

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра математического анализа и моделирования
Направление подготовки: 01.04.02 – Прикладная математика и информатика
Направленность (профиль) образовательной программы «Математическое и программное обеспечение информационных систем»

УТВЕРЖДАЮ
И.о. зав. кафедрой
_____ Н.Н. Максимова
«_____» _____ 2025 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: «Система интеллектуального анализа данных кардиомониторирования»

Исполнитель студент группы 31010м	_____	С.В. Колесников
	(подпись, дата)	
Научный руководитель доцент, канд. физ.-мат. наук	_____	Е.М. Веселова
	(подпись, дата)	
Руководитель научного содержания программы магистратуры доцент, канд. физ.-мат. наук	_____	Н.Н. Максимова
	(подпись, дата)	
Нормоконтроль	_____	В.О. Салмиянов
	(подпись, дата)	
Рецензент доцент, д-р физ.-мат. наук	_____	В.В. Нещименко
	(подпись, дата)	

Благовещенск 2025

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра математического анализа и моделирования

УТВЕРЖДАЮ
И.о. зав. кафедрой
_____ Н.Н. Максимова
« _____ » _____ 2025 г.

ЗАДАНИЕ

К магистерской диссертации студента Колесникова Семена Витальевича

1. Тема магистерской диссертации: Система интеллектуального анализа данных кардиомониторирования.

(утверждена приказом от 24.01.2025 г. № 162-уч)

2. Срок сдачи студентом законченной проекта: 24.06.2025 г.

3. Исходные данные к магистерской диссертации: формулировка темы научно-исследовательской работы, отчеты по практической подготовке (научно-исследовательской работе) за 1, 2, 3 семестры, отчет по преддипломной практике, средства автоматизации вычислений и интегрированная среда разработки программного обеспечения ППП MATLAB, учебная и научная литература.

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов): литературный обзор по теме интеллектуального анализа биомедицинских данных, инструментальная база и язык программирования Matlab, временные ряды (ЭКГ человека), открытые медицинские базы данных кардиомониторирования, спектральный и фрактальный виды анализа сигналов ЭКГ, простая модель нейросети классификации временных рядов.

5. Перечень материалов приложения (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.): скрипты разработанных программ, результаты работы программ в виде изображений.

6. Консультанты по магистерской диссертации: рецензент – Нецименко В.В., и.о. директора института компьютерных и инженерных наук, профессор кафедры физики ФГБОУ ВО «АмГУ», д-р физ.-мат. наук, доцент; нормоконтроль – Салмиянов В.О., ассистент.

7. Дата выдачи задания: 03.02.2025 г.

Руководитель магистерской диссертации: Веселова Елена Михайловна, доцент кафедры математического анализа и моделирования ФГБОУ ВО «АмГУ», канд. физ.-мат. наук, доцент

Задание принял к исполнению (03.02.2025): _____ С.В. Колесников

РЕФЕРАТ

Магистерская диссертация содержит 76 с., 31 рисунок, 9 приложений, 43 источника.

ЭЛЕКТРОКАРДИОГРАММА, ВРЕМЕННОЙ РЯД, КЛАССИФИКАЦИЯ, НЕЙРОСЕТЬ, ДАТАСЕТ

В работе исследуется задача предварительной диагностики сигналов электрокардиограммы человека на предмет наличия или отсутствия патологии.

Цель магистерской диссертации – разработка системы интеллектуальной диагностики и классификации динамических сигналов в приложениях к исследованию сердечно-сосудистой деятельности человека.

Данная тема работы является актуальной, так как объём собираемых данных при кардиомониторировании растёт, увеличивая нагрузку на специалистов, и, следовательно, снижаются качество и своевременность диагностики заболеваний сердечной системы.

В магистерской диссертации посредством применения спектральных методов анализа были выявлены характеристики сигналов электрокардиограммы человека. Сигналы, представляющие собой временные ряды, были взяты из открытой медицинской базы данных сети Интернет и преобразованы в массив числовых значений.

В работе проведен анализ методик и технологий, применяемых для обработки сигналов электрокардиограммы человека, из которых была выбрана простая модель нейросети классификации для достижения поставленной цели.

С использованием инструментария пакета прикладных программ Matlab разработано программное решение классификации сигналов электрокардиограммы человека на две категории: отсутствие патологии и её наличие.

СОДЕРЖАНИЕ

Введение	7
1 Биомедицинские сигналы, заданные временными рядами, и методы их интеллектуального анализа	11
1.1 Регистрация временных рядов при медицинской диагностике	11
1.2 Использование методов интеллектуального анализа для исследования динамических сигналов в биологии и медицине	16
1.2.1 Нейронные сети в биомедицине	16
1.2.2 Теоретические основы временных рядов	17
1.2.3 Современные методы классификации временных рядов	19
2 Нейронные сети как средство анализа динамических данных	21
2.1 Принцип работы нейросети при анализе данных	21
2.2 Типы анализа данных, основанные на нейронных сетях	24
2.3 Функционал работы с нейронными сетями ППП Matlab	25
2.4 Современное применение методов интеллектуального анализа в медицинских задачах	27
3 Медицинские базы данных как материал исследования и их анализ	33
3.1 Материальная база	33
3.2 Спектральный анализ: метод Фурье	36
3.3 R/S-анализ	38
3.4 Показатель Хигучи	41
3.5 Энтропия	43
3.6 Показатель Ляпунова	44
3.7 Квартиль Q1-Q4	45
3.8 Выводы по главе	46
4 Конфигурирование простой нейронной сети классификации временных рядов	48
4.1 Сведения о нейросети: структура, характеристики, области применения	48

4.2 Создание простейшей нейронной сети классификации временных рядов	50
4.3 Результаты обучения и тестирование нейросети	52
4.4 Выводы по главе	54
Заключение	55
Библиографический список	57
Приложение А. Скрипт преобразования Фурье в среде Matlab	63
Приложение Б. Скрипт R/S-анализа в среде Matlab	64
Приложение В. Скрипт расчёта показателя Хигучи в среде Matlab	66
Приложение Г. Скрипт расчёта энтропии в среде Matlab	67
Приложение Д. Скрипт расчёта показателя Ляпунова в среде Matlab	68
Приложение Е. Скрипт расчёта 1-4 квартилей в среде Matlab	69
Приложение З. Скрипт нейросети LVQ в среде Matlab	70
Приложение И. Скрипт функции TestDataSort	73
Приложение К	74

ВВЕДЕНИЕ

В современной медицине применяются различные методики мониторинга кардиальной функции организма. Они позволяют отслеживать работу сердечно-сосудистой системы, а также своевременно выявлять и предотвращать возможные заболевания и нарушения.

Однако объём собираемых данных при кардиомониторировании может быть очень большим, что затрудняет их анализ. В настоящее время наблюдается ускоренное развитие в области высоких технологий и рост объёмов данных в такой многоаспектной сфере, как медицина, что привело к активному внедрению интеллектуальных методов анализа, таких как статический анализ данных, спектральные методы, методы фрактального анализа, машинное обучение, искусственный интеллект, нейросети. В связи с чем разработка компьютерных систем диагностики и интеллектуального анализа данных кардиомониторирования представляет **актуальную** прикладную задачу [1-4].

Целью исследования является разработка системы интеллектуальной классификации динамических сигналов в приложениях к исследованию кардиальной деятельности человека.

Методологическую базу исследований составляют нейросетевые и нейро-нечёткие технологии анализа временных рядов с использованием инструментальных средств платформы ППП Matlab.

Объектом исследования являются динамические временные ряды, характеризующие кардиальную деятельность человека. Соответственно, **предметом исследования** – программные инструменты анализа динамических сигналов, разработанные на основе нейронных сетей.

Гипотеза научного исследования заключается в возможности разработки системы интеллектуального анализа кардиосигналов, позволяющей решать задачу предварительной диагностики на предмет наличия или отсутствия патологии у наблюдаемых пациентов.

Для достижения цели необходимо выполнение следующих **задач**:

– изучение научной и методической литературы по теме интеллектуального анализа биомедицинских данных: использование нейросетей для задач классификации данных, заданных временными рядами; классификация биомедицинских данных, описываемых временными рядами, методы интеллектуального анализа;

– получение расширенного опыта работы с инструментарием пакета ППП Matlab: создание скриптов программ и функций для анализа нестационарных временных рядов;

– поиск открытых медицинских баз данных кардиомониторирования в свободном доступе сети Интернет;

– проведение спектрального, фрактального видов анализа сигналов ЭКГ (методы Фурье- и вейвлет-разложения, R/S-анализ, анализ Хигучи), а также выявление статистических характеристик (энтропия, показатель Ляпунова, квартиль 1-4) на основе найденных медицинских данных;

– реализация простого примера конфигурирования нейросети; постановка задачи классификации временных рядов;

– получение результатов (метрик) обучения нейросети на основе датасета; проведение тестирования программного решения на новом датасете;

– анализ результатов работы программного решения классификации сигналов ЭКГ.

Научная новизна работы заключается в разработке и анализе модели простой нейронной сети классификации временных рядов, представленных кардиальной деятельностью человека (ЭКГ), на группы «норма» и «патология». В работе проводится анализ временных рядов различными математическими методами, а также подбирается архитектурное решение для разрабатываемой нейросети для получения максимальной точности классификации.

Практическая значимость исследования определяется тем, что созданная модель может быть пригодна на практике для классификации сигналов ЭКГ и внедрена в медицине для ускорения и увеличения качества анализа, повышая

достоверность результатов и снижая рабочую нагрузку специалистов в условиях постоянно растущего объёма собираемых данных при кардиомониторировании.

Диссертация состоит из введения, четырёх глав, заключения, списка использованных источников и приложений.

В первой главе проводится обзор видов биомедицинских сигналов, заданных временными рядами, и методы их интеллектуального анализа, широко применяемые сегодня.

Во второй главе приведен обзор нейронных сетей, применяемых в качестве средства анализа динамических данных, современные методы их разработки в среде ППП Matlab, а также известные практики их применения.

В третьей главе представлена экспериментальная часть в области материальной базы исследования, организации и математического анализа датасета, найденного в свободном доступе сети Интернет.

В четвертой главе изложено развёрнутое описание разработанной нейросети классификации сигналов ЭКГ, её обучение на датасете и ручной тест. Приведены наглядные результаты работы и метрики. Заключение содержит основные выводы по работе.

Результаты работы докладывались и обсуждались на четырёх **научных конференциях**: XXXIII научная конференция Амурского государственного университета «День науки», секция «Анализ данных и моделирование», г. Благовещенск, 18.04.2024; XXV Региональная научно-практическая конференция «Молодёжь XXI века: шаг в будущее», секция «Информационные технологии», г. Благовещенск, 22.05.2024; XXXIV научная конференция Амурского государственного университета «День науки», секция «Анализ данных и моделирование», г. Благовещенск, 13.03.2025; VIII Всероссийская национальная научная конференция молодых учёных «Молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований», г. Комсомольск-на-Амуре, 07-11.04.2025.

Доклад на тему «Нейросетевой метод анализа данных кардиомониторирования» отмечен **дипломом I степени** на секции «Анализ данных и моделирование» XXXIV научной конференции Амурского государственного университета «День науки», 13 марта 2025 г.

По теме магистерской диссертации **опубликовано шесть работ** [5-10]: одна статья в журнале, входящем в перечень ВАК, «Информатика и системы управления» (2024 г.); одна статья в журнале, входящем в РИНЦ, Вестник АмГУ (серия «Естественные и экономические науки» (2025 г.) и четыре работы в сборниках материалов конференций.

1 БИОМЕДИЦИНСКИЕ СИГНАЛЫ, ЗАДАННЫЕ ВРЕМЕННЫМИ РЯДАМИ, И МЕТОДЫ ИХ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА

1.1 Регистрация временных рядов при медицинской диагностике

Биомедицинские сигналы – это сигналы, которые измеряются и записываются с организмов живых существ, в том числе с людей. Эти сигналы представляют собой электрические, магнитные, акустические или механические изменения, которые отражают функционирование различных систем организма. Биомедицинские сигналы широко используются в медицинской диагностике, мониторинге состояния пациента и исследованиях.

Ниже представлены сведения о некоторых наиболее распространённых биомедицинских сигналах [1].

1) электрокардиография (ЭКГ): измеряет электрическую активность сердца для оценки его работы и выявления отклонений. Это методика регистрации и исследования электрических полей, образующихся при работе сердца. Результатом электрокардиографии является получение электрокардиограммы, которая является электрическим проявлением сократительной активности сердца (рисунок 1.1). Используется для оценки его работы и выявления отклонений.

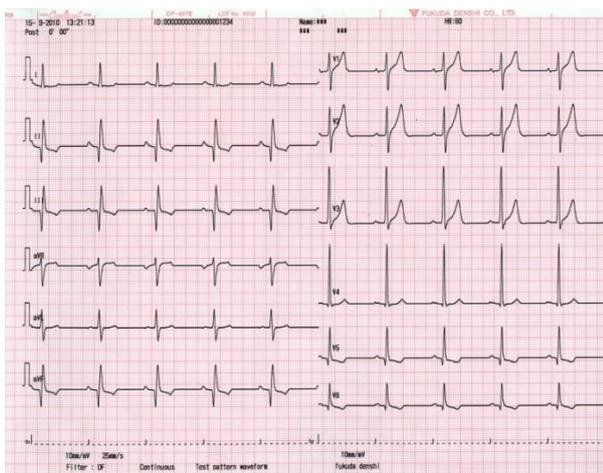


Рисунок 1.1 – Пример данных электрокардиограммы¹

¹ ARNAMEICAL [Электронный ресурс]. – Режим доступа : https://arnamedical.ru/doc/pic/prod_fukudadenshi/_jpg

2) электроэнцефалография (ЭЭГ): обеспечивает регистрацию электрической активности головного мозга (рисунок 1.2). Её изменения определяются функциональными изменениями центральной нервной системы, а по данным мощностных и пространственных характеристик ритмов головного мозга можно определить его дисфункции. Используется для диагностики эпилепсии, сна, комы и других состояний мозга.

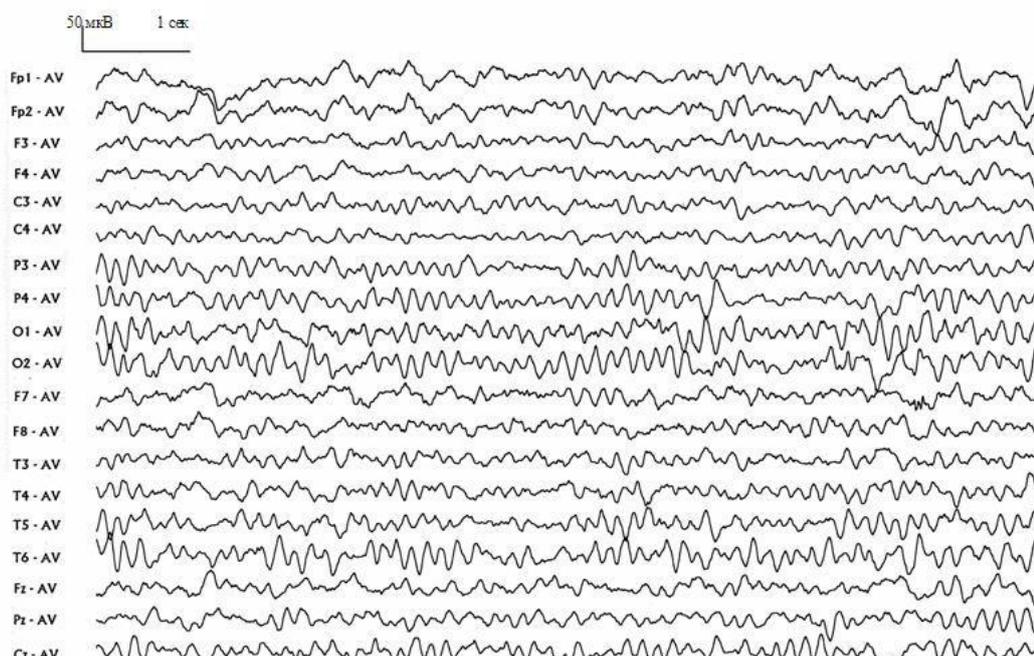


Рисунок 2.6. Мальчик 2 года. Здоров. На ЭЭГ в состоянии спокойного бодрствования регистрируется нестабильная синусоидальная альфа-активность индексом до 20%, частотой 6-8 кол/сек, преобладающая в теменно-затылочной области. Альфа-активность сочетается с тета- и дельта-колебаниями, имеющими наибольшую амплитуду в теменно-затылочных и задневисочных областях и диффузной, выраженной в виде групп волн, бета-активностью частотой 18-25 кол/сек, амплитудой до 10 мкВ.

Рисунок 1.2 – Пример данных электроэнцефалограммы

3) электромиография (ЭМГ): это регистрация электрических процессов в мышцах, то есть запись потенциалов действия мышечных волокон, которые заставляют их сокращаться. Поверхностная электромиограмма суммарно отражает разряды двигательных единиц, вызывающих сокращение (рисунок 1.3). Помимо этого, миограмма выступает как индикатор мышечного напряжения. Используется для выявления неврологических расстройств.

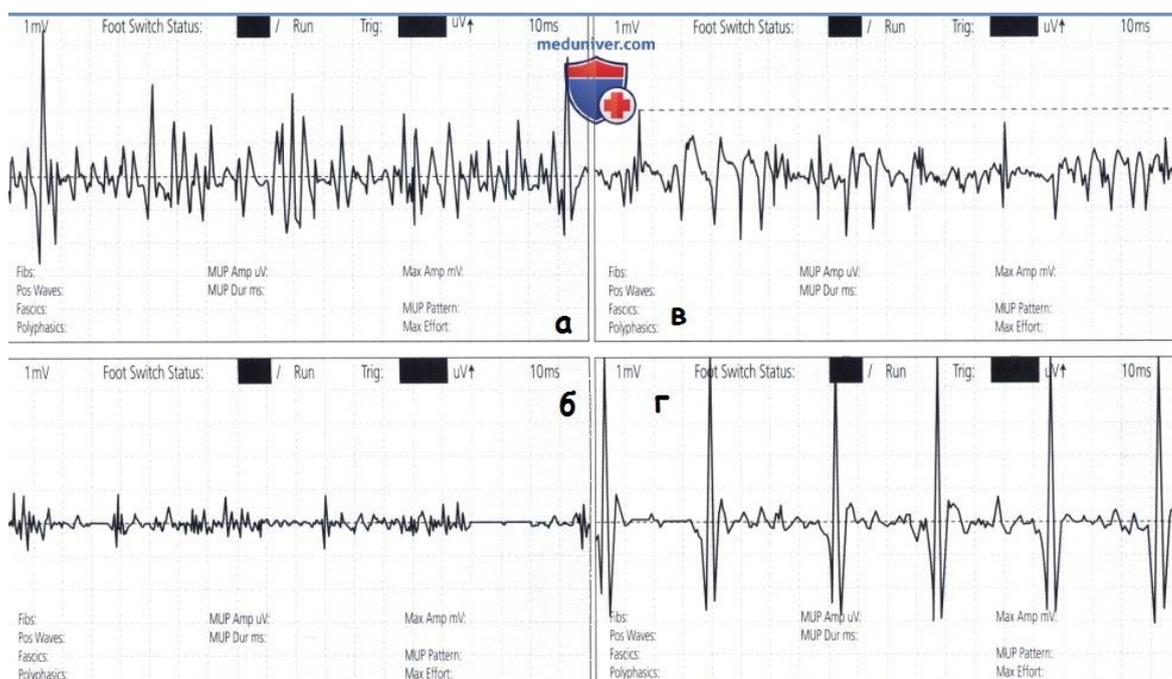


Рисунок 1.3 – Пример данных электромиограммы

4) фонокардиограмма представляет собой запись вибраций и звуковых сигналов, издаваемых при деятельности сердца и кровеносных сосудов (рисунок 1.4). Фонокардиограмма позволяет оценивать общее состояние сердца и кровеносных сосудов.

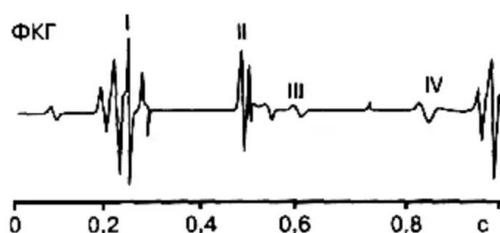


Рисунок 1.4 – Пример данных фонокардиограммы

5) сигналы потенциала действия: потенциал действия – это разновидность биопотенциала, возникающего на мембране электровозбудимых клеток в ответ на раздражение электрическим полем, химическим или другим стимулом. Потенциал действия является физиологической основой нервного импульса.

б) сигналы кожно-гальванической реакции: метод исследования кожно-гальванической реакции объединяет целый ряд показателей: уровень потенциала

кожи, реакцию потенциала, спонтанную реакцию потенциала, уровень сопротивления, реакцию сопротивления, спонтанную реакцию сопротивления. Возникновение электрической активности кожи обусловлено, главным образом, активностью потовых желёз в коже человека, которые в свою очередь находятся под контролем симпатической нервной системы.

7) сигналы реограммы: реография – это метод исследования центральной и периферической гемодинамики, основанной на изучении сопротивления тканей переменному электрическому току. Сигнал реограммы позволяет оценивать пульсирующую составляющую сопротивления тканей, которая возникает вследствие изменения интенсивности кровотока.

8) сигналы фотоплетизмограммы: фотоплетизмография в качестве измеряемой характеристики использует коэффициент светопропускания или светоотражения тканей. Регистрируемая плетизмограмма отражает изменения в объёме конечности или органа, вызванные изменениями количества находящейся в них крови.

9) спирограмма: записывает объём и скорость дыханий и используется в пульмонологии для диагностики и мониторинга функции лёгких (рисунок 1.5).

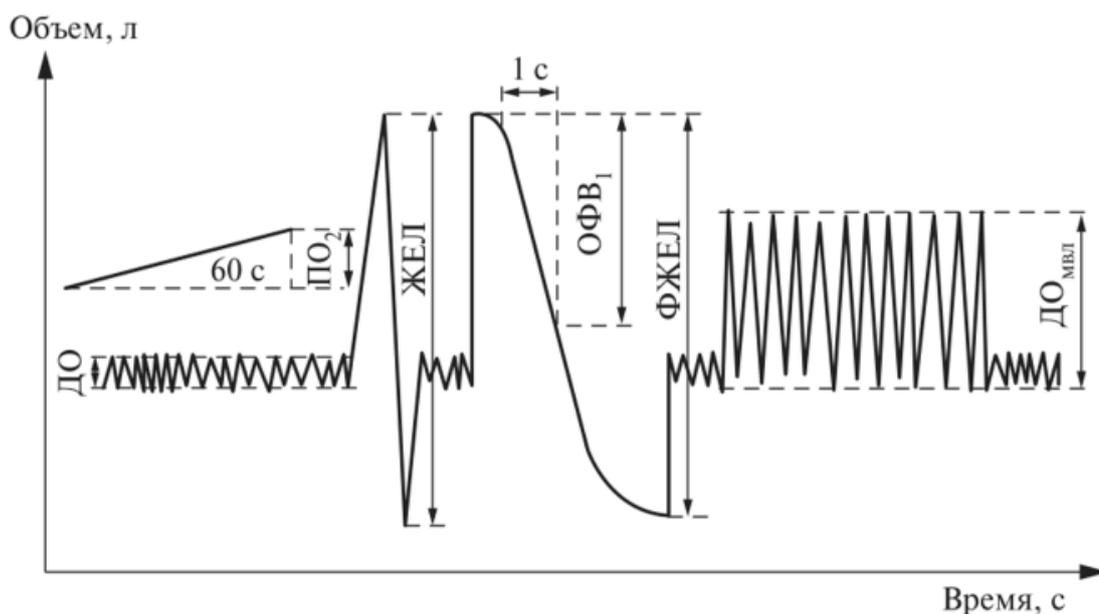


Рисунок 1.5 – Спирограмма

10) оксиметрия: измеряет уровень насыщения крови кислородом и часто используется в анестезиологии и реанимации.

11) артериальное давление: измерение давления в артериях для оценки сердечно-сосудистой системы.

12) речевой сигнал: чаще всего речевой сигнал рассматривается как сигнал для общения. Однако он может использоваться как диагностический, для исследования нарушений звукового и речевого трактов. Звук речи образуется за счёт прохождения выдыхаемого воздуха из легких в голосовой тракт (рисунок 1.6) [1].

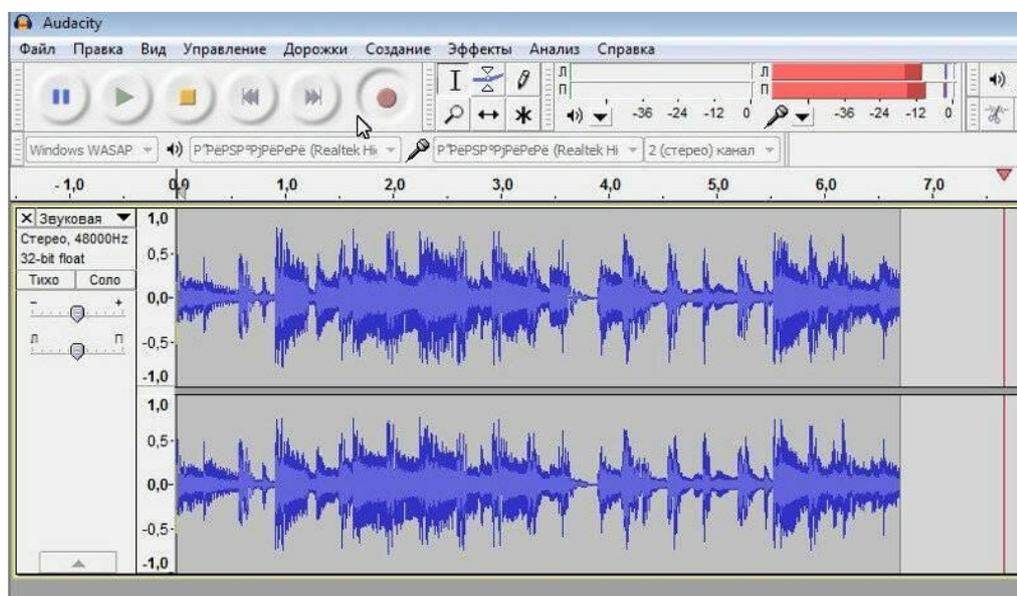


Рисунок 1.6 – Фиксация речевого сигнала

Каждый из этих сигналов отражает определенную физиологическую активность и имеет свои характеристики, такие как частота, амплитуда и форма. Измерение и анализ биомедицинских сигналов требуют специализированного оборудования и знаний в области сигнальной обработки и медицины.

Биомедицинские сигналы могут быть подвержены различным видам шумов и артефактов, которые могут исказить результаты измерений. Поэтому большое значение имеет обработка сигналов, которая включает фильтрацию, усиление, детектирование событий и многие другие методы обработки данных.

В настоящее время разрабатываются новые методы машинного обучения и искусственного интеллекта для более глубокого и точного анализа биомедицинских сигналов, что позволяет выявлять сложные взаимосвязи и паттерны, не всегда очевидные для человека. Применение передовых аналитических методов позволяет эффективно обрабатывать и интерпретировать сложные медицинские данные, что, в свою очередь, ведёт к повышению качества и доступности медицинского обслуживания [2, 3].

1.2 Использование методов интеллектуального анализа для исследования динамических сигналов в биологии и медицине

1.2.1 Нейронные сети в биомедицине

Нейронные сети позволяют автоматически находить сложные закономерности в заданной области и достигать высокой точности в решении данной задачи. Они широко используются во многих сферах, таких как финансы, прогнозирование погоды, обнаружение аномалий, медицинская диагностика и других. В современных медицинских и биомедицинских исследованиях существует несколько методов классификации данных, которые позволяют более эффективно работать с большим объемом информации и выявлять закономерности. Рассмотрим некоторые из этих методов [2-4, 11].

Первым и наиболее распространенным методом является *метод классификации на основе машинного обучения*. Он основан на создании модели, которая обучается на имеющихся данных и после этого может прогнозировать результаты для новых наблюдений. Для обучения модели используется алгоритм, который находит определенные закономерности в данных и строит математическую модель их описания. Этот подход позволяет классифицировать медицинские и биомедицинские данные с высокой точностью и скоростью.

Другой метод классификации, который часто применяется в медицинских исследованиях, – это метод временных рядов. Он используется для анализа данных, которые варьируются во времени, например, показатели сердечного ритма

или кровяного давления. Метод временных рядов позволяет исследователям выявлять тренды, цикличность или сезонные вариации в данных и использовать эту информацию для классификации и прогнозирования.

Также широко распространенный метод классификации – *метод глубокого обучения*. Он основан на использовании искусственных нейронных сетей, которые имитируют работу человеческого мозга. Глубокое обучение позволяет автоматически обнаруживать сложные структуры и закономерности в данных и использовать их для классификации. Этот метод активно применяется в медицинских и биомедицинских исследованиях, например, для классификации изображений рентгеновских снимков или для обнаружения опухолей на магнитно-резонансных томограммах.

Кроме того, применяются методы классификации на основе статистического анализа. Эти методы используют различные статистические признаки или характеристики данных, чтобы разделить их на классы. Например, можно использовать различные тесты или модели, чтобы определить, есть ли статистически значимые различия между двумя группами данных.

Наконец, в последние годы все большую популярность получают методы классификации на основе искусственного интеллекта, такие как использование нейронных сетей или генетических алгоритмов. Эти методы позволяют анализировать большие объемы данных и выявлять скрытые закономерности. Они используются для классификации данных с высокой точностью и для прогнозирования течения заболеваний или эффективности лечения [2-4, 11].

В целом, современные методы классификации медицинских и биомедицинских данных являются эффективными инструментами для анализа больших объемов информации. Они позволяют выявлять закономерности и улучшать диагностику, прогнозирование и лечение различных заболеваний. Благодаря постоянному развитию технологий и появлению новых методов, исследователям открываются все больше возможностей для более точного и эффективного анализа медицинских данных.

1.2.2 Теоретические основы временных рядов

Временные ряды представляют собой последовательности тех или иных данных, упорядоченных по критерию времени. Каждая точка данных соответствует определенному моменту времени. В медицинском и биомедицинском исследовании время является важным фактором, так как многие заболевания и физиологические процессы изменяются с течением времени. Поэтому методы классификации, специализированные на временных данных, становятся все более востребованными.

Метод временных рядов основан на анализе и понимании временного поведения данных. В основе метода лежит представление наборов данных как последовательности значений, полученных в течение определенного периода времени. Отслеживание данных во времени позволяет выявить закономерности, тренды, циклы и сезонности, которые могут быть полезны для классификации и прогнозирования. Эти подходы часто используются в комбинации с нейронными сетями для повышения точности и интерпретируемости результатов [4-6].

Временные ряды подразделяются на стационарные и нестационарные. Понятие стационарного временного ряда означает, что его среднее значение не изменяется во времени, то есть временной ряд не имеет тренда, дисперсия неизменная; статистические свойства не зависят от времени, поэтому результат легко предсказать, а вероятностные характеристики и показатели случайной величины неизменны. Например, рождаемость в той или иной стране не имеет выраженной сезонности, её рост или спад легко предсказать.

В случае с нестационарными временными рядами статистические свойства меняются со временем, то есть в промежутке времени одна или больше вероятностных характеристик показывает значительные изменения. Они показывают сезонные эффекты, тренды и другие структуры, которые зависят от временного показателя; им свойственна детерминистическая тенденция, а также случайное блуждание. Например, меняющееся число пассажиров на международных перелетах в зависимости от текущего сезона. Пример обоих видов временных рядов представлен на рисунке 1.7.

Временные ряды могут применяться в различных областях медицины и биомедицины:

1) мониторинг физиологических показателей: метод временных рядов позволяет анализировать и классифицировать данные, полученные от физиологических мониторов, таких как ЭКГ, пульсоксиметр или датчики кровяного давления. Это помогает в диагностике и прогнозировании сердечно-сосудистых заболеваний, а также контроле за состоянием пациента.

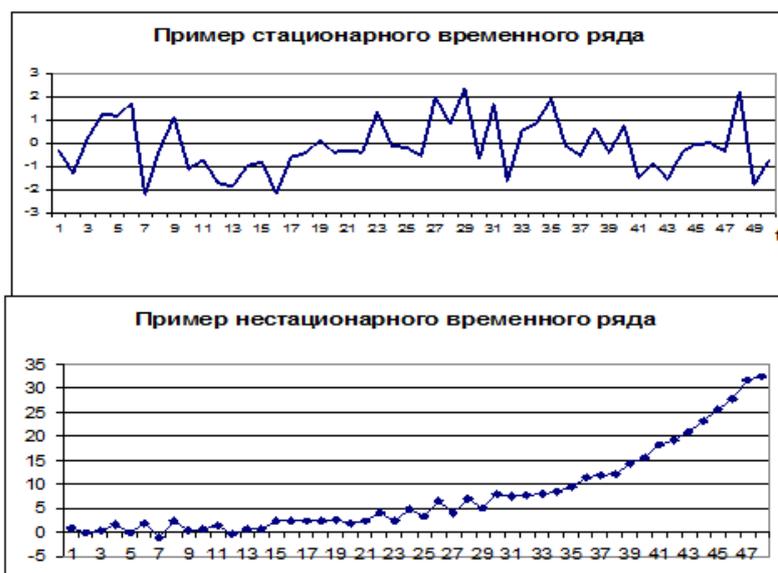


Рисунок 1.7 – Примеры стационарного и нестационарного временных рядов

2) изображения медицинских сканирований: метод временных рядов может быть применен к медицинским изображениям, таким как МРТ, КТ или УЗИ-сканы. Он позволяет анализировать изменения интенсивности пикселей во времени и классифицировать патологические изменения, такие как опухоли, кисты или воспалительные процессы.

3) анализ временных рядов лекарственного действия: метод временных рядов может использоваться для анализа данных, полученных во время клинических испытаний лекарственных препаратов. Он позволяет определить эффективность лекарственного вещества во времени и классифицировать его воздействие на пациентов [4, 11, 12].

1.2.3 Современные методы классификации временных рядов

Классификация временных рядов позволяет определить принадлежность ряда к определенному классу или категории.

Одним из наиболее распространенных подходов является использование *рекуррентных нейронных сетей* для обработки временных рядов. Они способны сохранять информацию о предыдущих состояниях нейронов и использовать ее для более точного предсказания и классификации данных. Long Short-Term Memory является одной из самых популярных архитектур RNN, позволяющей эффективно моделировать долгосрочные зависимости во временных рядах.

Другим часто используемым методом является *сверточная нейронная сеть (CNN)*. Хотя она изначально разработана для классификации изображений, её архитектура позволяет также работать с одномерными данными, такими как временные ряды. CNN способны автоматически извлекать важные признаки из данных и использовать их для классификации временных рядов.

Одна из недавних тенденций в классификации временных рядов – это использование глубоких нейронных сетей (DNN), которые состоят из нескольких слоев нейронов. Глубокие нейронные сети позволяют моделировать более сложные зависимости и достигать лучшей точности в классификации временных рядов. Они могут быть построены с использованием различных архитектур, включая комбинацию RNN и CNN, сети преобразования вейвлетов и другие.

Дополнительные методы, такие как *автоэнкодеры* и *вариационные автоэнкодеры*, также используются для классификации временных рядов. Автоэнкодеры позволяют извлекать самые релевантные признаки из данных путем сжатия и последующего восстановления, что способствует более эффективной классификации.

Обучение нейронных сетей для классификации временных рядов может осуществляться с использованием различных алгоритмов оптимизации, таких как градиентный спуск и его вариации, а также методы обратного распространения ошибки.

Кроме того, применяются методы объяснимого искусственного интеллекта (ХАИ), которые позволяют понять процесс принятия решений нейронных сетей, обеспечивая большую прозрачность и интерпретируемость [4, 11, 12].

2 НЕЙРОННЫЕ СЕТИ КАК СРЕДСТВО АНАЛИЗА ДИНАМИЧЕСКИХ ДАННЫХ

2.1 Принцип работы нейросети при анализе данных

Развитие технологии баз данных и систем управления базами данных, способствует росту объёма данных, хранящихся в базе. Ввиду этого многие компании используют технологию интеллектуального анализа данных, которая позволяет обрабатывать массивные базы данных и извлекать из них ценные сведения. Первоначально применение нейронных сетей в интеллектуальном анализе данных вызывало скептическое отношение ввиду недостатков, присущих нейронным сетям, таких как сложная структура, плохая интерпретируемость и долгое время обучения [13, 14]. Однако их преимущества, такие как, высокая допустимость к зашумленным данным и низкий коэффициент ошибок, непрерывное усовершенствование и оптимизация различных алгоритмов обучения сетей, алгоритма извлечения правил, алгоритма упрощения сетей, делают нейронные сети всё более и более перспективным направлением в интеллектуальном анализе [13, 14].

Области применения нейронных сетей обширны – автоматизация процессов распознавания образов, прогнозирование, адаптивное управление, создание экспертных систем, организация ассоциативной памяти, обработка аналоговых и цифровых сигналов, синтез и идентификация электронных цепей и систем [13, 15].

Таким образом, можно сказать, что использование нейронных сетей в технологии интеллектуального анализа данных является актуальным направлением, которое непрерывно развивается по пути устранения недостатков.

Анализ данных включает в себя три главные стадии: предварительная обработка данных, их анализ и последующая трактовка и интерпретация результатов (рисунок 2.1) [13, 14].



Рисунок 2.1 – Общий процесс анализа данных

Основанный на нейронной сети интеллектуальный анализ состоит из подготовки данных, извлечения правил и оценки правил (рисунок 2.2) [6, 14].

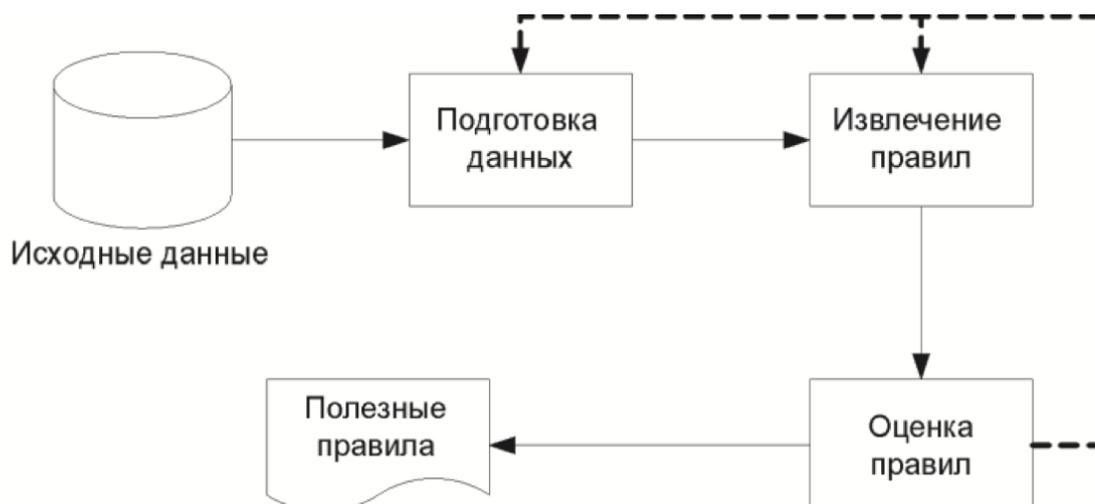


Рисунок 2.2 – Основанный на нейронной сети интеллектуальный анализ данных

Процесс подготовки данных играет решающую роль в анализе данных – он должен определить и обработать входящие данные, чтобы сделать их пригодными для конкретных методов интеллектуального анализа. Как правило, подготовка включает в себя четыре процесса [13]:

1) очистка данных (заполнение пустых значений, устранение зашумлённых данных, исправление несогласованности в данных);

2) выбор данных (определение расположения используемых в данном анализе данных);

3) предварительная обработка (является расширением процесса очистки данных, которые были выбраны);

4) выражение данных (преобразование после предварительной обработки в форму, которая может быть принята нейронной сетью).

Следует помнить, что анализ данных, основанный на нейронной сети, может работать только с числовыми данными (не символьными) [16].

Наиболее часто используемыми методами извлечения правил считаются метод чёрного ящика, метод извлечения нечётких правил, метод извлечения правил из рекурсивной сети, алгоритм извлечения правил двоичного входа и выхода (BIO-RE), алгоритм частичного извлечения правил (Partial-RE), метод Limited Relative Error (LRE) и алгоритм полного извлечения правил (Full-RE) [16].

Несмотря на то, что цель правил оценки зависит от конкретного применения, в общем они могут быть оценены в соответствии со следующими задачами:

1) найти оптимальную последовательность извлечения правил, за счёт чего в ряде определённых данных будут получены лучшие результаты;

2) проверить точность извлеченных правил;

3) определить количество знаний в нейронной сети, которые не были извлечены;

4) определить противоречия между извлеченными правилами и обученной нейросетью [16].

2.2 Типы анализа данных, основанные на нейронных сетях

Наиболее популярны среди множества типов интеллектуального анализа данных основаны на самоорганизующихся нейронных сетях и на нечётких сетях.

Анализ, основанный на самоорганизующейся сети, означает процесс обучения «без учителя», когда сеть учится понимать структуру данных. Идея такого принципа работы сети принадлежит финскому ученому Тойво Кохонену. Принцип работы этих сетей заключается во введении в правило обучения нейрона информации о его расположении, то есть составляются карты размещения нейронов. Эти сети устойчивы к зашумлённым данным, им свойственны быстрое неуправляемое обучение, возможность визуализации, возможность упрощения многомерной структуры. Область применения данного типа анализа весьма широка: классификация, кластерный анализ, прогнозирование, сжатие данных. Его недостатками определяются эвристичность алгоритма обучения, предопределенность числа кластеров [13].

Анализ, основанный на нечёткой нейронной сети, представляет собой использование существующей выборки данных для определения параметров функций принадлежности, выводы делаются на основе аппарата нечёткой логики, а для нахождения параметров функций принадлежности используются алгоритмы обучения нейронных сетей. Такие системы способны использовать заранее известную информацию, быстро обучаться, прогнозировать временные ряды, выполнять классификацию образов, интерпретировать накопленные знания, легко определять размер сети, аппроксимировать функции любой степени нелинейности, проводить параллельные вычисления. Выделяющимся достоинством является наглядность работы такой сети для пользователя. Область применения ограничена классификацией и прогнозированием, присущ недостаток априорности определения компонентов [13, 15].

Основное отличие сетей Кохонена от других типов нейронных сетей состоит в наглядности и удобстве использования, однако присущее им неуправляемое или неконтролируемое обучение позволяет задавать лишь значения вход-

ных переменных. Важнейшим преимуществом нейронечёткой сети является возможность построения одной сети для вычисления нескольких выходных значений по нескольким входным [13, 16], а также способность к логическому описанию процессов и ручной корректировке функций принадлежности [13, 17].

Из рассмотренных типов анализа данных, основанных на нейронных сетях, можно сделать вывод, что нейронные сети и системы нечёткой логики являются обязательным инструментом интеллектуального поиска и извлечения знаний, так как обладают способностью выявления значимых признаков и скрытых закономерностей в анализируемых экономических показателях [13].

2.3 Функционал работы с нейронными сетями ППП Matlab

MATLAB (от Matrix Laboratory) – это мощная среда и язык программирования высокого уровня, который широко используется инженерами, учеными, математиками, аналитиками и разработчиками для анализа данных, разработки алгоритмов, моделирования и вычислительного моделирования. Разработанный компанией MathWorks, MATLAB предлагает множество встроенных функций и инструментов, позволяющих быстро выполнять различные типы математических расчетов, обработку сигналов, машинное обучение, глубокое обучение и другие технические вычисления.

В MATLAB реализованы широкие возможности для работы с нейронными сетями с помощью специализированного инструментария, известного как Neural Network Toolbox или Deep Learning Toolbox.

Deep Learning Toolbox предоставляет алгоритмы, функции и приложения для моделирования и обучения глубоких нейронных сетей для решения различных задач, включая классификацию изображений, распознавание речи, прогнозирование и многие другие.

Основные возможности и компоненты Deep Learning Toolbox в MATLAB:

1) построение нейронных сетей:

1.1) поддержка различных архитектур нейронных сетей, таких как многослойные перцептроны (MLP), свёрточные нейронные сети (CNN), рекуррентные нейронные сети (RNN), LSTM (Long Short-Term Memory) и другие.

1.2) обширный набор слоёв: полносвязные, свёрточные, пулинговые (max и average), нормализующие, активационные и другие.

1.3) средства визуализации архитектуры нейронных сетей.

2) обучение и оценка моделей:

2.1) поддержка различных алгоритмов оптимизации, таких как SGD (Stochastic Gradient Descent), Adam, RMSprop и другие.

2.2) возможность использования графических процессоров (GPU) для ускорения процесса обучения.

2.3) инструменты для разделения данных на обучающие, валидационные и тестовые наборы.

2.4) автоматическое регулирование скорости обучения и ранняя остановка, чтобы предотвратить переобучение.

2.5) Визуализация процесса обучения с помощью графиков функции потерь и точности.

3) предобработка и аугментация данных:

3.1) встроенные функции для масштабирования, нормализации, кодирования и декодирования данных.

3.2) средства для аугментации данных, такие как поворот, масштабирование, обрезка и зеркальное отражение изображений для улучшения обобщающей способности модели.

4) Интеграция с MATLAB:

4.1) возможность интеграции с другими инструментами и тулбоксами MATLAB для предварительной обработки данных, выполнения статистического анализа, построения графиков и интерфейсов.

4.2) функции для импорта и экспорта моделей нейронных сетей из и в другие фреймворки глубокого обучения, например, TensorFlow и PyTorch.

5) прикладные решения:

5.1) готовые примеры использования нейронных сетей для различных задач машинного обучения.

5.2) поддержка трансферного обучения путём использования предварительно обученных моделей, таких как AlexNet, VGG, GoogleNet, ResNet и другие.

б) работа с аппаратным обеспечением:

б.1) возможность развёртывания обученных моделей на различных платформах, таких как серверы, мобильные устройства, IoT-устройства и даже FPGA или ASIC, с использованием MATLAB Coder и GPU Coder.

Это не все возможности Deep Learning Toolbox в MATLAB. Набор инструментов постоянно обновляется и расширяется, добавляя новые функции и улучшения, что делает его мощным инструментом для разработки и внедрения алгоритмов глубокого обучения.

2.4 Современное применение методов интеллектуального анализа в медицинских задачах

На сегодняшний день анализ и прогнозирование временных рядов используется во многих научных исследованиях. Основной целью анализа ВР является получение исчерпывающей информации, выявление всех возможных зависимостей из представленных данных: общий тренд развития, циклические и сезонные процессы, а также стохастическую (случайную) составляющую.

Одним из известных примеров применения нейросетей в медицинских задачах является исследование по оценке стадии сна на основе кардиореспираторных сигналов, полученных от датчика надгрудинного давления. Учёными из Эйндховенского технологического университета (Нидерланды) в домашних условиях с помощью портативной системы с датчиком супрастернального давления был проведён сбор кардиореспираторных данных (дыхательных усилий и вибраций сердца, передаваемых через трахею) у 100 взрослых людей с подозрением на нарушения сна, включая синдром апноэ, бессонницу и двигательные расстройства (рисунок 2.3).

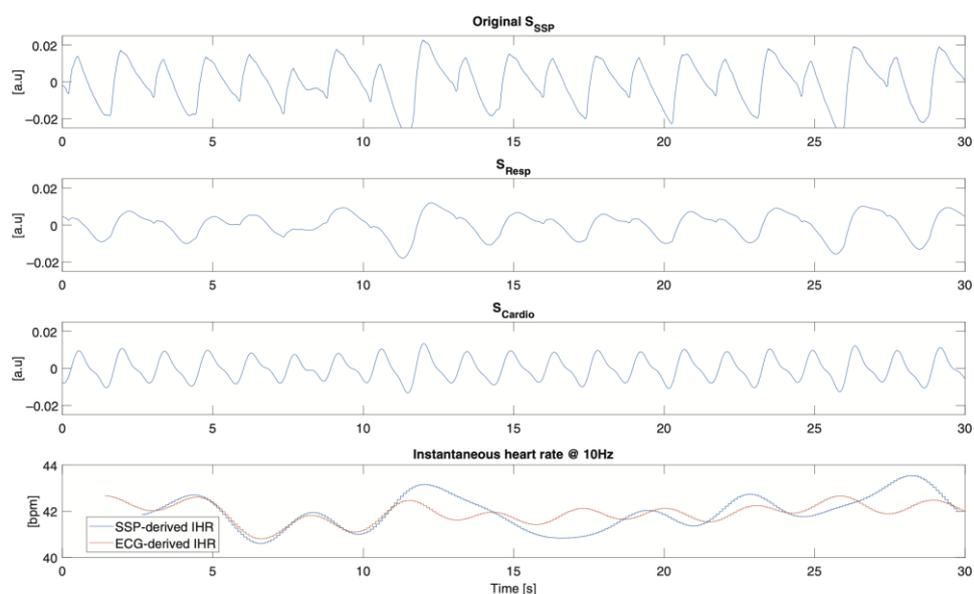


Рисунок 2.3 – Пример сигнала датчика супрастернального давления с кардиогенными колебаниями большой амплитуды

Далее сигналы, связанные с дыхательным усилием и сердечной деятельностью, были разделены и введены в нейронную сеть, обученную оценивать стадии сна. При использовании исходного смешанного сигнала результаты показали умеренное сходство с ручным подсчетом (рисунок 2.4): каппа Коэна составляет 0,53 при распознавании сна/Бодрствование/N1–N2/N3/быстрое движение глаз и 0,62 при распознавании во время сна/Бодрствование/Сон.

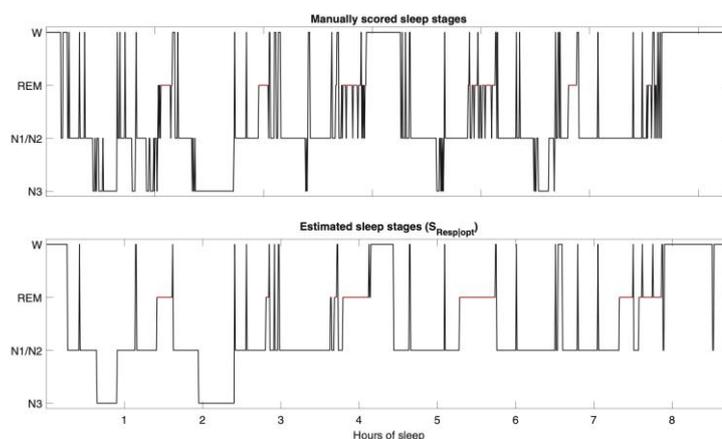


Рисунок 2.4 – Сравнение ручной оценки стадий сна (сверху) с оценкой нейросети (снизу)

Учёными был сделан вывод, что разделение двух сигналов и использование сердечного сигнала для оценки мгновенной частоты сердечных сокращений значительно улучшает процесс, достигая сходства в 0,63 и 0,71. Предложенный метод обеспечивает высокую точность, специфичность и чувствительность при выполнении различных задач по стадиям сна. Также учёными было сравнено общее время сна, рассчитанное с помощью того же метода, с проведённой вручную оценкой, – средняя ошибка составила 1,83 минуты, но с относительно большим доверительным интервалом ± 55 минут. Компактные системы, использующие насыщенный информацией сигнал датчика ССП, могут открыть новые способы клинической оценки, такие как вариабельность от ночи к ночи при обструктивном апноэ во сне и других нарушениях сна [18].

Похожее исследование стадий сна было проведено учёными Школы информационных наук и технологий Фуданьского университета (Шанхай, Китай). Учёные предположили, что эффективность автоматического определения стадий сна с использованием сигналов электроэнцефалографии (ЭЭГ) весьма высока, однако данный метод получения сигналов является громоздким и неудобным. Поэтому был предложен новый подход к определению стадий сна с использованием сигналов электроокулограммы (ЭОГ). Двухмасштабная свёрточная нейронная сеть сначала извлекает эпохальные временные эквивалентные функции из необработанных сигналов ЭОГ. Затем рекуррентная нейронная сеть захватывает долгосрочную последовательную информацию. Финальная архитектура разработанной нейросети обработки сигнала представлена на рисунке 2.5.

Предложенный метод был проверен на 101 образце данных о ночном сне из двух баз данных с открытым доступом (Монреальского архива исследований сна и Sleep-EDF) (рисунок 2.6).

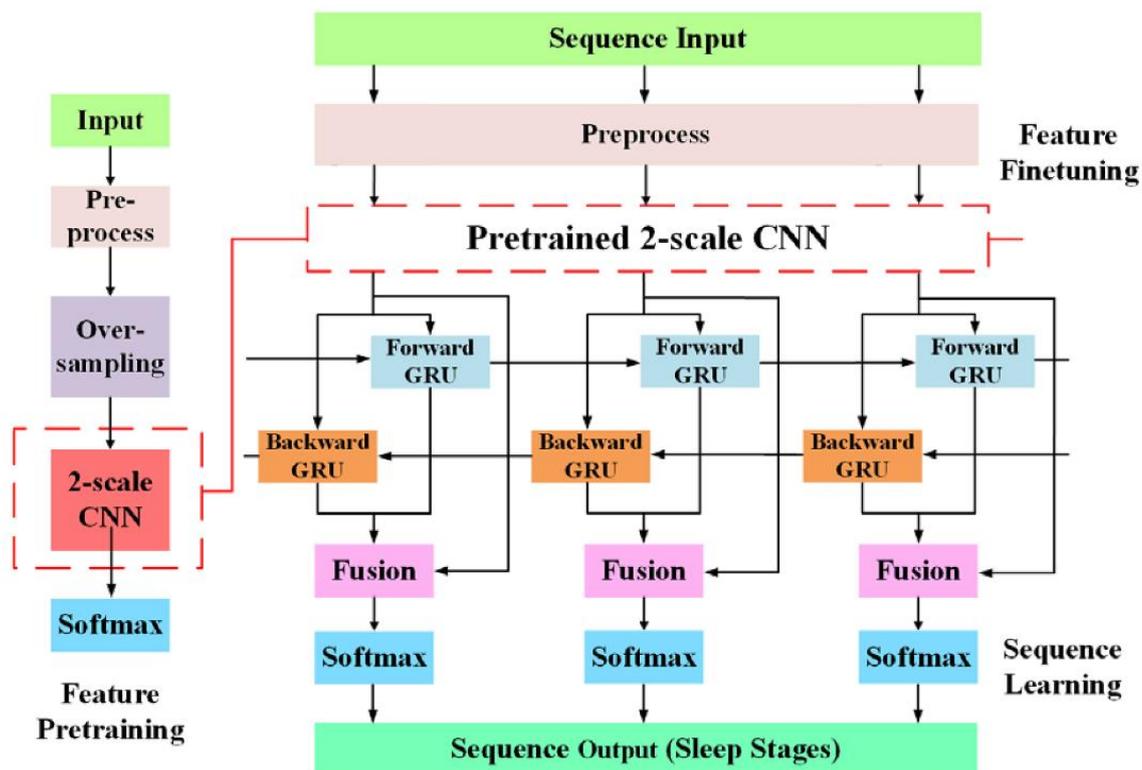


Рисунок 2.5 – Процесс обучения нейросети, учитывающий передискретизацию и обучение последовательности информации

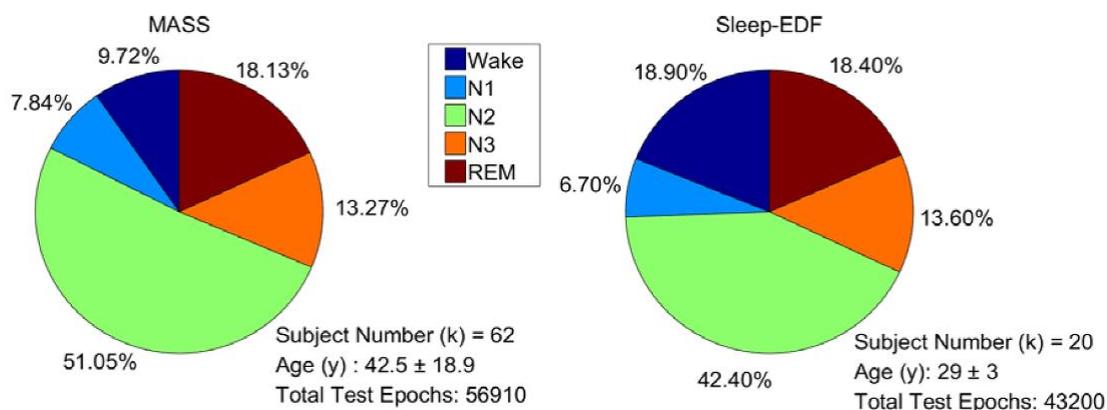


Рисунок 2.6 – Распределение данных из разных баз

Общая точность составила 81,2% и 76,3% соответственно. Результаты сопоставимы с моделями, обученными с помощью сигналов ЭЭГ. Кроме того, сравнение с шестью современными методами дополнительно демонстрирует эффективность предлагаемого подхода. В целом, это исследование открывает новые возможности для мониторинга сна [19].

В [20] авторами была построена сеть обратного распространения ошибки с учетом критериев чувствительности (79%) и специфичности (81%), что говорит о возможности использования модели для своевременной диагностики сердечно-сосудистых заболеваний. В [21] авторы применяли методы спектрального анализа при исследовании аритмий. Было выявлено, что аритмия проявляется в нерегулярной и хаотичной структуре сердечных циклов и уширении отдельных спектральных компонент, позволяя однозначно выявить патологию в работе сердца. В [22] авторами обучалась сеть долгой краткосрочной памяти (LSTM) для задачи мультиклассовой диагностики 13 паттернов электрокардиограммы, среди которых 12 сердечных ритмов и аритмий и инфаркт миокарда. Снятие сигналов сердечной деятельности производилось в 12 отведений. Были выявлены нерегулярности и хаотичности структуры сердечных ритмов и деформация спектральных компонент при использовании методов спектрального анализа. Результирующая точность классификации искомых классов составила от 94% до 99%, что говорит о полезности рассматриваемого метода в принятии клинических решений.

В [23] авторы проводили классификацию семи типов аритмий, собранных из пяти аккредитованных баз данных, используя вычисленную фрактальную размерность Хигучи для определения скрытых особенностей сигналов и нейронную сеть обратного распространения (BPN). Результирующая точность классификации аритмий и определение их местонахождения на сигнале электрокардиограммы представленным методом составила 98,83%. В [24] применялся фрактальный анализ на основе методов Херста и мультифрактального детрендированного флуктуационного анализа для диагностирования нормального ритма, частой экстрасистолии и фибрилляции предсердия. Было выявлено, что система регуляции сердечного ритма является сложной динамической системой. Данный факт следует учитывать при дальнейших исследованиях.

Из представленных выше результатов исследований других авторов следует, что теории фрактального и мультифрактального анализа включают в себя большой набор диагностических средств, позволяющих однозначно описать

сложную структуру и поведение сигналов. В авторском исследовании [25] приведены оценки фрактальной размерности Хигучи, применены метод мультифрактального флуктуационного анализа и вейвлет-преобразование для обнаружения структуры сигналов аускультации легких с пятью типами дыхательных шумов: везикулярный, бронхиальный и бронховезикулярный шум, свистящий и влажный хрип. Использование таких методов интеллектуального анализа, как нейронные сети и машинное обучение, встречается на сегодняшний день в большей части медицинских исследований по классификации патологических особенностей с впечатляющими результатами точности обученных моделей. Но, несмотря на достоинства многообразия выбора метода исследования, настоящее научное направление обладает и недостатками, которые невозможно игнорировать, так как это напрямую затруднит применение фрактальных и мультифрактальных методов, нейронных сетей и машинного обучения. Что касается первого аспекта, то некоторые методы фрактального анализа обладают значительными ограничениями в точности, нечувствительностью к сложным нестационарным временным рядам, что в последующем может привести к непредсказуемым результатам. Поэтому первой подзадачей в подобных исследованиях является установка оптимальной пары «сигнал – фрактальный метод» с наименьшим расхождением. Далее необходима предобработка временных рядов, которая включает в себя фильтрацию рассматриваемых сигналов и выделение интервалов диагностического интереса – «окон» или проведение технического анализа [25, 26, 27, 28]. Второй аспект, который касается нейронных сетей и машинного обучения, заключается в выделении либо нелинейных и/или специфических для данных признаков, либо узкого диагностического «окна», после чего подавать его на вход сети или модели [29, 30].

3 МЕДИЦИНСКИЕ БАЗЫ ДАННЫХ КАК МАТЕРИАЛ ИССЛЕДОВАНИЯ И ИХ АНАЛИЗ

3.1 Материальная база

Тема работы звучит как «Система интеллектуального анализа данных кардиомониторирования». Целью данной работы ставится разработка в среде Matlab системы для решения задач классификации методом нейросети биомедицинских данных, представленных временными рядами. На вход алгоритм принимает данные сигнала, после его обработки на выход поступает ответ: полученный сигнал показывает норму сердечно-сосудистой деятельности пациента или патологию. Примеры ЭКГ нормальной работы сердца и наличия патологии в его работе представлены на рисунках 3.1, 3.2, 3.3.

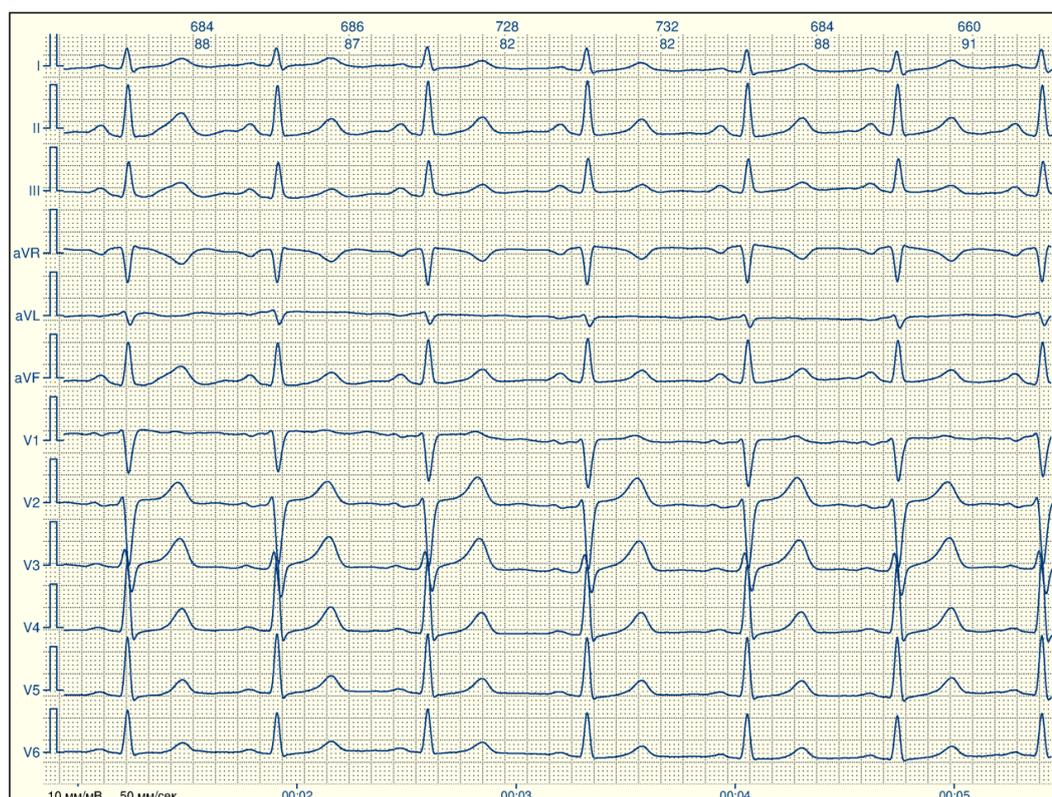


Рисунок 3.1 – ЭКГ нормальной работы сердца

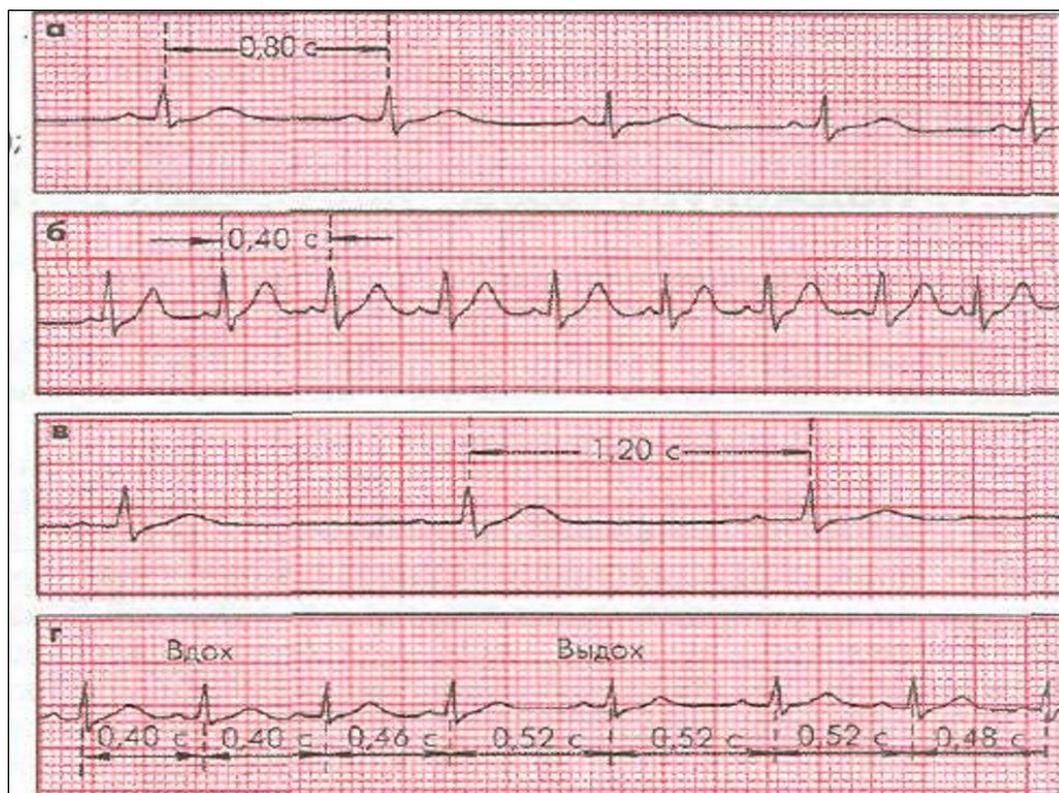


Рисунок 3.2 – Аритмия на ЭКГ

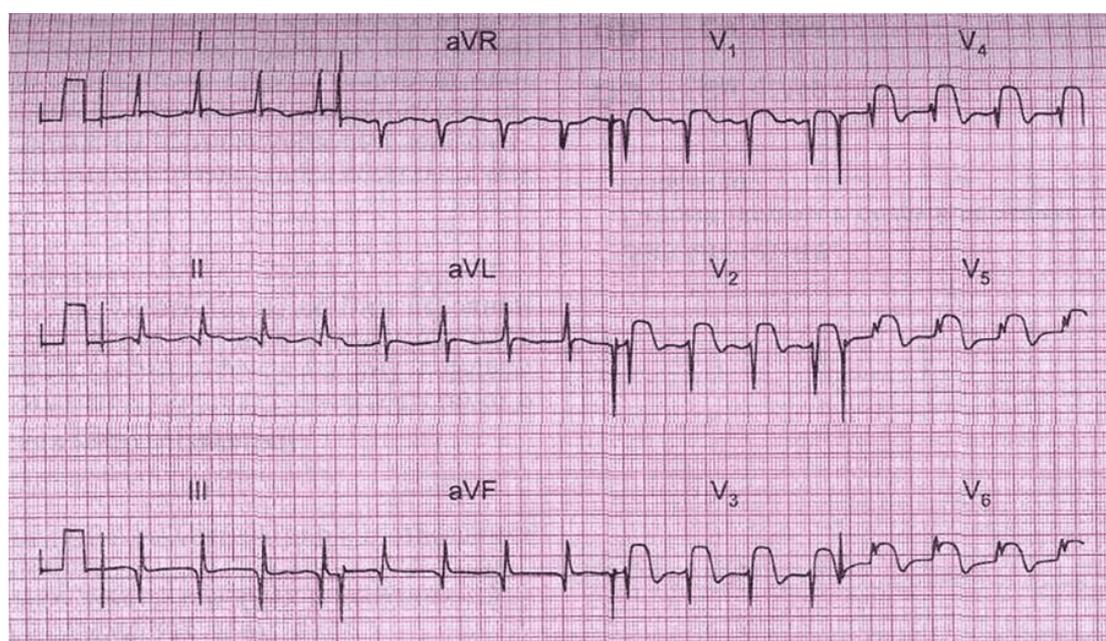


Рисунок 3.3 – Инфаркт миокарда на ЭКГ

В данной главе пойдёт речь о материале исследования, представленном временными рядами из медицинской базы данных кардиомониторирования, и его анализе: спектральном, фрактальном видах анализа сигналов ЭКГ (методы

Фурье- и вейвлет-разложения, R/S-анализ, анализ Хигучи), а также выявление статистических характеристик (энтропия, показатель Ляпунова, квартиль 1-4).

На данном этапе исследованию подлежит один участок между сердечными фазами (рисунок 3.4).

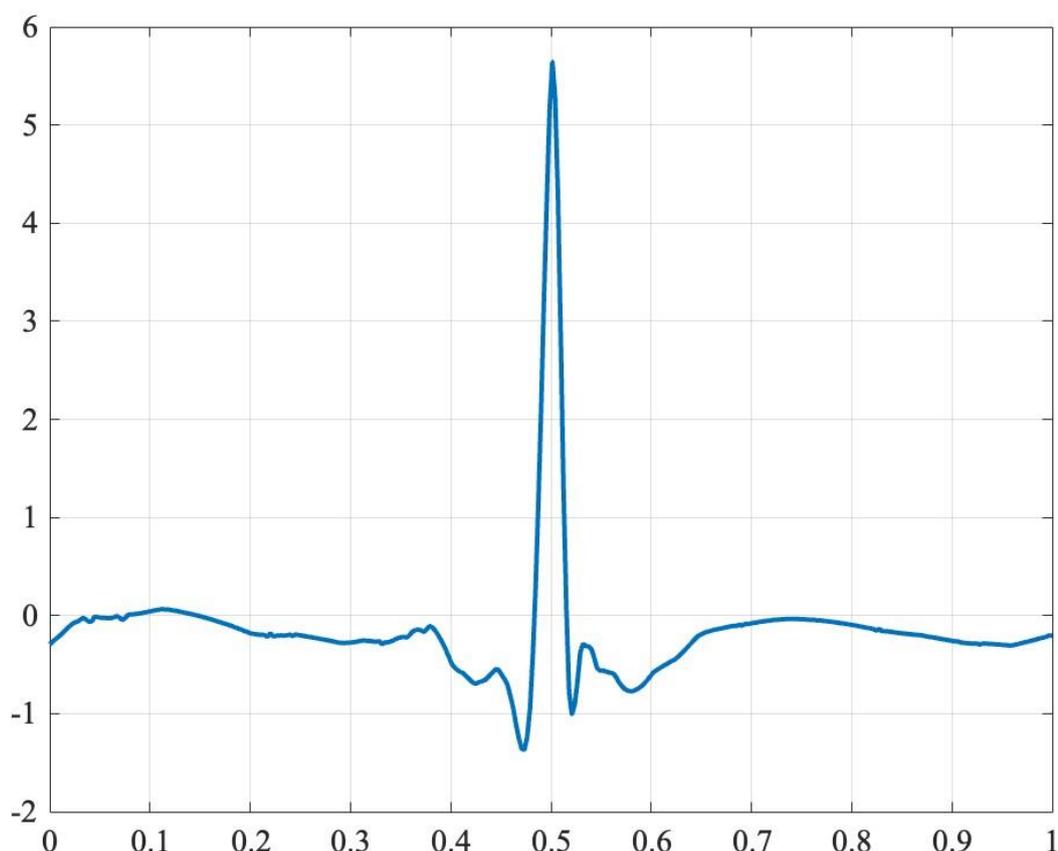


Рисунок 3.4 – Пример сигнала нормы на ЭКГ

Сложность анализа состоит в том, что число измерений на этом участке для разных данных может быть разным (на условном отрезке одной длины разное число моментов времени для отдельных экспериментов). Это особенно характерно для сигналов нестационарных временных рядов. Поэтому для основы проведения исследования был взят материал из открытой медицинской базы данных [31]. Из базы посредством программного обеспечения KT Result 3.0 [32] был извлечён объём данных, насчитывающий 50 сигналов, из которых 8 характеризуют предсердную экстрасистолию (Atrial premature beat), 8 – блокаду левой

ножки пучка Гиса (Left bundle branch block beat), 20 – нормальный ритм (Normal beat), 7 – преждевременное сокращение желудочков (Premature ventricular contraction), 7 – блокаду правой ножки пучка Гиса (Right bundle branch block beat) [8].

Эти сигналы были преобразованы в массив числовых значений [26]. Каждая строка массива (таблицы) насчитывает 360 дробных чисел для одного сигнала на едином промежутке времени (рисунок 3.5).

	350	351	352	353	354	355	356	357	358	359	360
1	-0.9769695	-0.9846135	-0.9911271	-0.9912163	-0.9882181	-0.9841399	-0.978679	-0.972664	-0.9661625	-0.958416	-0.9503605
2	-0.2743983	-0.2676153	-0.2604937	-0.2532787	-0.2452434	-0.2368657	-0.229991	-0.2241251	-0.2146101	-0.202842	-0.2044652
3	-0.1544797	-0.1644974	-0.148711	-0.1265107	-0.1426109	-0.1617307	-0.1531908	-0.1926209	-0.1536416	-0.1063434	-0.1317197
4	-1.4136703	-1.4009871	-1.3878543	-1.3774458	-1.3676776	-1.3584515	-1.3491094	-1.3369627	-1.3220182	-1.3109927	-1.3093912
5	-0.1197507	-0.1228553	-0.1261257	-0.1294519	-0.1309082	-0.1366918	-0.1357697	-0.1266187	-0.1558292	-0.1406488	-0.1273312
6	-0.0788623	-0.0587657	-0.0301218	-0.041941	-0.0637057	-0.0814019	-0.0787985	-0.0963531	-0.0630923	-0.0691817	-0.0773753
7	0.031539	0.03500507	0.05753494	0.0734842	0.09813983	0.12104566	0.15051116	0.19566349	0.16333384	0.2309445	0.25381679
8	0.41545876	0.40160655	0.39211577	0.38058841	0.35356851	0.32167078	0.29247653	0.26324942	0.24570208	0.22987138	0.20262932
9	-0.8296972	-0.8739184	-0.9136013	-0.953255	-0.991676	-1.0224104	-1.0506299	-1.0884395	-1.1257877	-1.1637499	-1.1973803
10	0.93428685	0.95950663	0.91275435	0.87398998	0.88172122	0.86880814	0.86732459	0.85487784	0.84028566	0.9242092	0.89244297
11											
12	0.08457509	0.04202764	0.07049055	0.08692429	0.07553158	0.0612396	0.0614696	0.10657925	0.1193038	0.01217601	0.11976863
13	0.37433854	0.4641717	0.48567153	0.53588595	0.52979614	0.46989304	0.50201599	0.50068967	0.602447	0.56557695	0.60096365
14	-0.1774266	-0.1765516	-0.1705594	-0.1644765	-0.1581997	-0.1554259	-0.1536702	-0.157409	-0.1692402	-0.1554764	-0.1641012
15	-0.5423134	-0.5381167	-0.5343495	-0.5305207	-0.5259872	-0.5211503	-0.5161925	-0.5111104	-0.5061191	-0.5011185	-0.4959755
16	-0.0005887	0.01466204	0.02953691	0.01721253	-0.0093185	-0.0008148	-0.0241682	-0.0375182	-0.0548964	-0.0737314	-0.0939877
17	-0.9728399	-0.9736345	-0.9355437	-0.8797807	-0.9148107	-0.9685389	-1.0387939	-0.9846496	-0.9851939	-0.9817274	-0.9916974
18	0.33895025	0.34852723	0.32721	0.36378859	0.3723116	0.38101263	0.39109931	0.40146479	0.41573595	0.41493445	0.44461894
19	-0.2859275	-0.2249849	-0.2504826	-0.2964197	-0.2924547	-0.2763661	-0.2617783	-0.2459803	-0.2487292	-0.2523991	-0.251276
20	-0.1431103	-0.1460665	-0.1404823	-0.1379471	-0.1343048	-0.1297736	-0.1246463	-0.1187913	-0.1126364	-0.1075783	-0.1027841
21	-0.2039398	-0.2013806	-0.1985263	-0.1954106	-0.1922384	-0.1898067	-0.1876569	-0.1854183	-0.1832318	-0.1811988	-0.1791478

Рисунок 3.5 – Фрагмент датасета

Такой вид представления временного ряда позволил приступить к проведению нескольких видов анализа.

3.2 Спектральный анализ: метод Фурье

Спектральный анализ – это методика в математике и инженерии, используемая для анализа характеристик сигналов в частотной области. Одним из наиболее популярных и мощных инструментов спектрального анализа является преобразование Фурье, которое позволяет разложить сложные сигналы на компоненты с различными частотами [33].

Преобразование Фурье – это математическая операция, которая берёт функцию $f(t)$ (обычно в зависимости от времени) и преобразует её в функцию

$F(\omega)$, представляющую амплитуды частот компонентов сигнала [33]. Также это преобразование называют операцией, сопоставляющей одной функции вещественной переменной другую (комплекснозначную) функцию вещественной переменной. Эта новая функция описывает коэффициенты («амплитуды») при разложении исходной функции на элементарные составляющие – гармонические колебания с разными частотами. Формула преобразования Фурье выглядит следующим образом:

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} X(t) e^{-it\omega} dt.$$

Основная идея преобразования Фурье состоит в том, чтобы выразить исходный сигнал как сумму синусоидальных волн различных частот. Это позволяет анализировать частотный состав сигнала и выявлять доминирующие частоты [33].

В пакете Matlab одномерное преобразование Фурье реализовано двумя функциями, выполняющие прямое и обратное быстрое преобразование Фурье (БПФ – функция нормированного значения амплитуды от частоты): *fft* и *ifft* соответственно [34, 35]:

$$c_n = \sum_{k=0}^{N-1} v_k e^{\frac{i2\pi nk}{N}}, \quad c_n = \frac{1}{N} \sum_{k=0}^{N-1} v_k e^{-\frac{i2\pi nk}{N}}.$$

На практике было применено прямое БПФ с целью локализации показателей частоты f по оси x и амплитуды A по оси y первых двух пиков сигнала (см. рисунок 3.6). Скрипт данной функции представлен в приложении А.

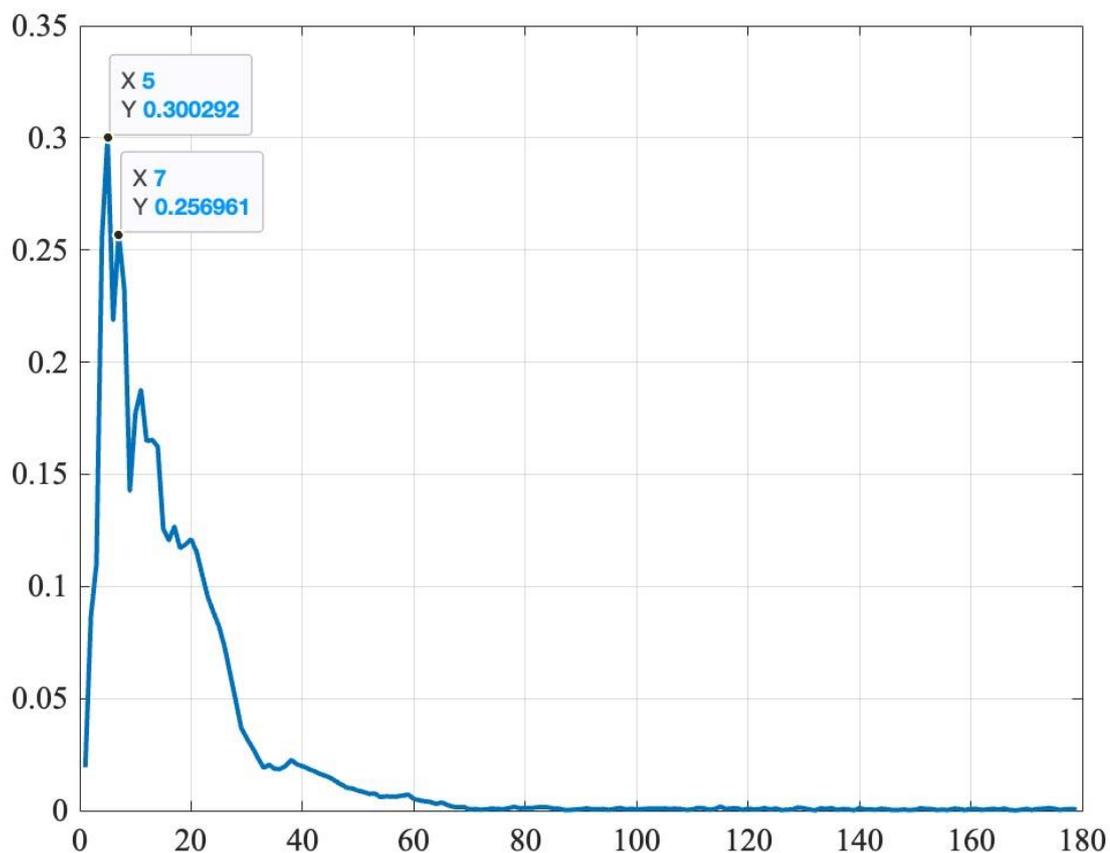


Рисунок 3.6 – Пример сигнала нормы на ЭКГ

Таким образом был получен первый набор массивов:

- 1) Fourier_A_peak1_norm;
- 2) Fourier_A_peak2_norm;
- 3) Fourier_A_peak1_patology;
- 4) Fourier_A_peak2_patology;
- 5) Fourier_f_peak1_norm;
- 6) Fourier_f_peak2_norm;
- 7) Fourier_f_peak1_patology;
- 8) Fourier_f_peak2_patology.

Каждый из них содержит 10 значений, дробных для амплитуды A (рисунок 3.7) и целых для частоты f (рисунок 3.8).

```
Fourier_A_peak1_norm = [2.0170 0.2930 0.2260 1.6120 0.3000 0.5610 0.2550 0.5430 1.4280 2.4340];  
Fourier_A_peak2_norm = [0.3610 0.2570 0.3760 0.3090 0.2570 0.4230 0.2590 0.2420 0.3870 0.2690];
```

Рисунок 3.7 – Пример массива для амплитуды А сигналов нормы

```
Fourier_f_peak1_patology = [1 1 2 1 4 1 2 3 4 3];  
Fourier_f_peak2_patology = [4 4 4 4 6 5 17 7 6 6];
```

Рисунок 3.8 – Пример массива для частоты f сигналов патологии

Это первые 8 из 16 характеристик временных рядов.

3.3 R/S-анализ

RS-анализ (Range-Scale Analysis) – это статистический метод фрактального анализа временных рядов, используемый для оценки долгосрочной зависимости временных рядов и анализа устойчивости систем, широко применяемый в экологии, экономике и других областях науки. Данный метод был введен в научную практику Х.Э. Хёрстом в контексте исследования особенностей колебаний водного стока реки Нила [36]. Позднее, Мандельброт [37] дополнил и расширил теорию на различные природные и экономические системы, улучшая её математическую базу и внедряя в различные дисциплины.

На практике RS-анализ – это мощный инструмент для исследования временных рядов, который может применяться в самых различных областях науки и техники. Его корректное применение позволяет глубже понять природу изучаемого процесса и выявить долгосрочные зависимости, что является важным аспектом для прогнозирования и планирования [36, 37].

Его преимуществами ставятся:

- 1) простота, эффективность и высокая информативность;
- 2) гибкость: возможность применения метода в различных дисциплинах делает его универсальным инструментом анализа долгосрочной зависимости временных рядов;
- 3) теоретическая база: существуют обширные теоретические и практические исследования, подтверждающие эффективность RS-анализа.

Анализ проводится следующим образом:

1) на вход принимается временной ряд $y(t)$ (длина $M+1$), который преобразуется в ряд z длиной M , используя логарифмическое соотношение:

$$z_i = \log\left(\frac{y_{i+1}}{y_i}\right), \quad i = 1, 2, 3, \dots, M$$

2) полученный ряд делится на A смежных периодов длины n таким образом, чтобы $A * n = M$. Каждый период обозначается I_a , где $a = 1, 2, \dots, A$. Каждый элемент в подпериоде обозначается z_k , где $k = 1, 2, \dots, n$. Для каждого подпериода вычисляется среднее значение:

$$E(I_a) = \frac{1}{n} \sum_{k=1}^n z_{k,a}$$

3) суммируются накопленные отклонения от среднего значения для всех подпериодов:

$$X_{k,a} = \sum_{i=1}^k (z_{i,a} - E(I_a)), \quad k = 1, 2, \dots, n$$

4) производится расчёт размаха для каждого подпериода:

$$R_{I_a} = \max(X_{k,a}) - \min(X_{k,a})$$

5) производится расчёт стандартного отклонения для каждого подпериода:

$$S_{I_a} = \sqrt{\frac{1}{n} \sum_{k=1}^n (z_{k,a} - E(I_a))^2}$$

6) для каждого диапазона выполняется нормализация, которая заключается в делении R/S :

$$\left(\frac{R}{S}\right)_n = \frac{1}{A} \sum_{a=1}^A \frac{R_{I_a}}{S_{I_a}}$$

7) повторение шагов 2-6 с увеличением n до тех пор, пока $n \leq M/2$;

8) строится график функции $\log(R/S(n))$ от $\log(n)$. Рассчитывается линейная регрессия:

$$\frac{R}{S} \propto n^H \Rightarrow \log\left(\frac{R}{S(n)}\right) = H \log(n) + c \Rightarrow D = 2 - Hu$$

Примером проведения R/S-анализа на данных ЭКГ представлен на рисунке 3.9.

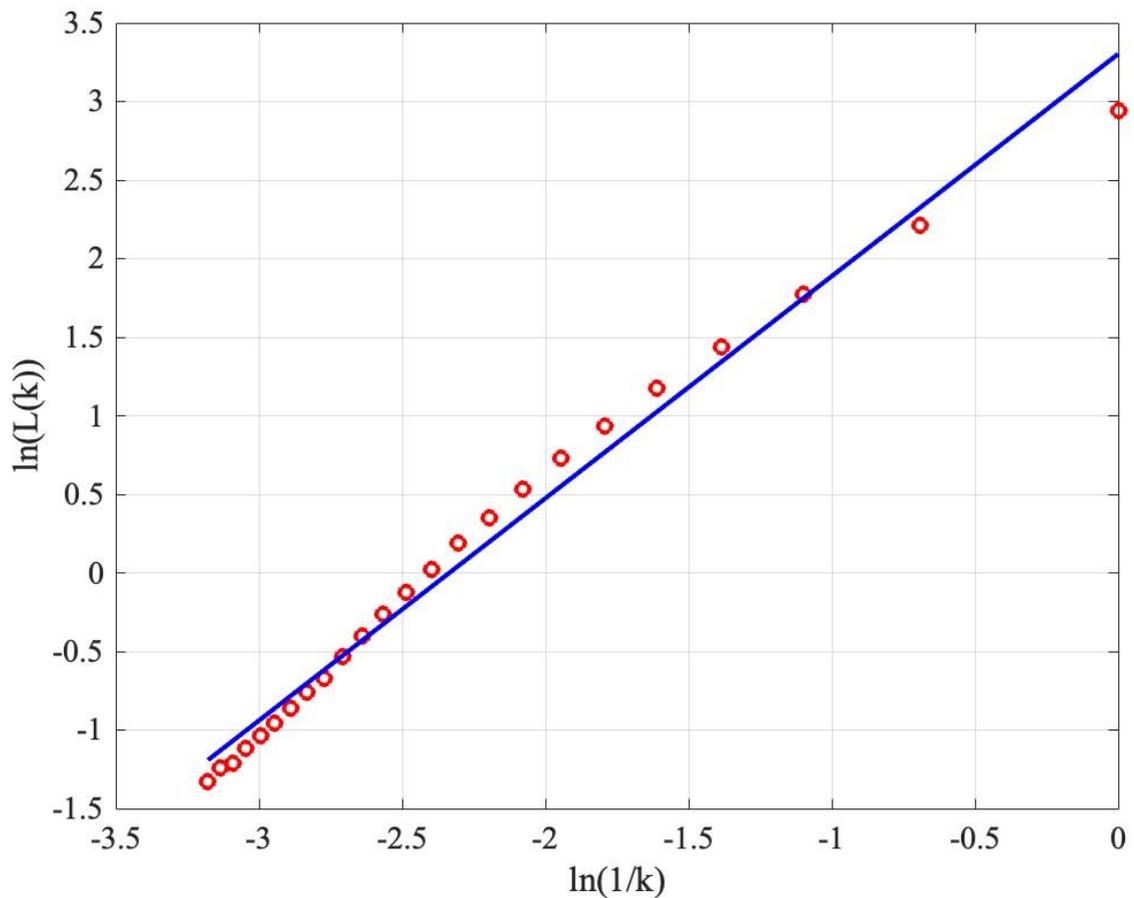


Рисунок 3.9 – Графическое представление R/S-анализа временного ряда нормы ЭКГ

С целью получения значений параметра Хёрста H и фрактальной размерности ($D = 2 - H$) данный метод был применён в среде Matlab (скрипт представлен в Приложении Б).

Таким образом было получено ещё 4 набора массивов, также содержащие по 10 дробных значений:

- 1) RS_h_norm;

- 2) RS_D_norm;
- 3) RS_h_patology;
- 4) RS_D_patology.

На данном этапе получено 12 массивов с характеристиками временных рядов.

3.4 Показатель Хигучи

Это метод, который используется для вычисления фрактальной размерности временных рядов. Он был предложен японским ученым Такео Хигучи в 1988 году и широко применяется в различных областях, таких как изучение финансовых рынков, биомедицинских сигналов, климатологии и других дисциплин, где анализ временных рядов играет важную роль. Особенно актуален для исследования сложных нерегулярных нестационарных сигналов [38]. Его алгоритм основан на оценке длины кривой через выделение сегментов k :

- 1) входной временной ряд $Y(t)$ разбивается на k новых рядов по правилу:

$$Y_m^k = \left\{ Y(m), Y(m+k), Y(m+2k), \dots, Y\left(m + \left\lfloor \frac{N-m}{k} \right\rfloor k\right) \right\}, \quad m = 1, 2, \dots, k$$

где m – начальное время, k – временная задержка в диапазоне от 1 до k_{max} , « $\lfloor \]$ » – целая часть от выражения в скобках.

- 2) вычисление длины кривых результирующих временных рядов:

$$L_m(k) = \frac{1}{k} \left(\sum_{i=1}^{\left\lfloor \frac{N-m}{k} \right\rfloor} |Y(m+ik) - Y(m+(i-1)k)| \right) \frac{N-1}{\left\lfloor \frac{N-m}{k} \right\rfloor k},$$

где $\frac{N-1}{\left\lfloor \frac{N-m}{k} \right\rfloor k}$ – нормировочный коэффициент;

Расчёт полной средней длины кривой:

$$\langle L(k) \rangle = \frac{1}{k} \sum_{m=1}^k L_m(k)$$

3) Расчёт фрактальной размерности:

$$k = [A_1 \sin(B_1 \cdot N + C_1) + A_2 \sin(B_2 \cdot N + C_2)]$$

Наглядный пример расчёта показателя Хигучи показан на рисунках 3.10-3.11, составленном на основе одного сигнала нормы ЭКГ.

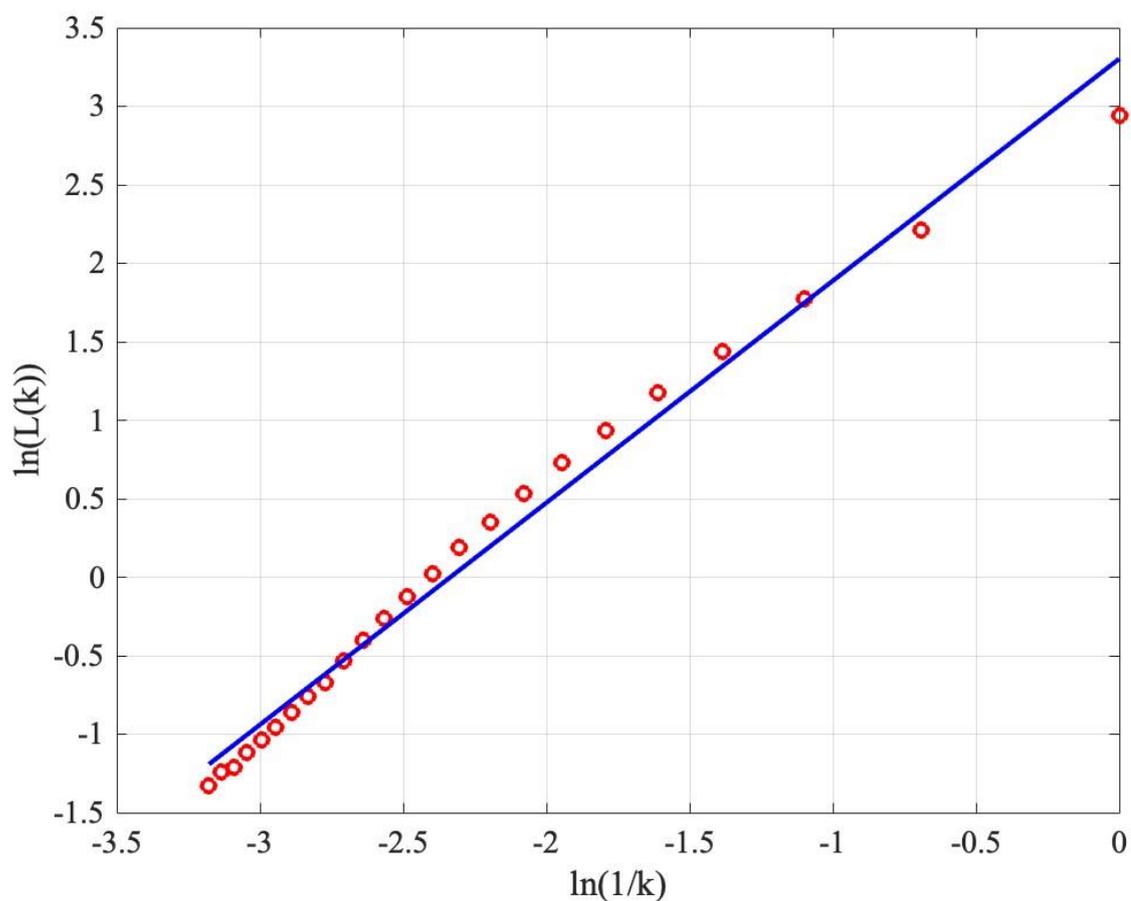


Рисунок 3.10 – Графическое представление показателя Хигучи на данных временного ряда (норма ЭКГ)

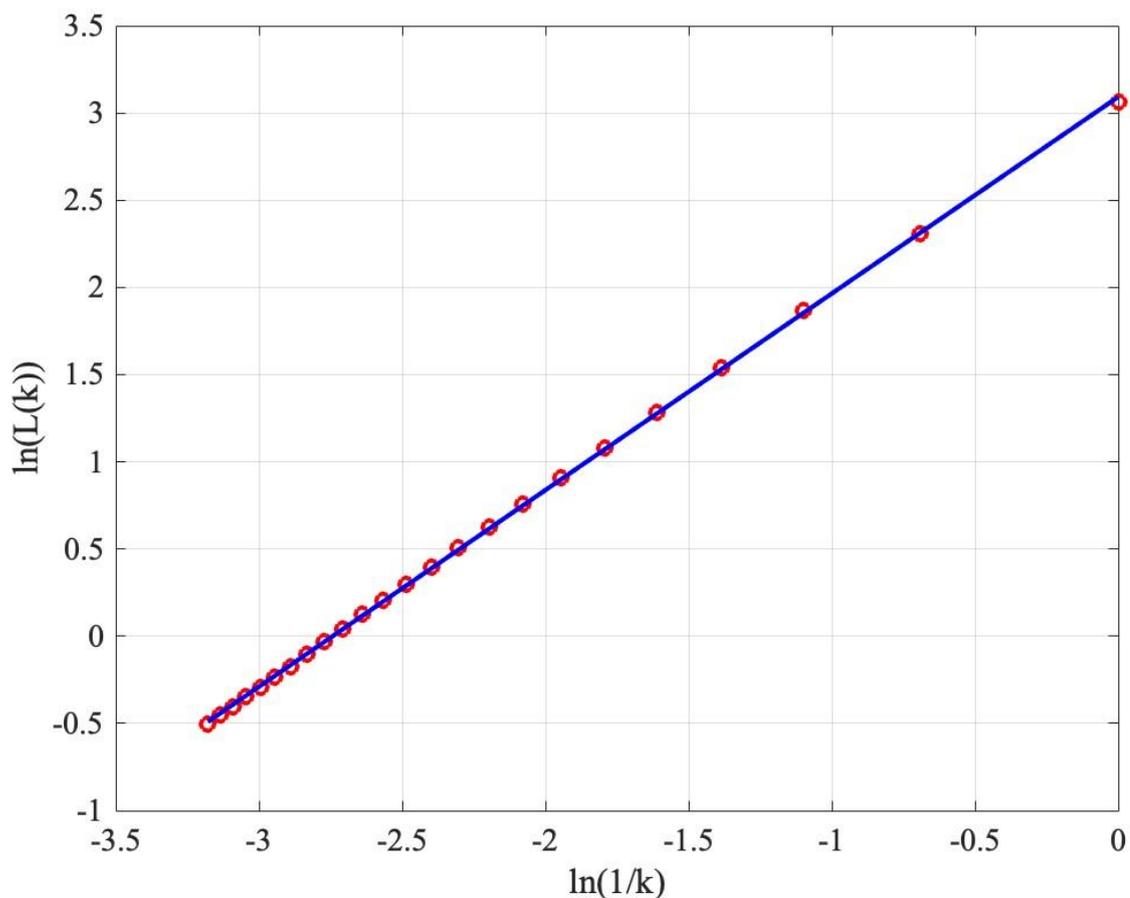


Рисунок 3.11 – Графическое представление показателя Хигучи на данных временного ряда (патология ЭКГ)

Реализованный в среде Matlab скрипт расчёта показателя Хигучи (приложение В) составил следующие массивы данных:

- 1) Khiguchi_norm;
- 2) Khiguchi_patology.

Итого, на данном этапе получено 14 массивов с характеристиками временных рядов.

3.5 Энтропия

Энтропия является фундаментальным понятием в теории информации и находит широкое применение в различных научных и инженерных областях. Она позволяет количественно измерить неопределённость и помогает более эффективно передавать, сжимать и кодировать данные. Понятие энтропии было

введено Клодом Шенноном в рамках его работы по теории информации, опубликованной в его знаменитой статье "A Mathematical Theory of Communication" в 1948 году [39]. Формула Шеннона для вычисления энтропии:

$$H(X) = -\sum_{i=1}^n p(x_i) \log_b p(x_i)$$

Данная формула была применена в среде Matlab, скрипт представлен в приложении Г.

Процесс расчёта состоит в следующем:

- 1) в переменную *data* принимается вектор данных;
- 2) рассчитывается общее количество событий, затем – вероятности каждого события как отношение количества каждого события к общему количеству событий; вероятности с нулевым значением отсеиваются, чтобы избежать логарифмирования нуля;

- 3) энтропия вычисляется с использованием формулы Шеннона;

- 4) значение энтропии выводится на экран и записывается в новый массив.

В результате было получено ещё 2 массива с расчётами энтропии для 10 сигналов нормы и 10 сигналов патологии:

- 1) Entropy_norm;

- 2) Entropy_patology;

Итого: 16 массивов.

3.6 Показатель Ляпунова

Показатель Ляпунова характеризует степень чувствительности динамической системы к начальным условиям. Это один из ключевых показателей для анализа устойчивости и хаотичности систем. Он позволяет оценить скорость расхождения (или сближения) траекторий в фазовом пространстве на основе временного ряда данных. В динамических системах даже небольшое различие в начальных условиях может приводить к существенно различным траекториям во времени, и показатель Ляпунова количественно описывает эту разницу [40].

Основные шаги при вычислении показателя на временных рядах:

1) подготовка временного ряда: имея временной ряд данных $x(t)$, где t – момент времени, нужно определить шаг времени Δt между измерениями;

2) построение траектории в фазовом пространстве: каждое значение временного ряда $x(t)$ становится координатой вектора состояния в фазовом пространстве. Это позволяет перейти от одномерного временного ряда к многомерной динамической системе;

3) вычисление расстояния между траекториями: для каждого момента времени t , рассчитывается расстояние между исходной траекторией и её ближайшими соседями в фазовом пространстве;

4) усреднение логарифмического приращения расстояния: рассчитывается логарифмическое изменение расстояния между соседними траекториями за каждый временной шаг. Затем это изменение суммируется и делится на количество шагов;

5) получение показателя Ляпунова: итоговый результат вычисляется как предел среднего логарифмического приращения расстояния при стремлении шага времени к нулю [40].

Скрипт, реализующий расчёт показателя Ляпунова, приведён в приложении Д. Программа на выход даёт вывод полученных значений на экран и/или их запись в новый массив. В итоге были получены 2 новых массива для обучения нейросети:

1) Lyapunov_norm;

2) Lyapunov_patology.

Итого, 18 массивов.

3.7 Квантиль Q1-Q4

За последнюю характеристику, которая может различать временные ряды друг от друга, был взят статистический показатель 1, 2, 3 и 4 квантилей. Это статистические меры, разделяющие упорядоченный набор данных на четыре равные части. Они широко используются для анализа распределения данных, в том числе временных рядов [41].

Первый квартиль (Q1): это значение, ниже которого находится 25% данных. В других терминах это 25 перцентиль. Позиция первого квартиля (Q1): $P\{Q1\} = 0.25*(n + 1)$;

Второй квартиль (Q2): это медиана данных, разделяющая набор на две равные части. 50% данных находятся ниже и 50% выше медианы. Позиция второго квартиля (Q2, медиана): $P\{Q2\} = 0.50*(n + 1)$;

Третий квартиль (Q3): это значение, ниже которого находится 75% данных. Это 75 перцентиль. Позиция третьего квартиля (Q3): $P\{Q3\} = 0.75*(n + 1)$

Четвертый квартиль (Q4): это не отдельное значение, а заключительный диапазон данных, то есть верхние 25% значений [41].

Процесс вычисления состоит в следующем:

- 1) сортировка данных: перед вычислением необходимо упорядочить данные во временном ряду по возрастанию;
- 2) определение количества элементов (n): общее число элементов в отсортированном временном ряду;
- 3) применение формул для нахождения позиций квартилей.

В среде разработки Matlab уже есть встроенная функция *quantile* для проведения расчётов данного статистического метода. Скрипт программы приведён в приложении Е. Новые значения были сгруппированы в 8 новых массивов:

- 1) Q1_norm;
- 2) Q2_norm;
- 3) Q3_norm;
- 4) Q4_norm;
- 5) Q1_patology;
- 6) Q2_patology;
- 7) Q3_patology;
- 8) Q4_patology.

Итого, 26 массивов.

3.8 Выводы по главе

В данной главе были описаны разные методы анализа временных рядов и их реализация с использованием возможностей ППП Matlab. 50 сигналов ЭКГ, представленных временными рядами, послужили материалом для проведения расчётов с целью получения новых значений (всего 26 массивов) посредством алгоритмов, способных отличить временные ряды друг от друга. Это важный критерий, способствующий успешному применению датасета в следующем этапе исследования – разработки простой нейросети классификации сигналов ЭКГ на две группы: «норма» и «патология».

4 КОНФИГУРИРОВАНИЕ ПРОСТОЙ НЕЙРОННОЙ СЕТИ КЛАССИФИКАЦИИ ВРЕМЕННЫХ РЯДОВ

4.1 Сведения о нейросети: структура, характеристики, области применения

Задачей данного этапа исследования ставится реализация простого примера конфигурирования нейросети, постановка перед ней задачи классификации временных рядов.

Для получения первого опыта разработки нейросети был выбран метод классификации, входящий в библиотеку ППП Matlab, – Learning Vector Quantization (LVQ, от английского «обучающиеся векторные квантования»). Это метод машинного обучения, основанный на конкурентном обучении. Он работает путём адаптации прототипов в пространстве признаков таким образом, чтобы они максимально точно представляли классы данных [42, 43]. Его достоинствами считаются:

1) простота реализации: алгоритм LVQ довольно прост в реализации и понятен с точки зрения теоретического фундамента;

2) интуитивная интерпретируемость: модель легко интерпретируема, так как основана на прототипах, которые можно визуализировать и анализировать;

3) эффективное использование ресурсов: LVQ требует меньше вычислительных ресурсов по сравнению с более сложными моделями, такими как глубокие нейронные сети;

4) известный радиус действия: прототипы имеют чётко определенный радиус действия, что упрощает понимание и контроль над процессом классификации.

5) подходит для многоклассовой классификации: LVQ хорошо работает для задач, где требуется классификация в несколько классов, из-за своей гибкости в настройке прототипов [42, 43].

К недостаткам относят:

1) чувствительность к начальному положению прототипов: начальное расположение прототипов может существенно повлиять на результаты, что требует тщательной инициализации;

2) конвергенция к локальным минимумам: алгоритм подвержен застреванию в локальных минимумах, что может привести к неоптимальной классификации;

3) чувствительность к параметрам: требуется настройка гиперпараметров, таких как скорость обучения, которые оказывают значительное влияние на качество модели;

4) требования к количеству прототипов: неправильный выбор количества прототипов на класс может негативно сказаться на производительности модели;

5) зависимость от масштабирования данных: алгоритм чувствителен к масштабированию признаков, поэтому требуется тщательная предобработка данных;

б) медленный при больших наборах данных: обучение может быть относительно медленным на больших наборах данных по сравнению с другими методами, такими как решающие деревья или методы на основе ансамблей [42, 43].

Схематичное изображение структуры нейросети представлено на рисунке 4.1 [42, 43].

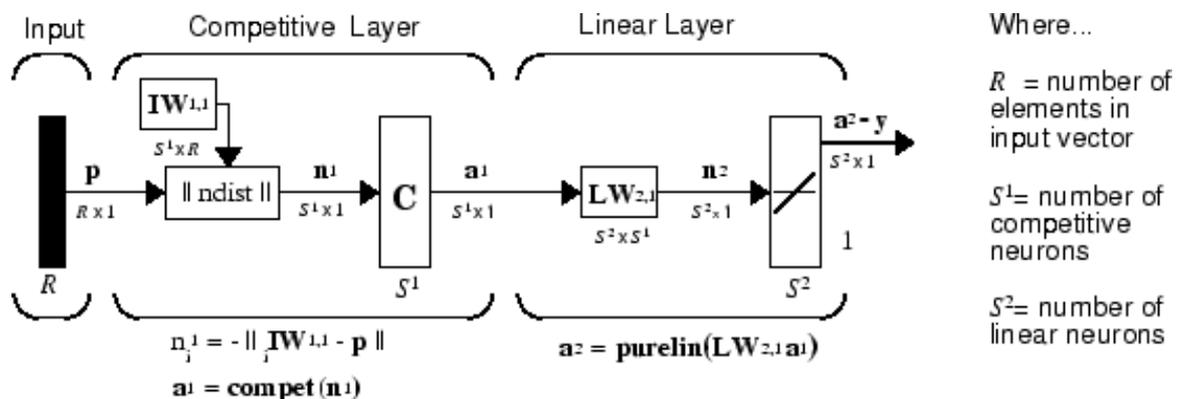


Рисунок 4.1 – Архитектура нейросети LVQ

Сеть LVQ состоит из двух слоёв, каждый из которых выполняет свою функцию. Входной слой Input принимает входящий вектор признаков размерности R . Никаких вычислений в этом слое не производится [42, 43].

Соревновательный слой Competitive Layer сравнивает входной вектор с эталонными (весовыми) векторами каждого нейрона этого слоя. Здесь вычисляется евклидово расстояние между входом и весами нейронов, определяется «победитель» – нейрон, чей вес ближе всего к входному вектору (конкурентная функция). Данный нейрон активируется, даёт на выход значение 1, остальные – 0 [42, 43].

Линейный слой Linear Layer получает «однократный» вход от нейрона-победителя, умножает этот сигнал на веса линейного слоя (матрица LW), преобразует результат с помощью линейной функции активации. На выходе формируется итоговая классификация (результат сети) [42, 43].

4.2 Создание простейшей нейронной сети классификации временных рядов

Свойственная данной нейросети простота позволила получить первый опыт конфигурирования модели для подготовленного заранее объёма данных. Учитывая упомянутый ранее фактор медленной производительности при больших наборах данных, для проведения обучения нейросети из исследованных ранее массивов сигналов ЭКГ случайным образом было выбрано 20, 10 из которых являются нормой ЭКГ, остальные 10 – патологией. Аналогично процессу, описанному в пункте 3.1, из выбранных сигналов в формате таблицы Excel был сформирован датасет, как на рисунке 3.5. Позже он будет импортирован в проект Matlab.

Скрипт конфигурирования нейросети весьма прост, он представлен в приложениях 3, И. Данный код содержит следующий функционал:

- 1) импорт датасета в массив *NewArray* и транспонирование (необходимо для корректного считывания сигналов нейросетью – в столбик);
- 2) подготовка вектора-строки, указывающей принадлежность каждого сигнала в датасете к группам нормы и патологии;

3) разделение данных на обучающую, валидационную и тестовую выборки в заданном процентном соотношении;

4) функция `newlvq(minmax(array), 4, [0.5 0.5], 0.01)` создаёт нейронную сеть LVQ с определёнными параметрами:

4.1) `minmax(array)` задаёт диапазон входных данных (минимальные и максимальные значения для каждого измерения; возвращаемое значение используется для нормализации данных входного слоя сети);

4.2) `4` указывает число нейронов в скрытом слое (их также называют *центроиды* или *прототипы*).

4.3) `[0.5 0.5]` задаёт начальное распределение прототипов среди классов, в данном случае 50% прототипов будут относиться к 1 классу (норма), 50% прототипов будут относиться ко 2 классу (патология);

4.4) `0.01` указывает начальный коэффициент обучения (стандартное значение);

4.5) параметр `net.trainParam.epochs = 50`; указывает количество итераций (эпох) обучения нейросети, здесь – 50; является опциональным, – при неиспользовании автоматически задаётся 1000 итераций обучения;

5) запуск обучения нейросети, подсчёт и вывод на экран метрик;

б) подготовка нового аналогичного датасета и ручное тестирование обученной нейросети. Выход из программы.

Таким образом, полный скрипт создаёт LVQ нейронную сеть с 4 прототипами, которые распределены по классам 50:50, и коэффициентом обучения 0.01. Схематично нейросеть выглядит так, как показано на рисунке 4.2.

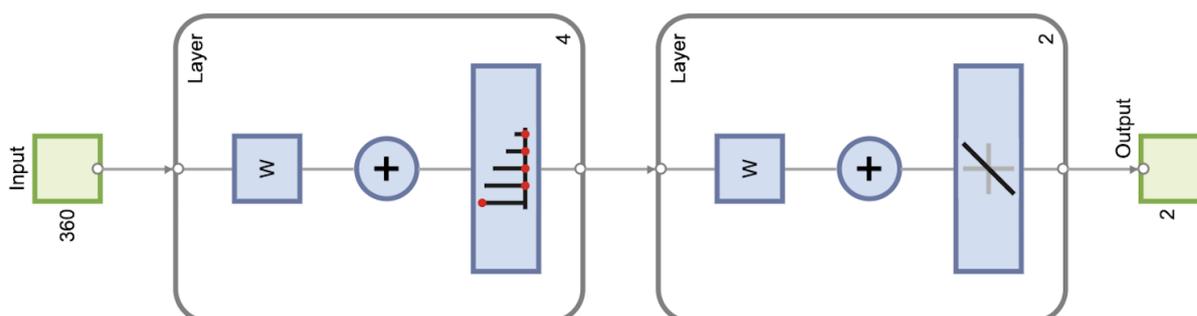
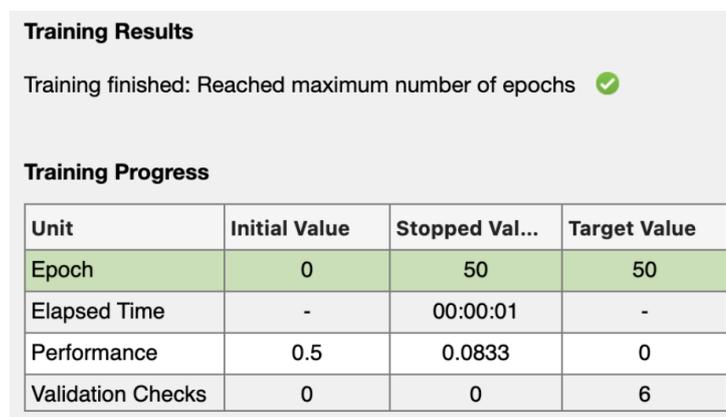


Рисунок 4.2 – Архитектура разработанной LVQ-нейросети

4.3 Результаты обучения и тестирование нейросети

С выполнением описанного выше процесса командой *Run* интерфейса Matlab был получен результат обучения нейросети, представленный на рисунках 4.3 и 4.4.



Training Results

Training finished: Reached maximum number of epochs ✔

Training Progress

Unit	Initial Value	Stopped Val...	Target Value
Epoch	0	50	50
Elapsed Time	-	00:00:01	-
Performance	0.5	0.0833	0
Validation Checks	0	0	6

Рисунок 4.3 – Результат обучения нейросети

Y =

1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	1	1
0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1	1	0	0

Рисунок 4.4 – Результат обучения нейросети

Здесь массив Y демонстрирует распределение 20 принятых сигналов на 2 группы с помощью двух комбинаций нулей и единиц по вертикали:

- 1) 1-0 – норма;
- 2) 0-1 – патология.

Однако результат обучения показал, что обучение прошло с ошибками: 10, 14, 16, 19 и 20 сигналы были неверно интерпретированы сконфигурированной нейросетью. Об этом также свидетельствуют сводки Test Validation Performance (от англ. «производительность обучения»), метрики Confusion Matrix (от англ. «матрица ошибок»), график Neural Network Training Receiver Operating Characteristic (от англ. «характеристики работы приёмника обучения нейронной сети»), представленные в приложении К. Приведём ниже краткие результаты обучения, выведенные на экран в окне команд Matlab (рисунок 4.5):

```

Training Accuracy (%):
91.6667

Training Confusion Matrix:
 6  0
 1  5

Validation Accuracy (%):
50

Validation Confusion Matrix:
 1  0
 2  1

Testing Accuracy (%):
50

Testing Confusion Matrix:
 2  1
 1  0

```

Рисунок 4.5 – Краткие результаты обучения

Данный вывод и приведённые метрики показывают, что в 25% случаев сконфигурированная нейросеть неправильно определяет категорию входящего сигнала.

Тем не менее, полученный результат требует проверки на новых данных – датасете, который нейросеть ещё не принимала для анализа. Аналогично датасету, предназначенному для обучения (рисунок 3.5), был составлен набор из 28 сигналов, первые 10 из которых – сигналы нормы ЭКГ, остальные 18 – патологии (рисунок 6).

ans =

```

 1  1  1  1  0  1  1  1  1  1  0  0  0  1  0  1  1  0  0  1  0  0  0  0  1  0  0  0
 0  0  0  0  1  0  0  0  0  0  1  1  1  0  1  0  0  1  1  0  1  1  1  1  0  1  1  1

```

Рисунок 4.6 – Результат теста нейросети

Результат демонстрирует, что 6 сигналов из 28 (примерно 21%) были неверно классифицированы, что сходится с результатом обучения.

Данный процент ошибки для такой области, как медицина, весьма велик. Предположительно это может быть связано со следующими причинами:

1) малый объём датасета: нейросеть обучена на гораздо меньшем числе паттернов, свойственных каждой группе сигналов; или же слишком большой объём вводимых в нейросеть данных (слишком много показателей в каждом из сигналов);

2) заданы не подходящие для поставленной задачи параметры конфигурирования нейросети;

3) наличие синтаксических ошибок в коде (например, повторное использование одной переменной для разных целей, приводящее к неправильным расчётам);

4) ошибки при составлении массивов статистических характеристик ввиду наличия человеческого фактора;

5) модель нейросети не подходит для решения задачи, поставленной в данной работе.

Каждый критерий имеет своё место и требует тщательной проработки.

4.4 Выводы по главе

В данной главе были описана реализация простого примера конфигурирования нейросети, поставлена задача классификации временных рядов, представленных сигналами сердечно-сосудистой деятельности человека. Нейросеть успешно завершила обучение и представила наглядный результат, демонстрирующий наличие ошибки, оценённой примерно в 25% случаев ошибочной идентификации входящего сигнала. Тем не менее данный результат не упраздняет допустимость применения выбранной в данной работе нейросети для поставленной задачи, но в то же время требует проверки и решения перечисленных ранее задач для более углубленной доработки или же опровержения приемлемости использования данной модели для классификации сигналов ЭКГ человека.

ЗАКЛЮЧЕНИЕ

Современные методики мониторинга кардиальной функции организма активно применяются в медицине, чтобы отслеживать работу сердечно-сосудистой системы, а также своевременно выявлять и предотвращать возможные заболевания и нарушения. Их назначением также является сокращение времени анализа, повышение его качества и повышения достоверности результатов в условиях постоянно растущего объёма данных мониторинга, что снижает нагрузку на специалистов, нивелируя человеческий фактор. Поэтому сегодня системы интеллектуальной классификации динамических сигналов в приложениях к исследованию кардиальной деятельности человека являются актуальным продуктом для разработки и внедрения для практического использования.

Исследование в рамках данной работы показало, что на сегодня существуют инструменты для разработки таких решений, причём не обязательно требующих глубоких знаний в области разработки и программирования. Для достижения поставленной цели был проведён ряд работ:

- 1) проведён литературный обзор по теме интеллектуального анализа биомедицинских данных;
- 2) освоена инструментальная база (ППП Matlab) и получен богатый опыт работы с инструментарием пакета;
- 3) в открытом доступе сети Интернет найдены медицинские базы данных ЭКГ (временные ряды), проведены спектральный, фрактальный виды анализа, выявлены статистические;
- 4) реализован простой пример конфигурирования нейросети, классифицирующей динамические временные ряды;
- 5) проведены её обучение и тестирование, получены наглядные результаты, сделаны выводы.

Несмотря на то, что разработанное программное решение работает чётко по заданному алгоритму, всегда есть место ошибке. Как известно, нейросеть –

это модель, «чёрный ящик», структура которой доподлинно известна её разработчику и профильным специалистам. Достигнутый результат вполне ожидаем с положительной точки зрения, так как в общем цель данной работы была достигнута: разработан вариант рабочей системы интеллектуальной классификации динамических сигналов в приложениях к исследованию кардиальной деятельности человека.

В завершение исследования приведём ряд задач, позволяющих **в перспективе** расширить данный труд:

- 1) провести обучение нейросети LVQ на более объёмном датасете;
- 2) подобрать более подходящие для поставленной задачи параметры конфигурирования нейросети;
- 3) провести исправление синтаксических ошибок в коде программного решения, в том числе при анализе и обработке датасета (при их наличии);
- 4) установить допустимость применения модели LVQ на практике для классификации временных рядов, представленных ЭКГ человека.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Кубланов, В.С. Анализ биомедицинских сигналов в среде MATLAB : учебное пособие / В. С. Кубланов, В. И. Борисов, А. Ю. Долганов. – Екатеринбург : Изд-во Урал. ун-та, 2016. – 120 с. – ISBN 978-5-7996-1813-1.

2 Сергеев, Н.Е. Системы искусственного интеллекта: учебное пособие : в 2 Ч. / Н.Е. Сергеев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования "Южный федеральный университет", Инженерно-технологическая академия. Таганрог: Издательство Южного федерального университета, 2018. – Ч.1. – 118 с. – ISBN 978-5-9275-2113-5.

3 Чучалин, А.Г. Пульмонология / под ред. Чучалина А.Г. – Москва : ГЭОТАР-Медиа, 2020. – 768 с. – ISBN: 978-5-9704-1076-9.

4 Бодрин, К.А. Использование технологий машинного обучения в медицине [Электронный ресурс] / К.А. Бодрин, А.А. Красноперова // Теория и практика современной науки. – 2018. – №10 (40). – Режим доступа : <https://cyberleninka.ru/article/n/ispolzovanie-tehnologiy-mashinnogo-obucheniya-v-meditsine>. – 17.12.2023.

5 Колесников, С.В. Применение спектральных методов анализа для выявления характеристик сигналов электрокардиограммы / С.В. Колесников, В.В. Евдокимова // «День науки» : материалы XXXIII научной конференции Амурского государственного университета (18 апреля 2024 г., Благовещенск). – Благовещенск : АмГУ, 2024. – С. 75-75.

6 Колесников, С.В. Применение машинного обучения, скейленговых и спектральных методов для интеллектуального анализа сердечных ритмов / С.В. Колесников, В.В. Евдокимова // Молодежь XXI века: шаг в будущее: матер. XXV регион. науч.-практ. конф. (Благовещенск, 22 мая 2024 г.) : в 2 томах. – Благовещенск : ФГБОУ ВО Амурская ГМА Минздрава России, 2024. – С. 469-470.

7 Евдокимова, В.В. Концепция гибридного интеллектуального анализа сердечных ритмов: применение спектральных методов и машинного обучения / В.В. Евдокимова, С.В. Колесников, Л.С. Афанасов, А.Г. Масловская // Информатика и системы управления, 2024, № 3(81). – С. 21-135

https://doi.org/10.22250/18142400_2024_81_3_121 (ВАК)

8 Колесников, С.В. Нейросетевой подход к анализу данных кардиомониторирования / С.В. Колесников, Е.М. Веселова // Вестник Амурского государственного университета. Серия: Естественные и экономические науки. – 2025. – № 109. – С. 17-20

9 Колесников, С.В. Применение простой нейросети в задаче анализа временных рядов / С.В. Колесников, Е.М. Веселова // «День науки»: материалы XXXIV научной конференции Амурского государственного университета (13 марта 2025 г., Благовещенск). – Благовещенск : АмГУ, 2025. – С. 53-54

10 Колесников, С.В. Применение простых нейросетей для классификации сигналов электрокардиограммы человека / С.В. Колесников, Е.М. Веселова // Сборник материалов VIII Всероссийской национальной научной конференции молодых учёных «Молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований» (07-11 апреля 2025 г., Комсомольск-на-Амуре). – Комсомольск-на-Амуре: КНАГУ, 2025. – С. 105-108

11 Карпов, О.Э. Информационные технологии, вычислительные системы и искусственный интеллект в медицине / О.Э. Карпов, А.Е. Храмов – М.: ДПК Пресс, 2022. – 480 с. – ISBN 978-5-91976-232-4

12 Резник, Е.В. Алгоритм ведения пациентов с острой сердечной недостаточностью [Электронный ресурс] / Е.В. Резник, Ю.В. Гаврилов, И.Г. Никитин // Архивъ внутренней медицины. – 2018. – Режим доступа : <https://cyberleninka.ru/article/n/algorithm-vedeniya-patsientov-s-ostroy-serdechnoy-nedostatocnostyu>. – 17.12.2023.

13 Манжула, В.Г. Нейронные сети Кохонена и нечеткие нейронные сети в интеллектуальном анализе данных / В.Г. Манжула, Д.С. Федяшов // Фундаментальные исследования, 2011. – № 4. – С. 108-114.

14 Xianjun, Ni Research of Data Mining Based on Neural Networks // World Academy of Science, Engineering and Technology. – 2008. – № 39. – с. 381-384.

15 Бирюков, Е.В. Практическая реализация нечеткой нейронной сети при краткосрочном прогнозировании электрической нагрузки [Электронный ресурс] / Е.В. Бирюков, М.С. Корнев // Портал магистров ДонНТУ. – 2005. – Режим доступа : http://www.masters.donntu.edu.ua/2006/kita/chuykov/library/library/article_5.htm – 18.12.2023.

16 Назаров, А.В. Нейросетевые алгоритмы прогнозирования и оптимизации систем / А.В. Назаров, А.И. Лоскутов – СПб.: Наука и Техника, 2003. – 384 с.

17 Ярушкина, Н.Г. Основы теории нечетких и гибридных систем: учебное пособие / Н.Г. Ярушкина. – М.: Финансы и статистика, 2004. – 320 с. – ISBN 5-279-02776-6.

18 Luca Cerina. A sleep stage estimation algorithm based on cardiorespiratory signals derived from a suprasternal pressure sensor / Luca Cerina, Sebastiaan Overeem, Gabriele B. Papini, Johannes P. van Dijk, Rik Vullings, Fokke van Meulen, Marco Ross, Andreas Cerny, Peter Anderer, Pedro Fonseca // Journal of Sleep Research, 2023. – № e14015. – С. 1-11.

19 Jiahao Fan. EOGNET: A Novel Deep Learning Model for Sleep Stage Classification Based on Single-Channel EOG Signal / Jiahao Fan, Chenglu Sun, Meng Long, Chen Chen, Wei Chen // Frontiers in Neuroscience, 2021. – № 15. – С. 1-13.

20 Качаева, Г.И. Использование нейросетевых методов для автоматического анализа электрокардиограмм при диагностике заболеваний сердечно-сосудистой системы / Г.И. Качаева, А.Г. Мустафаев // Вестник Дагестанского государственного технического университета. Технические науки. – 2018. – Т. 45 (2). – С. 114-124.

21 Захаров, С.М. Спектральный анализ аритмий сердца / С.М. Захаров, М. С. Захаров // Вопросы радиоэлектроники. – 2018. – № 2. – С. 120-127.

22 Chang, K.-C. Usefulness of multi-labelling artificial intelligence in detecting rhythm disorders and acute ST-elevation myocardial infarction on 12-lead electrocardiogram / K.-C. Chang, P.-H. Hsieh, et al // European Heart Journal – Digital Health, 2021. – V.2. – I.2. – P. 299–310.

23 Kiani, K. Classification of 7 arrhythmias from ECG using fractal dimensions / K. Kiani, K. Maghsoudi // Journal of Bioinformatics and Systems Biology, 2019. – V. 2(3). – P. 53–65. – DOI: 10.26502/jbsb.5107008.

24 Старченкова, К.С. Мультифрактальный анализ variability сердечного ритма для задач диагностики аритмий / К.С. Старченкова, Л.А. Манило // Мягкие вычисления и измерения: материалы XXVI Международной конференции, Санкт-Петербург, 24-26 мая 2023 года. – Санкт-Петербург: Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова, 2023. – С. 419 – 422.

25 Евдокимова, В.В. Система фрактальной диагностики скейлинговых и спектральных характеристик сигналов аускультации легких / В.В. Евдокимова, Е.Ю. Афанасьева, А.Г. Масловская // Информатика и системы управления. – 2023. – № 3(77). – С. 48-62. – DOI: 10.22250/18142400_2023_77_3_48.

26 Евдокимова, В.В. Спектральные и фрактальные характеристики дыхательных шумов / В.В. Евдокимова, Л.С. Афанасов, А.Г. Масловская // Вестник Амурского государственного университета. Серия: Естественные и экономические науки. – 2023. – № 101. – С. 18-29. – DOI: 10.22250/20730268_2023_101_18.

27 Масловская, А.Г. Алгоритмы мультифрактального вейвлет-анализа в задачах спецификации растровых изображений самоподобных структур / А.Г. Масловская, Л.С. Афанасов // Вестник Томского государственного университета. Управление, вычислительная техника и информатика, – 2020. – Т. 53. – С. 61-70.

28 Афанасов, Л.С. Применение методов частотно-временного и мультифрактального анализа для исследования динамических характеристик солнеч-

ной активности / Л.С. Афанасов, А.Г. Масловская // Вестник Амурского государственного университета. Серия: Естественные и экономические науки. – 2019. – № 87. – С. 29-33.

29 Fengying, M. An automatic system for atrial fibrillation by using a CNN-LSTM model / M. Fengying, Z. Jingyao, C. Wie, L. Wie, et al. // Hindawi. Discrete Dynamics in Nature Society, 2020. – V. 3. – P. 1–9. – DOI: 10.1155/2020/3198783.

30 Кардиология и спирометрия [Электронный ресурс]. – Режим доступа : https://files.btlnet.com/cor/documents/2aaa381d-f1c9-4822-9fcc-0047a1d82d72/BTL_cardio_spiro_CAT_RU404_1435299563_original.pdf. – 15.12.2023.

31 Goldberger, A. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals / A. Goldberger, L. Amaral, L. Glass, J. Hausdorff et al. / Circulation, 2000. – V. 101 (23). – P. 215-220. – DOI: 10.13026/C2F305.

32 КТ Result 3.0 [Электронный ресурс]. – Режим доступа : <https://www.holtershop.ru>. – 18.02.2024.

33 The Fourier Transform and its Applications [Электронный ресурс]. – Режим доступа : <https://see.stanford.edu/materials/lsoftaee261/book-fall-07.pdf>. – 12.03.2024.

34 Экспонента: Преобразования Фурье [Электронный ресурс]. – Режим доступа : <https://docs.exponenta.ru/R2019a/Matlab/math/fourier-transforms.html>. – 25.04.2024.

35 Экспонента: MATLAB и быстрое преобразование Фурье [Электронный ресурс]. – Режим доступа : <https://docs.exponenta.ru/R2019a/Matlab/math/fourier-transforms.html>. – 25.04.2024.

36 Hurst, H.E. Long-term storage capacity of reservoirs. // Transactions of the American Society of Civil Engineers, 1951. – № 116, p. 770-808. – DOI: 10.1061/TACEAT.0006518.

37 Mandelbrot, B. Fractional Brownian Motions, Fractional Noises and Applications / B. Mandelbrot, J.W. Van Ness // SIAM Review, 1968. – № 10. – p. 422-437. – DOI: 10.1137/1010093.

38 Higuchi, T. Approach to an irregular time series on the basis of the fractal theory. Physica D: Nonlinear Phenomena, 1988. – № 31(2). – p. 277-283.

39 Shannon Claude, E. A Mathematical Theory of Communication. / Bell System Technical Journal, 1948. – № 27 – p. 379-423, 623-656.

40 Wolf, A. Determining Lyapunov exponents from a time series / A. Wolf, J.B. Swift, H.L. Swinney, J.A. Vastano // Physica D: Nonlinear Phenomena, 1985. – № 16(3). – p. 285-317.

41 Freedman, D.A. Statistics / D.A. Freedman, R. Pisani, R.A. Purves // W.W. Norton & Company, 2007. – 715 p.

42 MathWorks Help Center Documentation [Электронный ресурс]. – Режим доступа : <https://www.mathworks.com/help/deeplearning/ug/learning-vector-quantization-lvq-neural-networks-1.html> – 13.05.2024.

43 Экспонента: Нейронные сети Изучения векторного квантования (LVQ) [Электронный ресурс]. – Режим доступа : <https://docs.exponenta.ru/deeplearning/ug/learning-vector-quantization-lvq-neural-networks-1.html> – 13.05.2024.

ПРИЛОЖЕНИЕ А

Скрипт преобразования Фурье в среде Matlab

```
% Получение значений А и f первых двух пиков спектров Фурье

% Норма на ЭКГ
for line=1:10
    A = normal_for_testing(line,:); % массив исходных данных
    N = length(A);
    i = 1:N;
    Tmax = 1; % длительность временного интервала в секундах
    c = fft(A); % вычисление спектра
    j = 2:N/2;
    Cm(j-1) = abs(c(j-1))/(N/2); % вычисление нормированного амплитудно-частотного спектра

    % Вычисление вектора частот в Гц
    Freq(j-1) = (j-1)/Tmax;
    figure();
    plot(Freq, Cm);
    grid on;
end
```

ПРИЛОЖЕНИЕ Б

Скрипт r/s-анализа в среде Matlab

```
%% Получение значений R/S-анализа

%% Норма на ЭКГ
test_RS_h_norm = [];
test_RS_D_norm = [];

for line=1:10
    y = normal_for_testing(line,.)+1000;    % массив исходных данных
    N = length(y);

    % Преобразование в ряд [b(t)]=[y(t)]-1
    for i=1:(N-1)
        b(i)=log(y(i+1)./y(i));
    end
    N1 = length(b);

    % figure()
    % w1=1:N1;
    % plot(w1,b,'r-','LineWidth',1)
    % grid on
    %%%%%%%%%%%%%%%
    n0=10;%начальное значение точек на одном подпериоде, окончательное по
алгоритму = N1/2
    for n=n0:(N1/2)%основной цикл программы - меняем значение точек в каждом
подпериоде
        A=floor(N1/n);%берем целое значение, число подпериодов длиной n
        T=zeros(A,n);%создаем (каждый раз у него будет разная размерность:
%число строк равно числу подпериодов A, а число столбцов - числу точек
%в одном подпериоде n) и обнуляем массив T - в алгоритме это матрица
N(k,a)
        %формируем массив T, записывая в каждую строку значения ряда b(t) на со-
ответствующем строке подпериоде
        for a=1:A
            for j=1:n
                T(a,j)=b(n*(a-1)+j);
            end
        end
        %вычисляем среднее значение для каждого подпериода - записываем в
%массив La
        for a=1:A
            L(a)=(1/n)*sum(T(a,:));
        end

        %формируем матрицу X(a,k) как массив накопленных отклонений на каж-
дом подпериоде
        for a=1:A
            for k=1:n
                X(a,k)=sum(T(a,1:k))-L(a);
            end
        end
    end
end
```

```

end
%вычисляем диапазон - формируем массив R таких значений для каждого
%подпериода
Ri=0;
for a=1:A
    maX(a)=max(X(a,:));
    miX(a)=min(X(a,:));
    Ri(a)=maX(a)-miX(a);
end

```

Продолжение Приложения Б

```

%вычисляем выборочное стандартное отклонение
% %для каждого подпериода - формируем массив S
% Si=0;
% for a=1:A
% Si(a)=(1./n)*sum(((T(a,1:n)-L(a)).^2).^0.5);
% end
Si=zeros(1,A);
for a=1:A
    S=0;
    for j=1:n
        S=S+(T(a,j)-L(a)).^2;
    end
    Si(a)=sqrt(S/n);
end
%получаем нормированный размах - формируем массив Q, в который
%последовательно для каждого разбиения записываем суммированный
норм. размах
Q(n-n0+1)=(1/A)*sum(Ri./Si);
end %конец основного блока программы

Y=log(Q);
n=n0:1:(N1/2);
Z=log(n);

ss=n0:0.1:(N1/2);%задаем промежуточные для МНК значения
s1=log(ss);
z1=polyfit(Z,Y,1);%строим полином первой степени, аппроксимирующий
%в двойном логарифмическом масштабе зависимость log(R/S) от n

h = z1(1);           %значение показателя Херста
test_RS_h_norm = [test_RS_h_norm, h];
D = 2-h;            %значение фрактальной размерности
test_RS_D_norm = [test_RS_D_norm, D];

z2=polyval(z1,s1);
figure()
plot(Z,Y,'ro',s1,z2,'b-','LineWidth',1)
xlabel('n')
ylabel('R/S')
grid on;

```

end

ПРИЛОЖЕНИЕ В

Скрипт расчёта показателя Хигучи в среде Matlab

```
%% Получение значений анализа Хигучи

%% Норма
test_Khiguchi_norm = [];

for line=1:10

    A = normal_for_testing(line,:); % массив исходных данных
    N = length(A);
    i = 1:N;

    % Получение фрактальной размерности Хигучи

    % максимальной степени сокращения / степени растяжения времени
    kmax = fix(129.9 * sin(0.00001292 * N + 0.04488) + 18.82 * sin(0.00006488 * N + 1.332));
    Lmk = zeros(kmax, kmax);
    % средняя длина кривой
    for k = 1:kmax
        for m = 1:k
            Lmi = 0;
            for i = 1:fix((N - m)/k)
                Lmi = Lmi + abs(A(m + i*k) - A(m + (i - 1)*k));
            end
            % нормировочный коэффициент
            Ng = (N - 1)/(fix((N - m)/k)*k);
            % длины кривых
            Lmk(m,k) = (Lmi * Ng)/k;
        end
    end
    % полная средняя длина кривой
    Lk = zeros(1, kmax);
    for k = 1:kmax
        Lk(1,k) = sum(Lmk(1:k,k))/k;
    end
    % рассчитать логарифмические значения для расчета наклона
    lnLk = log(Lk);
    lnk = log(1./(1:kmax));
    % полином первой степени, аппроксимирующий в двойном
    % логарифмическом масштабе зависимость log(L(k)) от k
    c = polyfit(lnk, lnLk, 1);
    s = log(1./(1:0.1:kmax));
    z = polyval(c, s);

    figure(Name = "Logarithmic dependence")
    plot(lnk, lnLk, 'ro', s, z, 'b-', 'LineWidth', 1)
    ylabel("ln(L(k))")
    xlabel("ln(1/k)")
    grid on;

    D = c(1);
```

```
test_Khiguchi_norm = [test_Khiguchi_norm, D];  
disp(['Фрактальная размерность Хигучи равна ', num2str(D)])  
end
```

ПРИЛОЖЕНИЕ Г

Скрипт расчёта энтропии в среде Matlab

```
%% Получение значений энтропии

%% Норма
test_Entropy_norm = [];

for line=1:10

    data = normal_for_testing(line,:);    % массив исходных данных

    % Сумма всех элементов для получения общего числа событий N
    total = sum(data);
    % Расчёт вероятностей каждого события
    probabilities = data / total;

    % Удаление нулевых значений для избежания логарифмирования нуля
    probabilities = probabilities(probabilities > 0);

    % Вычисление энтропии по формуле Шеннона
    entropy = -sum(probabilities .* log2(probabilities));

    % disp(['Энтропия данных ', num2str(line), ' сигнала: ', num2str(entropy)]);
    test_Entropy_norm = [test_Entropy_norm, entropy];

end
```

ПРИЛОЖЕНИЕ Д

Скрипт расчёта показателя Ляпунова в среде Matlab

```
%% Норма
test_Lyapunov_norm = [];

for line=1:10

    timeSeries = normal_for_testing(line,:); % массив исходных данных
    tau = 4; % задержка для фазовой реконструкции
    d = 3; % d - размерность вложения
    epsilon = 10; % epsilon - начальное расстояние между близкими траекториями
    deltaT = 1; % deltaT - шаг времени для измерения расстояния

    N = length(timeSeries);
    T = N - (d-1)*tau; % общее количество доступных вложенных точек

    % Формирование вложенных векторов
    phaseSpace = zeros(T, d);
    for i = 1:T
        for j = 1:d
            phaseSpace(i, j) = timeSeries((i-1)+(j-1)*tau+1);
        end
    end

    % Инициализируем расчет показателя Ляпунова
    sumLogDiv = 0;
    count = 0;

    % Анализируем каждую точку в фазовом пространстве
    for i = 1:T-deltaT
        % Находим ближайшую точку в заданном начальном промежутке
        distances = vecnorm(phaseSpace(i+1:min(i+epsilon,T),:) - phaseSpace(i,:), 2, 2);
        [~, idx] = min(distances);

        j = i + idx;

        if j + deltaT <= T
            initialDistance = norm(phaseSpace(i,:) - phaseSpace(j,:));
            evolvedDistance = norm(phaseSpace(i+deltaT,:) - phaseSpace(j+deltaT,:));

            if initialDistance > 0 && evolvedDistance > 0
                sumLogDiv = sumLogDiv + log(evolvedDistance / initialDistance);
                count = count + 1;
            end
        end
    end

    % Расчет показателя Ляпунова
    lyapunovExponent = sumLogDiv / (deltaT * count);

    % disp(['Показатель Ляпунова ', num2str(line), ' сигнала: ', num2str(lyapunovExponent)]);
end
```

```
test_Lyapunov_norm = [test_Lyapunov_norm, lyapunovExponent];
```

```
end
```

ПРИЛОЖЕНИЕ Е

Скрипт расчёта 1-4 квартилей в среде Matlab

```
%% Получение 1, 2, 3, 4 квартилей

%% Н о р м а
Q1_norm = [];
Q2_norm = [];
Q3_norm = [];
Q4_norm = [];

for line=1:10

    dataArray = normal_for_testing(line,:);    % массив исходных данных

    % Расчет квартилей
    Q1 = quantile(dataArray, 0.25); % Первый квартиль
    Q2 = quantile(dataArray, 0.50); % Второй квартиль (медиана)
    Q3 = quantile(dataArray, 0.75); % Третий квартиль
    Q4 = quantile(dataArray, 1.00); % Четвертый квартиль (максимальное значение)

    % Вывод результатов
    % disp(['Сигнал ', num2str(line)]);
    % fprintf('Первый квартиль (Q1): %.2f\n', Q1);
    % fprintf('Второй квартиль (Q2, медиана): %.2f\n', Q2);
    % fprintf('Третий квартиль (Q3): %.2f\n', Q3);
    % fprintf('Четвертый квартиль (Q4, максимум): %.2f\n', Q4);
    % fprintf('\n');

    Q1_norm = [Q1_norm, Q1];
    Q2_norm = [Q2_norm, Q2];
    Q3_norm = [Q3_norm, Q3];
    Q4_norm = [Q4_norm, Q4];

end
```

ПРИЛОЖЕНИЕ 3

Скрипт нейросети LVQ в среде Matlab

```
%% %%%%%%%%%%%  
%% %%%%%%%%%%%  
%% Подготовка датасета из mat-файла  
% Загрузка данных из файла-датасета  
% Датасет: 2 таблицы по 10 сигналов нормы и патологии ЭКГ, каждый в строку  
из 360 double-значений  
data = load('data_10x10.mat');  
  
% Получение всех полей (таблиц) из структуры данных  
fields = fieldnames(data);  
  
% Создание нового массива для переноса строк данных  
NewArray = [];  
  
% Перенос строк из каждого поля в новый массив  
% Подряд сначала сигналы нормы, затем сигналы патологии  
for i = 1:numel(fields)  
    fieldData = data.(fields{i});  
    % Проверяем, является ли значение таблицей или матрицей  
    if ismatrix(fieldData)  
        % Если да, добавляем каждую строку в массив  
        for row = 1:size(fieldData, 1)  
            NewArray = [NewArray; fieldData(row, :)];  
        end  
    else  
        error('Поле "%s" не является таблицей или матрицей. Убедитесь, что дан-  
ные формата числового массива.', fields{i});  
    end  
end  
% Преобразуем массив в читаемый для нейросети вид: теперь 1 столбец – 1  
сигнал из 360 значений (раньше была 1 строка), порядок сигналов прежний  
P = NewArray';  
  
% Отобразить подготовленный датасет (по желанию Ctrl+T – раскомментиро-  
вать две строки ниже)  
%                               Ctrl+R – закомментировать  
% disp('Данные датасета подготовлены для работы:');  
% disp(Dataset);  
  
% Готовим вектор-строку для распределения сигналов в датасете по груп-  
пам (1 – норма, 2 - патология)  
NormPat = [1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2];  
% Преобразование вектора NormPat в матрицу T с двумя строками  
T = ind2vec(NormPat);  
% Отобразить подготовленную матрицу (по желанию Ctrl+T – раскомментиро-  
вать строку ниже)  
%                               Ctrl+R – закомментировать  
% disp(T);
```

```

% Разделение данных на обучающую (60%), валидационную (20%) и тестовую
(20%) выборки
[trainInd, valInd, testInd] = dividerand(20, 0.6, 0.2, 0.2);
% Распределение по соответствующим массивам данных датасета Р и век-
тора-строки Т
P_train = P(:, trainInd);
T_train = T(:, trainInd);
P_val = P(:, valInd);
T_val = T(:, valInd);
P_test = P(:, testInd);

T_test = T(:, testInd);

```

Продолжение Приложения 3

```

%% Создание сети LVQ

% Создание новой сети встречного распространения требует параметров:
% - матрица минимальных и максимальных значений входных элементов
minmax(P)
% - число скрытых нейронов 4 – по-умолчанию (можно
варьировать)
% - вектор с элементами, указывающими долю каждого из классов [0.5
0.5] – 10 нормы и 10 патологии (НЕЛЬЗЯ ИЗМЕНЯТЬ)
% - величина коэффициента обучения 0.01 – значение по-
умолчанию (можно варьировать)
net = newlvq(minmax(P), 4, [0.5 0.5], 0.01);
% Задание количества итераций обучения (можно варьировать)
net.trainParam.epochs = 50;

% Настройка параметров деления данных
% (специальный код для настройки приёма нейросетью разгруппированного
датасета)
net.divideFcn = 'divideind';
net.divideParam.trainInd = trainInd;
net.divideParam.valInd = valInd;
net.divideParam.testInd = testInd;

% ОБУЧЕНИЕ НЕЙРОСЕТИ
net = train(net, P, T); % Закомментировать/стереть ";" для вывода системных дан-
ных о ходе обучения в окно команд
% Симуляция результата (вывод на экран), смотрим нижнюю строку: 0 – норма,
1 – патология
% Первые 10 сигналов – норма, вторые 10 сигналов – патология
Y = sim(net, P)

%% Подсчёт показателей работы нейросети
% Вычисление и вывод accuracy и confusion matrix для обучающих, валидационных и
тестовых данных
[accuracy_train, confMat_train] = calcAccuracyAndConfMat(net, P_train, T_train);

```

```

disp("Training Accuracy (%):");
disp(accuracy_train);
disp("Training Confusion Matrix:");
disp(confMat_train);

[accuracy_val, confMat_val] = calcAccuracyAndConfMat(net, P_val, T_val);
disp("Validation Accuracy (%):");
disp(accuracy_val);
disp("Validation Confusion Matrix:");
disp(confMat_val);

[accuracy_test, confMat_test] = calcAccuracyAndConfMat(net, P_test, T_test);
disp("Testing Accuracy (%):");
disp(accuracy_test);
disp("Testing Confusion Matrix:");
disp(confMat_test);
%% %%%%%%%%%%%
%% %%%%%%%%%%%

```

Продолжение Приложения 3

```

%% Ручное тестирование нейросети на НОВОМ датасете
% Датасет: 2 таблицы, подряд 10 сигналов нормы и 18 сигналов патологии
ЭКГ, каждый в строку из 360 double-значений
% Принцип подготовки массива сигналов для теста нейросети прежний, по-
этому выведен отдельным m-файлом
TestDataSort; % Смотри приложение И
% Данные датасета для теста нейросети подготовлены для работы, имя мас-
сива – TAr (TestArray).

% Скрипт для ввода чисел с клавиатуры
fprintf('Введите порядковый номер сигнала из датасета (целое число 1-28). Для
выхода введите 0.\n');

while true
    % Запрос числа у пользователя
    signal = input('\nСигнал №');

    % Проверка на выход из программы
    if signal == 0
        fprintf('\n\n—————\nЗавершение про-
граммы.\nВсего доброго!\n');
        break;
    % Проверка, что число в допустимом диапазоне
    elseif signal >= 1 && signal <= 28
        sim(net, TAr(:,signal))
    else
        disp('Ошибка! Введите число от 1 до 28, или 0 для выхода. ');
    end
end
end

```

```
%% %%%%%%%%%%%
%% %%%%%%%%%%%
%% %%%%%%%%%%%
%% %%%%%%%%%%%
% НЕ ИЗМЕНЯТЬ!
% Определение функции calcAccuracyAndConfMat для разбора данных первого да-
тасета на 3 группы
function [accuracy, confMat] = calcAccuracyAndConfMat(net, P, T)
    Y = net(P);
    [~, predictedClasses] = max(Y, [], 1);
    [~, trueClasses] = max(T, [], 1);
    accuracy = sum(predictedClasses == trueClasses) / numel(trueClasses) * 100;
    confMat = confusionmat(trueClasses, predictedClasses);
end
%% %%%%%%%%%%%
%% %%%%%%%%%%%
```


ПРИЛОЖЕНИЕ К

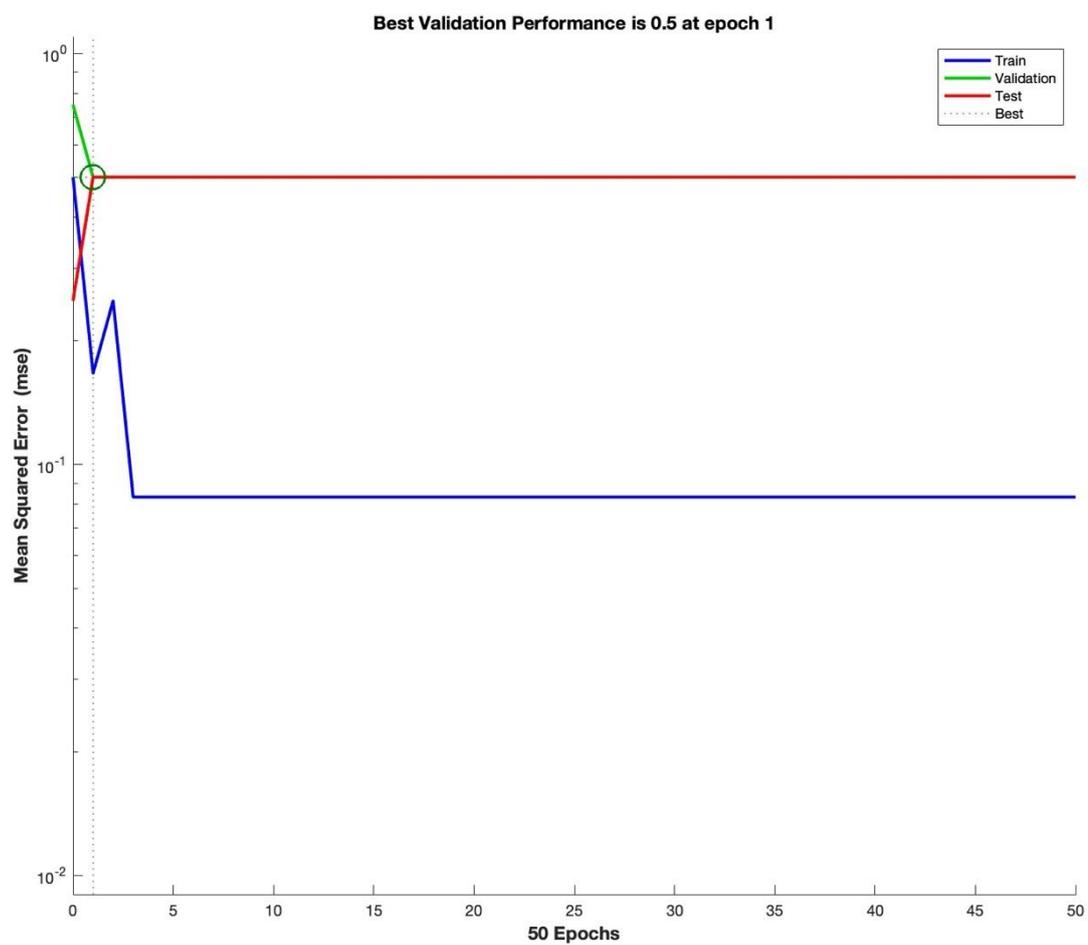


Рисунок К.1 – График лучшей производительности обучения

Продолжение Приложения К



Рисунок К.2 – Показатели ошибок обучения

Продолжение Приложения К

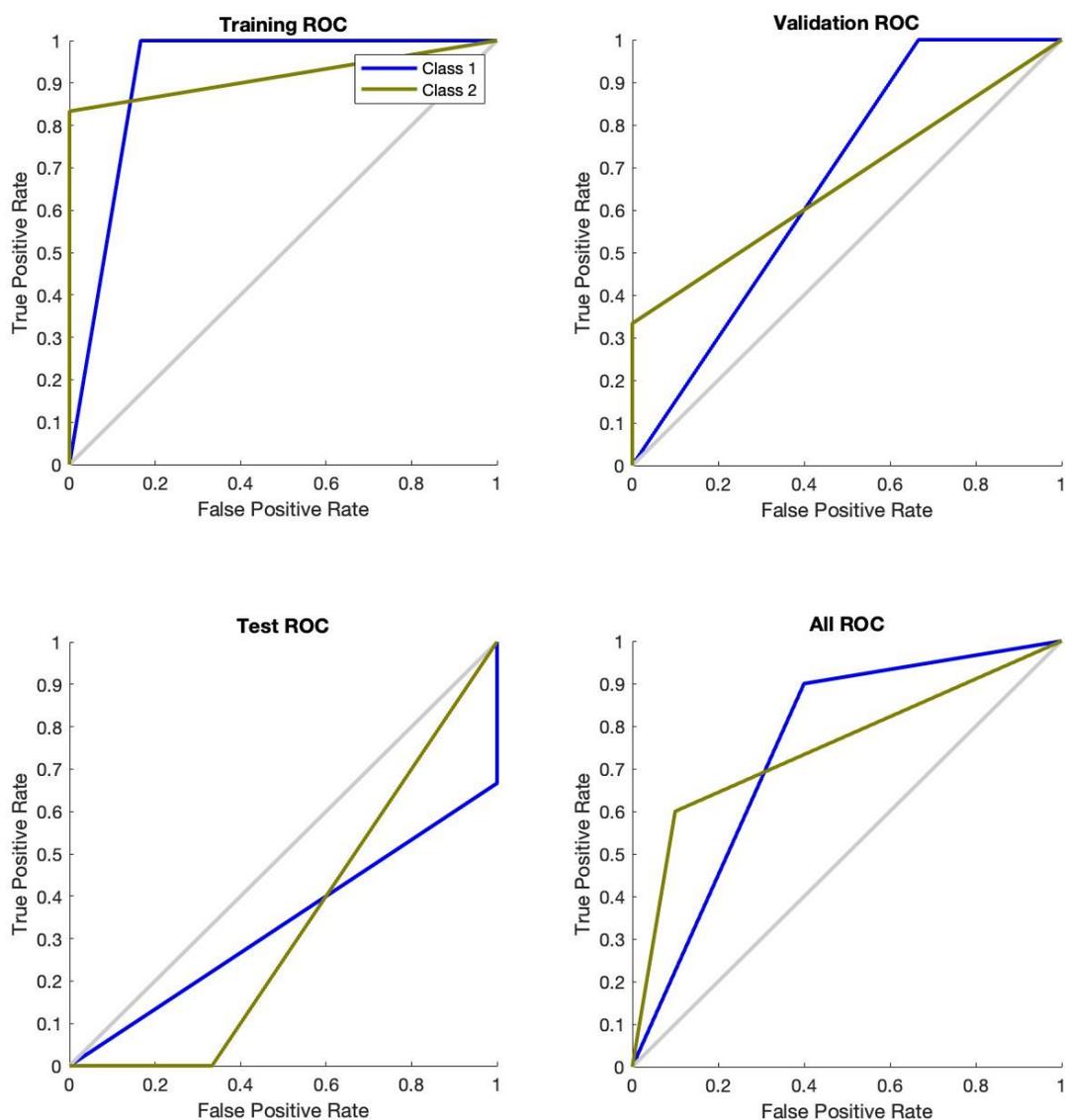


Рисунок К.3 – Показатели ошибок обучения