, алгоритм работает быстрее и просматривает все QRS комплексы, выделяя несколько кластеров. Анализ технико-экономических показателей не производился в рамках данной работы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению.
// Единая система программной документации. – М.: ИПК Стандартиформ, 2010.
2. Глубокое обучение/ С. Николенко, А. Кадури, Е. Архангельская. - М.: Питер, 2018. - 480 с.
3. Теорема Котельникова [Электронный ресурс] //URL:
https://ru.wikipedia.org/wiki/Теорема_Котельникова (Дата обращения: 14.05.2021, режим доступа: свободный).
4. Автокодировщик [Электронный ресурс] //URL:
<https://ru.wikipedia.org/wiki/Автокодировщик> (Дата обращения: 14.05.2021, режим доступа: свободный).
5. Benchmarking Performance and Scaling of Python Clustering Algorithms [Электронный ресурс] //URL:
https://hdbscan.readthedocs.io/en/latest/performance_and_scalability.html (Дата обращения: 16.05.2021, режим доступа: свободный).
6. How HDBSCAN Works [Электронный ресурс] //URL:
https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html (Дата обращения: 16.05.2021, режим доступа: свободный).
7. Performance Evaluation: Ball-Tree and KD-Tree in the context of MST / Hazarath Munaga, Venkata Jarugumalli [Электронный ресурс] //URL:
<https://arxiv.org/ftp/arxiv/papers/1210/1210.6122.pdf> (Дата обращения: 16.05.2021, режим доступа: свободный).
8. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun [Электронный ресурс] //URL: <https://arxiv.org/pdf/1502.01852.pdf> (Дата обращения: 17.05.2021, режим доступа: свободный).
9. Noise-Resilient Automatic Interpretation of Holter ECG Recordings/ Egorov Konstantin, Sokolova Elena, Avetisian Manvel, Tuzhilin Alexande [Электронный ресурс] //URL: <https://arxiv.org/pdf/2011.09303.pdf> (Дата обращения: 17.05.2021, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

10. Линейная интерполяция [Электронный ресурс] //URL: https://ru.wikipedia.org/wiki/Линейная_интерполяция (Дата обращения: 17.05.2021, режим доступа: свободный).
11. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION/ Diederik P. Kingma, Jimmy Lei Ba [Электронный ресурс] //URL: <https://arxiv.org/pdf/1412.6980.pdf> (Дата обращения: 17.05.2021, режим доступа: свободный).
12. CS231n: Convolutional Neural Networks for Visual Recognition [Электронный ресурс] //URL: <http://cs231n.stanford.edu> (Дата обращения: 17.05.2021, режим доступа: свободный).
13. PCA Analysis for GeneClassification [Электронный ресурс] //URL: <https://www.kaggle.com/kanav0183/pca-analysis-for-geneclassification> (Дата обращения: 18.05.2021, режим доступа: свободный).
14. Matplotlib [Электронный ресурс] //URL: <https://matplotlib.org> (Дата обращения: 17.05.2021, режим доступа: свободный).
15. PyTorch [Электронный ресурс] //URL: <https://pytorch.org> (Дата обращения: 17.05.2021, режим доступа: свободный).
16. NumPy [Электронный ресурс] //URL: <https://numpy.org> (Дата обращения: 17.05.2021, режим доступа: свободный).
17. Sklearn [Электронный ресурс] //URL: <https://scikit-learn.org/stable/> (Дата обращения: 18.05.2021, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ БИБЛИОТЕК И КЛАССОВ

Таблица 1. Описание и функциональное назначение библиотек

Библиотеки	Назначение
models	Библиотека, инициализирующий Dataloader и автоэнкодер
pytorch_models_2	Библиотека, реализующий автоэнкодер
pipeline_tools	Библиотека, содержащая функцию для обучения автоэнкодера
processing	Библиотека, реализующий предобработку сигнала
data_utils	Библиотека, реализующий работу с входными данными
dataset_creation	Библиотека, создающая обучающую выборку
main	Основной модуль

Таблица 2. Описание и функциональное назначение классов

Классы	Назначение
AutoEncoder	Класс, инициализирующий автоэнкодер
PytorchDS	Класс, организующий работу с DataSet
Conv1dBlock	Класс, реализующий pooling, нормализацию и свёртку
DownBlock	Класс, реализующий уменьшение размерности данных по координате времени
UpBlock	Класс, реализующий увеличение размерности данных по координате времени
ECGEncoder	Класс, реализующий последовательное уменьшение размерности данных по координате времени и создания bottleneck
ECGDecoder	Класс, реализующий увеличение размерности данных по координате времени, начиная от самого «узкого» мест
ECGAutoEncoder	Класс, реализующий автоэнкодер

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

ПРИЛОЖЕНИЕ 2

ТАБЛИЦЫ С ОПИСАНИЕМ МЕТОДОВ

Таблица 3. Описание методов класса train_advanced

имя	модификаторы доступа	тип	аргументы	назначение
main	public	void	-	Ввод данных с консоли
start	public	void	-	Инициализация данных перед началом обучения
expirement	public	void	ecg_path:str, ann_path:str	Кластеризация конкретного пациента на предобученной сети
process_patient	public	pandas.core.frame.DataFrame numpy.ndarray	int	Возвращает нарезанное на отрезки по 128 отсчётов ЭКГ

Таблица 4. Описание методов класса models.AutoEncoder

имя	модификаторы доступа	тип	аргументы	назначение
__init__	private	конструктор	channels_mult : int, bottleneck: int, batch_norm: bool, valid_size: float, seed:int, aug_params:dict, num_workers: int, batch_size:int, device: str, lr: float, num_epochs: int, log_file: str, weights_file: str, train_net:	Конструктор для инициализации автоэнкодера
Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

			bool, preload:str	
fit	public	void	ecgs: numpy.ndarra y	Метод, запускающий обучение
transform	public	numpy.ndarra y	ecgs: numpy.ndarra y	Метод, преобразующий данные
_run_epoch	private	list	dl:torch.utils.d ata.DataLoade r, train:bool	Метод, описывающий проход по одной эпохе обучения или валидации

Таблица 5. Описание методов класса models.PytorchDS

имя	модификаторы доступа	тип	аргументы	назначение
__init__	private	конструктор	ecgs:numpy.n darray, aug_params: dict	Конструктор, инициализирующий класс
__len__	private	int	-	Метод, возвращающий количество позиций в выборке
__getitem__	private	dict	index:in	Возвращает зашумлённый комплекс QRS
augment	public	numpy.ndarra y	ecg:numpy.nd array	Производит зашумление участка ЭКГ

Таблица 6. Описание методов класса pytorch_models_2.Conv1dBlock

имя	модификаторы доступа	тип	аргументы	назначение
__init__	private	конструктор	in_channels:int , out_channels:i nt, kernel_size:int, pooling:bool, batch_norm:bo ol	Производит инициализацию класса Conv1dBlock
forward	public	torch.Tensor	x:object	Реализует прямой проход по нейронной сети

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 7. Описание методов класса `pytorch_models_2.DownBlock`

имя	модификаторы доступа	тип	аргументы	назначение
<code>__init__</code>	private	конструктор	<code>in_channels:int</code> , <code>out_channels:int</code> , <code>kernel_size:int</code>	Производит инициализацию класса <code>DownBlock</code>
<code>forward</code>	public	<code>torch.Tensor</code>	<code>x:object</code>	Реализует прямой проход по нейронной сети

Таблица 8. Описание методов класса `pytorch_models_2.UpBlock`

имя	модификаторы доступа	тип	аргументы	назначение
<code>__init__</code>	private	конструктор	<code>in_channels:int</code> , <code>out_channels:int</code> , <code>kernel_size:int</code>	Производит инициализацию класса <code>UpBlock</code>
<code>forward</code>	public	<code>torch.Tensor</code>	<code>x:object</code>	Реализует прямой проход по нейронной сети

Таблица 9. Описание методов класса `pytorch_models_2.ECGEncoder`

имя	модификаторы доступа	тип	аргументы	назначение
<code>__init__</code>	private	конструктор	<code>in_channels:int</code> , <code>out_channels:int</code> , <code>bottleneck:int</code> , <code>batch_norm:bool</code>	Производит инициализацию класса <code>ECGEncoder</code>
<code>forward</code>	public	<code>torch.Tensor</code>	<code>x:object</code>	Реализует прямой проход по нейронной сети

Таблица 10. Описание методов класса `pytorch_models_2.ECGDecoder`

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

имя	модификаторы доступа	тип	Аргументы	назначение
<code>__init__</code>	private	конструктор	<code>in_channels:int</code> , <code>out_channels:int</code> , <code>bottleneck:int</code> , <code>batch_norm:bool</code>	Производит инициализацию класса ECGDecoder
<code>forward</code>	public	<code>torch.Tensor</code>	<code>x:object</code>	Реализует прямой проход по нейронной сети

Таблица 11. Описание методов класса `pytorch_models_2.AutoEncoder`

имя	модификаторы доступа	тип	аргументы	назначение
<code>__init__</code>	private	конструктор	<code>channels_mult:int</code> , <code>bottleneck:int</code> , <code>batch_norm:bool</code>	Производит инициализацию класса AutoEncoder
<code>forward</code>	public	<code>pytorch_models_2.AutoEncoder</code>	<code>x:object</code>	Реализует прямой проход по нейронной сети

Таблица 12. Описание методов библиотеки `pipeline_tools`

имя	модификаторы доступа	тип	аргументы	назначение
<code>valid_split</code>	public	<code>numpy.ndarray</code> , <code>numpy.ndarray</code>	<code>df:dict</code> , <code>valid_size:int</code> , <code>seed:int</code>	Разделяет выборку на train и valid
<code>get_data</code>	public	<code>anns_path:str</code> , <code>ecgs_path:str</code>	<code>anns:pandas.core.frame.DataFrame</code> , <code>ecg: numpy.ndarray</code>	Возвращает данные о пациентах, взятых из файла
<code>experiment_v2</code>	public	<code>params:dict</code>	<code>n_clusters:int</code> , <code>homogeneity:float</code> , <code>noise: float</code>	Запускает «эксперимент». Производит обучение и разбиение на кластеры. И сохраняет результат разбиения

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

				на кластеры в файл
--	--	--	--	--------------------

Таблица 13. Описание методов библиотеки processing

имя	модификаторы доступа	тип	аргументы	назначение
filter_signal	public	numpy.ndarray	ts: numpy.ndarray, rate: int, low_freq: int, high_freq: int, order: int	Подавляет частоты выше high_freq и частоты, которые ниже low_freq
process_ecg	public	numpy.ndarray	ecg: numpy.ndarray, trend: int, downsample: int, low_freq: int, high_freq: int, clip: int, divide_by: int	Преобразует сигнал, сглаживая частоты, обрезаая пики, удаление высокочастотного и низкочастотного шума, центрирования, передискретизации сигнала.
interp_holes	public	numpy.ndarray	arr: numpy.ndarray	Заполняет места отхождения электрода
get_slice	public	numpy.ndarray	ecg: numpy.ndarray, center: int, window: int, maxval: int, threshold: int	Возвращает QRS комплекс по значению R-пика

Таблица 14. Описание методов библиотеки data_utils

имя	модификаторы доступа	тип	аргументы	назначение
get_patients	public	list	-	Возвращает список пациентов (номера)
get_train	public	list	-	Возвращает id пациентов, которые вошли в обучающую выборку
get_test	public	list	-	Возвращает id пациентов, которые вошли в тестовую выборку
get_ecg	public	numpy.ndarray	patient: int, path: str	Возвращает ЭКГ пациента, по его id
get_ann	public	pandas.core.f	patient: int, path: str	Возвращает разметку

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

		rame.DataFrame		ЭКГ пациента по id
decode_ann	public	pandas.core.frame.DataFrame	signal: numpy.ndarray, path: str	Преобразует многомерный массив данных о пациенте в таблицу с именами параметров

Таблица 15. Описание методов библиотеки dataset_creation

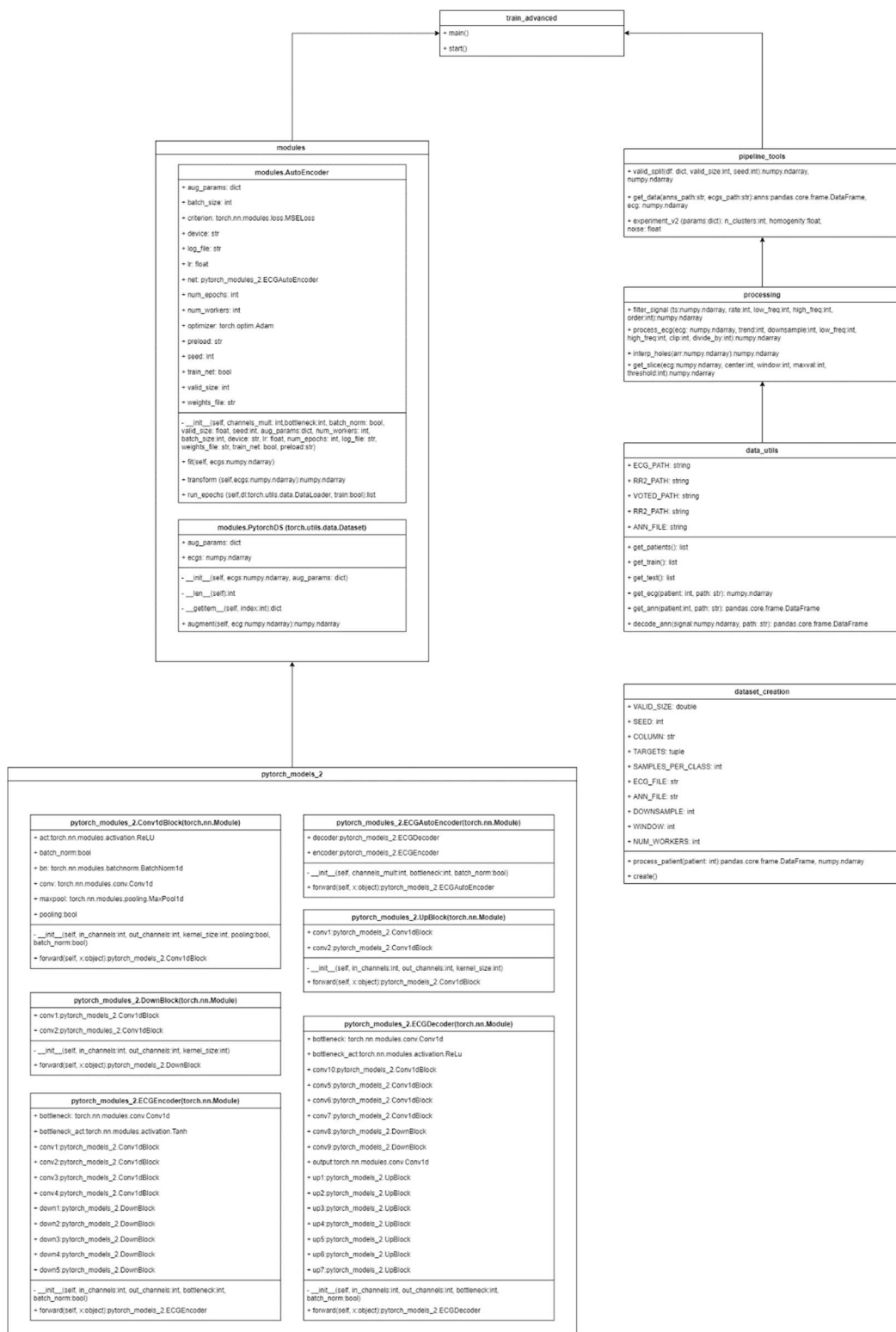
имя	модификаторы доступа	тип	аргументы	назначение
process_patient	public	pandas.core.frame.DataFrame numpy.ndarray	int	Возвращает сбалансированную выборку QRS комплексов одного пациента
create	public	void	-	Возвращает сбалансированную выборку всей обучающей выборки

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

ПРИЛОЖЕНИЕ 3

ДИАГРАММА UML



Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-				
Инв. № подл.	Подп. и	Взам. инв. №	Инв. № дубл.	Подп. и дата



This document was created with the Win2PDF "print to PDF" printer available at
<http://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

This page will not be added after purchasing Win2PDF.

<http://www.win2pdf.com/purchase/>