

Пояснительная записка  
к итоговому проекту на тему:

**«Диагностирование заболевания по форме ЭКГ»**

Автор: Ферапонтов Сергей

Группа: DLL-28

## Содержание

1	Введение в электрокардиографию.....	3
2	Постановка задачи.....	4
2.1	Определение рода исследуемых данных.....	4
2.2	Постановка задачи .....	4
3	Литературный обзор.....	6
4	Работа с данными .....	14
4.1	Выбранный датасет.....	14
4.2	Разметка и разбиение данных.....	16
4.3	Анализ данных .....	17
4.3.1	Распределение количества меток по примерам.....	17
4.3.2	Распределение частоты встречаемости классов.....	18
4.3.3	Метод балансировки обучающей выборки.....	19
4.3.4	Результаты балансировки .....	21
5	Построение моделей.....	23
5.1	Рекуррентная модель .....	24
5.2	Базовая свёрточная модель .....	25
5.3	Улучшенная свёрточная модель.....	26
6	Результаты.....	28
6.1	Дальнейшие пути развития .....	29
	Список литературы .....	31
	Приложение А. Сравнительная таблица рассмотренных моделей.....	33

# **1 Введение в электрокардиографию**

На сегодняшний день электрокардиография является ведущим методом диагностики сердечно-сосудистых заболеваний, которые, в свою очередь, являются ведущей причиной смерти в развитых странах. Аппарат измеряет разность потенциалов в 12 или 24 отведениях, производя таким образом график 12 или 24 величин, синхронизированных по времени [1]. Различные аппараты проводят измерения с различной частотой и точностью, поэтому на практике может потребоваться небольшая предобработка данных. Современные медицинские стандарты не предписывают учёта никаких дополнительных факторов (пол, возраст, вес и т.д.) при постановке диагноза. Отметим, что так как финальное решение всегда принимает лечащий врач, разрабатываемый алгоритм следует рассматривать как систему поддержки принятия решений, призванную автоматизировать работу врача функциональной диагностики, расшифровывающего ЭКГ.

Процесс обработки ЭКГ врачом функциональной диагностики включает в себя:

1. Анализ размеров зубцов и длин интервалов, анализ формы деформации, частоты сердечных сокращений и так далее;
2. По полученным данным определяются ЭКГ-признаки наличия некоторых заболеваний.

## 2 Постановка задачи

### 2.1 Определение рода исследуемых данных

Для начала отметим, что, автоматизировать ЭКГ-диагностику с применением машинного обучения возможно, как анализируя исходный временной ряд показаний прибора, так и анализируя измененные величины зубцов и интервалов. Остановимся на первом методе, так как для решения второго лучше подойдут алгоритмы, основанные на деревьях решений и им подобные. Кроме того, потенциальный алгоритм анализа временного ряда позволит избавиться от необходимости установки модуля измерений зубцов и интервалов, а также лучше оценить деформацию ЭКГ-сигнала.

Отметим, что измеренные прибором данные могут быть представлены как в виде временного ряда, так и в виде фотографии плёнки ЭКГ. Для работы используем именно представление в виде временного ряда, что более приближенно к автоматической обработке показаний аппарата ЭКГ.

Также важной особенностью данного типа данных является связь один ко многим между исходными данными и диагнозами (Multilabel), что затрудняет применение ряда базовых моделей.

### 2.2 Постановка задачи

Поставим нашей целью максимально автоматизировать работу врача функциональной диагностики и создать систему поддержки принятия решений, фактически повторяющую его функции.

Иными словами, целью нашего алгоритма будет по имеющимся данным предсказывать конкретные диагнозы, то есть решать задачу multilabel классификации временного ряда.

Входными данными алгоритма являются примеры, представляющие собой временные ряды из 12 признаков, что соответствует измерению в 12 отведениях. Каждому примеру соответствует набор диагнозов (количество классов  $\geq 1$ ).

В качестве метрик качества примем Accuracy, Precision и F1-метрику, за именем лучшего решения, будем рассматривать их относительно каждого диагноза в отдельности. Так как из-за специфики данных большая часть примеров будут отрицательными примерами для конкретного класса, accuracy интересует нас в меньшей степени, больший интерес представляет precision, отражающую точность классификации на положительных примерах и F1-метрика, соединяющая в себе как Precision, так и Recall, не рассматриваемый в данной работе.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FN} \quad (2)$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \quad (3)$$

### 3 Литературный обзор

Идея использования различных алгоритмов и методов машинного обучения для работы с ЭКГ не нова. Каждый исследователь, занимающийся данным вопросом, работает в пределах собственных допущений, особенностей данных и целей для моделей. Таким образом, особенностью данного направления является широкая доступность машинного представления показаний аппаратов ЭКГ, что отражается в наличии крупных датасетов, имеющих разнородные данные.

Как было сказано ранее, существует достаточное количество различных попыток построить нейросетевую модель для классификации сигнала ЭКГ, но к сожалению, многие из них представляют недостаточно убедительные результаты.

Структуру большинства моделей можно представить в виде следующей последовательности:

1. Предобработка данных в виде фильтрации высокочастотных шумов
2. Опциональное преобразование типов данных
3. Классификация

Предобработка осуществляется такими методами как вейвлет преобразование, преобразование Гильберта – Хуанга, фильтр высоких частот и т. д.. Теоретически, такое решение должно повысить качество классификации, но необходимость их применения сильно зависит от особенностей конкретного датасета. Также следует понимать опасность данного метода, мы не сможем провести честную валидацию на неизменённых данных, и в то же время, мы заменяем анализ исходных данных анализом фильтрованных данных. В некоторых случаях это может быть полезно модели, но при неудачном выборе параметров фильтрации мы можем потерять значащие признаки исходных данных. К сожалению, исследования на данную тему почти не касаются вопроса оптимизации фильтров и оценки их применимости, поэтому исключим данный этап из нашей собственной модели.

Преобразование типа данных может быть использовано для перехода от анализа одномерных данных к анализу двумерного представления в пространстве времени и частоты (time-frequency domain), что может быть осуществлено вейвлет преобразованием и преобразование Гильберта – Хуанга. Подобное решение имеет потенциал и может привести к повышению качества модели, так как упростит извлечение признаков из исходного временного ряда и позволит применять устоявшиеся архитектуры двумерных свёрточных нейросетей.

Перейдём к слою непосредственно обработки данных. Удивительно, но большинство разработанных моделей имеют достаточно простую архитектуру, представляющую собой полносвязный персептрон или простую 1D или 2D свёрточную нейросеть.

Рассмотрим несколько работ более подробно:

Интересным решением, использованном в работе [2] является применение GASF transformation для преобразования временного ряда в изображение. В остальном в данной работе сравнивается сеть на основе feed-forward и свёрточных слоёв. Результаты показывают увеличение точности при применении свёрточных сетей. Авторы используют классификацию на 4 суперкласса заболеваний и, к сожалению, не предоставляют никаких других метрик кроме accuracy и функции потерь на обучающем датасете. Далее, на примере собственной модели мы покажем, что метрика accuracy не всегда является информативной при большом дисбалансе классов.

Похожее решение было принято в работе [3]. В данном случае использовано преобразование Гильберта – Хуанга, результат которого обработан нейросетью VGG-16. Хотя совмещают нейросетевую модель со значительным объёмом пред- и постобработки (рис. 1), полученные ими значения точности, фактически равной 100% вызывают некоторые сомнения.

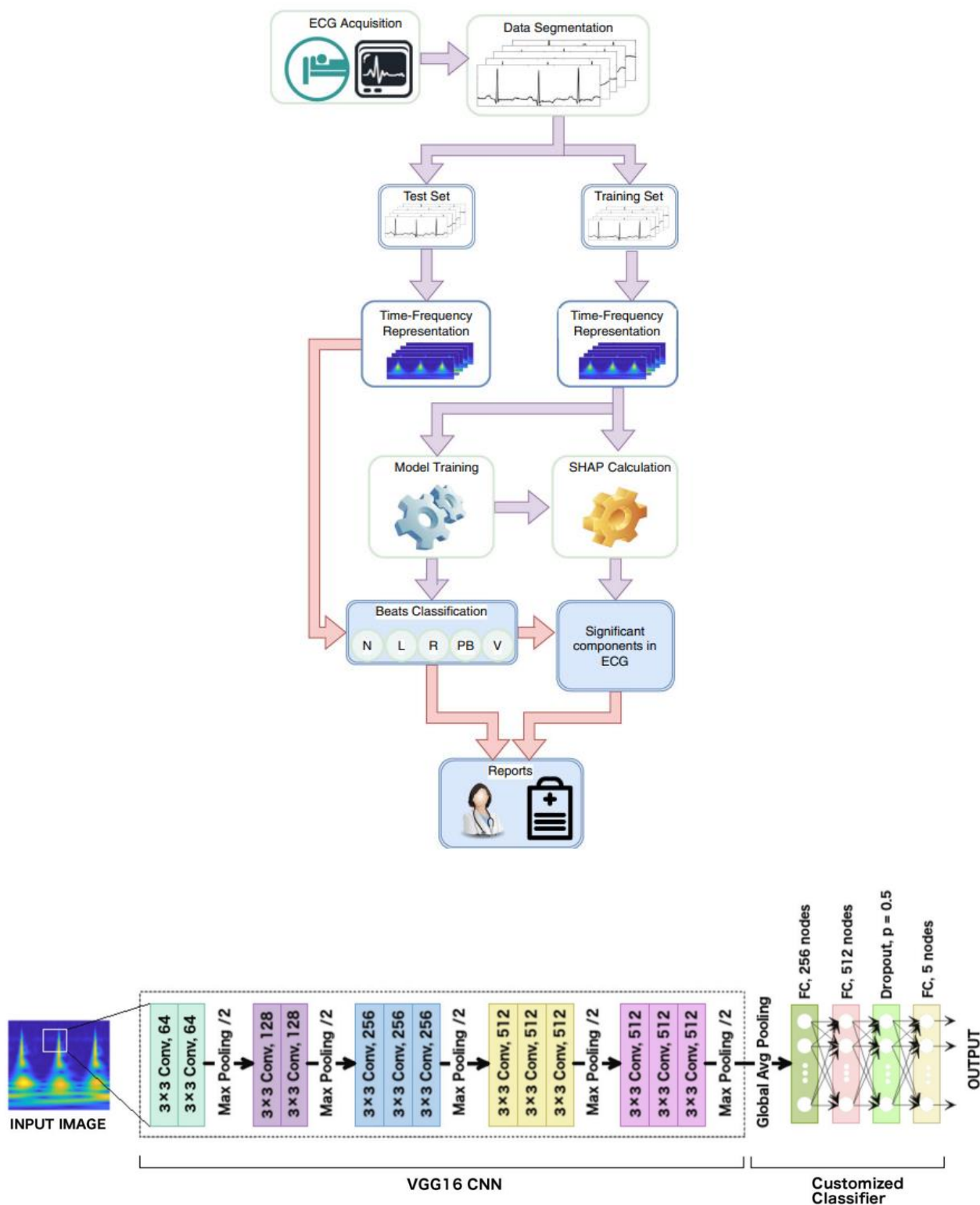


Рисунок 1 – Модель обработки данных [3]

Интересен подход работ, основанных на одномерных свёрточных сетях [4]. В данной работе также рассмотрено всего 5 классов, но для сравнения использо-



вано больше метрик. Предложенная авторами свёрточная сеть хоть и является достаточно базовой и содержит только свёрточные слои, pooling слои и один слой dropout (рис. 2), но всё же показала лучшие результаты чем другие методы машинного обучения, что свидетельствует о необходимости применения нейросетевых алгоритмов при анализе ЭКГ.

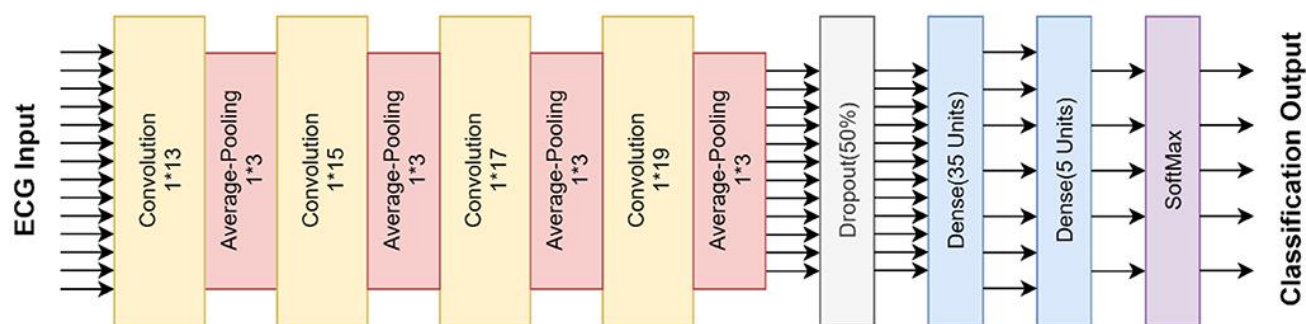


Рисунок 2 – Архитектура сети [4]

К сожалению, не все работы вызывают достаточное доверие к результатам [5]. Так, например, автор данной работы демонстрирует не просто подозрительное поведение графика зависимости функции потерь от эпохи, но и показывает на этом же графике видимое увеличение функции потерь на тестовой выборке, что свидетельствует о переобучении (рис. 3). Что не удивительно, так как предложенная автором модель имеет очень большую размерность (рис. 4).

Интересное решение, совмещающее свёрточные и рекуррентные сети можно найти в работе [6]. Отличительной частью данной работы является то, что авторы отмечают важность аугментации данных и балансировки изначального датасета, мы ещё увидим важность данных действий при рассмотрении собственной модели. К сожалению, авторы не предоставили метрик за исключением оценки конкретного соревнования.

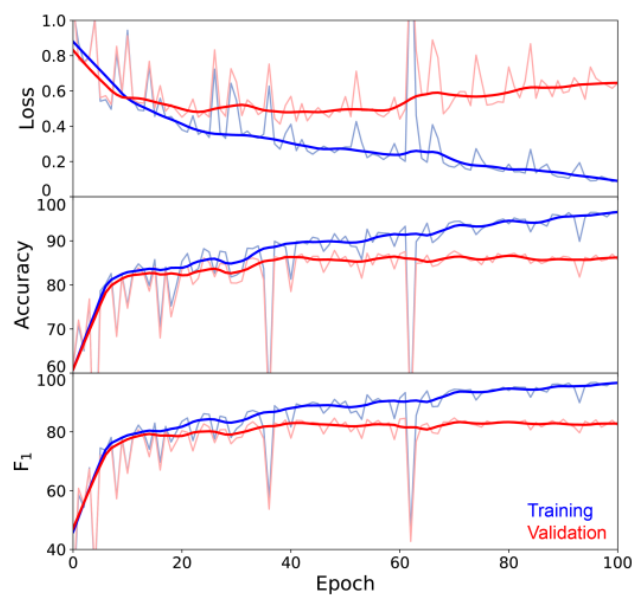


Рисунок 3 – Процесс обучения модели [5]

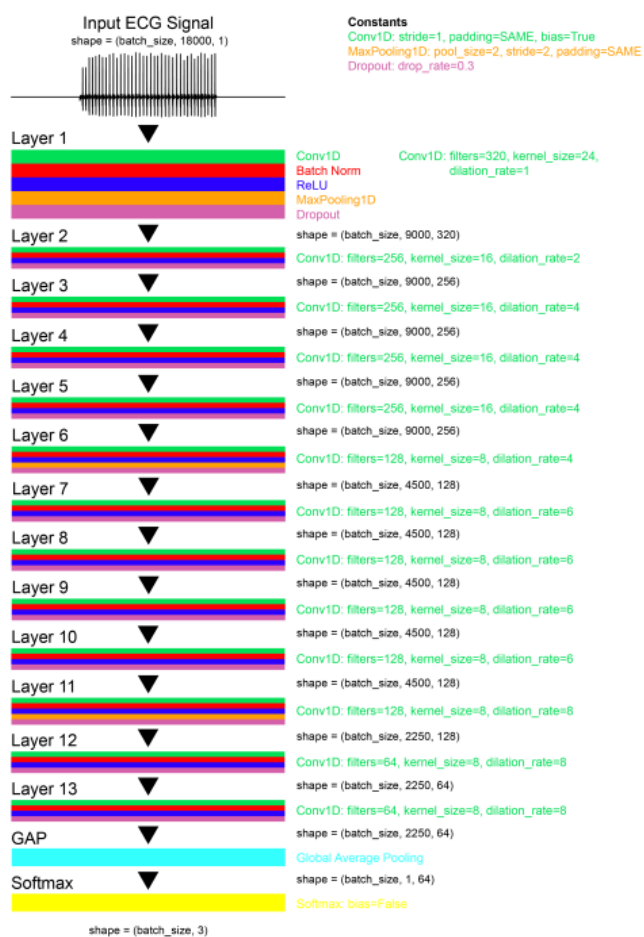


Рисунок 4 – Архитектура модели [5]

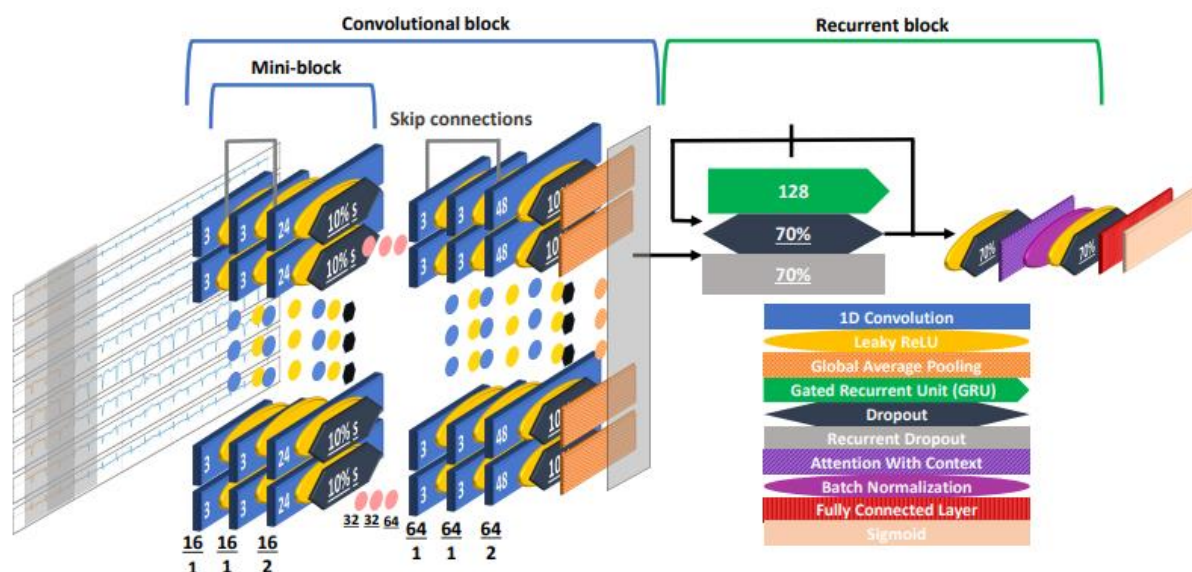


Рисунок 5 – Архитектура модели [6]

Наиболее близка к нашей задаче работа [7]. В данном случае авторы рассматривают задачу multilabel классификации конкретных диагнозов, используя свёрточную нейросеть (рис. 6). Полученные авторами значения метрики F1 представлены на рисунке 7. Будем ориентироваться на них ( $F1 \approx 0.7 \dots 0.95$ ) для оценки собственной модели, так как данная работа по своей сути является наиболее близкой к тому, что мы хотим получить.



	AUC	F1 score	Sensitivity	Specificity
<b>N</b>	1.0 (1.0, 1.0)	0.998 (0.994, 1.0)	1.0 (1.0, 1.0)	1.0 (0.999, 1.0)
<b>ST</b>	0.971 (0.965, 0.977)	0.858 (0.839, 0.877)	0.774 (0.743, 0.805)	0.998 (0.997, 0.998)
<b>SB</b>	0.901 (0.896, 0.906)	0.692 (0.683, 0.702)	0.561 (0.551, 0.571)	0.991 (0.99, 0.993)
<b>PAC</b>	0.949 (0.945, 0.952)	0.726 (0.714, 0.738)	0.647 (0.631, 0.664)	0.987 (0.985, 0.989)
<b>AR</b>	0.996 (0.993, 1.0)	0.827 (0.784, 0.87)	0.787 (0.703, 0.87)	0.998 (0.997, 0.999)
<b>AT</b>	0.986 (0.981, 0.99)	0.882 (0.871, 0.892)	0.897 (0.882, 0.913)	0.994 (0.992, 0.995)
<b>AFlutter</b>	0.993 (0.99, 0.995)	0.88 (0.867, 0.893)	0.898 (0.884, 0.912)	0.992 (0.991, 0.993)
<b>AFib</b>	0.991 (0.989, 0.993)	0.863 (0.848, 0.877)	0.873 (0.856, 0.889)	0.985 (0.982, 0.987)
<b>PJC</b>	0.991 (0.988, 0.994)	0.877 (0.862, 0.892)	0.907 (0.891, 0.923)	0.991 (0.99, 0.992)
<b>JR</b>	0.973 (0.969, 0.977)	0.854 (0.842, 0.865)	0.828 (0.807, 0.849)	0.994 (0.993, 0.995)
<b>PSVT</b>	0.982 (0.976, 0.987)	0.931 (0.927, 0.936)	0.895 (0.887, 0.904)	0.999 (0.998, 0.999)
<b>PVC</b>	0.963 (0.96, 0.967)	0.83 (0.819, 0.841)	0.758 (0.74, 0.775)	0.993 (0.992, 0.994)
<b>IVR</b>	0.997 (0.997, 0.998)	0.863 (0.847, 0.878)	0.939 (0.926, 0.951)	0.995 (0.994, 0.996)
<b>VT</b>	0.99 (0.986, 0.993)	0.938 (0.922, 0.955)	0.914 (0.895, 0.934)	0.998 (0.998, 0.999)
<b>AAPR</b>	1.0 (0.999, 1.0)	0.992 (0.983, 1.0)	0.986 (0.972, 1.0)	1.0 (1.0, 1.0)
<b>AVPR</b>	0.99 (0.988, 0.991)	0.917 (0.909, 0.924)	0.89 (0.88, 0.9)	0.997 (0.996, 0.997)
<b>LBBS</b>	0.984 (0.979, 0.989)	0.926 (0.915, 0.937)	0.88 (0.865, 0.896)	0.999 (0.998, 1.0)
<b>1<sup>st</sup> Degree AVB</b>	0.987 (0.983, 0.99)	0.878 (0.861, 0.894)	0.872 (0.854, 0.891)	0.991 (0.99, 0.993)
<b>Mobitz I AVB</b>	1.0 (1.0, 1.0)	0.989 (0.981, 0.997)	0.988 (0.976, 0.999)	1.0 (0.999, 1.0)
<b>WPW-B</b>	0.997 (0.995, 0.999)	0.941 (0.931, 0.951)	0.917 (0.899, 0.935)	0.998 (0.997, 0.999)
<b>WPW-A</b>	1.0 (1.0, 1.0)	0.972 (0.967, 0.978)	0.991 (0.983, 0.998)	0.997 (0.996, 0.997)
<b>Average</b>	0.983 (0.98, 0.986)	0.887 (0.875, 0.9)	0.867 (0.849, 0.885)	0.995 (0.994, 0.996)

Рисунок 7 – Результаты работы модели [7]

## 4 Работа с данными

### 4.1 Выбранный датасет

Исходя из решений, принятых в главе 1.2, выберем датасет, предназначенный для multilabel классификации заболевания по временному ряду ЭКГ. Отметим, что для дальнейшей работы предпочтительно выбрать датасет большего размера, чтобы уменьшить вероятность переобучения.

Наиболее подходящим для нашей задачи будет PTB-XL ECG dataset [8]. Данный датасет содержит 21837 размеченных примера ЭКГ, каждому из которых соответствует не менее одного класса, каждому классу соответствует степень уверенности в диагнозе. Кроме того, датасет содержит дополнительную информацию о пациентах, временной ряд в высоком (500 Гц) и низком (100 Гц) разрешении, стандартный набор измеряемых величин зубцов и интервалов и т.д. (рис. 8)

```
[2]: meta_df = pd.read_csv('ptbx1_database.csv', index_col=0)
     scp_df = pd.read_csv('scp_statements.csv')
     meta_df
```

	patient_id	age	sex	height	weight	nurse	site	device	recording_date	report	scp_codes	heart_axis	infarction_stadium1	infarction_stadium2	validated_by	second_opinion	initial_autogenerated_report
ecg_id																	
1	15709.0	56.0	1	NaN	63.0	2.0	0.0	CS-12 E	1984-11-09 09:17:34	sinusrhythmus periphere niederspannung	('NORM': 100.0, 'LVOLT': 0.0, 'SR': 0.0)	NaN	NaN	NaN	NaN	NaN	False
2	13243.0	19.0	0	NaN	70.0	2.0	0.0	CS-12 E	1984-11-14 12:55:37	sinusbradykardie sonst normales ekg	('NORM': 80.0, 'SBRAD': 0.0)	NaN	NaN	NaN	NaN	NaN	False
3	20372.0	37.0	1	NaN	69.0	2.0	0.0	CS-12 E	1984-11-15 12:49:10	sinusrhythmus normales ekg	('NORM': 100.0, 'SR': 0.0)	NaN	NaN	NaN	NaN	NaN	False
4	17014.0	24.0	0	NaN	82.0	2.0	0.0	CS-12 E	1984-11-15 13:44:57	sinusrhythmus normales ekg	('NORM': 100.0, 'SR': 0.0)	NaN	NaN	NaN	NaN	NaN	False
5	17448.0	19.0	1	NaN	70.0	2.0	0.0	CS-12 E	1984-11-17 10:43:15	sinusrhythmus normales ekg	('NORM': 100.0, 'SR': 0.0)	NaN	NaN	NaN	NaN	NaN	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	AT-60	2001-05-31	ventrikuläre extrasystole(n)	('NDT': 100.0, 'PVC':	...	...	...	...	...	...

Рисунок 8 – Исходный датасет

Для дальнейшей работы возьмём временные ряды с частотой 100 Гц, также оставим данные о диагнозе для классификации и номере дециля для дальнейшего разделения на обучающую и тестовую выборки. На рисунке 9 представлен отфильтрованный датасет. На рисунке 10 представлен пример временных рядов, являющихся входными данными для разрабатываемой модели.



ecg_id	scp_codes	strat_fold	filename_lr
1	{'NORM': 100.0, 'LVOLT': 0.0, 'SR': 0.0}	3	records100/00000/00001_lr
2	{'NORM': 80.0, 'SBRAD': 0.0}	2	records100/00000/00002_lr
3	{'NORM': 100.0, 'SR': 0.0}	5	records100/00000/00003_lr
4	{'NORM': 100.0, 'SR': 0.0}	3	records100/00000/00004_lr
5	{'NORM': 100.0, 'SR': 0.0}	4	records100/00000/00005_lr
...	...	...	...
21833	{'NDT': 100.0, 'PVC': 100.0, 'VCLVH': 0.0, 'ST...}	7	records100/21000/21833_lr
21834	{'NORM': 100.0, 'ABQRS': 0.0, 'SR': 0.0}	4	records100/21000/21834_lr
21835	{'ISCAS': 50.0, 'SR': 0.0}	2	records100/21000/21835_lr
21836	{'NORM': 100.0, 'SR': 0.0}	8	records100/21000/21836_lr
21837	{'NORM': 100.0, 'SR': 0.0}	9	records100/21000/21837_lr

21837 rows × 3 columns

Рисунок 9 – Необходимые для дальнейшей работы колонки датасета



Рисунок 10 – Пример плёнки ЭКГ, используемой моделью

## 4.2 Разметка и разбиение данных

Изначальный датасет содержит в себе метки о 71 классе, но не все они являются диагностическими. Некоторые классы свидетельствуют о нарушении формы или ритма сердечных сокращений, но не относятся ни к одному диагнозу.

Данное решение приводит к формированию небольшой группы примеров, для которых нет диагностического класса с ненулевой степенью уверенности, исключим такие примеры из дальнейшего рассмотрения.

На данном этапе примем решение рассматривать исключительно диагностические классы. Кроме того, использование степени уверенности в диагнозе в качестве целевой величины плохо отразится на выбранных нами метриках, так как потребуются адекватный учёт примеров с низкой уверенностью. Примем решение не учитывать степень уверенности и конвертировать любую ненулевую вероятность диагноза в исходных данных в метку наличия класса.

Разделим данные согласно примеру, приложенному авторами датасета: дециль 10 соответствует тестовой выборке, все остальные – обучающей.

Итого мы получим следующие наборы:

- Тестовая выборка: 2162 примера на 44 класса,
- Обучающая выборка: 19255 примера на 44 класса.



## 4.3 Анализ данных

### 4.3.1 Распределение количества меток по примерам

Так как каждому примеру может соответствовать более одного диагноза, построим распределение количества меток на диагнозе для обучающей (рис. 11) и тестовой (рис. 12) выборках.

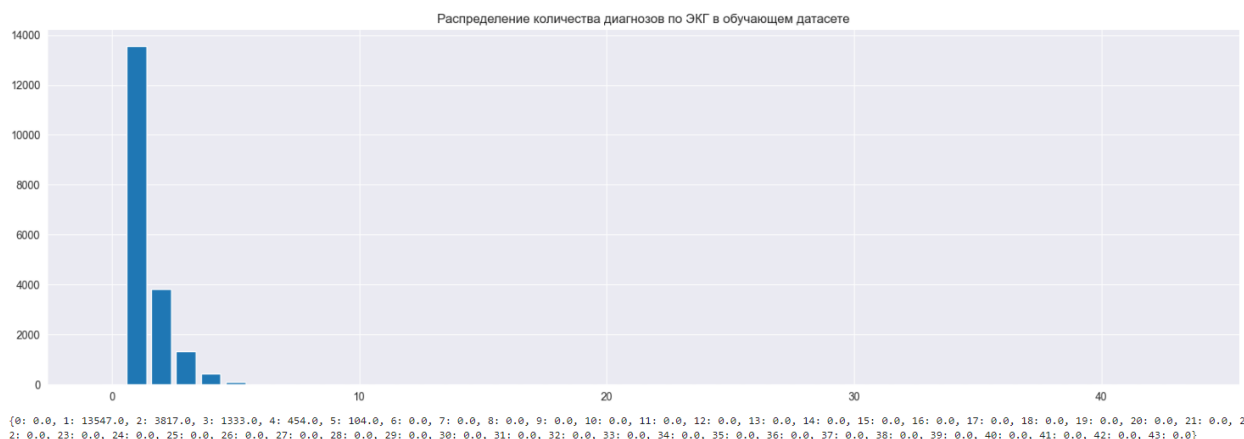


Рисунок 11 – Распределение количества меток на обучающих примерах



Рисунок 12 – Распределение количества меток на тестовых примерах

Как мы видим, задача действительно является multilabel классификацией: каждому примеру соответствует от одной до пяти меток. На данных гистограммах мы можем убедиться, что мы верно отфильтровали лишние примеры – не существует примеров без меток. Отметим, что распределение на обучающей и тестовой выборках идентичны, на этом этапе вмешательство не требуется.

### 4.3.2 Распределение частоты встречаемости классов

Соберём информацию о частоте встречаемости каждой из меток на обучающем (рис. 13) и тестовом (рис. 14) выборках.

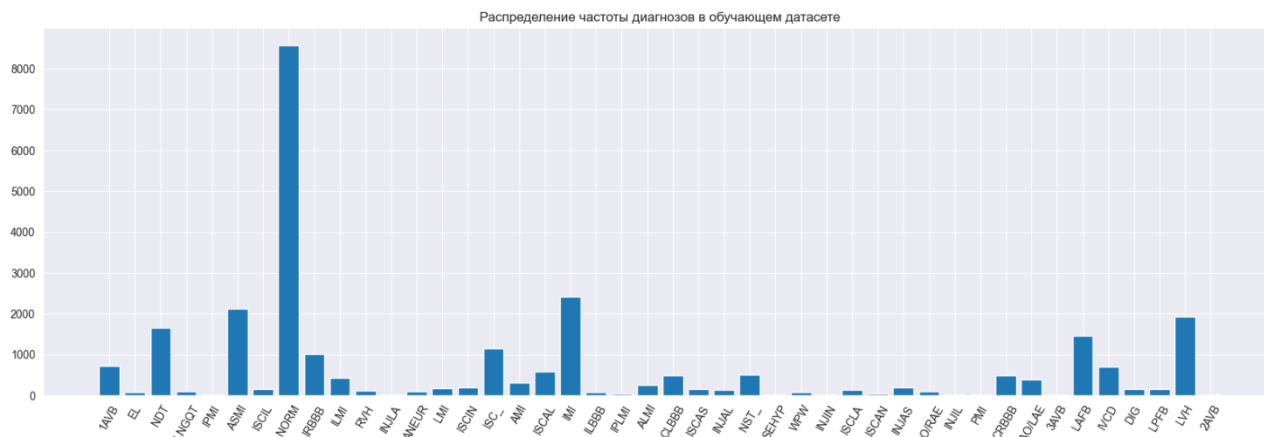


Рисунок 13 – Распределение частоты встречаемости диагнозов в обучающей выборке

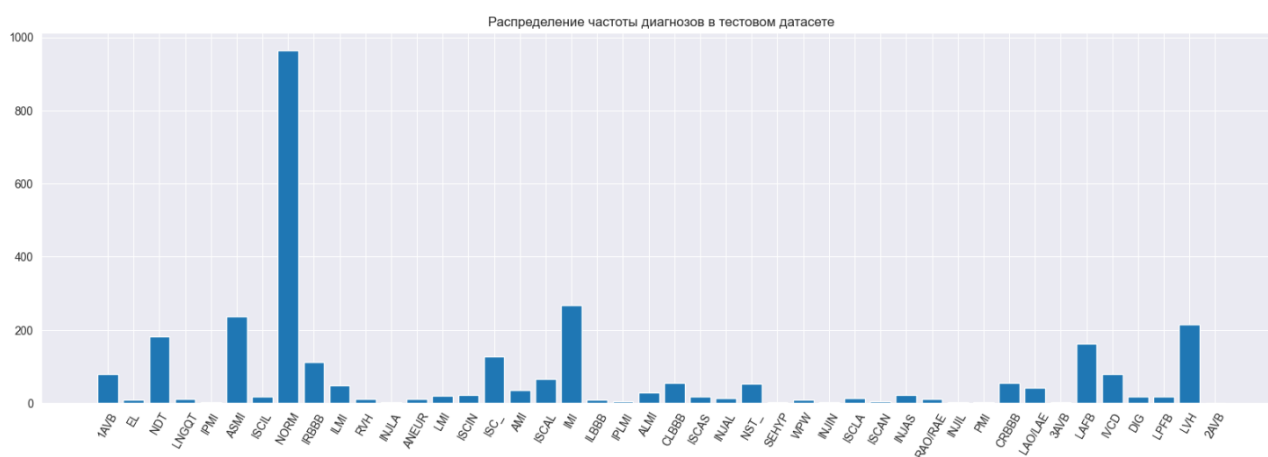


Рисунок 14 – Распределение частоты встречаемости диагнозов в тестовой выборке

Как мы видим, форма распределения совпадает для обеих выборок, это обеспечит более честную оценку по тестовой выборке.

В то же время отметим, что наблюдается очень серьёзный дисбаланс в репрезентации классов. В то время, как нормальные ЭКГ представлены очень широко, отдельные диагнозы встречаются от четырёх до нескольких сотен раз реже.

Данная проблема может очень плачевно сказаться на обучении нейросетевой модели по следующим причинам:

- Модель может начать игнорировать диагнозы и отмечать все примеры как нормальные, всё равно добиваясь хороших показателей;
- Модель может переобучиться на редких диагнозах из-за малого количества примеров;
- Модель может не успеть научиться на редких диагнозах из-за редкого попадания в обучающие батчи.

Чтобы сгладить возможные проблемы, вызванные дисбалансом классов, применим методики балансировки к обучающей выборке. Тестовую выборку оставим неизменной, чтобы сохранить информативность метрик

Так как данные методы могут достаточно сильно трансформировать обучающую выборку, а вместе с ней и качество работы модели, дальнейшие эксперименты будем проводить как на сбалансированных данных, так и на исходных.

#### 4.3.3 Метод балансировки обучающей выборки

Проблема работы с несбалансированными выборками не нова, существует несколько устоявшихся методов работы с подобными типами данных: присвоение весов, *downsampling*, *upsampling* и т.д. Ситуация также осложняется тем, что большое количество устоявшихся методов не адаптированы для работы с *multilabel* данными.

Сразу отбросим взвешенную функцию потерь, так как дисбаланс классов слишком большой, данный метод не решит проблемы редкости примеров некоторых классов в обучающих батчах.

Также отбросим метод *downsampling*, так как из-за того, что некоторые классы встречаются в сотни раз чаще других, придётся отбросить слишком большое количество примеров, что значительно снизит качество обучения.

Одним из возможных решений может быть upsampling обучающей выборки. Для реализации данного метода напишем собственный алгоритм. Его суть будет состоять в том, чтобы присвоить каждому примеру оценку его информативности в зависимости от редкости встреченных классов относительно их распределения в датасете:

$$I_i = \frac{1}{n} \sum_{j=1}^n -\ln(|sample_{i,j} - distr_j|)^2 \quad (4)$$

, где  $I_i$  – информативность  $i$ -го примера;  $sample_{i,j}$  – значение  $\{0, 1\}$  о наличии метки  $j$  класса  $i$  примера;  $distr_{i,j}$  – математическое ожидание встретить класс  $j$  во всей выборке.

Данная формула была получена по аналогии с вычислением объёма информации, данные по нескольким классам агрегируются через среднеквадратичное, чтобы большее внимание обращать на редкие классы.

Затем величину информативности  $[0, +\infty)$  преобразуем в вероятность копирования  $[0, 1]$ . Дополнительно введём настроечный параметр минимальной допустимой вероятности копирования, чтобы избежать ненужного увеличения уже достаточно распространённых классов. Получим следующую зависимость вероятности копирования от оценки информативности примера:

$$P_i = \begin{cases} 1 - \frac{1}{I_i + 1}, & 1 - \frac{1}{I_i + 1} > threshold \\ 0, & 1 - \frac{1}{I_i + 1} \leq threshold \end{cases} \quad (5)$$

, где  $P_i$  – вероятность копирования  $i$  примера,  $threshold$  – минимальная вероятность копирования.

Данный алгоритм выполняется итерационно, на каждой итерации пересчитывая распределение классов, таким образом алгоритм самостоятельно адаптируется, значительно увеличивая число редких классов, постепенно приводя выборку

к ровному распределению. Недостатком данного алгоритма является сам факт копирования примеров, что может привести к запоминанию редких классов и переобучению при работе с ними.

Алгоритмом, решающим проблему копирования, является SMOTE, для сравнения возьмём его адаптацию для работы с Multilabel данными MLSMOTE [9, 10].

#### 4.3.4 Результаты балансировки

Представим распределение частоты встречаемости классов в обучающей выборке до балансировки (рис. 15), после балансировки авторским методом (рис. 16) и после балансировки MLSMOTE (рис. 17).

Также отметим, что алгоритм MLSMOTE может сгенерировать примеры с отличным распределением количества меток по примерам, покажем это на рисунке 18.

Как мы видим, оба алгоритма значительно снижают дисбаланс выборки. К сожалению, доведение до полного баланса не представляется возможным, так как слишком мало примеров редких диагнозов.

Заметим, что в целом визуально алгоритм MLSMOTE создал немного менее гладкую выборку и сгенерировал больше примеров с диагнозами чем было нормальных ЭКГ. В дальнейшем мы увидим, какая из выборок лучше себя покажет.



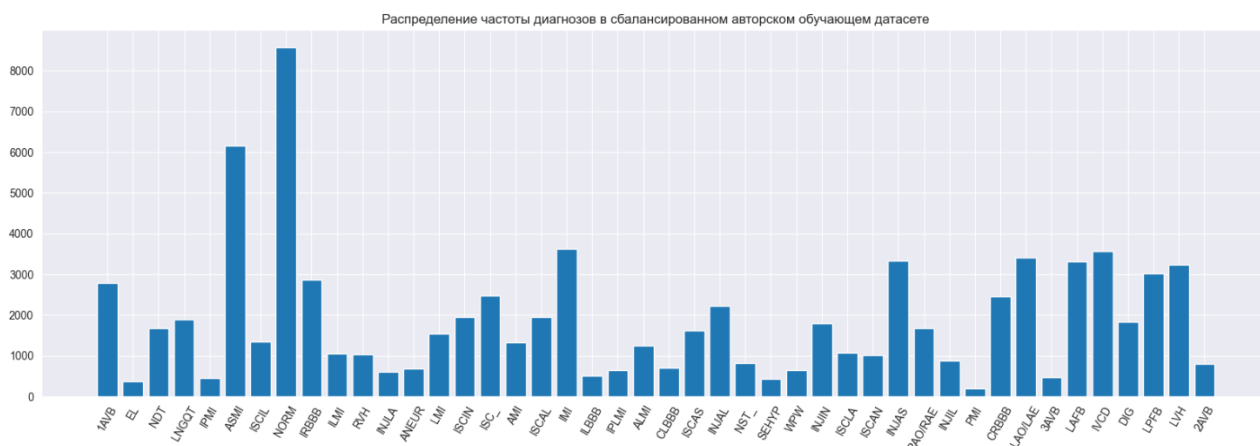


Рисунок 16 – Распределение классов в сбалансированной upsampling выборке



Рисунок 17 – Распределение классов в сбалансированной MLSMOTE выборке

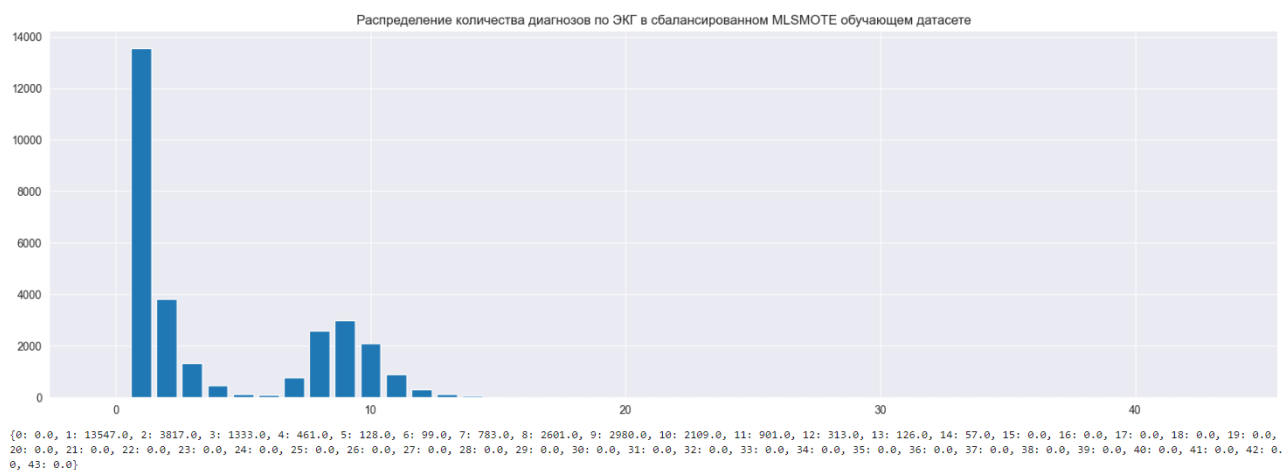


Рисунок 18 – Распределение количества меток на сбалансированных MLSMOTE примерах

## 5 Построение моделей

Как уже было сказано ранее, дальнейшее обучение проведено на трёх вариантах обучающей выборки. Среди основных методов обработки, используемых для обработки ЭКГ можно выделить рекуррентные и свёрточные нейросети. Рассмотрим три варианта архитектуры моделей.

На вход каждой модели подаётся батч ЭКГ, представляющих собой временной ряд из 12 признаков. На выходе каждой модели ожидается вектор из 44 значений, отвечающих за степень уверенности в наличии каждого из 44 возможных диагнозов.

Для подсчёта метрик качества необходимо привести выход модели к бинарному состоянию. Осуществим это с помощью применения сигмоидной функции к выходу и пороговой активации:

$$out' = \begin{cases} 1, & sigmoid(out) \geq threshold \\ 0, & sigmoid(out) < threshold \end{cases} \quad (6)$$

,где  $out'$  – обработанный выход модели  $\{0, 1\}$ ;  $out$  – необработанный выход модели  $(-\infty, +\infty)$ ;  $threshold$  – пороговое значение для отметки принадлежности к классу.

На практике в качестве порогового значения используем 0,5, эксперименты показали, что значения метрик слабо чувствительны к изменению данного значения.

В качестве функции потерь для каждой модели будем использовать BCEWithLogitsLoss, так как данная функция потерь адаптирована под задачи multilabel классификации [11]. Во всех случаях будем использовать градиентный спуск модификации ADAM.

## 5.1 Рекуррентная модель

Одним из способов обработки последовательности и учёта контекста является использование рекуррентных ячеек. Практика показывает, что использование ячеек GRU позволяет добиться наилучших результатов. К этому же выводу мы пришли, эмпирическим путём, проверяя различные варианты архитектуры.

В соответствии со спецификой данных, у нас нет нужды обрабатывать временной ряд в реальном времени, благодаря чему у нас есть возможность использовать двунаправленные ячейки. Также отметим, что для усложнения модели и улучшения обработки данных применены многослойные GRU ячейки, между которыми расположены dropout слои, позволяющие снизить влияние переобучения.

Используем 4-слойные GRU ячейку, между слоями dropout с вероятностью 0,25, на выходе линейный полносвязный слой. На рисунке 19 представлена архитектура полученной рекуррентной модели.

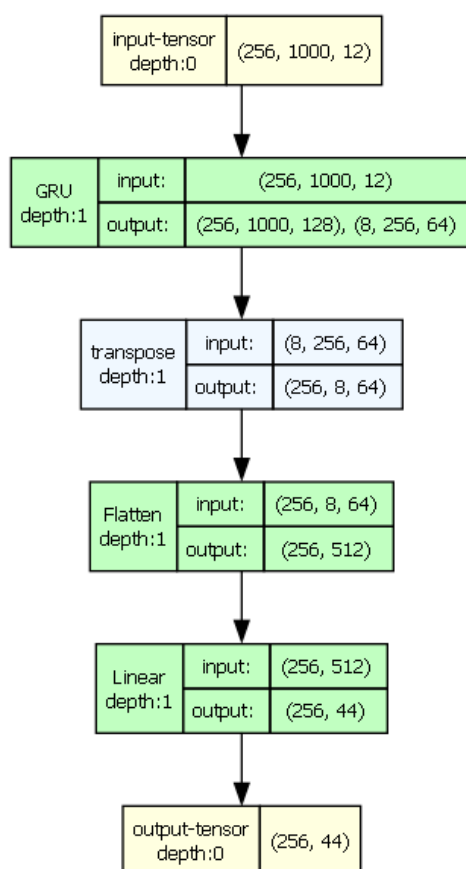


Рисунок 19 – Архитектура рекуррентной модели



## 5.2 Базовая свёрточная модель

Более популярным подходом к обработке сигнала ЭКГ являются свёрточные модели. Для начала построим более простую свёрточную модель, чтобы проанализировать положительный эффект от дальнейшего усложнения архитектуры.

Построим модель, состоящую из пяти одномерных свёрточных слоёв, последовательно увеличивающих число каналов с 12 до 512, затем уменьшающих их до 128. Также будем постепенно уменьшать длину последовательности, между свёрточными слоями разместим dropout слои регуляризации и слои MaxPool1d, позволяющие подсветить найденные свёрточными слоями паттерны. Для формирования выхода используем глобальный усредняющий пулинг и полносвязный слой. Архитектура данной модели изображена на рисунке 20.

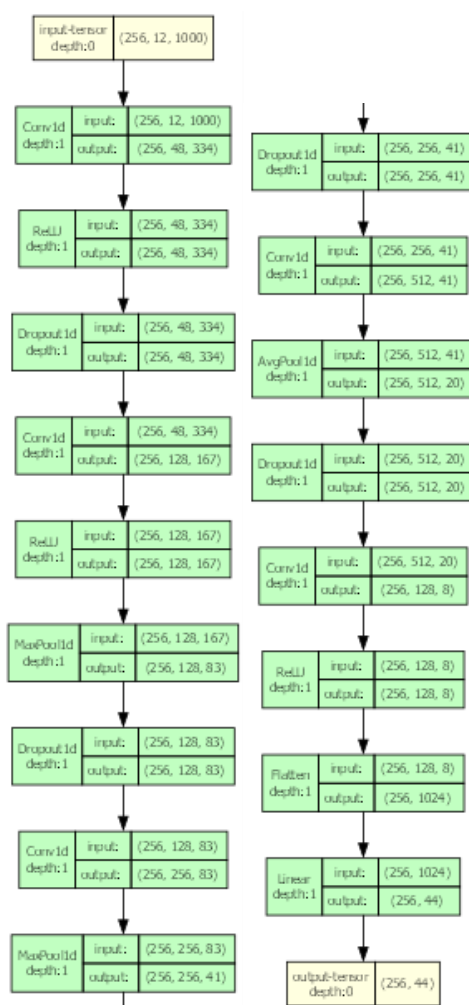


Рисунок 20 – Архитектура базовой свёрточной модели

### 5.3 Улучшенная свёрточная модель

Улучшение архитектуры свёрточной нейросети возможно несколькими способами. В данном случае заменим пять свёрточных слоёв на пять блоков, содержащих по три промежуточных свёрточных слоя с более плавным увеличением количества каналов. Также добавим в разработанные блоки skip-connection`ы и батч-нормализацию. Полученная архитектура в некоторой степени переносит архитектуру ResNet блоков в одномерную задачу обработки ЭКГ сигнала. Как и в предыдущем случае после всех блоков применим глобальный усредняющий пулинг и полносвязный слой для формирования выхода модели. Архитектура данной модели представлена на рисунках 21 и 22.

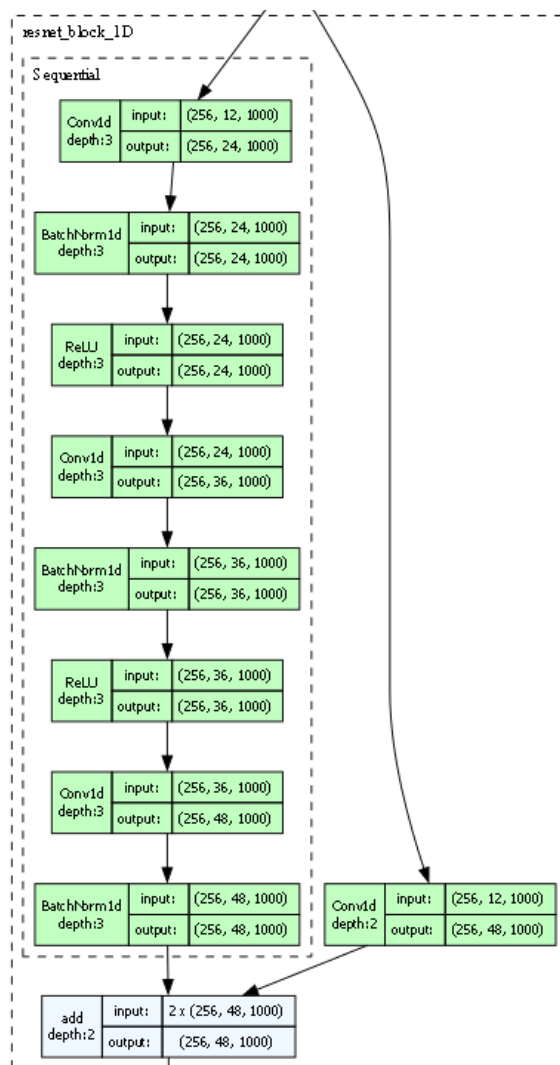


Рисунок 21 – Архитектура ResNet блока для одномерной задачи

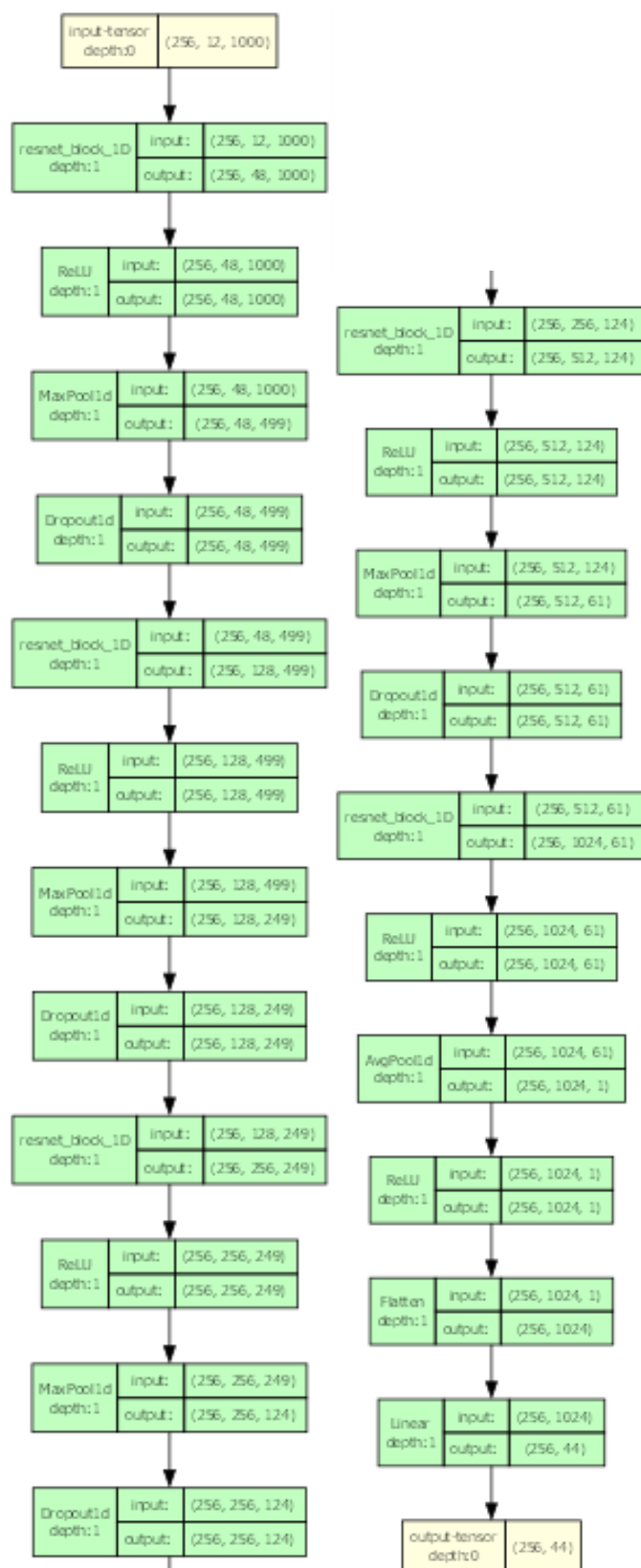


Рисунок 22 – Архитектура улучшенной свёрточной модели

## 6 Результаты

Как уже было сказано ранее, оценку качества работы моделей будем проводить по метрикам accuracy, precision и F1-score. Данные метрики рассчитаны на неизменённой тестовой выборке для каждого класса в отдельности. Сами модели были обучены на трёх вариантах обучающей выборки с различными вариантами балансировки данных. Итого было обучено 9 моделей, для каждой из которых посчитано 3 метрики для каждого из 44 классов. Общая таблица результатов расположена в приложении А.

Основу сравнительного анализа проведём относительно показаний метрики F1-score, так как она является наиболее информативной. Значения Accuracy не являются достаточно содержательными для редко встречающихся классов, что подтверждает тезис, выдвинутый в литературном обзоре.

Проанализируем результаты относительно различных способов балансировки обучающей выборки. Мы можем наблюдать достаточно интересные результаты, а именно отсутствие явного фаворита. В полученных результатах присутствуют классы, для которых оказались наиболее полезны как авторская балансировка, так и MLSMOTE, а в некоторых случаях наилучший результат наблюдался и при полном отсутствии балансировки.

При этом отметим, что для классов с наибольшим числом примеров лучшие результаты показал вариант с отсутствием балансировки, что можно объяснить внесением большого числа синтетических данных в обучающую выборку. Также по таблице видно, что балансировка не является идеальной и остались классы с чрезвычайно малым числом примеров, которые так и не были распознаны ни одной из моделей. Но отметим, что присутствуют и ситуации, когда исходной выборки оказалось недостаточно для качественного обучения распознаванию классов, что наблюдается, например, для классов «IVCD», «1AVB», «ALMI», «WPW»

и некоторых других. При этом чаще всего положительное влияние балансировки наблюдается для рекуррентной и улучшенной свёрточной моделей.

Анализируя показатели F1 метрики для разных архитектур, заметим, что базовая свёрточная модель практически никогда не превосходит улучшенную свёрточную модель. В то же время затруднительно выбрать предпочтительную модель среди рекуррентной и улучшенной свёрточной, в зависимости от класса победитель будет меняться.

Даже не учитывая очевидно неудачные классы, общие показатели F1 метрики оказались, хотя и близки, но всё же несколько ниже чем в аналогичном исследовании [7]. Это свидетельствует о возможности дальнейшего усовершенствования модели.

## **6.1 Дальнейшие пути развития**

Исходя из сказанного выше выделим следующие пути развития существующей модели.

Расширение обучающей базы модели. Эксперимент показал, что для разработки качественной системы поддержки принятия решения при вынесении диагноза нужно значительно большее количество примеров с редкими классами. Этого можно достичь, например, объединив несколько датасетов.

Усовершенствование алгоритма балансировки классов. Даже при использовании нескольких датасетов нормальных ЭКГ всё равно будет значительно больше. Также отметим, что уже текущий эксперимент показал, что теоретически применение алгоритма балансировки может значительно повысить качество работы для некоторых классов. Необходимо объединить и оптимизировать авторский метод выбора примеров и метод генерации синтетических данных MLS-MOTE.

Усовершенствование архитектуры нейросети. Исходя из эксперимента, в некоторых случаях предпочтительна оказалась свёрточная архитектура, а в некоторых – рекуррентная. Также эксперимент показал, что усложнение модели относительно базовой свёрточной архитектуры позволяет значительно поднять качество работы. В совокупности это свидетельствует о необходимости дальнейшего усложнения модели, чего можно достигнуть, например, объединив рекуррентную структуру и поиск признаков на основе свёрточной сети [6]. Альтернативой рекуррентной сети в данном случае может служить механизм внимания.

Также возможным путём развития может служить внедрение дополнительного канала поиска признаков, например, на основе вейвлет преобразования и использования двумерной свёрточной сети.

Все эти методы должны позволить применять подобную систему для автоматизации работы врача функциональной диагностики и создания полноценной системы поддержки принятия решений для кардиолога. Конечно же, в дальнейшем потребуются также система адаптации различных частот работы аппаратов ЭКГ, а также увеличение количества извлекаемой информации, но данные вопросы выходят за рамки предмета исследования.

*Исходный код, полная сравнительная таблица, рисунки архитектуры в хорошем разрешении и другие материалы расположены на*  
*<https://github.com/SergeiFG/ECG-Classifier>*

## Список литературы

- 1 Электрокардиография: учебн. пособие / В.В.Мурашко, А.В.Струтын-ский. – 18-е изд. – Москва : МЕДпресс-информ, 2022. – 360 с. : ил.
- 2 Sánchez, F. a. R., & Cervera, J. a. G. (2019). ECG classification using artificial neural networks. *Journal of Physics*, 1221(1), 012062. <https://doi.org/10.1088/1742-6596/1221/1/012062>
- 3 Rashed-Al-Mahfuz, M., Moni, M. A., Lió, P., Islam, S. M. S., Berkovsky, S., Khushi, M., & Quinn, J. M. (2021). Deep convolutional neural networks based ECG beats classification to diagnose cardiovascular conditions. *Biomedical Engineering Letters*, 11(2), 147–162. <https://doi.org/10.1007/s13534-021-00185-w>
- 4 Wu, M., Lu, Y., Yang, W., & Wong, S. Y. (2021). A study on arrhythmia via ECG signal classification using the convolutional neural network. *Frontiers in Computational Neuroscience*, 14. <https://doi.org/10.3389/fncom.2020.564015>
- 5 Goodfellow, S. D., Goodwin, A., Greer, R. O., Laussen, P. C., Mazwi, M., & Eytan, D. (2018). Towards understanding ECG rhythm classification using convolutional neural networks and attention mappings. *Machine Learning for Healthcare Conference*, 83–101. <http://proceedings.mlr.press/v85/goodfellow18a/goodfellow18a.pdf>
- 6 Sigurthorsdottir, H., Van Zaen, J., Delgado-Gonzalo, R., & Lemay, M. (2020). ECG classification with a convolutional recurrent neural network. *Computing in Cardiology (CinC)*, 2012. <https://doi.org/10.22489/cinc.2020.198>
- 7 Zhu, H., Cheng, C., Yin, H., Li, X., Zuo, P., Ding, J., Lin, F., Wang, J., Zhou, B., Li, Y., Hu, S., Xiong, Y., Wang, B., Wan, G., Yang, X., & Yuan, Y. (2020). Automatic multilabel electrocardiogram diagnosis of heart rhythm or conduction abnormalities with deep learning: a cohort study. *The Lancet Digital Health*, 2(7), e348–e357. [https://doi.org/10.1016/s2589-7500\(20\)30107-2](https://doi.org/10.1016/s2589-7500(20)30107-2)
- 8 PTB-XL ECG dataset. (2021, February 3). Kaggle. <https://www.kaggle.com/datasets/khyeh0719/ptb-xl-dataset/data> (дата обращения 05.02.2024)

- 9 Tolgadincer. (2020, September 28). Upsampling Multilabel Data with MLSMOTE. Kaggle. <https://www.kaggle.com/code/tolgadincer/upsampling-multilabel-data-with-mlsmote> (дата обращения 05.02.2024)
- 10 Charte, F., Rivera, A. J., Del Jesus, M. J., & Herrera, F. (2015). MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation. Knowledge-Based Systems, 89, 385–397. <https://doi.org/10.1016/j.knosys.2015.07.019>
- 11 BCEWithLogitsLoss — PyTorch 2.2 documentation. (n.d.). <https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html> (дата обращения 05.02.2024)
- 12 SergeiFG. (n.d.). GitHub - SergeiFG/ECG-Classifier. GitHub. <https://github.com/SergeiFG/ECG-Classifier>



## Приложение А. Сравнительная таблица рассмотренных моделей

	Количество примеров	GRU Исходная выборка accuracy_score	GRU Авторская балансировка accuracy_score	GRU MLSMOTE accuracy_score	Свёрточная базовая Исходная выборка accuracy_score	Свёрточная базовая Авторская балансировка accuracy_score	Свёрточная базовая MLSMOTE accuracy_score	Свёрточная продвинутая Исходная выборка accuracy_score	Свёрточная продвинутая Авторская балансировка accuracy_score	Свёрточная продвинутая MLSMOTE accuracy_score
LAFB	163	0,9611	0,9565	0,9574	0,9672	0,9551	0,9537	0,9676	0,9616	0,9658
IRBBB	112	0,9635	0,9542	0,9454	0,9528	0,9519	0,9426	0,9607	0,9621	0,9616
DIG	18	0,9917	0,9907	0,9866	0,9912	0,9907	0,9903	0,9912	0,9875	0,9894
ISCLA	14	0,9935	0,9931	0,9931	0,9935	0,9931	0,9921	0,9935	0,9926	0,9926
IVCD	79	0,9602	0,9616	0,9616	0,9621	0,9621	0,9611	0,9639	0,9611	0,9616
1AVB	80	0,9630	0,9570	0,9625	0,9621	0,9579	0,9607	0,9658	0,9667	0,9672
ISC_	128	0,9570	0,9533	0,9588	0,9537	0,9482	0,9528	0,9537	0,9561	0,9579
LAO/LAE	42	0,9806	0,9792	0,9810	0,9792	0,9746	0,9792	0,9801	0,9801	0,9796
3AVB	2	0,9991	0,9991	0,9991	0,9991	0,9991	0,9991	0,9991	0,9991	0,9991
IPLMI	5	0,9963	0,9977	0,9977	0,9977	0,9968	0,9977	0,9977	0,9972	0,9977
PMI	2	0,9991	0,9991	0,9986	0,9991	0,9991	0,9991	0,9991	0,9991	0,9991
IMI	268	0,8936	0,8936	0,8881	0,9006	0,8862	0,8802	0,9001	0,8904	0,8964
ALMI	29	0,9912	0,9880	0,9870	0,9870	0,9847	0,9884	0,9889	0,9894	0,9880
ISCAS	17	0,9917	0,9907	0,9755	0,9921	0,9894	0,9806	0,9912	0,9903	0,9907
LVH	214	0,9325	0,9325	0,9315	0,9274	0,9200	0,9246	0,9288	0,9278	0,9255
NORM	964	0,8645	0,8441	0,8571	0,8520	0,8358	0,8104	0,8728	0,8691	0,8464
ISCAN	4	0,9981	0,9972	0,9968	0,9981	0,9977	0,9958	0,9981	0,9981	0,9972
INJAS	22	0,9880	0,9870	0,9889	0,9898	0,9884	0,9898	0,9898	0,9912	0,9912
RVH	12	0,9935	0,9921	0,9898	0,9944	0,9870	0,9912	0,9949	0,9944	0,9944
ANEUR	10	0,9954	0,9949	0,9907	0,9954	0,9949	0,9926	0,9954	0,9940	0,9935
IPMI	3	0,9986	0,9981	0,9986	0,9986	0,9986	0,9986	0,9986	0,9986	0,9986
SEHYP	3	0,9986	0,9981	0,9991	0,9986	0,9986	0,9991	0,9986	0,9986	0,9991

ILMI	48	0,9787	0,9787	0,9621	0,9792	0,9764	0,9690	0,9820	0,9810	0,9778
LMI	20	0,9907	0,9894	0,9898	0,9907	0,9903	0,9907	0,9907	0,9907	0,9903
ISCIN	22	0,9903	0,9894	0,9903	0,9898	0,9884	0,9898	0,9889	0,9870	0,9903
AMI	35	0,9829	0,9833	0,9829	0,9838	0,9833	0,9820	0,9838	0,9829	0,9824
LNGQT	11	0,9949	0,9949	0,9935	0,9949	0,9949	0,9944	0,9940	0,9935	0,9935
2AVB	1	0,9995	0,9991	0,9995	0,9995	0,9995	0,9995	0,9995	0,9991	0,9995
EL	9	0,9958	0,9954	0,9940	0,9958	0,9954	0,9954	0,9958	0,9958	0,9958
NDT	183	0,9339	0,9158	0,9103	0,9214	0,9103	0,9098	0,9186	0,9149	0,8978
WPW	8	0,9968	0,9963	0,9963	0,9968	0,9963	0,9963	0,9963	0,9972	0,9981
NST_	52	0,9755	0,9722	0,9718	0,9722	0,9713	0,9709	0,9755	0,9695	0,9722
CLBBB	54	0,9935	0,9944	0,9935	0,9949	0,9917	0,9935	0,9921	0,9949	0,9972
CRBBB	54	0,9926	0,9884	0,9903	0,9917	0,9889	0,9884	0,9912	0,9917	0,9926
INJLA	2	0,9991	0,9991	0,9986	0,9991	0,9986	0,9991	0,9991	0,9991	0,9991
ISCAL	66	0,9685	0,9662	0,9625	0,9676	0,9676	0,9635	0,9685	0,9644	0,9690
ISCIL	18	0,9907	0,9907	0,9889	0,9917	0,9898	0,9898	0,9921	0,9907	0,9898
INJAL	14	0,9954	0,9917	0,9935	0,9944	0,9931	0,9931	0,9944	0,9917	0,9944
ILBBB	8	0,9949	0,9949	0,9958	0,9963	0,9940	0,9958	0,9958	0,9935	0,9954
RAO/RA E	10	0,9954	0,9944	0,9884	0,9954	0,9940	0,9912	0,9958	0,9958	0,9958
INJIN	2	0,9991	0,9986	0,9981	0,9991	0,9991	0,9991	0,9991	0,9991	0,9986
INJIL	2	0,9991	0,9986	0,9991	0,9991	0,9991	0,9991	0,9991	0,9991	0,9991
ASMI	236	0,9454	0,9376	0,9357	0,9426	0,9260	0,9195	0,9505	0,9463	0,9524
LPFB	18	0,9926	0,9926	0,9824	0,9935	0,9903	0,9889	0,9935	0,9931	0,9921

	Количество примеров	GRU Исходная выборка precision_score	GRU Авторская балансировка precision_score	GRU MLSMOTE precision_score	Свёрточная базовая Исходная выборка precision_score	Свёрточная базовая Авторская балансировка precision_score	Свёрточная базовая MLSMOTE precision_score	Свёрточная продвинутая Исходная выборка precision_score	Свёрточная продвинутая Авторская балансировка precision_score	Свёрточная продвинутая MLSMOTE precision_score
LAFB	163	0,7975	0,5828	0,6135	0,7178	0,6810	0,5092	0,7362	0,7362	0,7239
IRBBB	112	0,6161	0,4375	0,3571	0,6786	0,4018	0,1071	0,6696	0,3125	0,6339
DIG	18	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
ISCLA	14	0,0000	0,0000	0,0714	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
IVCD	79	0,0253	0,0759	0,0000	0,0380	0,0633	0,0253	0,0127	0,0253	0,0253
1AVB	80	0,0125	0,0750	0,0500	0,0750	0,0375	0,0125	0,1625	0,3500	0,3750
ISC_	128	0,5000	0,4531	0,5234	0,3125	0,3359	0,2969	0,3906	0,3828	0,4219
LAO/LAE	42	0,0000	0,0238	0,0238	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
3AVB	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,5000	0,0000
IPLMI	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
PMI	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
IMI	268	0,5522	0,5037	0,4515	0,5336	0,2724	0,2463	0,6269	0,6269	0,2537
ALMI	29	0,4138	0,2069	0,2069	0,0690	0,1724	0,1379	0,1724	0,3793	0,2069
ISCAS	17	0,0588	0,0000	0,0588	0,0000	0,0000	0,0000	0,0588	0,0000	0,0588
LVH	214	0,4953	0,4720	0,4626	0,3738	0,2850	0,2850	0,3832	0,3318	0,3318
NORM	964	0,8838	0,9118	0,9129	0,9315	0,9367	0,9793	0,8849	0,8786	0,9523
ISCAN	4	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
INJAS	22	0,1818	0,1818	0,1818	0,1364	0,1364	0,0909	0,3182	0,4545	0,1818
RVH	12	0,0833	0,1667	0,1667	0,0000	0,1667	0,0000	0,0833	0,0833	0,0833
ANEUR	10	0,0000	0,0000	0,1000	0,1000	0,0000	0,1000	0,0000	0,2000	0,2000
IPMI	3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
SEHYP	3	0,0000	0,0000	0,3333	0,0000	0,0000	0,3333	0,0000	0,0000	0,3333
ILMI	48	0,4792	0,2292	0,0625	0,1250	0,1042	0,0208	0,2500	0,5417	0,0625
LMI	20	0,0000	0,0000	0,0500	0,0000	0,0000	0,0500	0,0000	0,0500	0,0000

ISCIN	22	0,0455	0,0909	0,0455	0,0455	0,0000	0,0000	0,0455	0,0455	0,0455
AMI	35	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
LNGQT	11	0,0000	0,0000	0,0909	0,0000	0,0000	0,0000	0,1818	0,0909	0,0909
2AVB	1	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
EL	9	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
NDT	183	0,4262	0,3880	0,5027	0,1803	0,1858	0,0874	0,5683	0,4809	0,6066
WPW	8	0,1250	0,0000	0,0000	0,1250	0,1250	0,1250	0,0000	0,3750	0,5000
NST_	52	0,0000	0,0192	0,0192	0,0192	0,0769	0,0000	0,0000	0,0769	0,0385
CLBBB	54	0,8148	0,8333	0,8889	0,8704	0,8148	0,9074	0,7222	0,8333	0,9259
CRBBB	54	0,9074	0,7407	0,8333	0,9259	0,8704	0,7222	0,7778	0,8704	0,8333
INJLA	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
ISCAL	66	0,1667	0,0909	0,0909	0,1212	0,1212	0,0606	0,1061	0,2273	0,0909
ISCIL	18	0,0000	0,1111	0,1111	0,0000	0,0556	0,0556	0,0556	0,0556	0,0556
INJAL	14	0,4286	0,1429	0,2857	0,1429	0,2857	0,2857	0,4286	0,3571	0,4286
ILBBB	8	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,6250	0,1250
RAO/R AE	10	0,0000	0,1000	0,1000	0,0000	0,1000	0,1000	0,1000	0,1000	0,1000
INJIN	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,5000	0,0000	0,5000	0,0000
INJIL	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
ASMI	236	0,7161	0,6441	0,6780	0,6568	0,5127	0,3983	0,6356	0,6822	0,7458
LPFB	18	0,4444	0,2778	0,2778	0,2222	0,1667	0,2778	0,2778	0,2778	0,1667

	Количество примеров	GRU Исходная выборка f1_score	GRU Авторская балансировка f1_score	GRU MLSMOTE f1_score	Свёрточная базовая Исходная выборка f1_score	Свёрточная базовая Авторская балансировка f1_score	Свёрточная базовая MLSMOTE f1_score	Свёрточная продвинутая Исходная выборка f1_score	Свёрточная продвинутая Авторская балансировка f1_score	Свёрточная продвинутая MLSMOTE f1_score
LAFB	163	0,7558	0,6690	0,6849	0,7672	0,6959	0,6241	0,7742	0,7430	0,7613
IRBBB	112	0,6359	0,4975	0,4040	0,5984	0,4639	0,1622	0,6383	0,4605	0,6311
DIG	18	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
ISCLA	14	0,0000	0,0000	0,1176	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
IVCD	79	0,0444	0,1263	0,0000	0,0682	0,1087	0,0455	0,0250	0,0455	0,0460
1AVB	80	0,0244	0,1143	0,0899	0,1277	0,0619	0,0230	0,2600	0,4375	0,4580
ISC_	128	0,5792	0,5346	0,6009	0,4444	0,4343	0,4270	0,5000	0,5078	0,5427
LAO/LAE	42	0,0000	0,0426	0,0465	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
3AVB	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,5000	0,0000
IPLMI	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
PMI	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
IMI	268	0,5627	0,5400	0,5000	0,5709	0,3724	0,3376	0,6087	0,5864	0,3778
ALMI	29	0,5581	0,3158	0,3000	0,1250	0,2326	0,2424	0,2941	0,4889	0,3158
ISCAS	17	0,1000	0,0000	0,0364	0,0000	0,0000	0,0000	0,0952	0,0000	0,0909
LVH	214	0,5922	0,5805	0,5723	0,5047	0,4136	0,4281	0,5157	0,4765	0,4686
NORM	964	0,8533	0,8391	0,8507	0,8488	0,8357	0,8216	0,8612	0,8569	0,8469
ISCAN	4	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
INJAS	22	0,2353	0,2222	0,2500	0,2143	0,1935	0,1538	0,3889	0,5128	0,2963
RVH	12	0,1250	0,1905	0,1538	0,0000	0,1250	0,0000	0,1538	0,1429	0,1429
ANEUR	10	0,0000	0,0000	0,0909	0,1667	0,0000	0,1111	0,0000	0,2353	0,2222
IPMI	3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
SEHYP	3	0,0000	0,0000	0,5000	0,0000	0,0000	0,5000	0,0000	0,0000	0,5000
ILMI	48	0,5000	0,3235	0,0682	0,2105	0,1639	0,0290	0,3810	0,5591	0,1111
LMI	20	0,0000	0,0000	0,0833	0,0000	0,0000	0,0909	0,0000	0,0909	0,0000
ISCIN	22	0,0870	0,1481	0,0870	0,0833	0,0000	0,0000	0,0769	0,0667	0,0870

AMI	35	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
LNGQT	11	0,0000	0,0000	0,1250	0,0000	0,0000	0,0000	0,2353	0,1250	0,1250
2AVB	1	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
EL	9	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
NDT	183	0,5217	0,4383	0,4868	0,2797	0,2595	0,1410	0,5417	0,4889	0,5011
WPW	8	0,2222	0,0000	0,0000	0,2222	0,2000	0,2000	0,0000	0,5000	0,6667
NST_	52	0,0000	0,0323	0,0317	0,0323	0,1143	0,0000	0,0000	0,1081	0,0625
CLBBB	54	0,8627	0,8824	0,8727	0,8952	0,8302	0,8750	0,8211	0,8911	0,9434
CRBBB	54	0,8596	0,7619	0,8108	0,8475	0,7966	0,7573	0,8155	0,8393	0,8491
INJLA	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
ISCAL	66	0,2444	0,1412	0,1290	0,1860	0,1860	0,0920	0,1707	0,2804	0,1519
ISCIL	18	0,0000	0,1667	0,1429	0,0000	0,0833	0,0833	0,1053	0,0909	0,0833
INJAL	14	0,5455	0,1818	0,3636	0,2500	0,3478	0,3478	0,5000	0,3571	0,5000
ILBBB	8	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,4167	0,1667
RAO/RAE	10	0,0000	0,1429	0,0741	0,0000	0,1333	0,0952	0,1818	0,1818	0,1818
INJIN	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,5000	0,0000	0,5000	0,0000
INJIL	2	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
ASMI	236	0,7412	0,6925	0,6972	0,7143	0,6020	0,5193	0,7371	0,7352	0,7736
LPFB	18	0,5000	0,3846	0,2083	0,3636	0,2222	0,2941	0,4167	0,4000	0,2609