

НАО «Восточно-Казахстанский технический университет им. Д. Серикбаева»

УДК 004.89

На правах рукописи



ИСМУХАМЕДОВА АЙГЕРИМ МЭЛСАТОВНА

**Алгоритмическое обеспечение интеллектуальной системы поддержки
принятия клинических решений**

8D06101- Информационные системы (по отраслям)

Диссертация на соискание ученой степени
доктора философии (PhD)

Научные консультанты:
Увалиева И.М.,
PhD, ассоц.профессор

Бессмертный И.А.,
д.т.н., профессор

Республика Казахстан
Усть-Каменогорск, 2024

Содержание

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	4
ВВЕДЕНИЕ	5
1 АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ, ПРИМЕНЯЕМЫХ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ КЛИНИЧЕСКИХ РЕШЕНИЙ.....	11
1.1 Исследование процессов диагностики в информационно-клинической области.....	11
1.2 Методы применения машинного обучения в клиническом процессе и диагностике диабета	17
1.3 Эффективность использования и проблематика применения машинного обучения в клинических процессах.....	20
1.4 Вопросы применения систем поддержки принятия клинических решений	25
1.5 Модель диагностического процесса.....	29
1.6 Выводы первому разделу	39
2 АЛГОРИТМЫ И МОДЕЛИ ПОДДЕРЖКИ ПРИНЯТИЯ КЛИНИЧЕСКИХ РЕШЕНИЙ	41
2.1 Концептуальная модель процесса поддержки принятия клинических решений на основе методики EDA.....	41
2.2 Гибридный алгоритм поддержки клинических решений на основе андерсемплинга и автоматической оптимизация параметров.....	43
2.3 Алгоритм применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN	49
2.4 Алгоритм ансамблирования архитектур нейронных сетей для задач поддержки клинических решений.....	58
2.5 Выводы по второму разделу	61
3 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ И АРХИТЕКТУРНАЯ МОДЕЛЬ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ КЛИНИЧЕСКИХ РЕШЕНИЙ.....	63
3.1. Информационная модель интеллектуальной системы поддержки принятия клинических решений.....	63
3.2 Применение методики EDA на клинических данных эндокринологии и диабетологии.....	68
3.3 Экспериментальное исследование алгоритма поддержки клинических решений эндокринологии на основе технологии андерсэмплинга.....	77
3.4 Экспериментальное исследование алгоритма применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN.....	89
3.5 Оценка точности реализации алгоритма ансамблирования архитектур нейронных сетей LSTM и RNN для задач поддержки клинических решений	100
3.6 Архитектура системы поддержки принятия клинических решений	111

3.7 Выводы по третьему разделу	114
ЗАКЛЮЧЕНИЕ	116
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	117
ПРИЛОЖЕНИЕ А. СПРАВКА ОБ УЧАСТИИ В ПРОЕКТЕ ГРАНТОВОГО ФИНАНСИРОВАНИЯ	126
ПРИЛОЖЕНИЕ Б. СПРАВКА ОБ УЧАСТИИ В ПРОЕКТЕ «ЖАС ҒАЛЫМ»	127
ПРИЛОЖЕНИЕ В. АВТОРСКОЕ СВИДЕТЕЛЬСТВО «АЛГОРИТМ ПОДДЕРЖКИ КЛИНИЧЕСКИХ РЕШЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИИ АНДЕРСЭМПЛИНГА»	128
ПРИЛОЖЕНИЕ Г. АВТОРСКОЕ СВИДЕТЕЛЬСТВО «ПРОГРАММНЫЙ МОДУЛЬ ДИАГНОСТИРОВАНИЯ КЛИНИКО-ГЕМАТОЛОГИЧЕСКИХ СИНДРОМОВ».....	129
ПРИЛОЖЕНИЕ Д. АВТОРСКОЕ СВИДЕТЕЛЬСТВО «ДИАГНОСТИКА КЛИНИКО-ГЕМАТОЛОГИЧЕСКИХ СИНДРОМОВ НА ОСНОВЕ МОРФОЛОГИЧЕСКОЙ КЛАССИФИКАЦИИ»	130
ПРИЛОЖЕНИЕ Е. АКТ ВНЕДРЕНИЯ В ПРОИЗВОДСТВО С ТОО «ЮВЕНТАМЕД».....	131
ПРИЛОЖЕНИЕ Ж. АКТ ВНЕДРЕНИЯ В УЧЕБНЫЙ ПРОЦЕСС КАСУ ...	132
ПРИЛОЖЕНИЕ И. АКТ ВНЕДРЕНИЯ В УЧЕБНЫЙ ПРОЦЕСС ВКТУ Д.СЕРИКБАЕВА	134
ПРИЛОЖЕНИЕ К. ПОКАЗАТЕЛЬ ИНДЕКС ХИРША.....	136
ПРИЛОЖЕНИЕ Л. СОГЛАШЕНИЯ И ЛИЦЕНЗИЙ НА ИСПОЛЬЗОВАНИЕ ДАННЫХ	137
ПРИЛОЖЕНИЕ М. ТАБЛИЦА СВЯЗЕЙ БАЗЫ ДАННЫХ МІМІС ІІІ.....	138
ПРИЛОЖЕНИЕ Н. ИСХОДНЫЙ КОД	141

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ВОЗ - всемирная организация здравоохранения
ККСНВО - комитет по контролю в сфере науки и высшего образования
МНПК - международная научно-практическая конференция
ИИ – искусственный интеллект
МКБ-9 - девятая версия международной классификации болезней
МКБ-10 - десятая версия международной классификации болезней
МКБ-11 - одиннадцатая версия международной классификации болезней
СД - сахарный диабет
ППКР - поддержка принятия клинических решений
СППКР - системы поддержки принятия клинических решений
СППР - системы поддержки принятия решений
ЭПЗ - электронный паспорт здоровья
EHR - electronic health record (электронная медицинская карта)
ICD - International Classification of Diseases. МКБ
LSTM - долгая краткосрочная память (Long short-term memory)
RNN - рекуррентная нейронная сеть (recurrent neural network)
CNN - сверточная нейронная сеть (convolutional neural)
MDSS – medical decision support system

ВВЕДЕНИЕ

Актуальность темы исследования. С развитием информационных технологий медицинская индустрия переживает значительные трансформации, связанные с цифровизацией больших массивов данных. Эти изменения открывают новые возможности для улучшения качества медицинского обслуживания, однако также порождают вызовы, связанные с необходимостью эффективного анализа и интерпретации сложных и разнообразных данных. В частности, быстрое увеличение объёма доступной медицинской информации требует разработки новых подходов и методов обработки данных, которые могли бы справляться с этими задачами на качественно новом уровне.

Данное исследование вписывается в широкомасштабный проект по цифровизации медицинской отрасли в Республике Казахстан (Kazakhstan 2050 [1], eHealth [2], Digital Kazakhstan [3], Электронный паспорт здоровья (ЭПЗ) [4], реализация концепции Smart City) и касается реализации системы поддержки принятия клинических решений на базе искусственного интеллекта в рамках национальной системы здравоохранения. Такое внедрение инновационных технологий в медицине не только улучшит качество оказания помощи пациентам за счёт более ранней и точной диагностики, но и способствует оптимизации административных процессов, гарантируя надёжное и эффективное управление медицинскими данными.

Цифровая трансформация здравоохранения является процессом внедрения современных цифровых технологий и инноваций в сферу здравоохранения с целью повышения качества медицинских услуг, улучшения доступа к ним, оптимизации управления медицинскими процессами и повышения эффективности работы учреждений здравоохранения. Этот процесс включает в себя использование новых технологий для улучшения диагностики, лечения, мониторинга, а также управления данными и пациентами. Одним из ключевых аспектов цифровой трансформации здравоохранения является использование алгоритмов машинного обучения в анализе медицинских данных, например, для интерпретации изображений (рентген, МРТ, УЗИ), диагностики заболеваний, прогнозирования заболеваний и оптимизации лечения. Алгоритмы машинного обучения могут помочь в обработке больших объёмов данных, что ускоряет процесс принятия решений.

Применение искусственного интеллекта и машинного обучения в обработке данных пациентов с хронической ишемией и в других медицинских областях значительно улучшило результаты в диагностике, терапии и прогнозировании заболеваний [5]. Джулиан Мутц и Кэтрин Льюис использовали методы машинного обучения для определения биологического возраста на основе данных о психических характеристиках, связанных с ускоренным старением [6]. Искусственный интеллект и машинное обучение играют ключевую роль в разработке новаторских подходов к лечению болезни Паркинсона и ранней диагностике онкологических заболеваний с высокой

смертностью [7,8].

Теория и практика внедрения информационных технологий в медицинскую сферу, относительно диагностирования и корректировки лечения различных заболеваний рассмотрены в трудах казахстанских ученых М.Т.Ипалакова [9], М.Е.Мансурова [10,11], А.К.Мукашева [12], Н.П.Сапарходжаев, А.К. Мукашева [13,14] и других.

Исследования показывают, что большинство моделей машинного обучения достигают точности более 95% в диагностике хронического лимфоцитарного лейкоза, а также обеспечивают 100% точность в дифференциации этого заболевания от других патологий[15]. Методы машинного и глубокого обучения активно применяются для устранения диагностических пробелов в случаях наследственной аритмии [16]. Также важную роль эти технологии играют в выборе оптимальных алгоритмов для лечения рака молочной железы [17,18].

Для улучшения результатов исследователи часто используют комбинации из 5-6 моделей машинного обучения, включая базовые методы классификации в рамках глубокого обучения [19]. Индийские специалисты применяют ряд классификаторов машинного обучения, включая NB, KNN, SVM и алгоритмы деревьев решений, такие как ID3 и C4.5, для прогнозирования и диагностики диабета [20]. Коллеги из Пакистана демонстрируют применение шести известных алгоритмов машинного обучения, таких как SVM, KNN, логистическая регрессия, деревья решений, случайный лес и наивный Байес для прогнозирования диабета, достигая точности до 77% [21].

В рамках данного исследования для исследования процессов принятия клинических решений была выбрана сфера диабетологии. Сахарный диабет, широко известный как диабет, является серьёзной глобальной проблемой в области здравоохранения, затрагивающей миллионы людей по всему миру. Это заболевание достигло уровня эпидемии во многих регионах мира, при этом его распространённость продолжает расти. Согласно международным медицинским данным Всемирной организации здравоохранения, примерно 422 миллиона человек на глобальном уровне страдают от этого заболевания, что составляет приблизительно 6,028% от общей численности населения[22,23]. В контексте Республики Казахстан статистика ВОЗ показывает, что 11,5% населения страдают от сахарного диабета; среди них 11,7% составляют женщины и 11,3% - мужчины[24]. Национальный регистр Республики Казахстан за 2021 год фиксирует 317 597 зарегистрированных случаев сахарного диабета, включая 314 407 взрослых, 2 379 детей до 14 лет и около 811 подростков в возрастной категории от 15 до 17 лет [25].

Шведские ученые исследуют применение машинного обучения в профилактических программах по борьбе с диабетом 2 типа, выявляя ключевые рискованные факторы развития этого заболевания [26].

Таким образом, актуальность темы заключается в том, что машинное обучение и искусственный интеллект становятся ключевыми технологиями в

разработке методик и подходов в системах поддержки принятия медицинских решений. Государственные программы служат платформой для внедрения и тестирования этих передовых технологий, и чем обширнее база данных о пациентах, тем выше точность анализа и быстрее реализация новых уникальных программных продуктов и решений в области диагностики и прогнозирования заболеваний.

Актуальность диссертационной работы также подтверждается тем, что исследование выполнено в рамках НИР по договору №321/23-25 от 03.08.2023 по теме АР19679525 «Программный комплекс диагностики клиничко-гематологических синдромов для электронного паспорта здоровья», выполняемому в рамках бюджетной программы «Грантовое финансирование научных исследований» (Приложение А), а также НИР по договору №128/ЖГ 5-24-26 от 02.06.2024 по теме АР22683316 «Применение алгоритмов машинного обучения для систем поддержки принятия врачебных решений» выполняемому в рамках бюджетной программы «Грантовое финансирование молодых ученых по проекту «Жас ғалым» на 2023-2025 годы» (Приложение Б).

Объектом исследования является система поддержки принятия клинических решений.

Предметом исследования алгоритмическое обеспечение интеллектуальной системы поддержки принятия клинических решений в эндокринологии и диабетологии.

Идея работы - применение технологий машинного обучения и искусственного интеллекта для задач интеллектуальной поддержки процессов принятия клинических решений в эндокринологии и диабетологии, позволяющих внести вклад в реализацию глобальную стратегию Всемирной организации здравоохранения по электронному здравоохранению на 2020-2025 годы.

Цель исследования заключается в разработке алгоритмов интеллектуальной поддержки принятия клинических решений на основе алгоритмов машинного обучения.

Для достижения поставленной цели необходимо выполнить следующие **исследования** и решить **основные задачи**:

- исследование процессов диагностики в информационно-клинической области и особенностей систем поддержки принятия клинических решений;
- изучение вопросов эффективности использования и проблематики применения технологии машинного обучения для систем поддержки принятия клинических решений;
- разработка концептуальной модели процесса поддержки принятия клинических решений на основе методики EDA;
- разработка гибридного алгоритма поддержки клинических решений на основе андерсемплинга и автоматической оптимизации параметров;
- разработка алгоритма применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN;

- разработка алгоритма ансамблирования архитектур нейронных сетей для задач поддержки клинических решений;
- разработка информационной модели интеллектуальной системы поддержки принятия клинических решений;
- реализация методики EDA на клинических данных эндокринологии и диабетологии;
- проведение экспериментального исследования алгоритма поддержки клинических решений эндокринологии на основе технологии андерсэмплинга;
- оценка эффективности алгоритма прогнозирования диабета на основе модели глубокой нейронные сети с оптимизированными гиперпараметрами;
- оценка точности реализации алгоритма ансамблирования архитектур нейронных сетей LSTM и RNN для задач поддержки клинических решений;
- разработка архитектуры системы поддержки принятия клинических решений.

Научная новизна диссертационного исследования заключается в том, что впервые для повышения эффективности процессов поддержки принятия клинических решений в эндокринологии и диабетологии предложен комплекс алгоритмов, интегрирующий технологию андерсэмплинга и ансамблирования архитектур нейронных сетей LSTM и RNN.

Основные научные положения, выносимые на защиту:

- *гибридный алгоритм* поддержки клинических решений на основе андерсэмплинга и автоматической оптимизация параметров;
- *алгоритм* применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN;
- *алгоритм* ансамблирования архитектур нейронных сетей для задач поддержки клинических решений.

Практическая значимость результатов диссертационного исследования заключается в применимости предложенного комплекса алгоритмов, интегрирующий технологию андерсэмплинга и ансамблирования архитектур нейронных сетей LSTM и RNN для решения задач информационно-аналитической поддержки принятия управленческих решений при сопровождении технологических бизнес-процессов в эндокринологии и диабетологии.

Научно-обоснованные теоретические и экспериментальные результаты диссертационной работы использованы в научном проекте по теме «Программный комплекс диагностики клиничко-гематологических синдромов для электронного паспорта здоровья», выполняемого в рамках бюджетной программы «Грантовое финансирование научных исследований», о чем свидетельствует справка об участие в проекте (Приложение А).

Стоит отметить, что экспериментальные результаты работы использованы в научном проекте по теме «Применение алгоритмов машинного обучения для систем поддержки принятия врачебных решений», выполняемого в рамках конкурса на грантовое финансирование исследований молодых ученых по проекту «Жас ғалым» на 2024-2026 годы, о чем

свидетельствует справка об участие в проекте (Приложение Б).

Получены свидетельства о государственной регистрации прав на объекты авторского права №49449 от 04.09.2024 «Алгоритм поддержки клинических решений на основе технологии андерсэмплинга» (программа для ЭВМ) (Приложение В), так же №4737 от 01.08.2019 «Программный модуль диагностирования клиничко-гематологических синдромов» (Приложение Г) и №41784 от 05.01.2024 «База данных дифференциального диагностирования клиничко-гематологических синдромов на основе алгоритма морфологической классификации», выполнено в рамках проекта АР19679525 (Приложение Д).

Разработанная в диссертационной работе модули по методам управления большими данными, включая их сбор, хранение и обработку, на основе работы с медицинскими данными в контексте СППКР была успешно применена на базе производства ТОО «ЮвентаМед», о чем свидетельствует акт внедрении (Приложение Е).

Разработанная в диссертационной работе методы управление IT проектами и этические аспекты использования данных были внедрены в учебный процесс ОП 7М04104 «IT Менеджмент», КАСУ на 2024-2025 года в виде дисциплинах «Design and Implementation of Software System» и «Digital business modeling», о чем свидетельствует акт внедрении (Приложение Ж).

Результаты диссертационного исследования были внедрены в учебный процесс в 2023-2024 учебном году в следующих курсах лекционных и практических занятий ОП «Математическое и компьютерное моделирование», ВКТУ им. Д.Серикбаева, а именно в дисциплины «Моделирование биологических процессов» и «Основы нейронных сетей», о чем свидетельствует акт внедрения (Приложение И).

Методы исследования. В работе используется методы машинного и глубокого обучения, методы статистического анализа, методы обработки больших данных, теория принятия решений, а также методы экспериментального исследования и моделирования архитектуры системы.

Апробация результатов диссертационного исследования. Основные результаты диссертационной работы докладывались на научных семинарах кафедры «Информационные системы» ВКГТУ им. Д. Серикбаева и на следующих международных научно-практических конференциях: «ADVANCED SCIENCE» (Пенза, Россия, 2017 г.); «4th International Conference on Computer and Technology Applications» (Стамбул, Турция, 2018 г.); «4th International Conference on Engineering and MIS» (Стамбул, Турция, 2018 г.); «IV Международная научно-техническая конференция студентов, магистрантов и молодых ученых» (Усть-Каменогорск, Казахстан, 2018 г.); «XVIII международный научно-исследовательский конкурс «Лучшая научная статья 2018»» (Пенза, Россия, 2018 г.); «Computational and Information Technologies in Science, Engineering and Education: 9th International Conference, CITech 2018» (Усть-Каменогорск, Казахстан, 2018 г.); «Application of Information and Communication Technologies-AICT 2018» (Алматы, Казахстан, 2018 г.); «5th International Conference on Engineering and MIS» (Нур-Султан,

Казахстан, 2019 г.); «12th IEEE International Conference «Application of Information and Communication Technologies - AICT2019»» (Баку, Азербайджан, 2019 г.); «VI Международная научно-техническая конференция студентов, магистрантов и молодых ученых «Творчество молодых - инновационному развитию Казахстана»» (Усть-Каменогорск, Казахстан, 2020 г.); «2021 International Young Engineers Forum (YEF-ECE)» (2021 г.); «7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)» (Анкара, Турция, 2023 г.).

Личный вклад автора. Постановка проблемы, формализация всех рассмотренных задач, поиск методов и алгоритмов их решения, а также приведенные в диссертации научные и практические результаты, их анализ, формирование итоговых выводов осуществлены лично автором диссертации.

Публикации по теме диссертационного исследования. По теме диссертации опубликовано 28 научных работы, из них 8 в научных журналах, рекомендованных Комитетом по контролю в сфере науки и высшего образования МОН РК; 12 в трудах международных конференций; 3 публикации индексируются в базе данных SCOPUS; 1 монография [27].

Публикации, опубликованные в рамках научного исследования в научных изданиях Scopus и Web of Science были процитированы 49 раз (Приложение К).

Результаты экспериментального исследования алгоритмов интеллектуальной поддержки систем принятия клинических решений были описаны в статье на тему «Integrating machine learning in electronic health passport based on WHO study and healthcare resources», опубликованной в журнале «Informatics in Medicine Unlocked», имеющий в базе данных Scopus показатель процентиля по CiteScore равный 86 по направлению «Computer Science Applications».

Структура и объем диссертационной работы. Диссертация состоит из введения, трех глав, заключения, библиографии (174 наименований) и приложений.

1 АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ, ПРИМЕНЯЕМЫХ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ КЛИНИЧЕСКИХ РЕШЕНИЙ

1.1 Исследование процессов диагностики в информационно-клинической области

Изучение диагностических процессов как в информационной, так и в клинической областях — это сложно структурированная и быстро развивающаяся область исследований, пересекающая области здравоохранения, информационных технологий и науки о данных. Целью этой области является повышение точности, эффективности и общей результативности диагностики заболеваний или состояний за счет использования достижений в области обработки информации, клинических методологий и технологических инноваций.

Процессы клинической диагностики включают ряд систематических шагов, используемых медицинскими работниками для определения состояния здоровья пациента и выявления любых заболеваний или состояний, которые у него могут быть. Эти процессы объединяют историю болезни пациента, физические осмотры, диагностические тесты и клиническое обоснование. Обзор ключевых компонентов представлен в таблице 1.1.

Таблица 1.1 - Обзор ключевых компонентов процессов клинической диагностики

	Компонент	Процесс	Описание
1.	История пациента	История болезни	Сбор информации о прошлых заболеваниях, операциях, аллергиях и текущем лечении
		Семейный анамнез	Выявление генетической предрасположенности путем понимания проблем со здоровьем, преобладающих в семье пациента
		Социальный анамнез	Изучение образа жизни пациента, занятий, привычек и других факторов, которые могут повлиять на здоровье
		Основная жалоба	Документирование основной причины, по которой пациент обращается за медицинской помощью
		Обзор систем	Проведение систематического исследования функционирования различных систем организма (например, сердечно-сосудистой, дыхательной, желудочно-кишечной).
2.	Физический осмотр	Осмотр	Наблюдение за пациентом на предмет любых видимых признаков заболевания, таких как кожные заболевания, ненормальные движения или поза
		Пальпация	Ощупывание частей тела для выявления аномалий, таких как шишки, болезненность или увеличение органов

Продолжение таблицы 1.1.

		Перкуссия	Постукивание по поверхности тела для прослушивания издаваемых звуков, помогающее определить основную структуру
		Аускультация	Прослушивание внутренних звуков тела, таких как сердцебиение, звуки легких и кишечника, с помощью стетоскопа
		Измерение жизненно важных показателей	Проверка артериального давления, частоты сердечных сокращений, частоты дыхания и температуры для немедленной оценки состояния здоровья пациента
3.	Диагностическое тестирование	Лабораторные тесты	Анализ крови, мочи или других жидкостей организма для выявления отклонений. Общие тесты включают общий анализ крови (ОАК), уровень глюкозы в крови, уровень холестерина и функциональные тесты печени
		Визуализирующие исследования	Использование таких технологий, как рентген, компьютерная томография, МРТ и ультразвук, для визуализации внутренних структур и выявления отклонений
		Электродиагностические тесты	Запись электрической активности сердца (ЭКГ) или мозга (ЭЭГ) для диагностики связанных состояний
		Биопсия	Взятие образца ткани для микроскопического исследования с целью выявления таких заболеваний, как рак
4.	Дифференциальный диагноз	Клиническое обоснование	На основе собранной информации специалисты здравоохранения составляют список возможных диагнозов, известный как дифференциальный диагноз
		Исключение состояний	Использование дополнительных опросов, медицинских осмотров и диагностических тестов для сужения списка за счет исключения маловероятных состояний.
5.	Диагностика	Окончательный диагноз	Постановка наиболее вероятного диагноза на основе имеющихся данных
		Клиническое заключение	Объединение результатов диагностики с клиническим опытом и предпочтениями пациента для принятия обоснованных решений о его здоровье
6.	План лечения	Разработка плана	На основе диагноза создается план лечения, который может включать лекарства, изменение образа жизни, операции или другие методы лечения
		Обучение пациентов	Информирование пациентов об их состоянии, вариантах лечения и необходимых корректировках образа жизни
		Последующее наблюдение	Планирование регулярных посещений для отслеживания прогресса пациента и внесения любых необходимых корректировок в план лечения
7.	Документация	Медицинские записи	Ведение полных записей обо всех результатах, диагнозах, методах лечения и планах последующего наблюдения для обеспечения непрерывности ухода и юридической документации

Следуя этим шагам, специалисты здравоохранения стремятся поставить точный диагноз и назначить эффективное лечение, что в итоге улучшит результаты лечения пациентов.

В развивающейся области современного здравоохранения управление и анализ медицинских данных стали иметь решающее значение. Внедрение электронного паспорта здоровья (ЭПЗ) представляет собой шаг к улучшению обработки данных пациентов, предлагая комплексный инструмент для сбора, хранения и обработки медицинской информации. Это нововведение согласуется с Глобальной стратегией цифрового здравоохранения Всемирной организации здравоохранения (ВОЗ) на 2020-2025 годы [28,29] которая направлена на повсеместное улучшение здравоохранения с помощью цифровых технологий[30,31] с упором на равенство [32] и включение[33].

Несмотря на эти достижения, реализация таких стратегий в различных национальных условиях сопряжена с серьезными проблемами. В таблице 1.2 представлены основные трудности для информационной и клинической областей.

Конвергенция информационных технологий и клинической практики способствует более интегрированному подходу к диагностике. Например, искусственный интеллект и машинное обучение используются не только для анализа электронных медицинских данных, но также для интерпретации сложных медицинских изображений, прогнозирования прогрессирования заболевания и предложения персонализированных планов лечения[34]. Кроме того, носимые технологии и Интернет медицинских вещей (IoMT) устраняют разрыв между данными о состоянии здоровья пациентов и клинической диагностикой, позволяя осуществлять непрерывный мониторинг и раннее обнаружение потенциальных проблем со здоровьем.

В целом, изучение диагностических процессов в информационной и клинической областях занимает центральное место в продолжающейся трансформации здравоохранения. Оно обещает сделать диагностику более быстрой, точной и более ориентированной на пациента, что в итоге приведет к улучшению показателей здоровья и качества жизни пациентов. Для достижения этих целей необходим доступ к информации, касающейся пациента и клинических процессов. Пользуясь такими данными, появляется возможность провести более глубокий анализ и строить системы поддержки принятия клинических решений, позволяющие прогнозировать различные медицинские события и подбирать наилучшее персонализированное лечение. Соответственно тематикой данной работы является диагностика, основанная на данных.

Таблица 1.2 - Сложности реализации стратегий управления и анализа медицинских данных в информационной и клинической областях

	Область	Направление	Проблематика
1.	Информационная область	Диагностика, основанная на данных	С появлением аналитики больших данных диагностические процессы становятся все более управляемыми данными. Модели машинного обучения и алгоритмы искусственного интеллекта (ИИ) обучаются на обширных наборах данных для выявления закономерностей и аномалий, которые могут указывать на конкретные состояния здоровья или заболевания.
		Электронные медицинские карты (ЭМК).	Использование ЭМК произвело революцию в способах хранения, доступа и анализа информации о пациентах. Электронные медицинские записи предоставляют богатый источник данных для систем прогнозной аналитики и поддержки принятия решений, обеспечивая более точную и персонализированную диагностику[35–38].
		Телемедицина и дистанционная диагностика.	Расширение телемедицины открыло новые возможности для дистанционной диагностики пациентов. Это предполагает использование инструментов цифровой связи и устройств дистанционного мониторинга для сбора данных о пациентах и проведения первоначальной диагностической оценки без необходимости физического присутствия.
2.	Клиническая сфера	Точная медицина	Этот подход адаптирует лечение к индивидуальным характеристикам каждого пациента, включая генетику, окружающую среду и образ жизни. Объединив эту информацию с клиническими данными, поставщики медицинских услуг могут принимать более точные решения по диагностике и лечению.
		Передовые методы визуализации	Развитие более сложных технологий визуализации (таких как МРТ, КТ и ПЭТ) расширяет возможности диагностики заболеваний с большей точностью и детализацией, что часто приводит к раннему выявлению и улучшению результатов лечения пациентов.
		Тестирование на месте оказания медицинской помощи	Такое тестирование позволяет проводить диагностические тесты в месте оказания помощи пациенту или рядом с ним, обеспечивая немедленные результаты и позволяя быстро принимать решения. Это особенно полезно в чрезвычайных ситуациях или в условиях ограниченного доступа к комплексному лабораторному оборудованию.

Для качественной постановки диагноза с помощью машинного обучения и искусственного интеллекта необходимо понимать какие данные, в каком виде

нужны, а также как эти данные могут собираться [39]. Так, например авторами изучалось как можно использовать электронные медицинские записи для прогнозирования и прогрессирования диабета [40]. Наиболее эффективная модель машинного обучения смогла предсказать осложнение диабета с точностью 85% и выявить пациентов с риском быстрого прогрессирования. Анализ историй болезни пациентов и критериев отбора разработанных моделей позволяет прогнозировать заражение, выживаемость и риск критических состояний [41–45].

Основа создания систем, которые собирают, хранят и анализируют медицинскую информацию от пациентов из разных стран мира, лежит в сфере обширных данных. Обработка этого огромного количества данных, обычно называемых большими данными, дает возможность сформулировать методологии для прогнозирования таких факторов, как уровень заболеваемости, смертность, осложнения и многое другое [46]. Обработка больших данных позволила разработать интеллектуальные системы принятия решений, в том числе в медицине [41,47–50].

Анализ данных, машинное обучение и искусственный интеллект используются для улучшения диагностических процессов в различных областях медицины. Эффективность применения технологий машинного обучения в медицине подтверждена многочисленными исследованиями в различных областях медицины. Методы машинного обучения и глубокого обучения с присущей им способностью независимо извлекать ценную информацию из данных предлагают значительное преимущество в прогнозировании, что приводит к их широкому использованию в области медицины, что особенно подчеркивает важность и высокое значение методов глубокого обучения в здравоохранении [51–54].

Так, например, результаты проекта SPIN-UTI показали, что алгоритм Support Vector Machines (SVM) имеет точность 88% и является полезным инструментом для раннего прогнозирования пациентов с высоким риском внутрибольничных инфекции при поступлении в отделение интенсивной терапии [55].

Авторы утверждают, что применение искусственного интеллекта и машинного обучения (ML) способствовало улучшению результатов диагностики, лечения и прогноза для пациентов с CLTI [5]. Изучая диагностические свойства алгоритмов машинного обучения при заболеваниях периферических артерий, авторы приходят к выводу, что машинное обучение позволяет более точно классифицировать и прогнозировать заболевание [56]. Методологии машинного обучения используются для прогнозирования биологического возраста путем использования данных, связанных с идентифицируемыми психическими чертами, которые коррелируют с ускоренным старением [6]. К такому же выводу пришли авторы исследований болезни Паркинсона [7], онкологических заболеваний с высокой летальностью [8,15,17,18] и синдромов наследственной аритмии [16]. Использование алгоритмов глубокого обучения снижает вероятность

ложноположительного диагноза, тем самым устраняя негативное психологическое воздействие, с которым сталкивается пациент [57]. Использование методов машинного обучения и алгоритмов глубокого обучения при диагностике COVID-19 способствовало взятию пандемии под контроль [58–67].

Системы поддержки принятия клинических решений (СППР) представляют собой комплексные платформы, которые объединяют информацию о пациенте с клиническими знаниями и алгоритмами для повышения точности и эффективности диагностики, лечения и управления состоянием здоровья. Основные особенности таких систем включают следующие аспекты:

1) Диагностика, основанная на данных. СППР используют большие объемы данных и алгоритмы машинного обучения для анализа клинической информации, выявления паттернов и принятия решений на основе доказательных данных (использование электронных медицинских записей (ЭМК) для прогнозирования заболеваний, оценки рисков и выбора оптимальных методов лечения).

2) Персонализированная медицина. СППР могут адаптировать лечение в зависимости от индивидуальных характеристик пациента, включая его генетическую предрасположенность, образ жизни и медицинскую историю.

3) Интеграция с клиническими процессами. СППР поддерживают медицинских специалистов на всех этапах клинического процесса - от сбора данных о пациенте до разработки и реализации плана лечения. Они могут объединять данные из разных источников, таких как лабораторные исследования, результаты диагностических тестов и информация о применяемых лекарствах, для создания целостной картины состояния здоровья пациента.

4) Использование алгоритмов искусственного интеллекта и машинного обучения. Внедрение ИИ в СППР обеспечивает автоматизацию процесса диагностики и прогнозирования, что значительно повышает скорость и точность медицинских решений. Алгоритмы глубокого обучения, в частности, продемонстрировали высокую эффективность в таких областях, как диагностика диабета, распознавание изображений и прогнозирование прогрессирования заболеваний.

Таким образом, системы поддержки принятия клинических решений играют важную роль в повышении качества медицинской помощи, оптимизируя процесс диагностики и лечения за счет интеграции данных, применения передовых технологий и адаптации к индивидуальным потребностям каждого пациента.

1.2 Методы применения машинного обучения в клиническом процессе и диагностике диабета

Внедрение машинного обучения (МО) в клинической диагностике диабета с годами значительно изменилось благодаря более широкому применению искусственного интеллекта (ИИ) в здравоохранении. Использование машинного обучения в этом контексте восходит к концу 20-го века, а в 21-м веке оно набирает обороты благодаря достижениям в области вычислительной мощности и доступности данных [68].

Истоки применения машинного обучения для диагностики диабета начались с использования более простых статистических моделей и превратились в более сложные алгоритмы машинного обучения. На ранних стадиях для прогнозирования диабета на основе различных факторов риска, таких как возраст, вес, семейный анамнез и другие клинические показатели, обычно использовались статистические методы, такие как логистическая регрессия [69,70].

К концу 1990-х и началу 2000-х годов, когда хранение данных стало более эффективным, а вычислительная мощность увеличилась, исследователи начали изучать более сложные модели машинного обучения. В этот период были приняты такие алгоритмы, как деревья решений, нейронные сети и машины опорных векторов (SVM) [71–74]. Эти модели обеспечивали более детальный анализ сложных наборов данных, позволяя лучше выявлять закономерности, которые могли быть не очевидны для людей-наблюдателей или не обнаруживались с помощью более простых статистических тестов.

Ключевые события, повлиявшие на развитие применения машинного обучения в клиническом процессе и диагностике диабета представлены в таблице 1.3.

В последние годы, с появлением глубокого обучения, возможности машинного обучения в диагностике и лечении диабета еще больше расширились [53,75–77].

Модели глубокого обучения, которые особенно хороши при обработке больших объемов данных и распознавании сложных закономерностей, теперь используются для:

- анализ изображений сетчатки для выявления диабетической ретинопатии, распространенного осложнения диабета [78,79];
- прогнозирования уровня глюкозы в крови, используя данные непрерывного мониторинга уровня глюкозы [80,81];
- персонализации плана лечения на основе прогностических моделей, учитывающих уникальный профиль здоровья человека [82,83].

Прогнозирование диабета, особенно диабета 2 типа, стало важным направлением медицинских исследований в связи с ростом распространенности этого заболевания во всем мире [84]. Возможность предсказать, кто подвержен риску развития диабета, позволяет принимать

более ранние меры, которые могут отсрочить или даже предотвратить начало заболевания [85].

Таблица 1.3 - Ключевые события, повлиявшие на развитие применения машинного обучения в клиническом процессе и диагностике диабета

	Область	События
1.	Сбор данных и электронные медицинские записи (ЭМК).	Широкое внедрение электронных медицинских записей в 2000-х годах дало значительный толчок развитию подходов, основанных на данных, включая машинное обучение. Электронные медицинские записи представляют собой богатый, структурированный и доступный источник данных о пациентах, критически важный для обучения надежных моделей машинного обучения.
2.	Прогнозное моделирование	Модели машинного обучения стали использоваться не только для диагностики диабета, но и для прогнозирования его возникновения. Были разработаны модели для анализа сочетания исторических данных о состоянии здоровья, факторов образа жизни и генетической информации для прогнозирования индивидуального риска развития диабета.
3.	Интеграция нескольких типов данных.	Последние достижения позволили объединить различные типы данных, включая геномные данные, результаты лабораторных исследований и даже данные мониторинга в реальном времени с носимых устройств. Это позволяет получить целостное представление о состоянии здоровья пациента и разработать более персонализированные стратегии лечения диабета.

Исследования в этой области используют сочетание традиционных факторов риска и новых подходов, основанных на данных.

Исторически прогнозирование диабета основывалось на выявлении ключевых факторов риска, представленных в таблице 1.4.

Таблица 1.4 - Ключевые факторы риска заболевания диабетом

	Признак	Факторы риска
1.	Возраст	Пожилые люди подвергаются более высокому риску
2.	Семейный анамнез	Наличие близкого родственника с диабетом увеличивает риск.
3.	Вес	Избыточный вес или ожирение являются значительным фактором риска.
4.	Диета и физическая активность	Плохое питание и отсутствие физической активности повышают риск.
5.	Гестационный диабет	Наличие диабета во время беременности или при рождении крупного ребенка (более 9 фунтов) может увеличить риск для женщины.
6.	Этническая принадлежность	Некоторые этнические группы, такие как афроамериканцы, латиноамериканцы, коренные американцы и американцы азиатского происхождения, подвергаются более высокому риску[86–88].

С появлением больших данных и машинного обучения прогнозирование диабета значительно улучшается. Факторы, способствующие улучшению прогнозирования диабета представлены в таблице 1.5.

Таблица 1.5 - Факторы способствующие улучшению прогнозирования диабета

	Факторы	Описание
1.	Электронные медицинские карты (ЭМК)	ЭМК являются богатым источником данных о пациентах, включая демографические данные, истории болезни, результаты лабораторных исследований и многое другое. Модели машинного обучения могут анализировать эти данные, чтобы идентифицировать людей с высоким риском развития диабета [85,89–91].
2.	Модели машинного обучения	Для прогнозирования диабета применяются несколько методов машинного обучения, такие как логистическая регрессия, деревья решений, случайные леса и нейронные сети. Эти модели могут обрабатывать сложные взаимодействия между различными факторами риска более эффективно, чем традиционные статистические методы [92].
3.	Генетические маркеры	Исследования также изучали роль генетики в риске диабета. Выявляя конкретные генетические маркеры, ученые стремятся более точно предсказать вероятность развития диабета [93].
4.	Данные об образе жизни	Все чаще данные об образе жизни с носимых устройств (например, о физической активности и режиме сна) интегрируются с традиционными данными о здоровье для улучшения моделей прогнозирования диабета [94].
5.	Прогнозная аналитика в общественном здравоохранении	В более широком масштабе прогнозная аналитика используется для выявления групп населения, подверженных риску развития диабета, что позволяет проводить целевые вмешательства в области общественного здравоохранения [94,95].

Конечная цель прогнозирования диабета - обеспечить возможность раннего и целенаправленного вмешательства, которое может включать:

- Изменение образа жизни: улучшение питания, увеличение физической активности и достижение здорового веса могут значительно снизить риск развития диабета.

- Лекарства: в некоторых случаях могут быть рекомендованы лекарства для контроля уровня сахара в крови или устранения других факторов риска, таких как высокое кровяное давление или уровень холестерина.

- Мониторинг и обучение: Лицам, отнесенным к группе высокого риска, может быть полезен регулярный мониторинг и обучение тому, как снизить риск.

Модели машинного и глубокого обучения становятся все более востребованными для предсказания и диагностики диабета, поскольку они могут анализировать большие объемы данных и выявлять скрытые закономерности. Например:

- Различные методы машинного обучения применяются для диагностики диабета на ранних стадиях, что позволяет снизить риск осложнений и улучшить качество лечения [19,20].

- Сравнительный анализ подходов машинного и глубокого обучения для предсказания ранних стадий диабета показал, что глубинные модели обладают более высокой точностью в выявлении паттернов и предсказании риска заболевания [96].

- Использование машинного обучения для прогнозирования риска диабета среди шведского населения среднего возраста демонстрирует значительный потенциал в выявлении групп риска, что способствует более ранней профилактике заболевания [96].

Таким образом, прогнозирование диабета с помощью современных технологий машинного и глубокого обучения позволяет эффективно оценивать риски развития заболевания и принимать превентивные меры на ранних стадиях [96–98].

1.3 Эффективность использования и проблематика применения машинного обучения в клинических процессах

Интеграция машинного обучения (МО) в клинические процессы представляет собой революционный сдвиг в здравоохранении, предлагая потенциальные улучшения в эффективности, точности и результатах лечения пациентов. Однако, хотя использование МО в клинических условиях обещает значительные преимущества, оно также создает ряд проблем и трудностей, которые необходимо решить. Преимущества применения машинного обучения в клинических процессах представлены в таблице 1.6.

Таблица 1.6 - Преимущества применения машинного обучения в клинических процессах

	Преимущества применения	Описание
1.	Точность диагностики	Алгоритмы машинного обучения, особенно модели глубокого обучения, продемонстрировали замечательную точность в диагностике заболеваний по медицинским изображениям, таким как рентгеновские снимки, МРТ и компьютерная томография. Эти модели могут обнаруживать закономерности, которые трудно увидеть человеческому глазу.

Продолжение таблицы 1.6

2.	Предиктивная аналитика	Машинное обучение может прогнозировать исходы течения заболеваний пациентов, риски повторной госпитализации и потенциальные осложнения путем анализа огромных объемов исторических данных о пациентах. Это помогает в индивидуальном планировании ухода и профилактических мерах.
3.	Персонализация лечения	Анализируя данные пациентов и выявляя закономерности, модели машинного обучения могут рекомендовать персонализированные планы лечения, которые могут быть более эффективными для конкретного состояния отдельного пациента [99].
4.	Операционная эффективность	Алгоритмы машинного обучения могут оптимизировать работу больницы: от управления потоком пациентов до оптимизации планирования и сокращения времени ожидания, тем самым повышая общую эффективность медицинских услуг.
5.	Открытие и разработка лекарств	Машинное обучение ускоряет процесс открытия лекарств, прогнозируя потенциальную эффективность соединений, тем самым сокращая время и затраты, связанные с традиционными процессами разработки лекарств.

Проблемы и вызовы применения машинного обучения в клинических процессах представлены в таблице 1.7.

Таблица 1.7 - Проблемы и вызовы применения машинного обучения в клинических процессах

	Проблемы и вызовы	Описание
1.	Конфиденциальность и безопасность данных	Использование данных пациентов для обучения моделей машинного обучения вызывает серьезные опасения по поводу конфиденциальности и безопасности. Обеспечение конфиденциальности конфиденциальной медицинской информации имеет первостепенное значение.
2.	Качество и доступность данных	Производительность моделей машинного обучения во многом зависит от качества и количества данных, используемых для обучения. Неполные, неточные или нерепрезентативные данные могут привести к неправильным прогнозам или диагнозам. сокращая время и затраты, связанные с традиционными процессами разработки лекарств.

Продолжение таблицы 1.7

3.	Интерпретируемость и объяснимость	Многие модели машинного обучения, особенно сети глубокого обучения, работают как «черные ящики», из-за чего врачам трудно понять, как модели приходят к своим выводам. Отсутствие прозрачности может помешать доверию и принятию. сокращая время и затраты, связанные с традиционными процессами разработки лекарств.
4.	Интеграция в клинические рабочие процессы	Интеграция инструментов МО в существующие клинические рабочие процессы и обеспечение того, чтобы они дополняли, а не нарушали повседневную работу поставщиков медицинских услуг, является серьезной проблемой. сокращая время и затраты, связанные с традиционными процессами разработки лекарств.
5.	Нормативно-правовые и этические аспекты	Навигация по нормативно-правовой базе для приложений МО в здравоохранении является сложной задачей. Обеспечение соответствия этих инструментов всем этическим и юридическим требованиям имеет решающее значение для их широкого внедрения.

Исследователи с заметной точностью демонстрируют эффективность моделей машинного обучения в выявлении и прогнозировании заболеваний. Тем не менее, полезность этих технологий в тонкой области прогнозирования диабета остается недостаточно изученной. Частично это происходит из-за используемых алгоритмов. Например, в сфере альтернативных алгоритмов машинного обучения традиционные методы, такие как наивный Байес, логистическая регрессия и машины опорных векторов (SVM), приводят к экспоненциальному росту сложности вычислений из-за расширения данных, что приводит к неадекватным результатам [46,100–102] [7]. Напротив, древовидные алгоритмы предоставляют более надежную альтернативу, смягчая определенные ограничения, с которыми сталкиваются традиционные методы [47,48,103]. Ансамблевые методы и алгоритм дерева решений в машинном обучении предлагают подходы к высокоточной диагностике и прогнозированию рака молочной железы [104–106], сердечно-сосудистых заболеваний [107,108] и COVID-19 [109].

Решение представленных выше проблем требует междисциплинарного подхода, включающего сотрудничество между медицинскими работниками, специалистами по обработке данных, специалистами по этике и политиками. Усилия по повышению прозрачности и объяснимости моделей машинного обучения, установлению надежных мер конфиденциальности и безопасности данных, а также разработке беспристрастных, высококачественных наборов обучающих данных имеют решающее значение. Кроме того, постоянное обучение и повышение квалификации медицинских работников будет иметь жизненно важное значение для ответственного и этического использования всего его потенциала.

Поскольку технология ML продолжает развиваться, ее интеграция в здравоохранение обещает значительно улучшить клинические процессы. В

процессе анализа существующих методов и моделей было рассмотрено большое количество исследований Казахских и зарубежных авторов, посвященным применению алгоритмов машинного обучения в создании систем поддержки принятия клинических решений.

В частности, были рассмотрены работы, посвященные диагностике и профилактике диабета.

Все авторы отмечают, что для построения эффективной модели наиболее важным пунктом является определение подходящего набора данных. Несмотря на большое количество исследований и активное применение машинного обучения остаются пробелы в прагматическом анализе больших данных. Существующие исследования опираются на установленные маркеры заболеваний, отдельные типы медицинской информации или неполный набор данных. В таблице 1.8 представлены данные относительно сравнения результатов точности по итогам литературного обзора.

Несмотря на высокий уровень точности в работе моделей использование медицинских данных в системах поддержки принятия медицинских решений сопряжено с рядом проблем, которые могут повлиять на эффективность и надежность этих систем и также на точность моделей.

Медицинские данные часто поступают из разных источников, в том числе из разных больниц, клиник и систем электронных медицинских карт (ЭМК). Это может привести к несогласованности форматов и стандартов. Отсутствующие или неполные записи пациентов могут привести к неточным анализам и рекомендациям. Ошибки при вводе данных, устаревшая информация и неправильное кодирование могут подорвать надежность системы.

Обеспечение конфиденциальности данных пациентов имеет первостепенное значение. Нарушения могут привести к несанкционированному доступу к конфиденциальной медицинской информации. Соблюдение таких правил, как HIPAA (Закон о переносимости и подотчетности медицинского страхования) и GDPR (Общие правила защиты данных), требует надежных мер защиты данных, которые могут быть сложными в реализации и поддержании.

В разных системах здравоохранения используется различное программное обеспечение и форматы данных, что затрудняет беспрепятственную интеграцию и обмен данными. Отсутствие стандартизированной терминологии и протоколов может препятствовать эффективному обмену и интерпретации данных.

Медицинские данные часто являются многомерными и содержат множество переменных, которые необходимо анализировать одновременно. Многие медицинские записи содержат неструктурированные данные, такие как записи врача и описательные отчеты, которые трудно проанализировать с помощью обычных методов.

Таблица 1.8 - Представлены данные относительно сравнения результатов точности

Работа и исследователи	Модель	Точность
Прогнозное моделирование и аналитика диабета с использованием подхода машинного обучения [110]	SVM-Linear	89
Сравнительный анализ прогнозирования диабета на ранней стадии с использованием подхода машинного обучения и глубокого обучения [111]	XGboost	99
Подход машинного обучения на основе синтеза для прогнозирования возникновения диабета [112]	SVM-ANN ensemble	94.67
Прогнозирование диабета с использованием алгоритмов машинного обучения в здравоохранении [21]	SVM and KNN	77
Использование машинного обучения и искусственного интеллекта для улучшения выявления, лечения и результатов заболеваний периферических артерий [113]	Elastic net and Random Forest	87; 76
Прогнозирование диабета с использованием алгоритмов классификации [114]	Naive Bayes	76.30
Применение методов и технологий интеллектуального анализа данных для диагностики диабета [115]	C4.5 Decision Tree	90.62
Эмпирическое исследование консенсусной кластеризации для больших наборов данных ЭКГ[116]	Adaboost Decision Tree with Bagging	94.84
Ансамблевый подход машинного обучения для прогнозирования сахарного диабета II типа на основе показателей образа жизни[117]	Bagged DT	99.14
Сравнительное исследование, прогнозирование и развитие хронической болезни почек с использованием машинного обучения на основе клинических записей пациентов[118]	K-means clustering and RF	99.57
Сравнение алгоритмов машинного обучения для прогнозирования диабета [119]	KNN and AB	79.42
Прогнозирование диабета с помощью объединенного машинного обучения [120]	Fusion ML Decision	94.87
Ансамблевый подход к прогнозированию риска диабета на ранней стадии с использованием машинного обучения: эмпирическое исследование [121]	LR ensemble	77.60
Надежная модель прогнозной диагностики сахарного диабета с использованием алгоритмов машинного обучения [122]	Random Forest	82
Сравнительный анализ эффективности квантового машинного обучения с глубоким обучением для прогнозирования диабета [123]	Multilayer Perceptron	95
Прогнозирование диабета с использованием алгоритмов машинного обучения с выбором признаков и уменьшением размерности [124]	SVM with feed backward feature elimination	82.9

Исторические предвзятости в собранных данных могут привести к созданию предвзятых алгоритмов, которые увековечивают существующие различия в здравоохранении.

Алгоритмическая предвзятость модели машинного обучения могут создавать предвзятости на основе данных, на которых они обучаются, что потенциально может привести к несправедливым или дискриминационным результатам.

Крайне важно гарантировать, что системы, основанные на машинном обучении, предоставляет точные и клинически обоснованные рекомендации. Плохо проверенные системы могут привести к неправильному диагнозу и плану лечения. Системы нуждаются в тщательном тестировании и проверке в реальных условиях, чтобы подтвердить их эффективность в различных клинических условиях.

Медицинские работники могут сопротивляться внедрению новых технологий из-за знакомства с традиционными методами или скептицизма в отношении надежности системы. Необходимо пройти соответствующее обучение, чтобы пользователи могли эффективно взаимодействовать с системой и интерпретировать ее рекомендации.

Определение ответственности в случаях, когда MDSS предоставляет неверные рекомендации, может оказаться сложной задачей. Использование искусственного интеллекта и машинного обучения при принятии медицинских решений поднимает этические вопросы о том, в какой степени человеческое суждение должно быть дополнено или заменено технологиями.

Гарантия того, что система может обрабатывать большие объемы данных и масштабироваться с увеличением объема вводимых данных и спроса пользователей. Предоставление рекомендаций в реальном времени или почти в реальном времени требует значительных вычислительных ресурсов и эффективных алгоритмов.

Регулярные обновления и техническое обслуживание необходимы для поддержания системы в актуальном состоянии с учетом последних медицинских рекомендаций, методов лечения и открытий. Система должна иметь возможность адаптироваться к новым источникам данных и изменениям в медицинской практике с течением времени.

Решение этих проблем требует междисциплинарного подхода с участием специалистов здравоохранения, специалистов по обработке данных, инженеров-программистов и экспертов по правовым вопросам для разработки надежных систем.

Учитывая вышеперечисленные проблемы в данной работе, уделяется большое внимание обработке и подготовке используемых данных.

1.4 Вопросы применения систем поддержки принятия клинических решений

Системы поддержки принятия решений (СППР) в медицинской диагностике - это инструменты, предназначенные для помощи медицинским работникам в принятии клинических решений. Watson for Oncology использует искусственный интеллект, чтобы предоставить врачам научно обоснованные варианты лечения [125]. Он анализирует медицинские записи пациентов,

клинические исследования и мнения экспертов, чтобы помочь в принятии решений о лечении онкологических больных. DXplain - это система поддержки принятия клинических решений, которая помогает врачам диагностировать пациентов, предлагая возможные диагнозы на основе клинических данных пациента [126]. Она широко используется в медицинском образовании и клинической практике. Isabel Diagnosis Decision Support System - это диагностический инструмент, который помогает врачам составить список дифференциальных диагнозов, вводя клинические характеристики пациента [127]. Он использует большую базу данных заболеваний и симптомов, чтобы помочь в точной диагностике. QMR предоставляет подробную диагностическую информацию для широкого спектра заболеваний [128]. Это позволяет клиницистам вводить симптомы и данные для создания списка потенциальных диагнозов. MedCalc 3000 - это набор медицинских калькуляторов, инструментов поддержки принятия решений и клинических справочных материалов, предназначенных для поддержки научно обоснованной клинической практики [129]. PEPID предоставляет ряд инструментов поддержки принятия решений, включая информацию о лекарствах, медицинские калькуляторы и клинические справочники, которые помогают медицинским работникам принимать обоснованные клинические решения [130].

В области диагностики диабета также развиваются системы поддержки принятия решений [131]. AIDA - это прототип компьютерной системы, которая включает в себя модель взаимодействия глюкозы и инсулина при сахарном диабете I типа, а также основанную на знаниях систему для прогнозирования гликемии и выдачи рекомендаций по корректировке дозировки инсулина [132]. Авторы из Италии предлагают систему, представляющую собой пилотное приложение, специально разработанное для улучшения лечения хронического диабета 2 типа (СД2) [133]. Ее можно легко нацелить на эффективное лечение различных хронических заболеваний. DreamMed Advisor - это система поддержки принятия решений на основе искусственного интеллекта для управления диабетом [134]. Он анализирует данные непрерывного мониторинга уровня глюкозы, данные инсулиновой помпы и другую информацию, специфичную для пациента, чтобы предоставить персональные рекомендации по дозировке инсулина. mySugr - это мобильное приложение, предлагающее поддержку в лечении диабета [135]. Оно помогает пользователям отслеживать уровень глюкозы в крови, прием лекарств и другие показатели здоровья. Приложение обеспечивает анализ и поддержку принятия решений, чтобы помочь пользователям эффективно контролировать диабет.

Системы поддержки принятия медицинских решений (СППМР) предлагают множество преимуществ, которые повышают качество оказания медицинской помощи и улучшают результаты лечения пациентов. В таблице 1.9 представлены ключевые из них.

Таблица 1.9 - ключевые преимущества применения СППМР в медицинской сфере

	Преимущества применения	Результаты	Описание
1.	Повышенная точность диагностики	Поддержка принятия клинических решений	СППМР может анализировать огромные объемы данных и предоставлять научно обоснованные рекомендации, помогая врачам ставить более точные диагнозы.
		Раннее выявление заболеваний	Выявляя закономерности и тенденции в данных о пациентах, СППМР может помочь в раннем выявлении заболеваний, обеспечивая своевременное вмешательство.
2.	Индивидуальные планы лечения	Индивидуальные рекомендации	СППМР может учитывать индивидуальные характеристики пациентов и истории болезни, чтобы рекомендовать персонализированные планы лечения.
		Оптимизированное управление приемом лекарств	Системы могут предлагать лучшие лекарства и дозировки с учетом конкретного состояния пациента и реакции на предыдущее лечение.
3.	Повышенная эффективность	Оптимизированный рабочий процесс	СППМР может автоматизировать рутинные задачи, такие как ввод и извлечение данных, что позволяет медицинским работникам больше сосредоточиться на уходе за пациентами.
		Снижение административной нагрузки	Благодаря интеграции с электронными медицинскими записями СППМР может сократить время, затрачиваемое на выполнение административных задач.
4.	Доказательная практика	Доступ к новейшим рекомендациям	СППМР может предоставить врачам новейшие клинические рекомендации и результаты исследований, гарантируя, что уход за пациентами основан на самых последних фактических данных.
		Стандартизация медицинской помощи	Эти системы помогают стандартизировать протоколы лечения среди различных поставщиков и учреждений, что приводит к более последовательному и надежному лечению.
5.	Управление рисками и безопасность пациентов	Сокращение ошибок	Обеспечивая поддержку принятия решений, СППМР может помочь уменьшить количество человеческих ошибок при диагностике и лечении.
		Оповещения и напоминания	Системы могут выдавать предупреждения о потенциальных неблагоприятных взаимодействиях лекарств, аллергии или других рисках, повышая безопасность пациентов.

6.	Расширение взаимодействия с пациентами	Обучение пациентов	СППМР может предоставить пациентам информацию об их состоянии и вариантах лечения, помогая им принимать обоснованные решения относительно ухода за ними.
		Инструменты самоконтроля	Некоторые системы включают в себя инструменты, позволяющие пациентам контролировать свое здоровье, что приводит к улучшению лечения заболеваний и улучшению результатов.
7.	Экономия средств	Сокращение количества ненужных тестов	Предоставляя точную диагностическую поддержку, СППМР может помочь избежать ненужных тестов и процедур, сокращая расходы на здравоохранение
		Эффективное использование ресурсов	Оптимизация планов лечения и улучшение результатов лечения пациентов могут привести к более эффективному использованию ресурсов здравоохранения.
8.	Аналитика, основанная на данных	Управление здоровьем населения	СППМР может анализировать данные больших групп пациентов для выявления тенденций и улучшения стратегий общественного здравоохранения.
		Клинические исследования	эти системы могут поддерживать клинические исследования, предоставляя информацию на основе реальных данных, способствуя открытию новых методов лечения и вмешательств.
9.	Улучшение процесса принятия клинических решений	Поддержка сложных случаев	СППМР может помочь в сложных случаях, синтезируя большие объемы данных и предлагая потенциальные диагнозы и методы лечения, которые врачи, возможно, не рассматривали
		Повышенная уверенность в диагностике	Подтверждая клинические результаты научно обоснованными рекомендациями, СППМР может повысить уверенность врачей в своих решениях.

Интеграция СППМР в клиническую практику может изменить здравоохранение, сделав его более точным, эффективным и ориентированным на пациента.

Несмотря на широкое эффективное применение СППР в медицине существуют определенные проблемы и вызовы, которые стоит рассмотреть и учитывать при дальнейших разработках и внедрении. СППР иногда могут выдавать неточные или ненадежные рекомендации из-за ограничений базовых алгоритмов или неточностей данных. Это может привести к неправильному диагнозу или неправильному плану лечения, что может нанести вред пациентам. Клиницисты могут стать чрезмерно зависимыми от СППР, что

потенциально снижает их диагностические навыки и клиническое суждение. Это может привести к менее персонализированному уходу и потенциальному снижению клинического опыта с течением времени. Поэтому необходимо внедрять грамотные программы обучения медицинского персонала, а также привлекать медицинских специалистов к сотрудничеству, консультированию и исследованиям. Также важной проблемой является интеграция СППР с существующими системами электронных медицинских карт и другой ИТ-инфраструктурой больницы. Это может оказаться сложной и дорогостоящей задачей. Также плохая интеграция может привести к сбоям в рабочем процессе и снижению общей эффективности СППР.

Диабет - это динамическое состояние, которое может быстро меняться в зависимости от различных факторов, включая диету, физические упражнения и стресс. СППР не всегда может идти в ногу с этими быстрыми изменениями, что может привести к устаревшим или неактуальным рекомендациям. Поэтому так важно выявлять новые взаимосвязи между показателями пациента и уметь среагировать на изменения с помощью современных информационных технологий. К тому же точная диагностика и лечение диабета зависят от высококачественных и полных данных пациентов. Неполные или некачественные данные могут привести к неточным результатам СППР, влияя на качество медицинской помощи, оказываемой пациентам с диабетом. Диабет проявляется по-разному у каждого пациента, и такие факторы, как сопутствующие заболевания, взаимодействие лекарств и генетические факторы, могут влиять на лечение. СППР не всегда может учесть эту изменчивость, что приводит к выработке обобщенных, а не персонализированных рекомендаций.

1.5 Модель диагностического процесса

В современном мире, где здоровье населения становится одним из приоритетных направлений международной политики, особое значение приобретают принципы устойчивого развития в медицине. Эти принципы направлены на обеспечение высококачественного, доступного и эффективного медицинского обслуживания, способствующего улучшению здоровья на глобальном уровне.

Одним из ключевых аспектов устойчивого развития в медицине является использование стандартизированных медицинских протоколов, разработанных и одобренных Всемирной организацией здравоохранения (ВОЗ). Эти протоколы предназначены для того, чтобы гарантировать, что каждый пациент, независимо от своего местоположения и социально-экономического статуса, получает лечение, соответствующее самым высоким международным стандартам.

Применение унифицированных медицинских протоколов позволяет достичь нескольких ключевых целей устойчивого развития, такие как повышение качества лечения, обеспечение его доступности для широких

слоев населения и повышение эффективности использования медицинских и финансовых ресурсов.

Во всем мире ежедневно фиксируются многочисленные случаи заболеваний, требующие квалифицированного медицинского вмешательства для их диагностики и лечения. Врачи-специалисты применяют дифференциальную диагностику, опираясь на свой клинический опыт и современные методы лабораторных и инструментальных исследований. Этот комплексный подход позволяет точно установить диагноз и разработать оптимальную стратегию лечения для каждого пациента, что является ключевым для успешного восстановления здоровья. Алгоритм постановки диагноза состоит из 4 этапов согласно википедии [136].

Этапы алгоритма с пояснениями представлены в Таблице 1.10.

Таблица 1.10 - Этапы постановки диагноза

Этапы	Название	Описание
1.	Анамнез или опрос пациента	<ul style="list-style-type: none"> – жалобы пациента – история болезни – история жизни
2.	Физические методы осмотра	<ul style="list-style-type: none"> – осмотр пациента – температура – ощупывание – перкуссия (постукивание) – аускультация (прослушивание)
3.	Предварительный диагноз	Врач назначает на основе пунктов 1 и 2 лабораторные исследования + может быть еще инструментальные исследования (ЭКГ, томография и т.д.)
4.	Окончательный диагноз	На основании исследований и умозаключений специалиста (путем дифференциальной диагностики)

Процесс диагностики в медицине представляет собой сложную и повторяющуюся последовательность шагов, которым следуют медицинские работники для выявления основной причины симптомов и признаков у пациента. Обобщенная модель диагностического процесса представлена на рисунке 1.1.

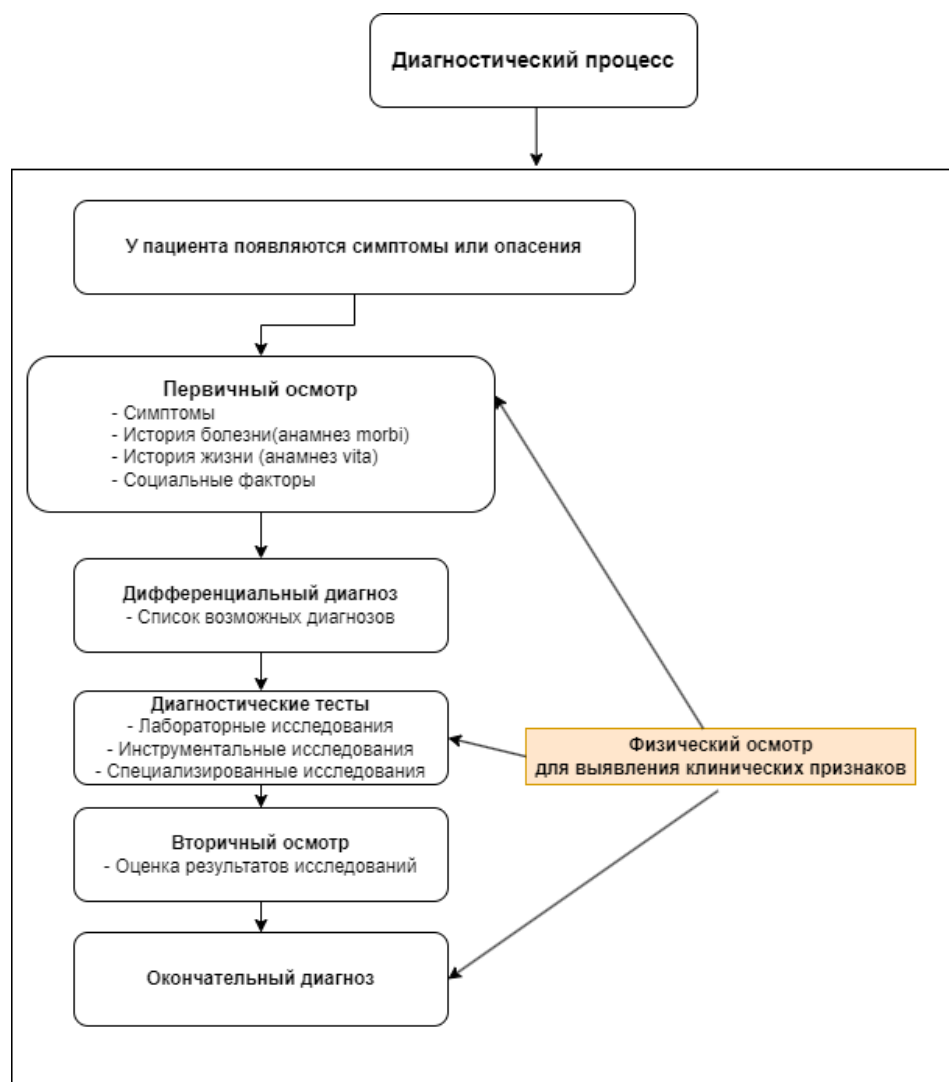


Рисунок 1.1 - Обобщённая модель диагностического процесса

Процесс начинается, когда у пациента появляются симптомы или опасения. Врач собирает исчерпывающую информацию об истории болезни пациента, включая симптомы, перенесенные заболевания, семейный анамнез и любые значимые социальные факторы или факторы окружающей среды. Проводится медицинский осмотр для выявления клинических признаков, которые могут помочь сузить круг возможных диагнозов. На основании собранного анамнеза и результатов физикального обследования врач составляет список возможных диагнозов, известный как дифференциальный диагноз. Для уточнения дифференциальной диагностики и приближения к окончательному диагнозу могут быть назначены различные диагностические тесты. Они могут включать лабораторные анализы, визуализационные исследования (такие как рентген, компьютерная томография, магнитно-резонансная томография) и другие специализированные тесты, имеющие отношение к симптомам. Клиницист оценивает результаты диагностических тестов в контексте истории болезни пациента и результатов физикального обследования. Этот этап может подтвердить диагноз, исключить определенные состояния или потребовать дальнейшего обследования.

Располагая достаточной информацией и результатами анализов, клиницист часто может поставить окончательный диагноз, который заключается в выявлении заболевания или состояния, объясняющего симптомы и признаки пациента. Данный процесс соответствует алгоритму постановки клинического диагноза, начиная от анамнеза, физического осмотра, предварительного диагноза, дополнительных методов обследования, до окончательного диагноза и назначения лечения.

Понимание структуры данных на каждом этапе процесса постановки диагноза и лечения важно для эффективного управления медицинской информацией. Структура данных, которая обычно используется на каждом из этапов медицинского процесса представлена в таблице 1.11.

Таблица 1.11 - Структура данных используемые в медицинских процессах

№	Процесс	Данные	Формат данных
1	Первичный прием и анамнез	Основные демографические данные (имя, возраст, пол), история болезни (анамнез заболеваний, анамнез жизни), жалобы пациента.	Текстовые записи, электронные медицинские карты (ЭМК), анкеты
2	Физический осмотр	Результаты физического осмотра, включая показатели, такие как температура тела, давление, осмотренные характеристики (цвет кожи, наличие отеков и т. д.).	Текстовые записи в ЭМК, таблицы.
3	Предварительный диагноз	Записи о предполагаемых диагнозах на основе симптомов и результатов осмотра.	Текстовые записи, возможно использование специализированных кодов заболеваний (например, ICD-10)
4	Дополнительные методы обследования	Лабораторные методы: результаты анализов крови, мочи и т. д. Инструментальные методы: изображения (рентген, МРТ), записи ЭКГ и т. д.	Лабораторные методы: числовые значения, таблицы, графики. Инструментальные методы: цифровые изображения, видео, графики.
5	Окончательный диагноз	Конкретный диагноз, установленный на основе всех доступных данных.	Текстовые записи, коды диагнозов.
6	Назначение лечения	План лечения, включая медикаменты, процедуры, рекомендации по изменению образа жизни.	Текстовые инструкции, дозировки лекарств, расписания лечения.
7	Следование лечению и мониторинг	Отчеты о прогрессе пациента, побочные эффекты, отклик на лечение.	Текстовые записи, диаграммы, таблицы с данными о состоянии здоровья.
8	Повторный визит и оценка	Информация о текущем состоянии здоровья, результаты повторных исследований.	Текстовые записи, обновленные медицинские изображения, таблицы.

На каждом этапе важно поддерживать целостность, конфиденциальность и доступность данных. Интеграция информационных систем в медицине, таких как электронные медицинские карты и системы управления данными, способствует оптимизации управления данными, облегчая доступ, обмен и анализ медицинской информации. Это в свою очередь улучшает качество лечения и повышает его эффективность.

Для предварительной или ранней диагностики заболеваний можно выделить особенно важные этапы, которые представлены в таблице 1.12.

Таблица 1.12 - Важные этапы предварительной диагностики и форматы данных

№	Процесс	Особенности	Данные	Формат данных
1	Первичный прием и анамнез	На этом этапе важно собрать подробный анамнез, включая историю болезни и жизни пациента. Информация о предыдущих заболеваниях, наследственности, жалобы пациента и другие данные могут указывать на риск развития определённых заболеваний	Основные демографические данные (имя, возраст, пол), история болезни (анамнез заболеваний, анамнез жизни), жалобы пациента.	Текстовые записи, электронные медицинские карты (ЭМК), анкеты
2	Физический осмотр	Физический осмотр включает методы, такие как осмотр, пальпация, перкуссия и аускультация. Эти методы могут выявить первые физические признаки заболевания, даже если клинические симптомы ещё не проявились полностью.	Результаты физического осмотра, включая показатели, такие как температура тела, давление, осмотренные характеристики (цвет кожи, наличие отеков и т. д.).	Текстовые записи в ЭМК, таблицы.

Продолжение таблицы 1.12

3	Дополнительные методы обследования	Лабораторные методы:		
		Ранняя диагностика: Различные анализы крови могут выявить изменения, указывающие на начальные стадии заболеваний, даже если другие симптомы отсутствуют. Например, повышенные уровни определённых маркеров в крови могут указывать на ранние стадии воспаления, инфекции или онкологические заболевания.	Результаты анализов крови, мочи и т. д.	Числовые значения, таблицы, графики.
		<i>Инструментальные методы:</i>		
		Раннее выявление: Методы визуализации, такие как УЗИ, рентген, и МРТ, могут выявить структурные изменения в органах и тканях, которые могут быть признаками начальных стадий заболевания. Это особенно важно для заболеваний, таких как рак, сердечно-сосудистые заболевания и другие хронические состояния.	Изображения (рентген, МРТ), записи ЭКГ и т. д.	Цифровые изображения, видео, графики.

Из анализа представленной таблицы 1.12 следует, что специалисты преимущественно оперируют данными текстового и числового форматов. Важно отметить, что рамки данного исследования не охватывают задачи, связанные с обработкой изображений, вследствие чего методы инструментальной визуализации не применяются. Исследование сфокусировано на раннем распознавании сахарного диабета, основное внимание уделено анализу гематологических параметров крови и сбору анамнеза пациентов. Этот подход позволяет акцентировать внимание на ключевых биомаркерах, связанных с ранними стадиями развития сахарного

диабета, обеспечивая тем самым научно обоснованную базу для дальнейших клинических исследований и разработки диагностических протоколов.

На данном этапе диссертационного исследования нужно понять природу данных, для этого нужно оцифровать модель диагностического процесса, с точки зрения работы с данными, для предварительной(ранней) диагностики сахарного диабета.

В связи с чем, прежде всего, необходимо определить, какие именно данные нужно собрать на каждом из-под этапов. Для сахарного диабета исходные данными выступают следующие показатели:

- жалобы пациента, например такие как частое мочеиспускание, жажда, потеря веса без причины, усталость и др.;
- история болезни, например наличие сахарного диабета у ближайших родственников, предыдущие диагнозы гипергликемии и т.д.;
- история жизни, к примеру образ жизни, питание, физическая активность, вредные привычки, условия работы и т.д.;
- лабораторные показатели сахарного диабета (СД), были проанализированы согласно МКБ (международного классификатора заболеваний) и медицинским протоколам;
- данные лабораторные показатели нужны чтобы понимать, как определить проверочную выборку из фрейма данных.

Из медицинского протокола согласно МКБ 11, можно получить данные по СД, которые представлены в таблице 1.13.

Таблица 1.13 - Данные по СД из МКБ 11

Наименование показателей	Тип данных	Диапазон значений
Пол	Строковый	<ul style="list-style-type: none"> • F - female (женский) • M - male (мужской)
Возраст	Вещественный	<ul style="list-style-type: none"> • Возрастные группы: • <40, 40 - 49, • 50 - 59, 60 >
Симптомы	Логический	<ul style="list-style-type: none"> • 1 - присутствует • 0 - отсутствует
Глюкоза	Вещественный	<ul style="list-style-type: none"> • Измерения в моль/л • NORM <5,6 • RISK $\geq 5,6$ и $\leq 6,0$ • preddiabet $\geq 6,1$ и $\leq 6,9$ • SD $\geq 7,0$ • NONE - не определялось
Гликозилированный гемоглобин HbA1c	Вещественный	<ul style="list-style-type: none"> • NORM <5,5% • RISK $\geq 5,5$ и $\leq 5,9$ % • preddiabet $\geq 6,0$ и $\leq 6,4$ % • SD2 $\geq 6,5$% • NONE - не определялось

Продолжение таблицы 1.13

рН Крови	вещественный	<ul style="list-style-type: none"> • NORM <7,3 • Abnorm > 7,3 • None - не определялось
Глюкоза в моче	логический	<ul style="list-style-type: none"> • NORM - Нормально • abnorm - превышено
Наследственность	логический	<ul style="list-style-type: none"> • 1 - присутствует • 0 - отсутствует
Ожирение	логический	<ul style="list-style-type: none"> • 1 - присутствует • 0 - отсутствует
болезнь	строковый	<p>Какая стадия диабета:</p> <ul style="list-style-type: none"> • NORM - норма; • RISK- В зоне риска; • preddiabet - Преддиабет • SD1 - СД1; • SD2 - СД2

Алгоритм метода формирования модели данных для системы поддержки принятия клинических решений (СППКР) для диагностики заболевания, такого как сахарный диабет, на основе данных о госпитализациях и диагностических записях представлен на рисунке 1.2.



Рисунок 1.2 - Алгоритм метода формирования модели данных для СППКР

Исходя из рисунка 1.2, можно сказать, что алгоритм включает в себя все необходимые шаги от подготовки данных до обучения и тестирования модели глубокого обучения (таблица 1.14).

Таблица 1.14 - Этапы анализа данных

Шаг	Процесс	Особенности
Сбор и подготовка данных	Импорт данных	Загрузите данные о госпитализациях и диагностических записях
		Конвертируйте временные метки в соответствующий формат
	Предварительная обработка данных	Очистите данные, удаляя несущественные столбцы, такие как идентификаторы и личные данные пациентов
		Интеграция данных
Анализ и визуализация данных	Исследование данных	Проведите визуализацию ключевых аспектов данных, например, распределение продолжительности пребывания в больнице или типов диагнозов
		Идентифицируйте и удалите выбросы или аномалии в данных
Предобработка данных для моделирования	Нормализация и кодирование	Примените нормализацию к числовым данным для уменьшения разброса значений
		Кодируйте категориальные переменные с помощью техник, таких как one-hot encoding
Построение и обучение модели глубокого обучения с использованием жадного алгоритма	Конфигурация и инициализация модели	Разделите данные на обучающий и тестовый наборы для последующего обучения и оценки модели
		Определите начальную архитектуру нейронной сети, выберите начальные функции активации и оптимизаторы
	GridSearch модели	Инициализируйте базовую модель с минимальным количеством слоёв и нейронов для создания простого прототипа
		Примените сеточного алгоритма (GridSearch) для пошагового улучшения модели. На каждом шаге добавляйте слои или нейроны, изменяйте параметры сети (например, скорость обучения, функции активации), исходя из предыдущих результатов обучения
		После каждого изменения проводите оценку модели на валидационном наборе данных для определения, улучшает ли добавление нового элемента или изменение параметра общую производительность модели
		Продолжайте процесс до тех пор, пока дополнительные изменения не перестанут приносить заметное улучшение в производительности или пока не будет достигнут заданный предел сложности модели

Продолжение таблицы 1.14

	Финальная настройка и валидация модели	После определения оптимальной конфигурации сети, проведите финальное обучение модели на полном обучающем наборе данных
		Используйте кросс-валидацию и другие методы валидации, такие как проверочный набор данных, для окончательной оценки модели и подтверждения её эффективности и устойчивости
Тестирование и оценка модели	Оценка модели	Примените модель к тестовому набору данных для оценки её производительности
		Используйте метрики, такие как точность, recall, precision и F1-score для оценки качества модели
Оптимизация и деплоймент	Тонкая настройка модели	Настройте гиперпараметры модели для улучшения её производительности на основе результатов тестирования
		Проверьте модель на новых данных или в реальных клинических условиях для окончательной валидации
	Развертывание модели	Интегрируйте модель в клиническую СППКР для использования в реальных условиях для поддержки врачей в принятии решений

Данный алгоритм обеспечивает комплексный подход к разработке модели глубокого обучения для предсказания медицинских условий, обеспечивая высокую точность и надёжность в клиническом применении.

1.6 Выводы первому разделу

Несмотря на значительный прогресс, остаются проблемы, такие как повышение точности прогностических моделей, обеспечение их применимости в различных группах населения и интеграция прогностических инструментов в клиническую практику. Наше исследование было сосредоточено на совершенствовании этих моделей с использованием более полных наборов данных, изучении новых биомаркеров для прогнозирования риска и разработке персонализированных стратегий вмешательства на основе индивидуальных профилей риска.

Прогнозирование диабета с помощью передовых методов представляет собой важный шаг вперед в борьбе с этой растущей проблемой общественного здравоохранения, давая надежду на более эффективные стратегии профилактики и, в итоге, снижая бремя диабета на отдельных лиц и системы здравоохранения во всем мире.

Несмотря на то, что современные СППКР предлагают значительные потенциальные преимущества в улучшении диагностики и лечения диабета,

представленные в главе недостатки подчеркнули необходимость тщательного внедрения, постоянной оценки и интеграции с клиническим опытом для обеспечения оптимальных результатов лечения пациентов. В данном исследовании уделялось особое внимание персональным данным пациента в диагностике диабета, что является его отличительной чертой в сравнении с существующими СППР в области диагностики диабета.

В распространенной литературе подчеркивается постепенное внедрение цифровых технологий здравоохранения, при этом стратегические цели сосредоточены на управлении, ресурсном потенциале и безопасности данных. Однако литература показывает пробел в прагматическом анализе больших данных для информирования о предоставлении медицинской помощи. В данном исследовании, была сделана попытка разъяснить методологию, объединив различные модели машинного обучения с методами классификации глубокого обучения для прогнозирования наличия диабета.

Задачей диссертационного исследования являлась бинарная классификация с точки зрения классификационного моделирования. Множественные критерии оценки для выбора моделей были использованы с методами настройки гиперпараметров для создания более надежных алгоритмов классификации. Также стоит отметить, что одной из подзадач было изучение различных методов моделирования для анализа и сравнения производительности пяти методов древовидной ансамблевой классификации и трех алгоритмов классификации на основе глубокого обучения. Для достижения этой цели использовались очистка данных, объединение и слияние таблиц данных для получения желаемых функций или атрибутов пациентов.

Благодаря тщательной предварительной обработке данных, в исследовании также была изучена история болезни пациентов и уровни госпитализации, чтобы повысить прогностические возможности модели. В этом исследовании были использованы характеристики или признаки, доступные из общей истории болезни пациента и информации об уровне госпитализации. Отчеты пациентов использовались для маркировки наличия диабета у пациента, поскольку содержали достоверную информацию о заболеваемости.

2 АЛГОРИТМЫ И МОДЕЛИ ПОДДЕРЖКИ ПРИНЯТИЯ КЛИНИЧЕСКИХ РЕШЕНИЙ

2.1 Концептуальная модель процесса поддержки принятия клинических решений на основе методики EDA

Концептуальная модель процесса поддержки принятия клинических решений на основе методики Exploratory Data Analysis (EDA) данных эндокринологии и диабетологии включает в себя несколько ключевых этапов, которые обеспечивают эффективное принятие решений на основе анализа данных. В этом процессе используются различные методы анализа данных, машинного обучения и визуализации для извлечения полезной информации и прогнозов.

Исследовательский анализ данных представляет собой важнейший этап работы с медицинскими данными, особенно в таких областях, как эндокринология и диабетология. Методика EDA включает несколько ключевых этапов, каждый из которых играет важную роль в подготовке данных для дальнейшего машинного обучения или статистического анализа [137]. Классическая методика реализации EDA включает следующие 7 обобщённых этапов анализа данных:

- импорт и подготовка данных;
- нормализация и стандартизация данных;
- агрегация данных;
- кодирование категориальных данных;
- статистический анализ данных;
- оценка качества моделей и метрики;
- моделирование [138].

В рамках данного исследования основной целью является выявление ключевых закономерностей в клинических данных пациентов с заболеваниями эндокринной системы, в частности, с диагнозом «сахарный диабет». Это позволяет проводить углубленный анализ данных, который способствует оптимизации диагностики, улучшению прогнозов заболеваний и принятию более обоснованных медицинских решений.

Представленный в рамках диссертации, процесс поддержки принятия клинических решений, основанный на анализе медицинских данных, состоит из 7 этапов. На первом этапе осуществляется сбор необходимых медицинских данных из различных источников и их интеграция для дальнейшего анализа. На втором этапе проводится предварительный анализ собранных данных для выявления ключевых паттернов, аномалий и других значимых характеристик. Далее применяется алгоритм андерсемплинга для балансировки набора данных, что необходимо для улучшения качества модели, особенно в случае, когда классы (например, наличие или отсутствие диабета) неравномерно представлены. На четвертом этапе для прогнозирования вероятности развития диабета у пациентов предложено использовать алгоритмы глубоких

нейронных сетей с оптимизированными гиперпараметрами. Далее для улучшения точности прогноза и поддержки врачей в принятии решений будет применяться алгоритм ансамблирования архитектур нейронных сетей, таких как CNN (сверточные нейронные сети), LSTM (долгосрочная память) и RNN (рекуррентные нейронные сети). На завершающих этапах реализуется подготовка визуальных отчетов и представление информации пользователю, чтобы сделать результаты анализа более доступными и понятными. На последнем этапе медицинские работники принимают обоснованные решения на основе предоставленной информации и анализа.

Этапы процесса поддержки принятия клинических решений на основе EDA методики данных эндокринологии и диабетологии представлены на рисунке 2.1.



Рисунок 2.1 - Этапы процесса поддержки принятия клинических решений на основе EDA

Применение EDA в процессе поддержки принятия клинических решений является важным шагом к улучшению качества медицинского обслуживания. Он позволяет выявлять закономерности, оптимизировать лечение и повышать безопасность пациентов, что в конечном итоге приводит к лучшим результатам в области здравоохранения.

2.2 Гибридный алгоритм поддержки клинических решений на основе андерсемплинга и автоматической оптимизация параметров

Гибридный алгоритм поддержки клинических решений на основе андерсемплинга и автоматической оптимизации параметров является важным инструментом для работы с несбалансированными медицинскими данными. В реальной клинической практике данные о пациентах часто бывают несбалансированными. Например, в данном исследовании было больше данных о пациентах, у которых не диагностировано заболевание «сахарный диабет», в сравнение с пациентами, у которых диагностировано данное заболевание.

Андерсэмплинг представляет собой процесс, при котором класс с большим количеством данных (например, пациенты без диабета в нашем случае) сокращается, чтобы его количество стало сопоставимым с количеством наблюдений меньшего класса (пациенты с диабетом). Такая последовательность действий помогает сбалансировать классы и сделать обучение модели более равномерным [139]. Данная несбалансированность данных может приводить к снижению точности моделей машинного обучения, поскольку модели склонны игнорировать редкие события. Для устранения данной проблемы в рамках данного исследования предложен алгоритм, основанный на технологии андерсэмплинга.

Применение андерсэмплинга в задачах поддержки клинических решений за счет уравнивания классов позволяет получить ряд следующих преимуществ:

- позволяет уменьшить количество примеров доминирующего класса, что помогает модели уделять больше внимания меньшинству;
- за счет балансировки классов модель становится менее склонной к переобучению на большинстве данных и начинает лучше улавливать закономерности минорного класса, что особенно важно в задачах, связанных с обнаружением редких событий;
- обучение на меньшем объеме данных требует меньше вычислительных ресурсов и времени;
- предотвращение ситуационных искаженных результатов, когда модель слишком сосредоточена на предсказании доминирующего класса за счет игнорирования важной информации о редком классе.

Обобщённый типовой алгоритм андерсэмплинга содержит следующие этапы:

- сбор и подготовка данных;

- определение количества наблюдений каждого класса;
- выбор количества для андерсэмплинга;
- случайный выбор наблюдений доминирующего класса;
- создание сбалансированного набора данных;
- обучение модели;
- оценка производительности [140].

В рамках данного исследования технология андерсэмплинга используется для задач поддержки принятия клинических решений для повышения эффективности решения проблемы несбалансированных данных, которая является частой в медицинских исследованиях и диагностике. В задачах клинического принятия решений несбалансированные данные могут возникать, когда редкие заболевания или осложнения встречаются значительно реже по сравнению с нормальными состояниями или распространёнными заболеваниями. Например, в медицинских задачах часто встречается ситуация, когда данные о пациентах включают большое количество примеров здоровых пациентов или распространённых заболеваний, тогда как данные о редких патологиях или осложнениях присутствуют в меньшинстве.

Применение технологии андерсэмплинга для СПКР позволит сбалансировать классы и улучшить предсказательную способность моделей машинного обучения. Андерсэмплинг в задачах СПКР может помочь улучшить распознавание редких или критически важных состояний (например, острых заболеваний или осложнений), уменьшая влияние доминирующего класса (здоровые пациенты или менее серьезные заболевания). Балансировка данных сделает модели машинного обучения более чувствительными к случаям, которые встречаются редко, но являются клинически значимыми.

Обобщённый алгоритм адаптации технологии андерсэмплинга для СПКР представлен на рисунке 2.2.

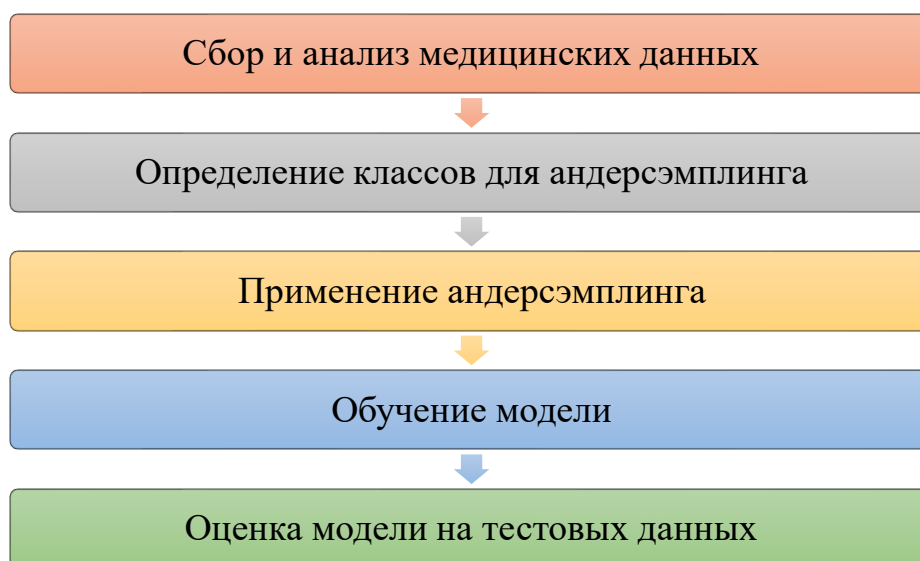


Рисунок 2.2 - Этапы адаптации андерсэмплинга в СПКР

Андерсэмплинг в задачах поддержки принятия клинических решений может быть полезен для балансировки данных, особенно когда речь идет о редких, но клинически важных состояниях. Этот подход помогает избежать склонности моделей к доминирующим классам, что повышает точность предсказаний и улучшает результаты в сложных медицинских ситуациях. Важно тщательно адаптировать технику андерсэмплинга и использовать её в комбинации с другими методами для максимальной эффективности.

Для работы с несбалансированными данными в рамках данного исследования был использован метод Tomek Links - метод удаления лишних данных. Данный метод определяет пары наблюдений, где одно принадлежит классу с избыточным количеством данных (пациенты без диабета), а другое - класс с меньшим количеством данных (пациент с диабетом). Такие пары располагаются близко друг к другу по пространству признаков, и одно из наблюдений в паре удаляется, что позволяет уменьшить количество наблюдений в классе с избыточными данными.

Гибридный алгоритм поддержки клинических решений на основе андерсэмплинга и автоматической оптимизации параметров представлен на рисунке 2.3.

Метод Tomek Links является одним из алгоритмов для очистки данных и борьбы с дисбалансом классов, который можно применять для андерсэмплинга [141]. Основная идея метода заключается в удалении «пограничных» точек из доминирующего класса, которые находятся слишком близко к точкам из минорного класса. Это позволяет уменьшить шум в данных и лучше разделить классы.

В выборе D содержится n примеров, каждый из которых является парой (x_i, y_i) , где $x_i \in \mathbb{R}^d$ - это вектор признаков, а $y_i \in \{0, 1\}$ - метка класса (0 - минорный класс, 1 - доминирующий класс).

Близость точек определяется следующим образом. Для каждого примера x_i из доминирующего класса вычисляется его ближайший сосед x_j (по евклидову расстоянию или другой метрике расстояния), который принадлежит минорному классу (2.1) [141]:

$$dist(x_i, x_j) = \min_{x_k \in D, y_k \neq y_i} \|x_i - x_k\|_2 \quad (2.1)$$

где $\|\cdot\|_2$ - это евклидово расстояние.

Пара точек (x_i, y_i) называется Tomek Link, если выполняются условия (2.2):

$$\begin{cases} \forall x_k \in D: dist(x_i, x_j) < dist(x_i, x_k) \\ dist(x_i, x_j) < dist(x_i, x_k) \end{cases}, \quad (2.2)$$



Рисунок 2.3 - Гибридный алгоритм поддержки клинических решений на основе андерсэмплинга и автоматической оптимизации параметров

То есть расстояние между (x_i, y_i) должно быть минимальным для обеих точек. Если пара (x_i, y_i) образует Tomek Link, и точка x_i принадлежит доминирующему классу, то она считается «пограничной» и может быть удалена. Это позволяет удалить шумовые точки или точки, которые сложно классифицировать, тем самым улучшая разделимость классов [142].

После применения метода Tomek Links выборка D становится чище, и точки, которые мешали разделению классов, удаляются. Очищенные данные используются для дальнейшего обучения модели, и классы становятся лучше разделимыми. Это может уменьшить дисбаланс классов и повысить качество модели.

Таким образом, андерсэмплинг в задачах поддержки принятия клинических решений может быть полезен для балансировки данных, особенно когда речь идет о редких, но клинически важных состояниях. Этот подход помогает избежать склонности моделей к доминирующим классам, что повышает точность предсказаний и улучшает результаты в сложных медицинских ситуациях. Важно тщательно адаптировать технику андерсэмплинга и использовать её в комбинации с другими методами для максимальной эффективности.

Нормализация в данном исследовании была проведена с целью сохранить исходное распределение. Использовался метод нормализации Min-Max Scaling, формула 2.3

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.3)$$

Для оценки качества моделей машинного обучения, используемых для предсказания заболеваний, таких как диабет, применяются различные метрики. Основные метрики включают в себя точность (accuracy), полноту (recall), F1-мера и AUC-ROC. Данные метрики позволяют объективно оценивать качество работы моделей и определить, насколько эффективно они предсказывают заболевания. Ниже перечислены метрики и используемые для их вычислений формулы:

1. Точность (Accuracy)

Accuracy показывает долю правильных ответов алгоритма. Данная метрика бесполезна в задачах, где классы не равномерны. Точность определяет общую корректность модели и рассчитывается как сумма истинных предсказаний над всеми предсказаниями, представлена в формуле 2.4:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

где TP - количество истинных положительных предсказаний, TN - количество истинных отрицательных предсказаний, FP - количество ложноположительных предсказаний, и FN - количество ложноотрицательных предсказаний.

2. Точность предсказания (Precision)

Precision показывает, как часто модель правильно предсказывает положительный класс, из всех предсказанных положительных примеров. В медицинских задачах важно минимизировать FP, чтобы не выдавать ложные диагнозы 2.5:

$$Precision = \frac{TP}{TP + FP} \quad (2.5)$$

3. Полнота (recall)

Полнота показывает, какая доля истинных положительных случаев была корректно предсказана моделью, представлена в формуле 2.6:

$$Recall = \frac{TP}{TP + FN} \quad (2.6)$$

4. F1-мера

F1-мера объединяет точность и полноту в одну метрику, что позволяет сбалансировано оценивать модели, особенно при работе с несбалансированными данными. Данная метрика представлена в формуле 2.7:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.7)$$

Эти метрики вычисляются на основе матрицы смешения (confusion matrix), которая представляет собой таблицу, показывающую, сколько правильных и неправильных предсказаний сделала модель. Матрица ошибок разделена на четыре квадранта и представлена на рисунке 2.4:

- Истинные положительные результаты (TP - true positive): экземпляры, правильно предсказанные как положительные,
- Истинные негативы (TN - true negative): Экземпляры, правильно предсказанные как отрицательные,
- Ложноположительные результаты (FP - false positives): случаи, которые были неверно предсказаны как положительные, часто называемые ошибкой типа I,
- Ложные отрицательные результаты (FN - false negatives): Экземпляры, неверно предсказанные как отрицательные, часто называемые ошибкой II типа.

		Predicted class	
		Class = Yes	Class = No
Actual Class	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Рисунок 2.4 - Матрица ошибок или таблица непредвиденных обстоятельств

AUC-ROC - это одна из наиболее часто используемых метрик для оценки качества моделей классификации. Она помогает оценить, насколько хорошо модель различает классы.

Определения:

ROC-кривая (Receiver Operating Characteristic) - это график, который строится по результатам бинарной классификации. По оси X откладывается доля ложноположительных срабатываний (False Positive Rate, FPR), а по оси Y - доля истинноположительных срабатываний (True Positive Rate, TPR).

AUC (Area Under Curve) - это площадь под ROC-кривой. Она измеряет способность модели отделять один класс от другого. Чем ближе значение AUC к 1, тем лучше модель справляется с классификацией.

2.3 Алгоритм применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN

Переход к использованию технологий глубокого обучения для прогнозирования диабета является логичным следствием необходимости повышения точности и надежности прогностических моделей в медицинских приложениях. После успешной подготовки и анализа данных, включая балансировку и предобработку данных с использованием традиционных методов машинного обучения, следующий шаг включает развертывание более сложных и мощных моделей глубокого обучения. Этот переход обусловлен несколькими ключевыми факторами, улучшающими моделирование сложных медицинских данных:

1) Глубокое обучение способно автоматически определять релевантные признаки из данных, что существенно упрощает процесс подготовки данных и повышает потенциал обнаружения новых, ранее не учтенных взаимосвязей в данных. То есть алгоритмы глубокого обучения способствуют автоматическому извлечению признаков [143].

2) Медицинские данные часто содержат неструктурированную информацию, такую как медицинские изображения, текстовые записи и временные ряды. Глубокое обучение предоставляет инструменты, такие как сверточные и рекуррентные нейронные сети, для эффективной работы с такими типами данных [144].

3) Нейронные сети могут моделировать более сложные зависимости между признаками и классами, что часто приводит к повышению точности и обобщающей способности моделей по сравнению с традиционными методами [145].

4) Глубокое обучение активно развивается и предлагает множество новых методологий и архитектур, которые могут быть адаптированы и оптимизированы для специфических задач диагностики и прогнозирования в медицине [146].

Методы глубокого обучения считаются одним из новейших и высокоразвитых решений для различных контролируемых и неконтролируемых проблем обучения и являются одной из самых динамичных областей исследований и приложений этого столетия. Основным недостатком использования машинного обучения решается методами глубокого обучения или искусственными нейронными сетями. Проблема поиска наиболее подходящей функции обучения и передачи с растущим числом признаков и классифицированных наборов является проблемой и в глубоком обучении. Но поскольку реальные проблемы представлены большими объемами и многомерными наборами данных, глубокое обучение играет важную роль в этой области. Алгоритм обратного распространения обучает нейронную сеть, а метод градиентного спуска (GDM) используется для уменьшения среднеквадратической ошибки или категориальной перекрестной

энтропии в случае классификации между выходом сети и фактической частотой ошибок [147–149].

Но с внедрением и недавними разработками в соответствующей комбинации различных методов с точки зрения обучения, построения архитектуры и функций активации, глубокие нейронные сети стали наиболее успешным алгоритмом, поддерживаемым алгоритмом обратного распространения почти в каждой области [150–153].

Три основные различные архитектуры глубоких нейронных сетей, которые были реализованы в данном исследовании:

- сверточные нейронные сети - CNN (Convolutional Neural Network);
- рекуррентная нейросеть с долгой краткосрочной памятью - LSTM (Long Short-Term Memory);
- рекуррентные нейронные сети - RNN (Recurrent Neural Network).

CNN способны автоматически выбирать признаки из входных данных. Модель сверточной нейронной сети может помочь в разработке современных моделей для классификации аудио. Сверточные слои CNN действуют как скользящее окно над входной матрицей, которое на каждом шаге скольжения, обычно называемом шагом, сверточный слой уменьшает подматрицу в окне до одного выходного значения [154]. Это преобразование выполняется на каждом шаге, и относительные положения выходных значений сохраняются. Поэтому в конце входная матрица преобразуется в меньшую выходную матрицу.

В данном исследовании предложен алгоритм применения метода Grid Search для задач поддержки принятия клинических решений на основе модели классификации CNN, который включает несколько этапов, представленных на рисунке 2.5.

После загрузки и балансировка данных *на втором этапе* производится кодирования категориальных данных в числовом формат. Для этого в данном алгоритме применяется метод One-Hot Encoding [155]. В медицинских данных часто встречаются категориальные признаки, такие как диагнозы, симптомы, пол пациента и т. д. Метод One-Hot Encoding (ONE) позволяет преобразовать эти категориальные данные в числовой формат, который может быть использован в моделях машинного обучения. Каждый уникальный класс представляется отдельным бинарным признаком, что устраняет проблемы с порядком или иерархией данных.



Рисунок 2.5 - Алгоритм применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN

Пусть C - это набор категорий для некоторой категориальной переменной. Например, пусть $C = \{c_1, c_2, \dots, c_k\}$ представляет собой k уникальных категорий.

Для каждой категории c_i будет создан бинарный вектор длины k (формула 2.8) [13]:

$$\text{ONE}(c_i) = \begin{cases} 1, & \text{если } c = c_i \\ 0, & \text{если } c \neq c_i \end{cases} \quad (2.8)$$

где c - значение кодируемой переменной.

Применение One-Hot Encoding в медицинских данных является важным шагом для повышения качества и точности моделей машинного обучения. Этот метод помогает преобразовать категориальные данные в формат, подходящий для анализа, и обеспечивает лучшую интерпретируемость результатов.

На третьем этапе алгоритма реализуется построение модели CNN на основе архитектуры Sequential. Данная архитектура является предназначена для последовательных нейронных сетей, где каждый слой передает данные в следующий слой без сложных разветвлений [156,157]. Это идеальный вариант для простых архитектур и быстрого прототипирования моделей.

На рисунке 2.6 ниже показан основной механизм функционирования сверточных нейронных сетей [158].

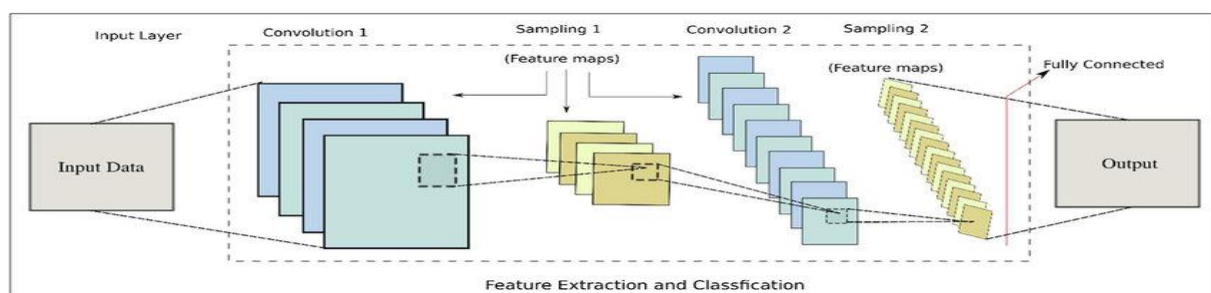


Рисунок 2.6 - Механизм функционирования сверточных нейронных сетей

Математическая модель сверточной нейронной сети может быть представлена следующим уравнением (2.9) [159]:

$$n^{[l]} = \left[\frac{n^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right] n_c^{[l-1]} \quad (2.9)$$

где $n^{[l]}$ - глубина (количество каналов) выходного объема в слое l ; $n^{[l-1]}$ - глубина входного объема в предыдущем слое (слое $l-1$); $p^{[l]}$ - заполнение, добавленное к входному объему в слое l ; $f^{[l]}$ - размер рецептивного поля (фильтра/ядра) в слое l ; $s^{[l]}$ - шаг, используемый в слое l для операции свёртки

; $n_c^{[l-1]}$ - высота или ширина входного объема в слое l-1 (при условии квадратного ввода).

Данная модель (2.9) описывает размеры последнего слоя l с размером фильтра f, отступом p, шагом s и используется для вычисления глубины выходного объема в текущем слое на основе размеров входного объема, отступа, размера фильтра, шага и глубины входа.

Уравнение (2.10) является математическим описанием значения шага, с которым ядро перемещается во время свертки:

$$stride = s^{[l]} \quad (2.10)$$

Количество весов рассчитывается путем умножения размера фильтра на глубину входа и глубину выхода (2.11):

$$weights = f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]} \quad (2.11)$$

где $f^{[l]}$ - размер рецептивного поля (фильтра/ядра) в слое l; $n_c^{[l-1]}$ - глубина входного объема в слое l-1; $n_c^{[l]}$ - глубина выходного объема в слое l.

Эти веса являются обучаемыми параметрами, которые CNN корректирует во время обучения.

Математическая модель (2.12) используется для вычисления размера выходного тома определяется высотой, шириной и глубиной выходного тома в текущем слое:

$$output = n_h^{[l]} \times n_w^{[l]} \times n_c^{[l]} \quad (2.12)$$

где $n_h^{[l]}$ - высота выходного объема в слое l; $n_w^{[l]}$ - ширина выходного объема в слое l; $n_c^{[l]}$ - глубина выходного объема в слое l.

Для вычисления параметров смещения вводятся в каждый канал выходного объема, обеспечивая дополнительный обучаемый параметр, который может помочь CNN фиксировать различные закономерности и взаимосвязи в данных, используется следующая модель (2.13):

$$bias = 1 \times 1 \times 1 \times n_c^{[l]} \quad (2.13)$$

где $n_c^{[l]}$ - глубина выходного объема в слое l.

Архитектура Sequential в нейронных сетях представляет собой последовательную композицию слоев, через которые данные последовательно проходят, начиная от входного слоя до выходного. Математически это можно

выразить как составление нескольких функций, каждая из которых соответствует одному слою нейронной сети [160].

Предположим, что архитектура состоит из L слоев. Каждый слой l применяет некоторую нелинейную или линейную операцию к входным данным, и выход слоя l используется как вход для слоя $l+1$. Последовательно такие слои можно описать с помощью функций f_l , где l - номер слоя.

Пусть входные данные x_0 - входные признаки или векторы для модели.

Каждый слой l представлен функцией f_l , которая может включать линейную операцию (например, матричное умножение с весами W_l) и нелинейную активацию (например, функцию ReLU, сигмоидальную или softmax-активацию) (2.14):

$$x_l = f_l(x_{l-1}) = \sigma(W_l x_{l-1} + b_l) \quad (2.14)$$

где: W_l - матрица весов слоя l , b_l - вектор смещений (bias), σ - функция активации (например, ReLU, сигмоида и т.д.), x_{l-1} - выход с предыдущего слоя.

Последний слой f_L выдаёт окончательный выход модели (2.15):

$$\hat{y} = f_L(f_{L-1}(\dots f_1(x_0) \dots)) \quad (2.15)$$

Здесь последовательность применений функций f_1, f_2, \dots, f_L соответствует последовательности слоев модели.

Таким образом, Sequential модель формализуется как последовательное применение нескольких линейных и нелинейных функций, которые преобразуют входные данные к выходным [161].

При построении модели CNN для предотвращения переобучения применяется метод регуляризации Dropout. Данный метод регуляризации заключается в случайном «выключении» (занулении) некоторого процента нейронов на каждом шаге обучения, что делает сеть более устойчивой к переобучению и повышает её способность к обобщению на новых данных.

Пусть x это входной вектор для слоя нейронной сети, а W - матрица весов этого слоя. При обычном прямом распространении (без Dropout) выход слоя можно описать как (2.16) [162]:

$$y = Wx + b \quad (2.16)$$

где W - матрица весов, b - вектор смещений, x - входной вектор.

С применением Dropout на этапе обучения происходит случайное отключение части нейронов. Пусть p - вероятность того, что нейрон сохраняется (не обнуляется). Тогда Dropout вводится с помощью маски m , состоящей из случайных значений, которые принимают значения 1 (нейрон активен) с вероятностью p и 0 (нейрон "выключен") с вероятностью $1-p$.

Математически маску можно записать как (2.17):

$$m_i \sim \text{Bernoulli}(p) \quad (2.17)$$

где m_i — это элемент маски m , который принимает значение 1 с вероятностью p и 0 с вероятностью $1-p$. Для каждого входного нейрона x_i создается такая маска.

Во время обучения выход слоя с применением Dropout можно описать следующим образом (2.18):

$$\tilde{x} = m \odot x \quad (2.18)$$

m - маска Dropout, \odot - поэлементное умножение (Hadamard product), x - входной вектор.

Это означает, что некоторые элементы вектора x умножаются на 0 (отключены), а остальные остаются без изменений. После применения Dropout к входным данным выход слоя, будет следующим (2.19):

$$y = W\tilde{x} + b = W(m \odot x) + b \quad (2.19)$$

Вектор x модифицируется маской m , которая случайным образом обнуляет его элементы.

Таким образом, метод регуляризации Dropout помогает уменьшить взаимозависимость нейронов и тем самым снижает риск переобучения, обеспечивая более устойчивое обучение модели.

На шестом этапе алгоритма осуществляется подбор оптимальных гиперпараметров на основе метода Grid Search. Grid Search является одним из методов подбора оптимальных гиперпараметров для машинного обучения, позволяющему перебрать все возможные комбинации гиперпараметров из заданного диапазона значений и выбрать те, которые дают наилучшие результаты на основе некоторой метрики качества [163].

Пусть имеется набор гиперпараметров, каждый из которых принимает различные значения (2.20):

$$\begin{aligned} H_1 &\in \{h_{1,1}, h_{1,2}, \dots, h_{1,m_1}\} \\ H_2 &\in \{h_{2,1}, h_{2,2}, \dots, h_{2,m_2}\} \\ &\dots \\ H_k &\in \{h_{k,1}, h_{k,2}, \dots, h_{k,m_k}\} \end{aligned} \quad (2.20)$$

где: H_1, H_2, \dots, H_k - это гиперпараметры, $h_{i,j}$ - это возможные значения гиперпараметра H_i , m_i - количество значений гиперпараметра H_i .

В итоге, задача Grid Search заключается в том, чтобы перебрать все комбинации (2.21):

$$\{(h_{1,i_1}, h_{2,i_2}, \dots, h_{k,i_k}) \mid i_1 \in [1, m_1], i_2 \in [1, m_2], \dots, i_k \in [1, m_k]\} \quad (2.21)$$

Таким образом, для каждой комбинации можно оценить модель с помощью метода перекрестной проверки или на валидационном наборе данных.

При обучении нейронной сети очень важно правильно инициализировать веса, чтобы градиенты во время обучения распространялись эффективно. Плохая инициализация может привести к проблемам, таким как затухание или взрыв градиентов. Эти проблемы затрудняют обучение, особенно в глубоких сетях. На этом же этапе для повышения эффективности CNN модели применяется инициализатор весов `he_normal` [164]. Инициализатор весов `he_normal` является методом инициализации весов, который был предложен как эффективный инструмент при работе с нейронными сетями, использующими нелинейные функции активации, такие как ReLU и её вариации (например, Leaky ReLU).

Схемы работы RELU и ELU представлены на рисунках 2.7 и 2.8 соответственно.

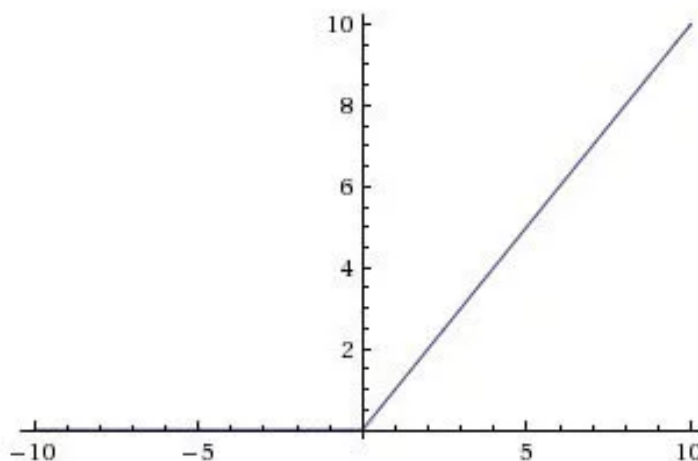


Рисунок 2.7 - Схема работы RELU

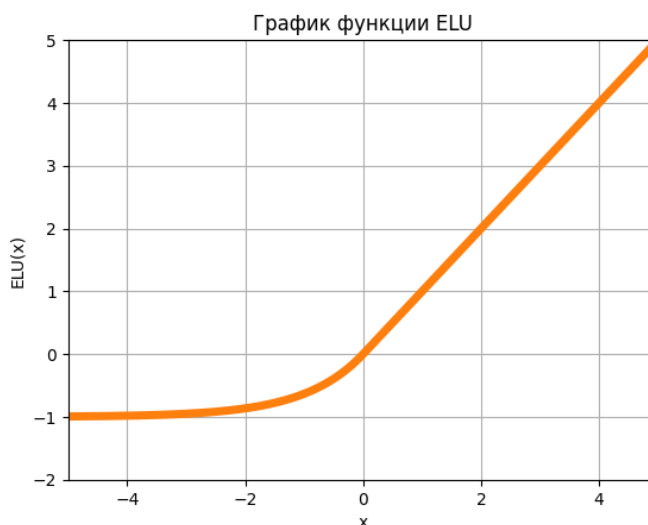


Рисунок 2.8 - Схема работы ELU

Пользуясь определением, становится понятно, что ReLu(Rectified linear unit) возвращает значение x , если x положительно, и 0 в противном случае. Схема работы приведена на слайде

ELU (Exponential Linear Unit) - это функция активации. Она представляет собой измененную версию ReLU (Rectified Linear Unit), которая помогает ускорить обучение глубоких нейронных сетей и справляется с проблемой "мертвых нейронов" (dead neurons). ELU работает так же, как и ReLU, возвращая исходное значение входа, если он больше нуля. Однако, если значение входа меньше или равно нулю, то ELU использует экспоненциальную функцию, чтобы получить значение, которое ближе к нулю, чем значение, возвращаемое ReLU. Это позволяет избежать "мертвых нейронов" и ускорить обучение глубоких нейронных сетей.

Кроме того, ELU имеет свойство гладкости, которое так же помогает избежать проблемы "взрывающегося градиента" (exploding gradient), которая может возникать при использовании других функций активации, таких как ReLU. Это делает ELU более стабильной и более эффективной функцией активации для обучения глубоких нейронных сетей, представлено в формуле 2.22:

$$ELU(x) = \begin{cases} x, & \text{если } x > 0 \\ \alpha(e^x - 1), & \text{если } x \leq 0 \end{cases} \quad (2.22)$$

где α — это параметр, который может быть установлен в значение 1 по умолчанию.

Инициализатор `he_normal` помогает избежать этих проблем, обеспечивая лучшую инициализацию для функций активации ReLU, которые зануляют отрицательные значения [164]. Это достигается за счет более подходящего масштабирования начальных весов в зависимости от числа входов в нейрон.

Инициализация весов `he_normal` использует нормальное распределение (гауссовское распределение) с нулевым средним и дисперсией, зависящей от количества входов в нейрон. Формула для распределения выглядит следующим образом (2.23) [165]:

$$W \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_{in}}}\right) \quad (2.23)$$

где:

- W - весовые коэффициенты нейронной сети;
- \sim - обозначение того, что веса W следуют какому-то распределению (в данном случае нормальному распределению);

– $N(\mu, \sigma)$ - это нормальное распределение с параметрами: $\mu=0$ - среднее значение распределения (веса центрированы вокруг нуля); $\sqrt{\frac{2}{n_{in}}}$ - стандартное отклонение распределения. Это значение зависит от количества входов в нейрон.

– n_{in} - количество входов (входных нейронов) в данный нейрон.

Модель (2.23) показывает, что каждый вес в сети выбирается случайно из нормального распределения с нулевым средним значением и стандартным отклонением, равным $\sqrt{\frac{2}{n_{in}}}$.

Таким образом, использование инициализатора `he_normal` для весов — это также улучшение, поскольку он хорошо согласуется с активационной функцией ELU. Этот метод инициализации, разработанный для учета размера предыдущего слоя, может способствовать более эффективному обучению.

На последнем этапе, при неудовлетворительном уровне точности улучшенной модели, предусмотрены процедуры регуляризации и оптимизации модели. Для этого используется метод `ReduceLROnPlateau`, позволяющий автоматически уменьшает скорость обучения, если качество модели (например, её ошибка или точность) перестаёт улучшаться в течение определённого количества эпох. Данный метод встроен в библиотеки Keras и PyTorch. Таким образом, в зависимости от поведения ошибки (метрики), метод `ReduceLROnPlateau` динамически уменьшает скорость обучения, чтобы помочь сети лучше адаптироваться в случае застревания в локальном минимуме.

Для оценки модели вычисляются следующие показатели: Loss, Accuracy, Precision, Recall и F1-score. Для проверки модели используется матрица ошибок, которая визуализирует, где модель ошибается.

2.4 Алгоритм ансамблирования архитектур нейронных сетей для задач поддержки клинических решений

Алгоритм ансамблирования архитектур нейронных сетей для задач поддержки клинических решений - это подход, при котором несколько различных нейронных сетей объединяются для повышения точности, устойчивости и надежности системы [166–168]. Ансамблирование позволяет компенсировать ошибки отдельных моделей и улучшать общие результаты за счет комбинирования их предсказаний. В задачах поддержки клинических решений это особенно важно, поскольку малейшие ошибки в диагнозе или прогнозе могут привести к серьезным последствиям.

В системе поддержки принятия клинических решений по прогнозированию заболевания диабета будет использован ансамбль из нескольких архитектур:

- сверточные нейронные сети (CNN);
- рекуррентные нейронные сети (RNN);

- долгая краткосрочная память (LSTM).

Объединение предсказаний моделей CNN, RNN, LSTM повысит точность прогнозирования и поможет более точно предсказывать развитие диабета у пациента.

В рамках диссертационного исследования для координации работы ансамбля применяется класс DirichletEnsemble, который использует метод распределение Дирихле для определения оптимальных весов каждой модели, учитывая их индивидуальные показатели производительности [169].

Распределение Дирихле является многомерным обобщением бета-распределения. Этот метод моделирует вероятности для нескольких категорий, например, для классов в задаче классификации. Таким образом распределение Дирихле описывает распределение вероятностей, которое может быть использовано для моделирования вероятностей K -классов.

Обозначим вероятность для каждого из K классов через p_1, p_2, \dots, p_K , где

$$p_1 + p_2 + \dots + p_K = 1$$

Распределение Дирихле для K -мерного вектора вероятностей задается следующим образом (2.24) [170]:

$$P(p_1, p_2, \dots, p_K | \alpha_1, \alpha_2, \dots, \alpha_K) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K p_i^{\alpha_i - 1} \quad (2.24)$$

где $\alpha_1, \alpha_2, \dots, \alpha_K$ - то гиперпараметры, Γ - гамма-функция, которая является обобщением факториала.

Пусть имеется N моделей, каждая из которых предсказывает вероятности принадлежности объекта к классам p_1, p_2, \dots, p_K для i -модели, где $i = 1, 2, \dots, N$. Применяя метод Dirichlet Ensemble можно комбинировать эти предсказания, используя апостериорное распределение, которое моделируется как распределение Дирихле [171].

Для агрегирования предсказаний каждой модели используем апостериорное распределение Дирихле. Вероятности для каждого класса j в финальном предсказании можно записать как модель (2.25) [172]:

$$p_j = \frac{\sum_{i=1}^N \alpha_i * p_{ij}}{\sum_{i=1}^N \alpha_i} \quad (2.25)$$

где: p_{ij} - предсказанная вероятность класса j -й моделью i , α_i - вес (или гиперпараметр), который регулирует вклад модели i в итоговое предсказание.

Гиперпараметры $\alpha_1, \alpha_2, \dots, \alpha_K$ могут быть обновлены на основе апостериорного распределения или через методы оптимизации, такие как градиентный спуск. Эти параметры можно интерпретировать как апостериорные вероятности доверия к каждой модели.

Основная идея заключается в том, что распределение Дирихле генерирует распределения вероятностей для классов, тем самым моделируя неопределенность в предсказаниях.

Dirichlet Ensemble может применяться для медицинских данных, где есть неопределенность в предсказаниях различных моделей. Например, при классификации диагнозов по результатам анализов несколько моделей могут давать различные предсказания, и использование метода распределения Дирихле позволит более точно объединить эти предсказания, учитывая неопределенность каждой модели и её вклад.

Этапы алгоритма ансамблирования архитектур нейронных сетей с использованием Dirichlet Ensemble представлены на рисунке 2.9.

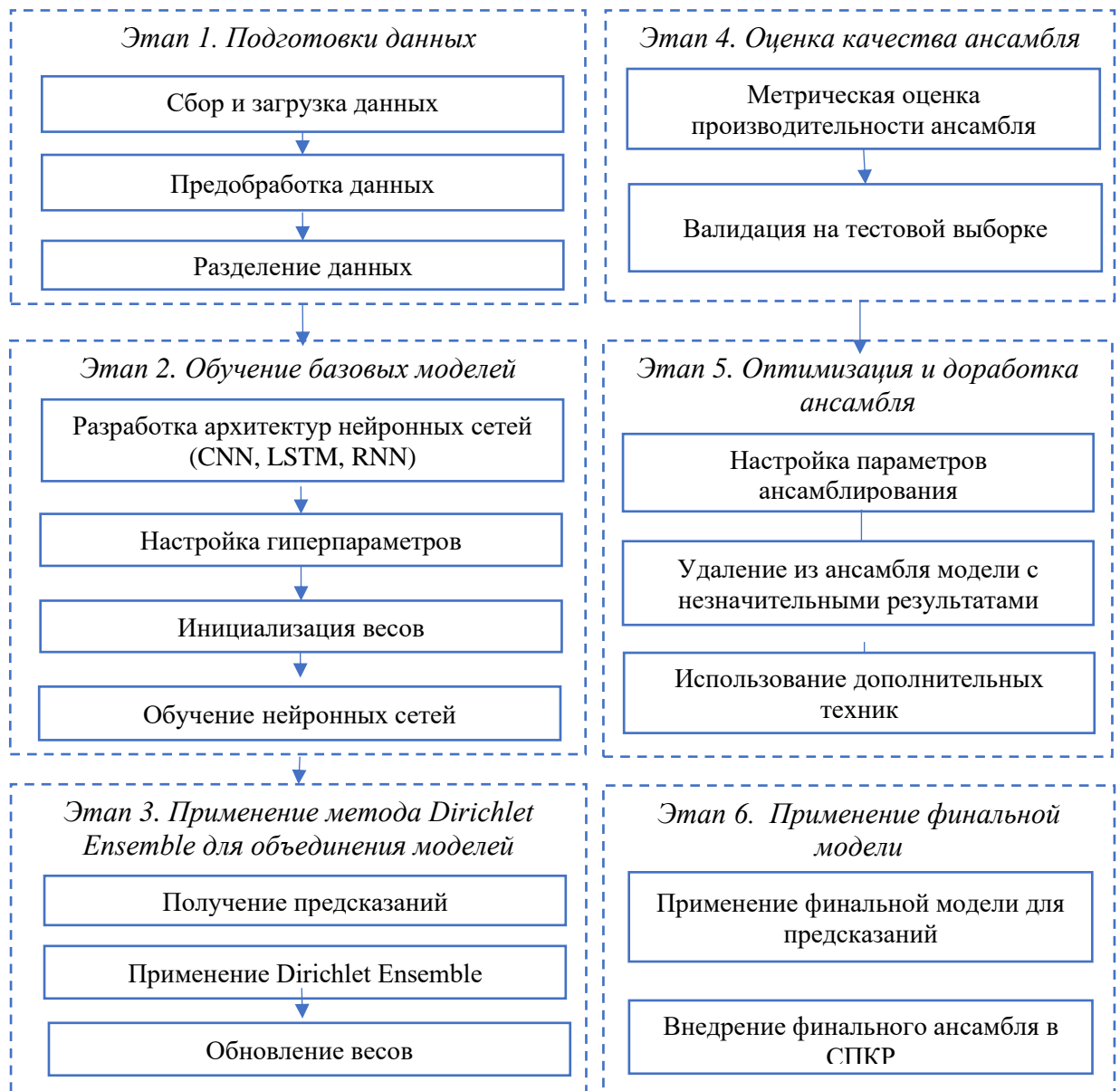


Рисунок 2.9 - Алгоритм ансамблирования архитектур нейронных сетей для задач поддержки клинических решений

В ансамбле классификаторов задач поддержки клинических решений каждая модель предсказывает вероятность принадлежности объекта к тому или иному классу. Dirichlet Ensemble объединяет предсказания таким образом, чтобы учитывать неопределенность, и распределяет вероятности предсказаний между классами на основе апостериорного распределения, построенного на основе распределения Дирихле. Веса, которые присваиваются предсказаниям каждой модели, основаны на распределении Дирихле, что позволяет более гибко и эффективно агрегировать результаты.

Можно выделить следующие преимущества ансамблирования для клинических решений:

- ансамблирование позволяет уменьшить ошибку предсказания, поскольку слабые стороны одной модели могут быть компенсированы другой моделью.

- модели, обученные на различных подвыборках данных или с использованием разных архитектур, могут быть более устойчивы к шумам в медицинских данных.

- ансамблирование улучшает общую надежность предсказаний, что критически важно для принятия клинических решений, особенно в случаях жизнеугрожающих состояний.

Этот подход делает систему более гибкой и устойчивой, что важно в условиях высокой неопределенности и неоднородности медицинских данных.

2.5 Выводы по второму разделу

В результате тщательного изучения процесса диагностирования и постановки диагноза была сформулирована концептуальная модель процесса поддержки принятия клинических решений.

Предложенная концептуальная модель процесса поддержки принятия клинических решений основана на методике Exploratory Data Analysis (EDA) данных эндокринологии и диабетологии. Данная модель включает в себя несколько следующих ключевых этапов, каждый из которых играет важную роль в подготовке данных для дальнейшего машинного обучения или статистического анализа: сбор и интеграция медицинских данных; исследовательский анализ данных; балансировка набора данных эндокринологии и диабетологии на основе алгоритма андерсэмплинга; прогнозирование диабета на основе алгоритма глубокой нейронной сети с оптимизированными гиперпараметрами; поддержка ПКР эндокринологии и диабетологии на основе алгоритма ансамблирования архитектур нейронных сетей CNN, LSTM, RNN; визуализация и формирование отчетов; предоставление информации пользователю; принятие решения ЛПР.

Был предложен гибридный алгоритм поддержки клинических решений на основе андерсэмплинга и автоматической оптимизации параметров, который является важным инструментом для работы с несбалансированными медицинскими данными. Применение технологии андерсэмплинга для СПКР

позволит сбалансировать классы и улучшить предсказательную способность моделей машинного обучения.

Был разработан алгоритм применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN. Данный алгоритм состоит из 8 этапов и интегрирует следующие методы и модели:

- метод One-Hot Encoding для конкатенации числовых и категориальных признаков;
- модель Sequential для упрощенного добавления слоев нейронной сети;
- метод регуляризации Dropout для предотвращения переобучения;
- метода Grid Search для подбора оптимальных гиперпараметров нейронной сети;
- метод he_normal для применения инициализатора весов;
- метод ReduceLROnPlateau для регулировки скорости обучения.

Предложен алгоритм ансамблирования архитектур нейронных сетей для задач поддержки клинических решений при котором несколько различных нейронных сетей объединяются для повышения точности, устойчивости и надежности системы. Разработанный алгоритм ансамблирования позволяет компенсировать ошибки отдельных моделей и улучшать общие результаты за счет комбинирования их предсказаний. Для координации работы ансамбля применялся класс DirichletEnsemble, который использует метод распределение Дирихле для определения оптимальных весов каждой модели, учитывая их индивидуальные показатели производительности.

В результате применения методов исследовательского анализа данных, нормализации и использования алгоритмов машинного обучения можно значительно улучшить процесс диагностики и прогнозирования заболеваний в эндокринологии и диабетологии. Применение различных метрик для оценки качества моделей позволяет объективно оценить их эффективность и выбрать наиболее подходящие для клинической практики.

3 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ И АРХИТЕКТУРНАЯ МОДЕЛЬ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ КЛИНИЧЕСКИХ РЕШЕНИЙ

3.1. Информационная модель интеллектуальной системы поддержки принятия клинических решений

Информационная модель интеллектуальной системы поддержки принятия клинических решений (СППР) играет ключевую роль в организации и управлении данными, процессах принятия решений и интеграции знаний, необходимых для диагностики и лечения пациентов.

Модель позволяет структурировать медицинские данные:

- информационная модель позволяет обеспечить систематизацию и структуризацию данных, поступающие из разных источников;
- интегрировать клинических протокола и знания;
- автоматизировать аналитические процессы;
- обеспечить поддержку принятия клинических решений и персонализацию лечения.

То есть информационная модель интеллектуальной СППР способствует улучшению качества медицинских решений, повышению эффективности лечения и снижению рисков для здоровья пациентов.

На начальном этапе диссертационного исследования производилась оценка данных, полученных из открытых источников. Однако в рамках данного исследования все данные были извлечены из открытой базы данных MIMIC-III (Medical Information Mart for Intensive care III) [173].

Для информационного обеспечения и реализации предложенных в рамках диссертации алгоритмов поддержки принятия клинических решений была проектирована реляционная модель базы данных, которая является фрагментом MIMIC-III и состоящая из 26 таблиц: patients, admissions, diagnoses_icd, d_icd_diagnoses, prescriptions, transfers, callout, cptevents, drgcodes, icustays, inpatientevents_cv, inpatientevents_mv, procedureevents_mv, caregivers, labevents, d_labitems, services, outpatientevents, chartevents, d_items, datatimeevents, noteevents, microbiologyevents, procedures_icd, d_icd_procedures.

Таблицы содержат данные о пациентах, которые находились в отделениях интенсивной терапии медицинского центра Beth Israel Deaconess Medical Center [174].

Таблицы связаны идентификаторами, которые обычно имеют суффикс «ID». Например, SUBJECT_ID относится к уникальному пациенту, HADM_ID относится к уникальному случаю поступления в больницу, а ICUSTAY_ID относится к уникальному случаю поступления в отделение интенсивной терапии.

Описания данных представлены в таблицах 3.1 - 3.4.

Таблица 3.1 - Метаданные БД MIMIC-III, хранящие информацию о пациентах

№	Объекты	Описание
1	ADMISSIONS	Каждая уникальная госпитализация для каждого пациента в базе данных (определяет HADM_ID)
2	CALLOUT	Информация о том, когда пациент был пройден этап выписки из ОИТ и когда пациент был фактически выписан
3	ICUSTAYS	Каждое уникальное ОИТ пребывание в базе данных (определяет ICUSTAY_ID)
4	PATIENTS	Каждый уникальный пациент в базе данных (определяет SUBJECT_ID)
5	SERVICES	Клиническая служба, в рамках которой зарегистрирован пациент
6	TRANSFERS	Движение пациента от кровати к кровати в пределах больницы, включая входы и выходы ОИТ

Зарегистрированные события, такие как заметки, лабораторные испытания и баланс жидкости, хранятся в серии таблиц событий. Например, таблица OUTPUTEVENTS содержит все измерения, связанные с выходными данными для данного пациента, в то время как таблица LABEVENTS содержит результаты лабораторных испытаний пациента.

Таблица 3.2 - Метаданные БД MIMIC-III

№	Название таблицы	Данные в таблице
1	CAREGIVERS	каждый обслуживающий персонал, который записал данные в базу данных (определяет CGID)
2	CHARTEVENTS	все графические наблюдения за пациентами
3	DATETIMEEVENTS	все записанные наблюдения, которые являются датами
4	INPUTEVENTS_CV	приемы пациентов, контролируемых с помощью системы Philips CareVue в отделении интенсивной терапии
5	INPUTEVENTS_MV	приемы пациентов, контролируемых с помощью системы Imdsoft Metavision в том числе и в ОИТ
6	NOTEEVENTS	примечания относительно пребывания пациента в отделении интенсивной терапии, включая примечания по уходу, примечания врача, рапорты ЭКГ, отчеты по визуализации и сводки о выписке
7	OUTPUTEVENTS	выходные данные по пациентам в ОИТ
8	PROCEDUREEVENTS_MV	процедуры пациентов для подмножества пациентов, которые наблюдались в ОИТ с использованием системы Imdsoft MetaVision

Таблицы с префиксом «D_» являются словарными таблицами и предоставляют определения для идентификаторов. Например, каждая строка CHARTEVENTS ассоциирована с одним ITEMID, который представляет измеряемую концепцию, но не содержит фактическое название измерения. Присоединив CHARTEVENTS и D_ITEMS по ITEMID, можно идентифицировать концепцию, представленную данным ITEMID.

Каждому ICUSTAY_ID соответствует один HADM_ID и один SUBJECT_ID. Каждый HADM_ID соответствует одному SUBJECT_ID. Один SUBJECT_ID может соответствовать нескольким HADM_ID (несколько госпитализаций одного и того же пациента), и несколько ICUSTAY_ID (несколько ICU остается либо в пределах одной и той же госпитализации, или через несколько госпитализаций, или оба). В таблице 3.2 содержатся данные, собранные в ОИТ.

Следующие таблицы содержат данные, собранные в системе больничных записей (таблица 3.3).

Таблица 3.3 - Метаданные БД MIMIC-III о больничных записях

№	Название таблицы	Данные таблицы
1	CPTEVENTS	Процедуры, записанные как коды текущей процедурной терминологии
2	DIAGNOSES_ICD	Поставленные диагнозы, закодированные с использованием системы Международной статистической классификации заболеваний и связанных с ними проблем со здоровьем (МКБ)
3	DRGCODES	Данные, которые используются больницей для выставления счетов
4	LABEVENTS	Лабораторные измерения для пациентов как больницы, так и амбулаторных учреждений
5	MICROBIOLOGYEVENTS	микробиологические измерения и чувствительность из базы данных больницы
6	PRESCRIPTIONS	Выписанные пациенту лекарства
7	PROCEDURES_ICD	Процедуры пациентов, закодированные с использованием системы Международной статистической классификации болезней и связанных с ними проблем со здоровьем (МКБ)

Справочные БД MIMIC-III представлены в таблице 3.4.

Таблица 3.4 - Справочные метаданные БД MIMIC-III

№	Название таблицы	Описание
1.	D_CPT	Высокоуровневый словарь кодов текущей процедурной терминологии
2.	D_ICD_DIAGNOSES	Словарь Международной статистической классификации болезней и связанных с ними проблем со здоровьем (МКБ), коды связанные с диагнозами
3.	D_ICD_PROCEDURE S	Словарь Международной статистической классификации болезней и связанных с ними проблем со здоровьем (МКБ) коды, касающиеся процедур
4.	D_ITEMS	Словарь ITEMIDs, появляющийся в базе данных MIMIC, за исключением тех, которые относятся к лабораторным испытаниям
5.	D_LABITEMS	Словарь ITEMIDs в базе данных лаборатории, которые относятся к лабораторным испытаниям

Информационная модель интеллектуальной системы поддержки принятия клинических решений представлена на рисунке 3.1.

База данных MIMIC-II содержит также множество производных таблиц, которые упрощают использование базы данных. Но, в то же время, создатели MIMIC-III приняли сознательное решение не включать любые производные таблицы или вычисляемые параметры, насколько это возможно. Вместо этого они рекомендуют сообществу создавать сценарии, которые можно запускать для создания этих таблиц или параметров, и обмениваться ими. Это имеет много преимуществ: он сохраняет различие между необработанными данными и вычисляемыми данными, поощряет пользователей проверять скрипты, которые получают данные, и позволяет использовать столько скриптов, сколько возможно, не загромождая базу данных для всех пользователей.

Дополнительная информация предоставлена в приложение М Таблицы связей БД MIMIC III.

Информационная модель интеллектуальной системы поддержки принятия клинических решений служим главным элементом для реализации анализа данных. После импорта данных из внешних хранилищ производится предобработка данных, целью которой является подготовка данных для дальнейшего анализа. На этом этапе данные очищаются от ошибок, пропущенных значений и аномалий. Также проводится нормализация данных и их преобразование в формат, удобный для анализа. Далее на подготовленных данных реализуется разведочный и статистический анализ данных. Данные подходы позволяет исследовать основные характеристики распределений и возможных взаимосвязей, аномалий. Этот этап также включает визуализацию данных. Данный этап является важной частью подготовки чистых и хорошо структурированных данных, для более сложных статистических и машинно-обучаемых анализов. Статистический анализ, обеспечивает тщательное понимание данных, что крайне важно для последующего формирования и оценки моделей машинного обучения.

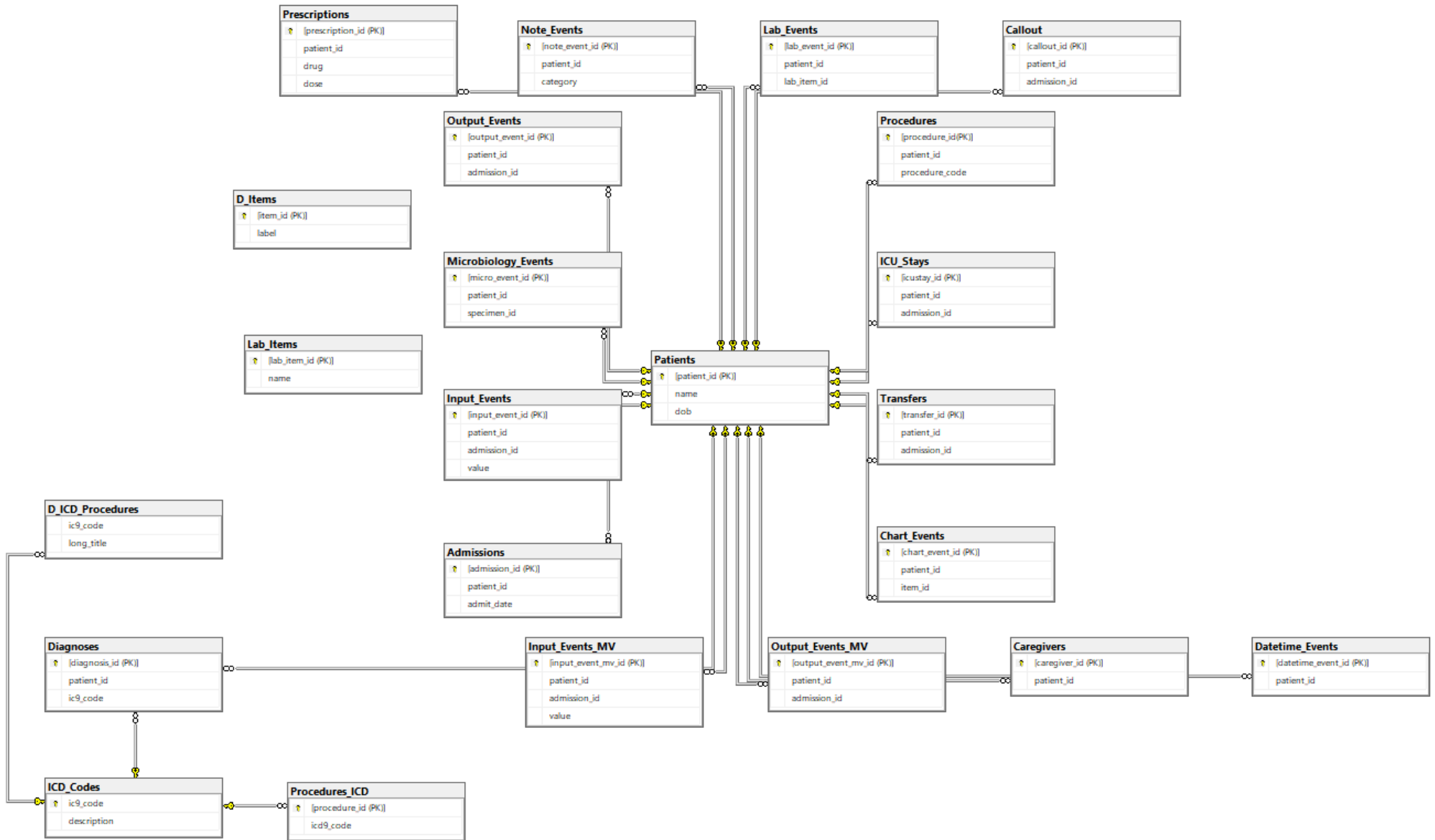


Рисунок 3.1 - Информационная модель интеллектуальной системы поддержки принятия клинических решений

Для предобработки «сырых» данных информационной модели интеллектуальной СППКР используются методы описательной статистики, реализуется анализ распределений и взаимосвязей, а также многомерный анализ данных. Указанные методы необходимо применить для очистки данных и для формирования гипотез.

Для проверки гипотез применяется описательный анализ данных, который включает расчет основных статистических показателей для каждой переменной, таких как среднее значение, медиана, стандартное отклонение, минимум и максимум. Эти метрики помогут получить общее представление о распределении каждого признака, включая лабораторные показатели и количество госпитализаций на каждого пациента. Следующим шагом будет анализ распределения ключевых переменных. Используя гистограммы и ящики с усами, можно визуализировать распределение числа госпитализаций на каждого пациента и основные лабораторные показатели. Это позволит выявить аномалии и потенциальные выбросы в данных.

Для понимания связей между различными переменными целесообразно использовать корреляционный анализ. Корреляционные матрицы могут быть полезны для выявления зависимостей между лабораторными показателями и частотой госпитализаций, а также между типами заболеваний (МКБ коды) и лабораторными результатами.

Для более глубокого анализа взаимосвязей между переменными можно применить многомерные статистические методы, такие как главные компоненты (РСА) для снижения размерности или кластеризация для группировки пациентов с похожими клиническими профилями.

Таким образом, информационная модель системы поддержки принятия клинических решений играет ключевую роль в анализе данных: после импорта данные проходят предобработку, включающую очистку, нормализацию и подготовку для анализа. Дальнейший статистический и разведочный анализ позволяет выявлять основные характеристики данных, взаимосвязи и аномалии для последующего построения и оценки моделей машинного обучения.

3.2 Применение методики EDA на клинических данных эндокринологии и диабетологии

3.2.1 Исследовательский анализ данных клинических данных эндокринологии и диабетологии

EDA анализ будет состоять из нескольких ключевых шагов:

ШАГ 1. Импорт данных и библиотек. Используя Python и библиотеки, такие как pandas и numpy, начинается с загрузки исходных данных, что является основой для всех последующих операций.

ШАГ 2. Предварительный просмотр данных. Первичное изучение данных с помощью методов `head()` или `.describe()` для получения представления о структуре и базовых статистиках.

ШАГ 3. Обработка пропущенных значений. Идентификация и обработка пропущенных значений с помощью методов, таких как `dropna()`, что критично для обеспечения качества анализа.

ШАГ 4. Нормализация и стандартизация данных. Применяется для кодов МКБ, где важно привести все коды к единому стандарту, облегчая агрегацию и анализ.

ШАГ 5. Агрегация данных. Группировка данных по ключевым полям, таким как идентификаторы пациентов и госпитализации, что позволяет анализировать данные на уровне отдельных пациентов или эпизодов лечения.

ШАГ 6. Визуализация. Создание графиков и диаграмм для визуального представления распределений, трендов и возможных аномалий в данных.

Анализ взаимосвязей показал, что анализ и нормализация кодов ICD или МКБ являются ключевыми для понимания распределения и частоты различных заболеваний среди госпитализированных пациентов, а также для выявления взаимосвязей между типами заболеваний и другими клиническими характеристиками пациентов. Нормализация и стандартизация кодов ICD улучшают согласованность данных и упрощают дальнейший анализ связей между диагнозами и исходами лечения, что может влиять на предсказания и решения в контексте машинного обучения.

На рисунке 3.2 представлена взаимосвязи между диагнозом и госпитализацией.

	SUBJECT_ID	HADM_ID	ICD9_CODE
0	2	163353	V30.01;V05.3;V29.0
1	3	145834	038.9;785.59;584.9;427.5;410.71;428.0;682.6;42...
2	4	185777	042.;136.3;799.4;276.3;790.7;571.5;041.11;V09....
3	5	178980	V30.00;V05.3;V29.0
4	6	107064	403.91;444.0;997.2;276.6;276.7;285.9;275.3;V15.82

Рисунок 3.2 - Результаты анализа взаимосвязи между диагнозом и госпитализацией

Далее требовалось провести анализ взаимосвязей «резюме выписок» из больницы. Для этого были определены взаимосвязи пациентов и «резюме выписок» (рисунок 3.3). Резюме выписки - это описательный документ для передачи клинической информации о том, что произошло с пациентом в больнице. Это очень важно для того, чтобы сообщить врачам первичного звена и другим амбулаторным специалистам, какие последующие мероприятия необходимо провести для пациента. Были выполнены очистка и объединение текстовых отчетов о выписке из больницы. Основная цель - сгруппировать и объединить основные отчеты и дополнения к ним для каждого пациента.

	subject_id	text
0	0	CHIEF COMPLAINT: , Blood in urine.,HISTORY OF ...
1	1	HISTORY OF PRESENT ILLNESS: , A 71-year-old fe...
2	2	CHIEF COMPLAINT:;1. Infection.;2. Pelvic pai...
3	3	Dear Sample Doctor;Thank you for referring Mr...
4	4	REASON FOR ADMISSION: , Sepsis.;HISTORY OF PRE...

Рисунок 3.3 - Результаты анализа взаимосвязи между пациентами и «резюме выписок»

Следующим этапом работы было установить связь между диагнозом и пациентом (рисунок 3.4). На данном этапе происходит подготовка к анализу данных, путем загрузки и первичного просмотра информации о диагнозах и пациентах, что является первым шагом в обработке и анализе медицинских данных. В коде производится фильтрация данных о диагнозах с целью выделения случаев сахарного диабета, а затем проводится анализ полученного подмножества.

	ROW_ID	SUBJECT_ID	GENDER	DOB	DOD	DOD_HOSP	DOD_SSN	EXPIRE_FLAG
0	1	2	M	2138-07-17 00:00:00	NaN	NaN	NaN	0
1	2	3	M	2025-04-11 00:00:00	2102-06-14 00:00:00	NaN	2102-06-14 00:00:00	1
2	3	4	F	2143-05-12 00:00:00	NaN	NaN	NaN	0
3	4	5	M	2103-02-02 00:00:00	NaN	NaN	NaN	0
4	5	6	F	2109-06-21 00:00:00	NaN	NaN	NaN	0

Рисунок 3.4 - Результаты анализа взаимосвязи диагнозов и пациентов

Результаты анализа показывают, что из 651047 диагнозов 16454 (около 2.53%) связаны с сахарным диабетом, при этом уникальных пациентов с таким диагнозом - 10318.

Далее демонстрируется структура данных о случаях сахарного диабета. Это позволяет более детально изучить данные, включая коды диагнозов, идентификаторы пациентов и другую связанную информацию, специфичную для сахарного диабета (рисунок 3.5). Данный процесс позволяет сузить общий набор данных о пациентах до тех, кто имеет диагноз сахарного диабета, для дальнейшего анализа или исследования, например, изучения демографических характеристик пациентов с сахарным диабетом, анализа сопутствующих заболеваний, оценки исходов лечения и так далее.

Unnamed: 0	SUBJECT_ID	HADM_ID	SEQ_NUM	ICD9_CODE	SHORT_TITLE	LONG_TITLE	Diabetes	
66	66	13	143045	3.0	25000	DMII wo cmp nt st uncnt	Diabetes mellitus without mention of complicat...	True
94	94	18	188822	1.0	25080	DMII oth nt st uncnt	Diabetes with other specified manifestations, ...	True
108	108	20	157681	3.0	25000	DMII wo cmp nt st uncnt	Diabetes mellitus without mention of complicat...	True
128	128	21	111970	13.0	25000	DMII wo cmp nt st uncnt	Diabetes mellitus without mention of complicat...	True
130	130	21	109451	12.0	25000	DMII wo cmp nt st uncnt	Diabetes mellitus without mention of complicat...	True

Рисунок 3.5 - Список пациентов с диагнозом «Сахарный диабет»

После полученного фрейма данных о пациентах с заболеванием СД нужно обогатить дата фрейм данными, связанными с заболеваниями, детальной информацией о их диагнозах, что может быть использовано для дальнейшего анализа, исследований и принятия медицинских решений (рисунок 3.6).

ER	DOB	DOD	DOD_HOSP	DOD_SSN	EXPIRE_FLAG	DIAGNOSES	ICD9_CODE	SEQ_NUM	HADM_ID
F	2127-02-27 00:00:00	NaN	NaN	NaN	0	[Pure hypercholesterolemia, Unspecified essent...	[2720, 4019, 4111, 41401, 25000]	[5.0, 4.0, 2.0, 1.0, 3.0]	[143045, 143045, 143045, 143045, 143045]
M	2116-11-29 00:00:00	NaN	NaN	NaN	0	[Pure hypercholesterolemia, Right bundle branc...	[2720, 4264, 4019, 78321, 78057, 47829, V170, ...]	[12.0, 11.0, 10.0, 9.0, 8.0, 7.0, 13.0, 4.0, 3...]	[188822, 188822, 188822, 188822, 188822, 188822, 188822...]
F	2107-06-13 00:00:00	NaN	NaN	NaN	0	[Coronary atherosclerosis of native coronary a...	[41401, 4111, 25000, 2724, 4019]	[1.0, 2.0, 3.0, 4.0, 5.0]	[157681, 157681, 157681, 157681]
M	2047-04-04 00:00:00	2135-02-08 00:00:00	2135-02-08 00:00:00	2135-02-08 00:00:00	1	[Abscess of liver, Intestinal infection due to...	[5720, 00845, 70709, 6823, 5119, 99592, 99859, ...]	[10.0, 9.0, 5.0, 7.0, 6.0, 11.0, 8.0, 12.0, 4...]	[111970, 111970, 111970, 111970, 111970, 111970, 11197...]
M	2100-05-31 00:00:00	NaN	NaN	NaN	0	[Diabetes mellitus without mention of complica...	[25000, 41401, 41041, 53081]	[4.0, 2.0, 1.0, 3.0]	[161859, 161859, 161859, 161859]

Рисунок 3.6 - Обогащенные данные пациентов с СД

В конце проведенной работы готовый фрейм данных, с применением методов разведочного анализа данных, записывается в файл `diabetes_patients_final.csv`.

3.2.2 Статистический анализ на наборе данных по диабету

Представленный ниже график позволяет наглядно увидеть распределение пациентов с диабетом по полу, что может быть полезно для выявления половых различий среди пациентов с диабетом и может служить важной информацией для дальнейших исследований и анализа заболеваемости. Далее выводится статистика по общему количеству пациентов с диабетом и их распределению по гендеру, а затем эта информация визуализируется с помощью круговой диаграммы. На рисунке 3.7 показано распределение пациентов с диабетом по полу, определенное путём фильтрации фрейма данных `diabete_patients` по столбцу 'GENDER' и подсчёта соответствующих записей.

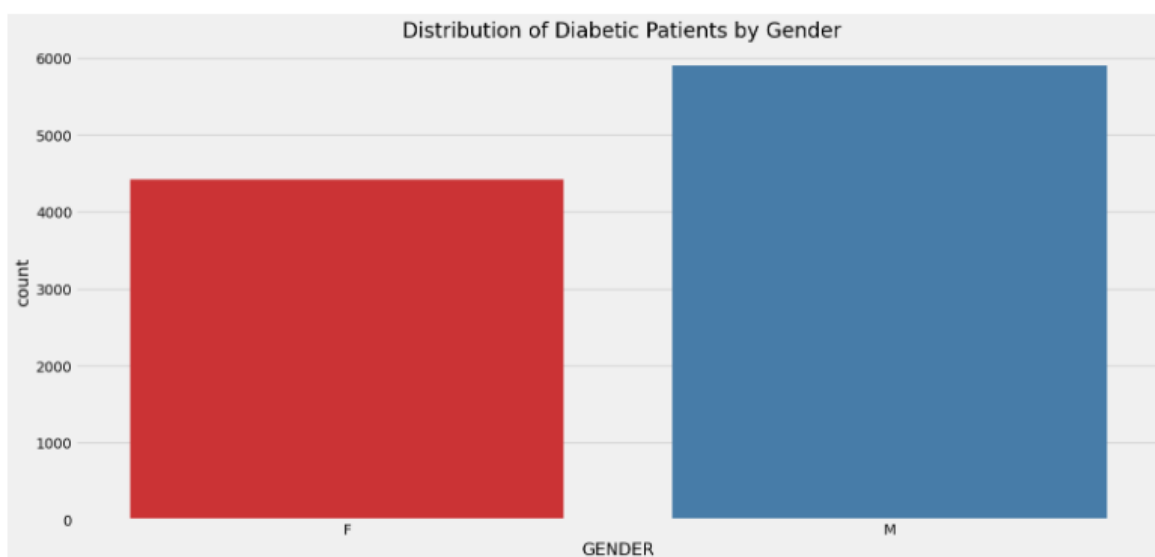


Рисунок 3.7 - Распределение пациентов с диабетом по полу

Результаты соотношения мужчин и женщин больных СД позволяет быстро оценить соотношение между количеством мужчин и женщин с диабетом, что может быть полезно при анализе распределения заболеваемости по гендеру (рисунок 3.8).

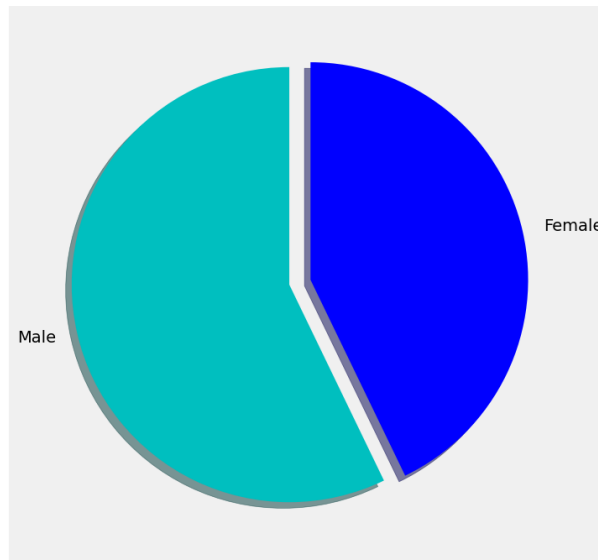


Рисунок 3.8 - Результаты соотношения мужчин и женщин больных СД

На рисунке 3.9, происходит визуализация горизонтальной столбчатой диаграмма, которая показывает отследить наиболее часто встречающиеся краткие наименования (заголовки) диагнозов среди пациентов с диабетом, используя данные из фрейма данных.

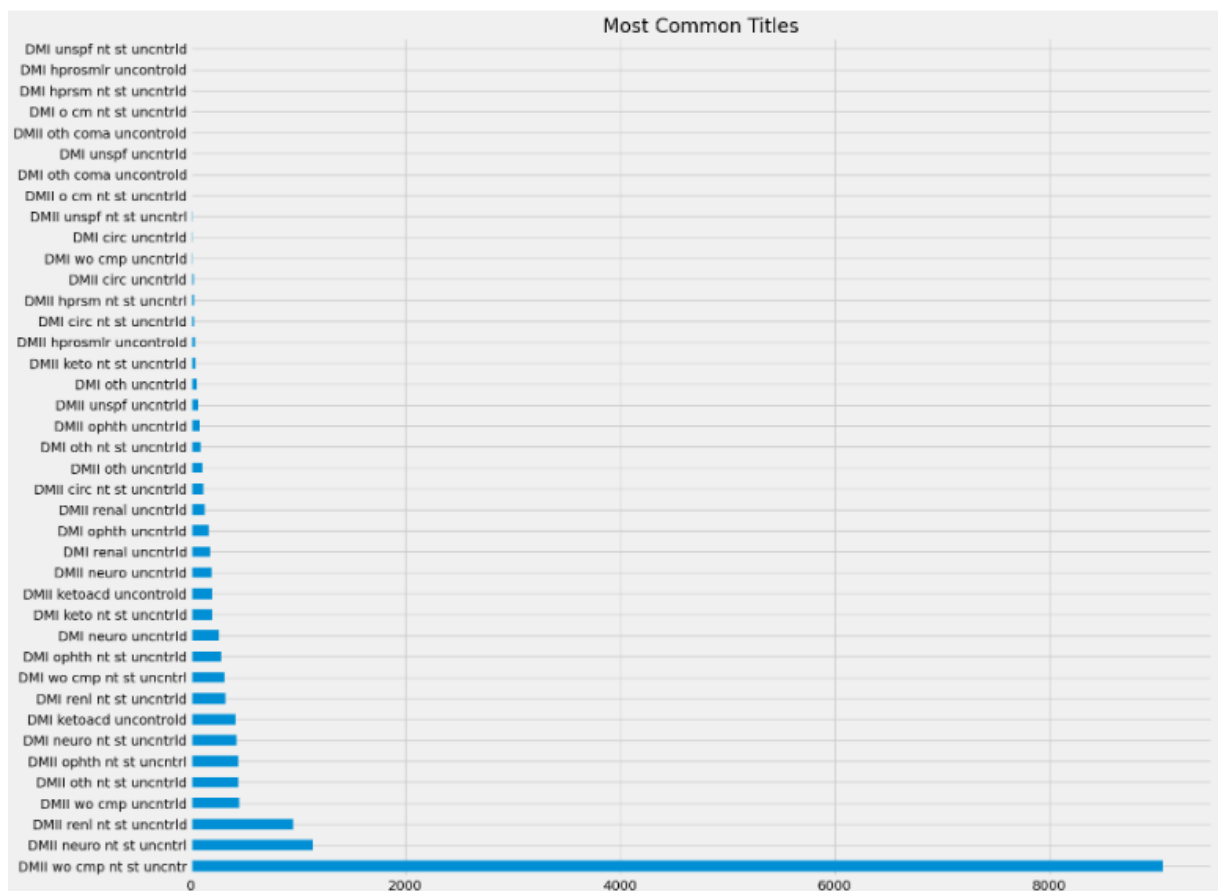


Рисунок 3.9 - Результаты анализа диагнозов среди пациентов

Приведенная на рисунке 3.9 визуализация, позволяет легко идентифицировать наиболее частые диагнозы среди исследуемой группы пациентов, выделяя те, которые встречаются особенно часто.

Далее нужно провести две статистики, связанные с диагнозами сахарного диабета в наборе данных, а затем строится столбчатая диаграмма для сравнения количества диабетических и недиабетических случаев.

Процент диагнозов сахарного диабета, рассчитывается отношение количества диагнозов сахарного диабета к общему количеству диагнозов в наборе данных. Результат показывает, что примерно 2.53% всех диагнозов в наборе данных относятся к сахарному диабету (рисунок 3.10).

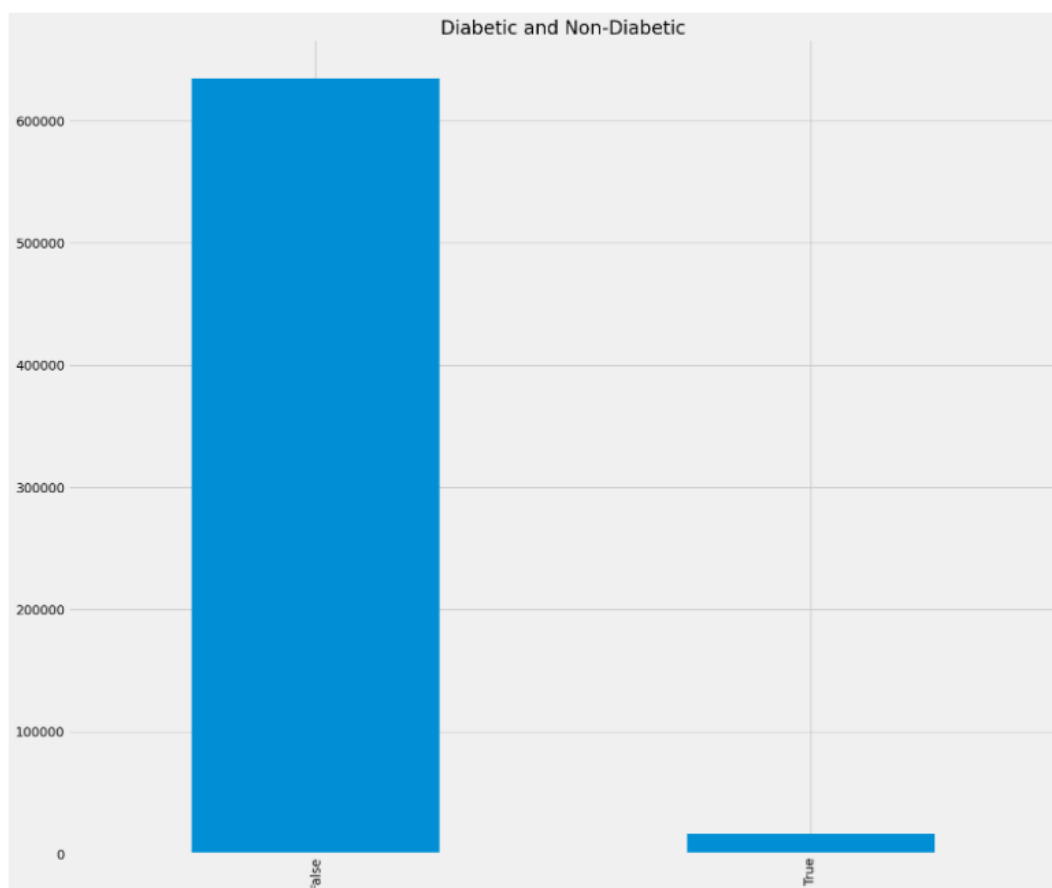


Рисунок 3.10 - Соотношение пациентов с заболеванием СД к пациентам с прочими заболеваниями

Диаграмма эффективно иллюстрирует различие между количеством диагнозов сахарного диабета и всеми другими диагнозами, предоставляя важное визуальное представление о распределении диагнозов в исследуемом наборе данных.

3.2.3 Анализ клинических записей по пациентам

В начале оценивается файл с данными о пациентах, для это выводятся записи по пациентам из файла patients.csv. Происходит соединение таблиц patients.csv и noteevents.csv для выявления детальной информации. Затем строится график распределения пациентов по гендерному признаку, мужчина и женщина, который показывает количество мужчин и женщин в наборе данных. Далее загрузили данные из файла noteevents.csv и отобразили первые пять записей, чтобы получить представление о структуре данных. Эти записи содержат столбцы Unnamed: 0, subject_id, chartdate, category и text (рисунок 3.11).

	Unnamed: 0	subject_id	chartdate	category	text
0	0	0	01/01/2086	Urology	CHIEF COMPLAINT: , Blood in urine.,HISTORY OF ...
1	1	0	01/01/2086	Emergency Room Reports	CHIEF COMPLAINT: , Blood in urine.,HISTORY OF ...
2	2	0	01/01/2086	General Medicine	CHIEF COMPLAINT: , Blood in urine.,HISTORY OF ...
3	3	0	01/01/2086	General Medicine	CHIEF COMPLAINT: Followup on hypertension an...
4	4	0	01/01/2086	Consult - History and Phy.	CHIEF COMPLAINT: , Blood in urine.,HISTORY OF ...

Рисунок 3.11 - Структура данных таблицы noteevents

Следующим шагом было необходимо рассчитать длину каждой записи в столбце text и построили распределение длин документов. Это было сделано для визуальной оценки распределения длины записей, что видно на гистограмме (рисунок 3.12).

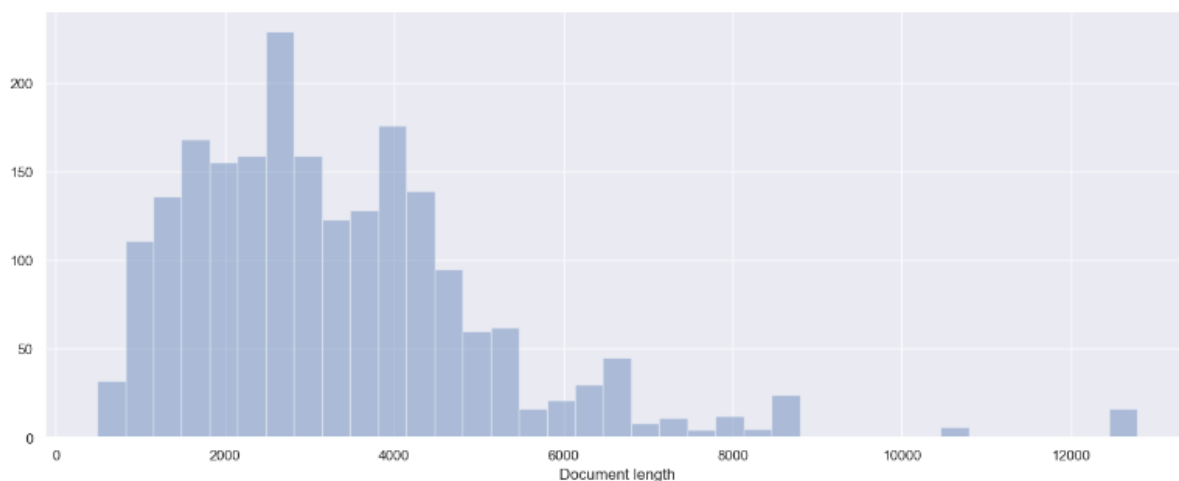


Рисунок 3.12 - Результаты анализа длины документов

После выполнения очистки данных, а именно были удалены самые короткие и самые длинные записи, это 5% записей, чтобы исключить выбросы, которые могли бы исказить анализ. Затем вновь построили гистограмму распределения длины документов без этих выбросов. Данная очистка была проделана путем нахождения минимальных и максимальных длин документов, которые использовались в качестве порогов для очистки данных.

Так же применили условие фильтрации к исходным данным, исключив записи, длина которых выходила за установленные границы.

После очистки данных было проведено сравнение размеров до и после очистки. Сравнили размер нового набора данных (noteevents) с исходным (noteevents_original), обнаружив, что число записей сократилось с 2130 до 1914.

С помощью гистограммы (рисунок 3.13) было визуализировано распределение количества документов на пациента, что позволяет увидеть, сколько записей приходится на каждого пациента в датасете. Эта визуализация показывает, что большинство пациентов имеют относительно небольшое количество записей.

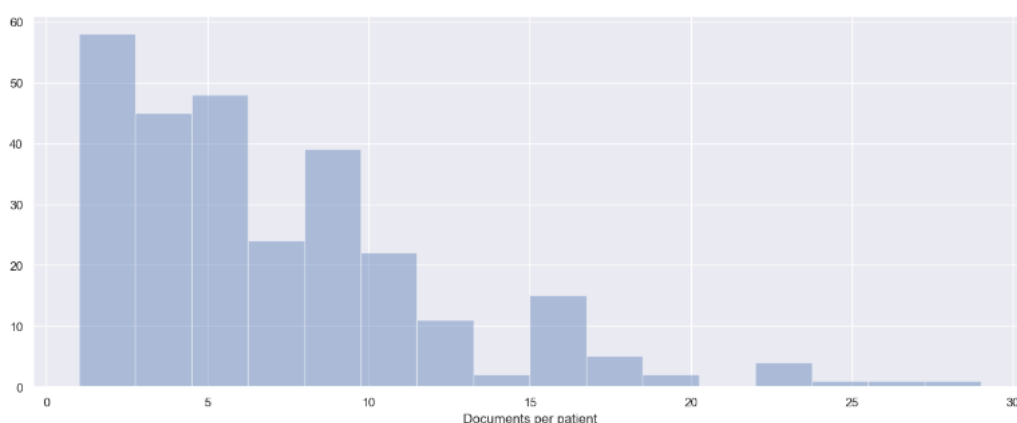


Рисунок 3.13 - Визуализации документов на каждого пациента

Просмотрев общие данные, было решено удалить данные по 1% пациентов с наименьшим и наибольшим количеством документов. Это сделано для удаления возможных выбросов, которые могли бы исказить общую картину. Например, если некоторые пациенты имеют необычайно много записей, они могут непропорционально влиять на статистический анализ. Были так же проведены пересчитывание и очистка списка. Исходя из пороговых значений, был создан набор keep_subject_id, который содержит subject_id пациентов, не попадающих под критерии удаления. Затем происходит фильтрация исходного DataFrame noteevents, оставляя только записи, относящиеся к этим пациентам (рисунок 3.14).

Unnamed: 0	subject_id	chartdate	category	text
6	6	1 01/01/2079	General Medicine	HISTORY OF PRESENT ILLNESS; The patient is a ...
7	7	1 01/01/2079	Rheumatology	HISTORY OF PRESENT ILLNESS; , A 71-year-old fe...
8	8	1 01/01/2079	Consult - History and Phy.	HISTORY OF PRESENT ILLNESS; The patient is a ...
9	9	2 01/01/2037	Consult - History and Phy.	CHIEF COMPLAINT;1. Infection,.2. Pelvic pai...
10	10	2 01/01/2037	Dermatology	SUBJECTIVE; This is a 29-year-old Vietnamese...

Рисунок 3.14 - Пример данных после очистки и фильтрации

Проведя сравнительный анализ размеров данных, можно сказать, что количество записей после очистки сократилось с 2130 до 1305, что демонстрирует, что некоторые данные были удалены в результате процесса фильтрации и очистки.

Была проведена визуализация распределения документов по их категориям представлена на рисунке 3.15. Для каждой категории записей показано количество соответствующих документов. Например, категория с самым большим количеством записей занимает самую левую позицию на диаграмме и выделена другим цветом.

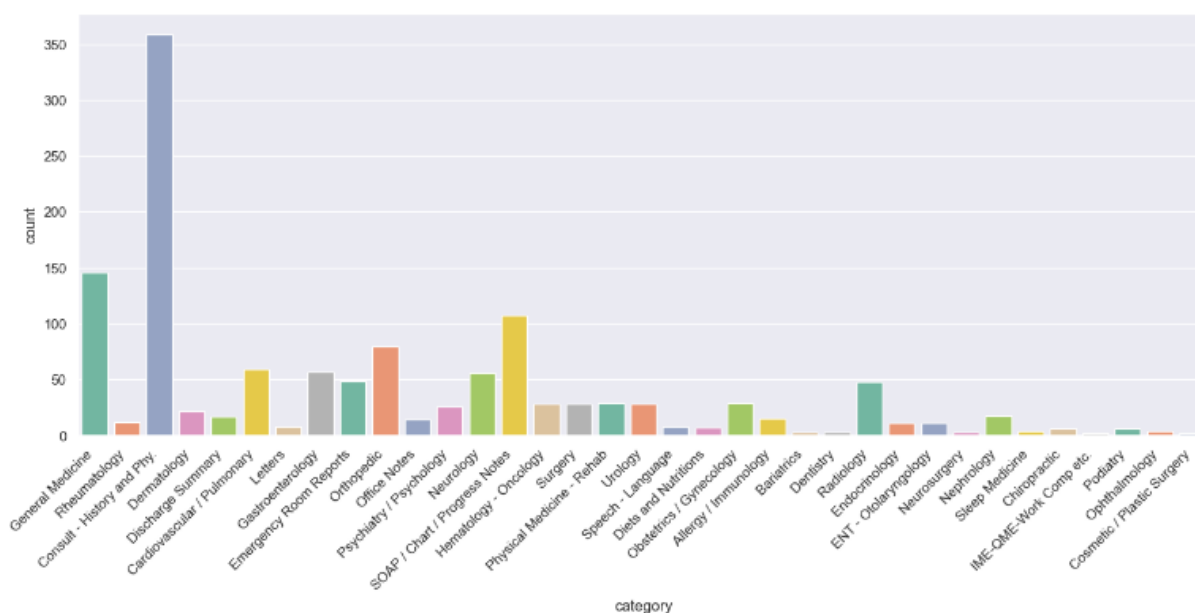


Рисунок 3.15 - Визуализация распределения документов по их категориям

Представленная визуализация на рисунке выше, позволяет легко идентифицировать наиболее и наименее распространенные типы документов в наборе данных. Такой анализ может быть полезен для понимания, какие категории медицинских записей являются наиболее частыми и могут потребовать дополнительного внимания при анализе данных больницы.

Далее идет подготовка данных для более детального анализа из набора таблиц под названием «NOTEEVENTS». Детальный анализ связан с исходами лечения и управления лечением пациентов после выписки.

3.3 Экспериментальное исследование алгоритма поддержки клинических решений эндокринологии на основе технологии андерсэмплинга

3.3.1 Этапы предобработки данных

Данное исследование предполагает анализ не только лабораторных показателей пациентов, но также и текстовые данные из медицинских записей, диагнозов или отчетов.

На этапе предобработки были выполняли такие действия, как очистка текста (удаление HTML-тегов, чисел, специальных символов и т.д.), приведение к нижнему регистру, удаление стоп-слов, лемматизация и стемминг.

После очистки текстовых данных было важно понять, какие слова являются наиболее значимыми, что и достигается с помощью таких инструментов, как облака слов (рисунок 3.16).



Рисунок 3.16 – Примеры облака слов

Облако слов на рисунке 3.16 отражает наиболее значимые термины после предобработки текста. В исследование использовались три разных подхода для создания текстовых характеристик:

Count Vectoriser - метод отображение частоты слов без учета взаимосвязей между словами (первый пример на рисунке 3.16);

TFIDF Vectoriser - метод представления важности слов с учетом не только их частоты, но и редкости в документе, что позволяет выделить более значимые слова (второй пример на рисунке 3.16);

Compound Vectoriser (Count + TFIDF) – комбинирование методов Count Vectoriser и TFIDF Vectoriser, то есть учтены обе метрики (третий пример на рисунке 3.16).

Из рисунка 3.16 можно понять, что наиболее частые слова, такие термины, как "heart failure", "unspecified", "essential hypertension", и т.д., являются наиболее часто встречающимися. Это может говорить о том, что данные, вероятно, связаны с хроническими заболеваниями, особенно сердечно-сосудистыми, и с гипертонией.

После подготовки данных начинается этап моделирования, который может включать настройку гиперпараметров, обучение моделей и оценку их производительности. Итоговый набор данных или фрейм данных приведен на рисунке 3.17.

Следующим шагом в исследование является балансировка набора данных для создания равного количества случаев с диагнозом диабета и без него.

Финальный фрейм данных имеет равное количество записей для пациентов с диабетом и без, что составляет 31938 случаев. Это создает сбалансированный набор данных, который позволит построить модель, не предвзято относящуюся к какому-либо классу (с диабетом или без).

gender	age_at_admission	marital_status	ethnicity	admission_type	diagnosis	lab_value_num	lab_label	fluid	category	family_history	obesity_status
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	164.0	Glucose	Blood	Blood Gas	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	194.0	Glucose	Blood	Blood Gas	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	1.8	Creatinine	Blood	Chemistry	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	2.0	Creatinine	Blood	Chemistry	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	65.0	Glucose	Blood	Chemistry	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	134.0	Glucose	Blood	Chemistry	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	null	Glucose	Urine	Hematology	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	7.5	pH	Urine	Hematology	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	1.7	Creatinine	Blood	Chemistry	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	1.7	Creatinine	Blood	Chemistry	No	No
M	68	WIDOWED	WHITE	EMERGENCY	INFECTED LEFT FOOT,DIABETES	1.8	Creatinine	Blood	Chemistry	No	No

Рисунок 3.17 - Итоговый фрейм данных для моделирования перед балансировкой

Сбалансированный набор данных (рисунок 3.18) помогает обеспечить более точную и справедливую оценку производительности модели предсказания.

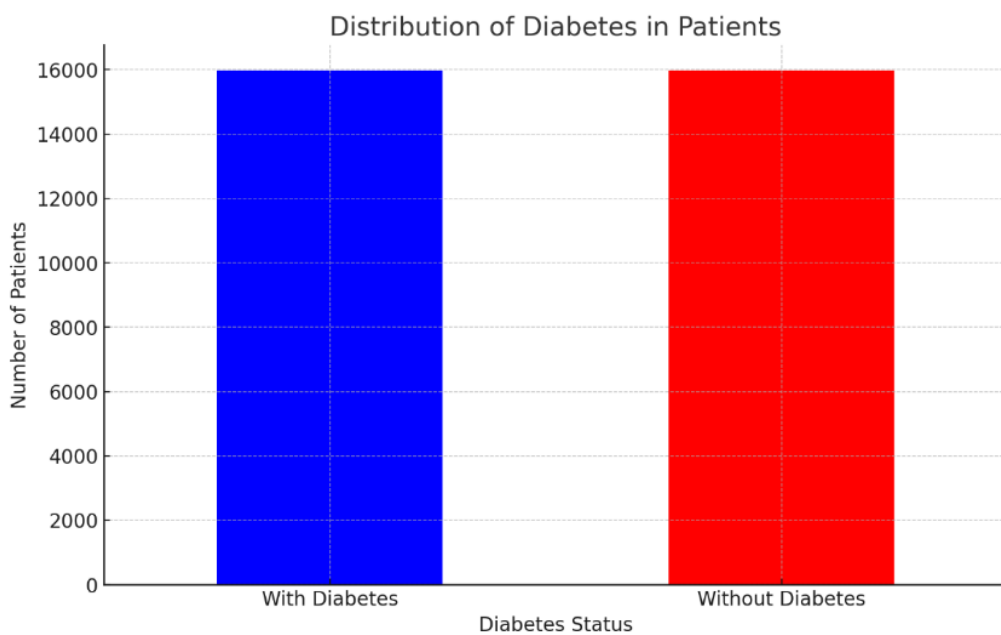


Рисунок 3.18- Сбалансированный набор данных

На графике 3.18 показано распределение пациентов по наличию диабета. Можно увидеть, что количество пациентов с диабетом и без диабета в выборке сбалансировано, что является важным для обучения машинных моделей, чтобы избежать смещения в сторону одного из классов.

3.3.2 Создание матрицы признаков

Для создания матрицы признаков использовались основные категории параметров для определения диабета, представленные в таблице 3.5.

Таблица 3.5 - Основные категории параметров для определения диабета

№	Категории	Показатели
1	Демографические данные пациента	age_at_admission - возраст на момент госпитализации gender - пол пациента ethnicity - этническая принадлежность marital_status - семейное положение
2	Лабораторные анализы	Glucose - Уровень глюкозы в крови HbA1c - Гликозилированный гемоглобин pH - количество показателя в крови Глюкоза в моче Холестерин (cholesterol) - можно добавить уровень общего холестерина и липопротеинов. Blood_urea_nitrogen (мочевина крови) - для оценки функции почек. Creatinine (креатинин) - для оценки функции почек, особенно важно для пациентов с диабетом. Triglycerides (триглицериды) - связаны с диабетом и метаболическим синдромом. Указывать на состояние питания и функцию печени: total_protein - общий белок albumin - альбумин
3	Vitae параметры (жизненные показатели)	heart_rate - частота сердечных сокращений blood_pressure - артериальное давление, включая систолическое и диастолическое давление respiratory_rate - частота дыхания temperature - температура тела spo2 - сатурация кислорода
4	Диагностические коды и история заболеваний	diagnosis - основной диагноз при поступлении, можно анализировать текст на наличие указаний на диабет. Icd_code (коды ICD в diagnoses_icd) - можно добавить коды ICD для диабета (например, E10-E14). История заболеваний (например, наличие сердечно-сосудистых заболеваний, хронических заболеваний почек).
5	Медикаментозная терапия	Информация из таблицы prescriptions: Назначенные препараты , такие как инсулин, гипогликемические препараты (например, метформин). Тип лечения (medication_type) - инсулин, пероральные сахароснижающие препараты и т.д.
6	Процедуры и вмешательства	Коды процедур (procedures_icd) - процедуры, связанные с лечением диабета Процедуры диализа - для пациентов с диабетической нефропатией
7	Антропометрические показатели	height (рост) и weight (вес) - для расчета индекса массы тела (BMI). Bmi (Body Mass Index) - расчет на основе роста и веса, важно для оценки ожирения.

Продолжение таблицы 3.5

8	Показатели функции печени и электролиты	alanine_aminotransferase (АЛТ) и aspartate_aminotransferase (АСТ) - показатели функции печени. Электролиты и их баланс: sodium - натрий, potassium - калий, chloride - хлорид(32)
9	Симптомы	Аномальная жажда и сухость во рту Нехватка энергии и крайняя усталость Постоянное чувство голода Внезапная потеря веса Расплывчатость зрения Внезапная потеря веса

Стоит отметить, что в качестве матрицы признаков (X) используется табличная структура, где каждая строка представляет индивидуальное наблюдение (в данном случае пациента), а каждый столбец - определенный признак (характеристику), который может быть использован для предсказания.

Количество признаков равно 38, данные признаки были сформированы из МКБ предоставляемые ВОЗ и медицинских протоколов республиканского уровня.

А в качестве вектора меток (Y) представлен столбец, содержащий метки классов или значения, которые модель должна предсказать, на основе данных в матрице признаков.

Этапы формирования матрицы X и вектора Y можно описать следующими шагами:

- ШАГ 1 - Создание матрицы признаков X;
- ШАГ 2 - Создание вектора меток Y;
- ШАГ 3 - Кодирование категориальных признаков;
- ШАГ 4 - Выбор числовых признаков;
- ШАГ 1 - Конкатенация признаков.

Конечный результат этих действий - матрица признаков X с 38 признаками и вектор меток Y для 31938 пациентов, готовые к использованию в моделях машинного обучения для предсказания наличия диабета (рисунок 3.18).

На рисунке 3.19 показан график распределение категориальных признаков после их преобразования в числовые с помощью метода one-hot encoding. Каждая колонка на графике представляет собой отдельный признак после кодирования, и высота колонки отображает количество раз, которое каждый признак встречается в данных.

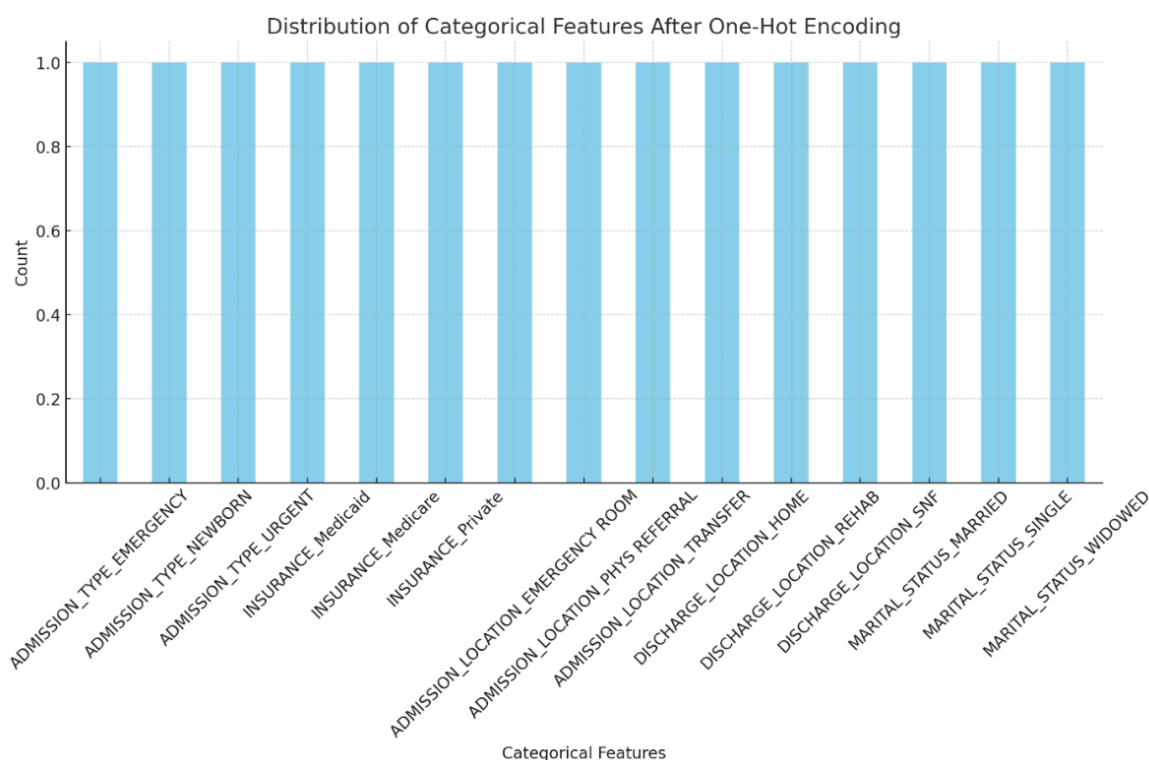


Рисунок 3.19- Результаты распределения категориальных признаков после их преобразования

3.3.3 Проблема задачи классификации и её решение

Поскольку данное исследование посвящено задаче классификации пациентов по состоянию здоровья возникает необходимость решения проблемы перекрытия классов. Перекрытие классов влечёт за собой низкую точность и надёжность работы моделей машинного обучения, а также приводит к частым ошибкам.

Термин «перекрытие классов» используется в машинном обучении для описания ситуации, когда различные классы (группы) данных не четко разделены друг от друга, и их характеристики или признаки совпадают или очень близки. Это может привести к тому, что модель машинного обучения испытывает трудности с правильным классифицированием новых примеров, поскольку она не может легко определить, к какому классу относится каждый пример.

Решением в такой ситуации будет служить уменьшение влияния перекрытия. Для уменьшения влияния перекрытия классов могут использоваться различные методы, включая:

1. Выбор или создание более информативных признаков, которые лучше разделяют классы.
2. Применение техник андерсэмплинга и оверсэмплинга для изменения распределения классов в тренировочных данных.
3. Использование более сложных моделей или настройка гиперпараметров, которые могут лучше справляться с перекрытием классов.

Для диагностирования СД или преддиабетного состояния в исследовании используются гематологические признаки, а именно лабораторные результаты анализов пациентов. Они взяты из открытого источника и содержат в себе информацию о пациентах, поступивших в отделение интенсивной терапии. Эти пациенты уже имеют какие-либо заболевания, но исходя из принципов уменьшения влияния перекрытия классов, согласно первому пункту для осуществления более точной классификации необходимы также данные здоровых пациентов. Это поможет сформировать более информативные признаки. Необходимые данные пациентов и эффективная точная классификация поможет выявить скрытые зависимости между состоянием пациента и его лабораторными данными, что может послужить прорывом в диагностировании сахарного диабета. так же скрытые зависимости.

Практическим путём было выявлено, что на улучшение классификации данных влияет применение техники андерсэмплинга и отсутствие применения каких-либо техник. Техника оверсэмплинга не использовалась, поскольку в медицинских задачах не может фигурировать большое количество синтетических данных. Соотношение больных без диабета и больных диабетом в имеющихся данных составляет 552336 к 15969 пациентов.

Из пункта три следует, что можно создать более сложную модель, которая может справляться с перекрытием классов лучше.

3.3.4 Подход к балансировке классов

Для последующего обучения машинных моделей применяются два разных подхода к балансировке классов в предобработке данных. Основные отличия данных подходов заключаются в методе балансировки и влиянии на модель. Метод балансировки в первом подходе использует данные в их исходном распределении без вмешательства в баланс классов, в то время как второй подход активно изменяет баланс классов через удаление Tomek Links для уменьшения перекрытия классов и потенциального улучшения модели.

Первый подход - без андерсэмплинга.

Данный подход заключается в трех ключевых моментах:

1) Обработка пропусков. Применяется заполнение пропущенных значений медианой по каждому столбцу. Это стандартный метод обработки пропущенных данных, который помогает поддерживать статистическую целостность данных.

2) Разделение данных. Данные разделяются на обучающую и тестовую выборки с использованием функции `train_test_split` из `sklearn`. Процент тестовой выборки составляет 30% от всего датасета.

3) Баланс классов. Не применяется никакой метод балансировки классов перед разделением данных, данные используются в том виде, в каком они есть после предварительной обработки.

Второй подход - андерсэмплинг. В данном подходе стоит отметить так же три ключевых пункта:

- Использование Tomek Links. Применяется андерсэмплинг с помощью метода TomekLinks из библиотеки imblearn. Этот метод удаляет так называемые Tomek Links - пары близко расположенных примеров разных классов. Это помогает уменьшить перекрытие классов и улучшить качество классификации.

- Преобразование данных. После формирования финальной матрицы признаков X из числовых и закодированных категориальных признаков происходит их андерсэмплинг с помощью fit_resample, что приводит к изменению исходного размера набора данных в зависимости от удаления Tomek Links.

- Разделение данных. Так же, как и в первом фрагменте, данные разделяются на обучающую и тестовую выборки, но уже после андерсэмплинга, что может привести к изменению распределения и характеристик данных.

Проведенный эксперимент показал, что данный метод возможен, но его результаты оказались хуже, чем результаты, полученные при использовании второго подхода. Поэтому далее балансировка данных в исследование будет осуществляться на основе андерсэмплинга.

Влияние на модель отражается в том, что при андерсэмплинге можно улучшить способность модели к классификации, особенно если данные изначально имели сильное перекрытие между классами. Однако это может привести к потере важной информации, если данные не обладают значительным перекрытием классов.

Так же была проведена нормализация данных. Нормализация данных необходима для приведения признаков к одному масштабу, поскольку алгоритмы машинного обучения чувствительны к диапазону входных данных. В данном исследовании был использован метод нормализации Min-Max Scaling. Он был выбран с целью сохранить исходное распределение данных.

3.3.5 Использование метода андерсэмплинга в обучении моделей

С помощью метода Tomek Links, происходит балансировка классов в данных, что может улучшить производительность моделей машинного обучения на несбалансированных наборах данных. После балансировки классов проводится разделение на обучающий и тестовый наборы, затем выполняется кросс-валидация с различными классификаторами.

На рисунке 3.20 представлен график результатов кросс-валидации для алгоритмов DecisionTree, AdaBoost, RandomForest, ExtraTrees и GradientBoosting, показывая средние значения точности и стандартные отклонения.

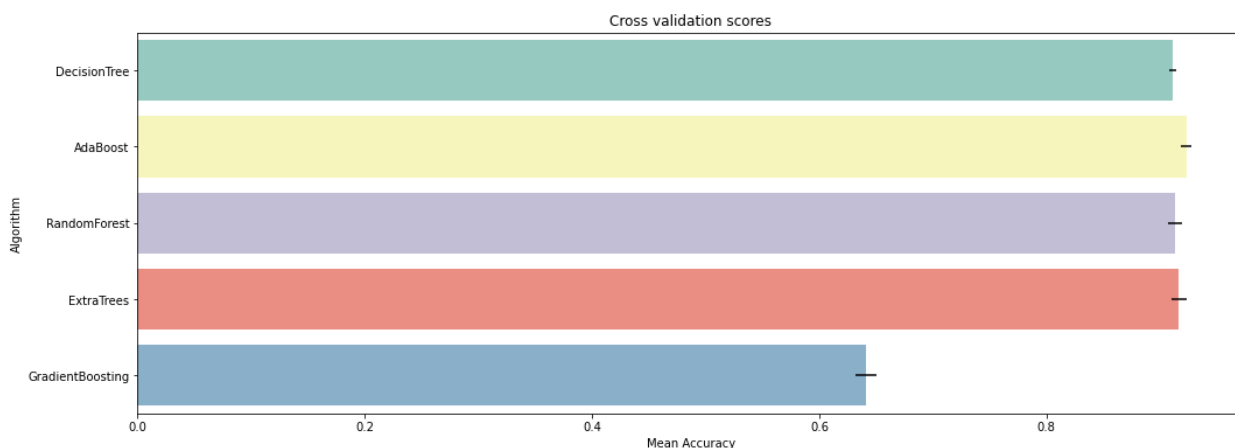


Рисунок 3.20- График результатов кросс-валидации

По результатам кросс-валидации, представленным в таблице 3.6 можно оценить, какие из моделей наилучшим образом справляются с задачей классификации в условиях текущего набора данных.

Таблица 3.6 - Результаты кросс-валидации алгоритмов

	CrossValMeans	CrossValerrors	Algorithm
0	0.911332	0.002996	DecisionTree
1	0.923008	0.004382	AdaBoost
2	0.913211	0.005768	RandomForest
3	0.916611	0.006340	ExtraTrees
4	0.640987	0.00891	GradientBoosting

Исходя из предоставленных таблицы значений кросс-валидации и рисунка 3.20, можно сделать следующие выводы:

- AdaBoost показывает наивысшую среднюю точность (0.923008) среди всех алгоритмов, что указывает на его эффективность в данном наборе данных. При этом стандартное отклонение (0.004382) остается относительно низким, что свидетельствует о стабильности модели на разных подвыборках данных.

- ExtraTrees и RandomForest также демонстрируют высокую среднюю точность (0.916611 и 0.913211 соответственно) с умеренными стандартными отклонениями (0.006340 и 0.005768). Это подчеркивает их надежность и способность хорошо обобщать данные, делая их подходящими кандидатами для дальнейшего использования в проекте.

- DecisionTree достигает средней точности 0.911332 с минимальным стандартным отклонением (0.002996), что делает его сравнительно стабильным и эффективным, хотя и немного уступает другим ансамблевым методам.

- GradientBoosting показывает значительно ниже среднюю точность (0.640987) с относительно высоким стандартным отклонением (0.00891). Это может указывать на проблемы с переобучением или недостаточной адаптацией модели к используемым данным, требующие дополнительного

анализа и возможно корректировки подхода к обучению или параметров модели.

3.3.6 Комплексный подход для андерсэмплинга

Используя класс `EstimatorSelectionHelper`, проводится обширный поиск по сетке (`Grid Search`) для различных ансамблевых классификаторов с целью нахождения лучших гиперпараметров на основе точности (`accuracy`) как метрики оценки. Создается экземпляр класса `EstimatorSelectionHelper` с заданными моделями и их соответствующими параметрами для поиска.

`Grid Search` запускается для каждой модели с указанными параметрами. Для каждой модели выводится сообщение о начале выполнения, количестве комбинаций параметров и количестве итераций, которые будут выполнены.

По завершении поиска для каждой модели собираются и сохраняются результаты в словаре `grid_searches`, который ассоциирован с ключами моделей.

Этот подход позволяет автоматизировать процесс выбора модели, оптимизируя несколько моделей и их параметров одновременно и предоставляя удобный способ сравнения их производительности. В результате получаем подробные данные о производительности каждой модели, что помогает принимать информированные решения о выборе модели для дальнейшего использования.

На рисунке 3.21 представлены результаты, полученные с помощью класса `EstimatorSelectionHelper`, который был использован для проведения `Grid Search` и кросс-валидации для ряда классификаторов. Результаты сортированы по максимальному значению точности (`max_score`).

Для каждого классификатора указаны следующие метрики:

- `min_score`: Минимальная достигнутая оценка.
- `mean_score`: Средняя оценка всех итераций кросс-валидации.
- `max_score`: Максимальная достигнутая оценка.
- `std_score`: Стандартное отклонение оценок, отражающее стабильность модели.

- Параметры модели, такие как `max_depth`, `learning_rate`, `n_estimators` и `max_features`.

	estimator	min_score	mean_score	max_score	std_score	max_depth	learning_rate	n_estimators	max_features
41	GradientBoostingClassifier	0.911884	0.914016	0.916111	0.001726	9	NaN	500	auto
37	GradientBoostingClassifier	0.87008	0.875908	0.880143	0.00425985	7	NaN	500	auto
40	GradientBoostingClassifier	0.861719	0.868956	0.877607	0.00656254	9	NaN	200	auto
39	GradientBoostingClassifier	0.868765	0.870366	0.871877	0.00127209	9	NaN	500	sqrt
36	GradientBoostingClassifier	0.802912	0.811529	0.823502	0.00873425	7	NaN	200	auto
38	GradientBoostingClassifier	0.79427	0.802073	0.807985	0.00575673	9	NaN	200	sqrt
33	GradientBoostingClassifier	0.7845	0.791301	0.804152	0.0090921	5	NaN	500	auto
35	GradientBoostingClassifier	0.791639	0.794589	0.799079	0.0032269	7	NaN	500	sqrt
34	GradientBoostingClassifier	0.724847	0.731964	0.739808	0.00612949	7	NaN	200	sqrt
32	GradientBoostingClassifier	0.710005	0.719094	0.734079	0.0106756	5	NaN	200	auto
31	GradientBoostingClassifier	0.703805	0.710828	0.717077	0.00544594	5	NaN	500	sqrt
29	GradientBoostingClassifier	0.677219	0.686498	0.692748	0.00669148	3	NaN	500	auto
30	GradientBoostingClassifier	0.665853	0.672752	0.678753	0.00530458	5	NaN	200	sqrt
17	RandomForestClassifier	0.665195	0.670246	0.678565	0.00592684	9	NaN	500	NaN
16	RandomForestClassifier	0.661343	0.666145	0.674056	0.00563667	9	NaN	200	NaN
27	GradientBoostingClassifier	0.635698	0.6472	0.660436	0.0101734	3	NaN	500	sqrt
28	GradientBoostingClassifier	0.644058	0.650989	0.654894	0.00491412	3	NaN	200	auto
24	ExtraTreesClassifier	0.642179	0.646104	0.653673	0.00535281	9	NaN	200	NaN
25	ExtraTreesClassifier	0.639737	0.644257	0.652921	0.0061285	9	NaN	500	NaN

Рисунок 3.21 - Результат работы GridSearchCV

На основании этих данных можно сделать вывод о том, какие настройки параметров классификатора показывают лучшие результаты на данном наборе данных. В частности, GradientBoostingClassifier с max_depth=9, n_estimators=500 и max_features='auto' показал высокую среднюю и максимальную точность, что делает его потенциально хорошим выбором для дальнейшего использования и более глубокого тестирования.

3.3.7 Финальная оценка модели

Для каждой модели были протестированы различные классификаторы с уточнёнными гиперпараметрами и оценены дополнительные метрики, такие как точность (accuracy), F1 score, recall и precision на обучающем и тестовом наборах данных.

Выведенные значения точности для тестовой выборки (Test Accuracy) и других метрик в полученных результатах на самом деле не равны 1.0, как указано в выводе перед каждым отчётом о классификации. Вместо этого, значения, которые следует учитывать, находятся внутри самих отчётов о классификации и матрицах ошибок для каждой модели. Эти результаты показывают, как модели работают на практике.

Например, для GradientBoosting. Точность на тестовом наборе составила 0.92. F1 score для класса True составил 0.91, что указывает на хорошее сочетание precision и recall. Матрица ошибок показывает, что модель совершила 116 ошибок первого рода (False Positive) и 669 ошибок второго рода (False Negative) для класса True. Аналогичные результаты представлены для

других моделей, и по ним видно, что GradientBoosting превосходит другие модели по всем основным метрикам.

Эти результаты могут быть использованы для выбора наилучшей модели для задачи классификации, при этом GradientBoosting является самым многообещающим кандидатом на основании данных метрик. При выборе модели для реального применения необходимо также учитывать компромисс между различными типами ошибок (например, как важно минимизировать ошибки первого или второго рода) и требования к вычислительной сложности и скорости работы модели.

На рисунке 3.22 представлены метрики для различных классификаторов, протестированных на наборе данных. Перечислены показатели точности на обучающем наборе (Train Accuracy) и тестовом наборе (Test Accuracy), а также метрики Precision, Recall и F1 Score на тестовом наборе данных.

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
DecisionTree	0.658347	0.641202	0.684453	0.526458	0.595148
AdaBoost	0.590222	0.585055	0.623501	0.433333	0.511308
RandomForest	0.688853	0.671467	0.696948	0.608958	0.649989
ExtraTrees	0.658481	0.643394	0.671887	0.563125	0.612717
GradientBoosting	0.958356	0.916301	0.970132	0.859375	0.911401

Рисунок 3.22 - Результаты оценки различных классификаторов машинного обучения

GradientBoostingClassifier значительно превосходит другие модели по всем показателям, что делает его лучшим выбором из протестированных алгоритмов для данной задачи классификации.

На рисунке 3.23 показаны результаты оценки пяти различных классификаторов: DecisionTree, AdaBoost, RandomForest, ExtraTrees и GradientBoosting. Каждый классификатор оценивается по пяти метрикам: точность обучения (Train Accuracy), точность тестирования (Test Accuracy), точность (Precision), полнота (Recall) и совокупная метрика F1 (F1 Score).

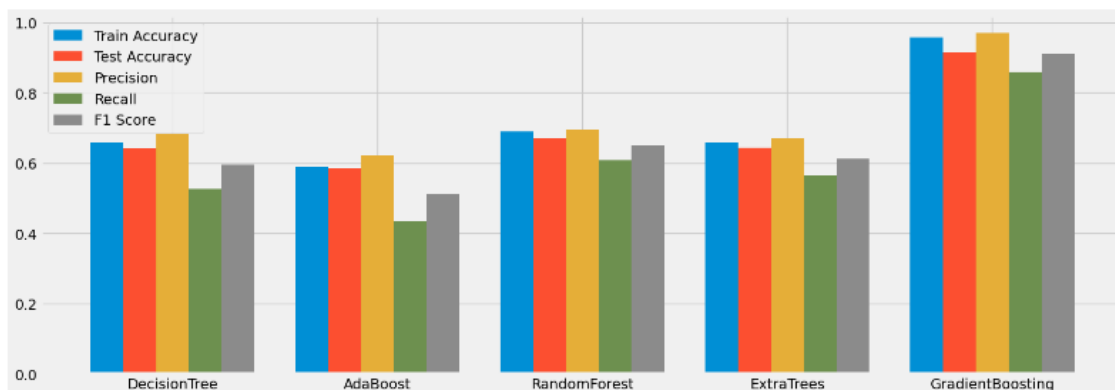


Рисунок 3.23- График сравнения различных метрик качества для выбранных алгоритмов машинного обучения

GradientBoostingClassifier явно выделяется, показывая самые высокие результаты по всем пяти метрикам. Это указывает на его способность эффективно обобщать и прогнозировать на новых данных, что делает его предпочтительным выбором для данной задачи классификации.

Модели DecisionTree и AdaBoost показывают схожие показатели по всем метрикам, хотя AdaBoost имеет несколько более низкие значения по сравнению с DecisionTree. RandomForest и ExtraTrees показывают схожую производительность, но они уступают GradientBoosting.

3.4 Экспериментальное исследование алгоритма применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN

3.4.1 Подготовка данных для экспериментального исследования

Экспериментальное исследование алгоритма применения метода Grid Search для задач поддержки принятия клинических решений на основе модели CNN начинается с импорта необходимых библиотек, таких как Keras и TensorFlow. Данные библиотеки позволяют строить сложные архитектуры нейронных сетей и методы оптимизации.

После того как все необходимые библиотеки загружены, загружаются данные и создается окончательный фрейм данных. Параллельно при создании фрейма данных проводится удаление пропусков, балансировка данных, объединение и перемешивание данных случайным образом, сбрасываются индексы для удаления любой предыдущей упорядоченности. Итого получается:

- Количество пациентов без диабета: 552336
- Количество пациентов с диабетом: 15969
- Размер финального сбалансированного DataFrame: 31938 строк на 13 колонок

Осуществляется подготовка данных для последующего обучения модели машинного обучения. Процесс начинается с исключения целевой переменной Diabetes из набора данных, сохраняя её значения в вектор меток Y для обучения. Для признаков, участвующих в моделировании, производится трансформация категориальных переменных в числовой формат с помощью метода One-Hot Encoding, что позволяет преобразовать категориальные атрибуты в формат, подходящий для анализа. Одновременно, выбранные числовые признаки извлекаются непосредственно из исходного фрейма данных. Результатом этих операций является итоговая матрица признаков X, которая объединяет как числовые, так и закодированные категориальные данные, и готова к использованию в алгоритмах машинного обучения.

Конкатенация числовых и обработанных категориальных признаков в единую матрицу обеспечивает комплексное представление данных, необходимое для эффективного обучения моделей. Финальная проверка размеров матриц X и Y подтверждает их соответствие, с 31938 экземплярами

и 38 признаками, что указывает на успешную подготовку данных к фазе моделирования. Эти шаги важны для обеспечения качества и последовательности данных перед их использованием в аналитических моделях. Далее выборка делится на тренировочную и тестовую выборки.

3.4.2 Реализация модели CNN для задачи бинарной классификации

Следующим шагом в исследование является создание глубокой нейронной сети для задачи бинарной классификации.

Сеть строится с использованием последовательной модели Sequential, которая позволяет добавлять слои в стековом порядке.

Последовательная модель или Sequential в Keras, это способ построения модели, где слои соединяются строго последовательно. А стековый порядок означает, что каждый слой добавляется в конце списка существующих слоев, и каждый слой имеет один выход и один вход.

Данный метод был выбран так как метод Sequential упрощает построение модели, так как нужно определить только типы слоев и их параметры в нужном порядке, связи между слоями устанавливаются автоматически моделью. Типы слоёв и их описание представлены в таблице 3.7.

Таблица 3.7 - Архитектура исходной модели нейронной сети

Слой	Описание
1	Принимает входные данные размерности, равной количеству признаков в обучающем наборе (<code>X_train.shape[1]</code>). Этот слой состоит из 38 нейронов, использует функцию активации ReLU и инициализатор весов <code>glorot_normal</code> .
2 и 3	Следующие два слоя содержат 64 и 32 нейрона соответственно, также используя ReLU и инициализатор <code>glorot_normal</code> для весов. Эти слои увеличивают абстрактность признаков, извлекаемых из данных.
Dropout	Добавлены два слоя Dropout с коэффициентом 0.3 после третьего и пятого слоёв. Это техника регуляризации, предназначенная для предотвращения переобучения путём случайного обнуления части весов нейронов во время обучения.
4 и 5	Слой с 8 нейронами продолжает обработку признаков, а последний слой состоит из одного нейрона с активацией <code>sigmoid</code> , подходящей для бинарной классификации, где результат будет представлять вероятность принадлежности к одному из двух классов.

В итоге получается модель, структура сети, включая типы слоёв, форму выходных каждого слоя и количество обучаемых параметров. Пример описания модели представлен на рисунке 3.24. Общее количество параметров в модели - 15,601, все из которых являются обучаемыми.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 38)	4992
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 32)	2080
dropout (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 8)	264
dropout_1 (Dropout)	(None, 8)	0
dense_4 (Dense)	(None, 1)	9

Total params: 15,601
Trainable params: 15,601
Non-trainable params: 0

Рисунок 3.24 - Пример архитектуры слоев нейронной сети для первой модели

Модель обучается на данных, разделённых на тренировочные и тестовые, с 500 эпохами обучения. Каждая эпоха представляет один проход по всему тренировочному набору данных. В процессе обучения модель стремится минимизировать значение функции потерь (loss) и повысить точность (accuracy) как на тренировочных, так и на тестовых данных.

Графики на рисунке 3.25 показывают динамику изменения значения функции потерь и точности на тренировочных и валидационных (тестовых) данных исходной модели. Хотя точность на тренировочных данных увеличивается со временем, значение функции потерь на валидационных данных начинает расти после улучшения, что может указывать на переобучение модели.

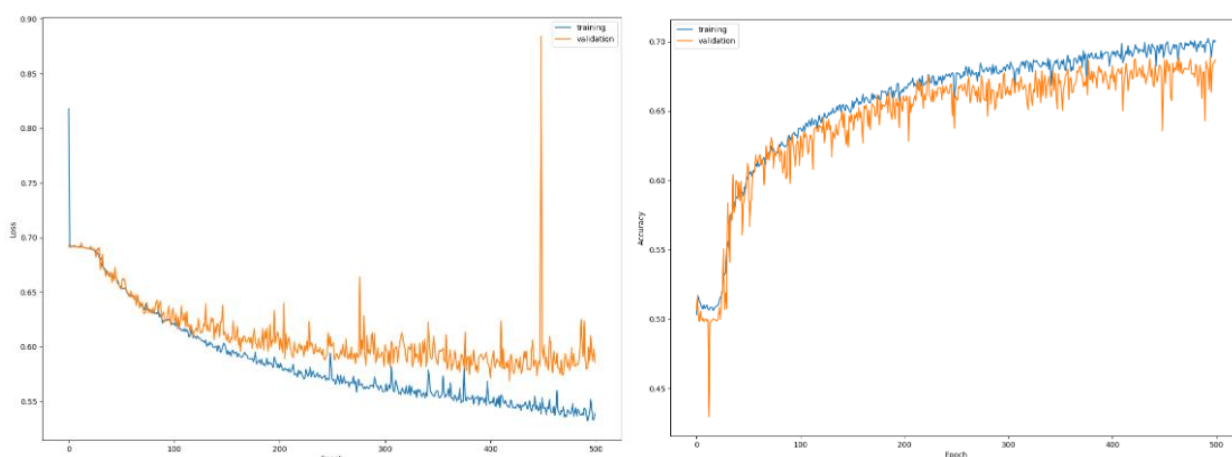


Рисунок 3.25 - График потерь и точности исходной модели

По завершении обучения проводится оценка модели на тестовом наборе данных.

Результаты Classification Report и Confusion Matrix (рисунок 3.26), представленные показывают, что модель правильно классифицирует всех пациентов без диабета (True Negatives), но не справляется с распознаванием пациентов с диабетом (False Negatives). Такой результат говорит о сильном дисбалансе в предсказаниях модели, при этом модель почти не имеет способности к корректной классификации положительного класса.

Classification Report				
	precision	recall	f1-score	support
0.0	0.49	1.00	0.66	4731
1.0	0.00	0.00	0.00	4851
Accuracy			0.49	9582
Macro avg	0.25	0.50	0.33	9582
Weighted avg	0.24	0.49	0.33	9582

Confusion Matrix

[[4731 0]
 [4851 0]]

Рисунок 3.26 - Результаты Classification Report и Confusion Matrix исходной модели

Отсутствие правильных предсказаний для класса 1 и идеальная классификация для класса 0 могут указывать на проблемы в обучении или дисбаланс классов. Модель, возможно, слишком сильно оптимизирована для идентификации одного класса в ущерб другому, что делает её непригодной для практического применения в случаях, где оба класса имеют клиническое значение.

3.4.3 Улучшение модели

Изменяем архитектуру сети используя функцию активации ELU (Exponential Linear Unit) вместо ReLU в каждом слое, что может помочь справиться с проблемой "умирающих нейронов", к которой может приводить ReLU. Функция ELU имеет некоторые преимущества, такие как более гладкая кривая активации и нелинейность, что может помочь в обучении более сложных шаблонов.

Использование инициализатора `he_normal` для весов - это также улучшение, поскольку он хорошо согласуется с активационной функцией ELU. Этот метод инициализации, разработанный для учета размера предыдущего слоя, может способствовать более эффективному обучению.

Делаем сеть глубже добавляя несколько слоев различной мощности. Это может помочь в изучении более сложных взаимосвязей в данных, но также увеличивает риск переобучения, против чего можно защититься с помощью слоев Dropout. Модель обучается и получаем следующие результаты, представленные на рисунках 3.27 и 3.28.

Classification Report				
	precision	recall	f1-score	support
0.0	0.49	1.00	0.66	4731
1.0	0.00	0.00	0.00	4851
Accuracy			0.49	9582
Macro avg	0.25	0.50	0.33	9582
Weighted avg	0.24	0.49	0.33	9582

Confusion Matrix

[[4774 0]
[4808 0]]

Рисунок 3.27 - Результаты Classification Report и Confusion Matrix улучшенной модели

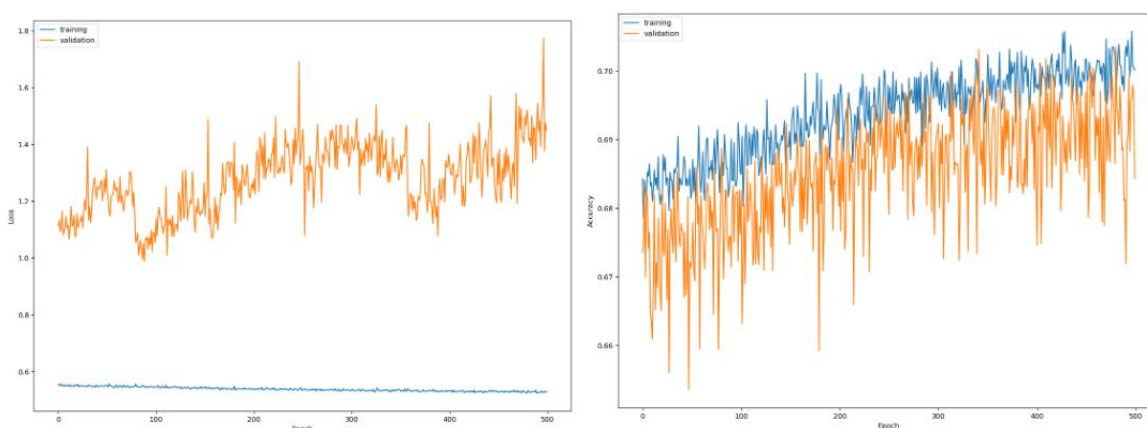


Рисунок 3.28 - График потерь и точности улучшенной модели

Из результатов и графиков можно сделать несколько важных выводов о текущем состоянии вашей модели глубокого обучения:

1) Наблюдается значительный разрыв между потерями на обучающем и валидационном наборах данных. Это может указывать на переобучение, поскольку модель хорошо работает на обучающих данных, но значительно хуже на непривычных, валидационных данных.

2) Точность на обучающем наборе постепенно увеличивается, что указывает на то, что модель обучается. Однако точность на валидационной выборке колеблется и, в конечном итоге, ухудшается, что также может быть признаком переобучения.

3) Как видно из отчета о классификации и матрицы ошибок, модель предсказывает только один класс (0.0), что свидетельствует о серьезной проблеме с балансом или предвзятостью в модели.

На данном этапе видно, что не хватает регуляризации, оптимизации гиперпараметров сети и использования ранней остановки.

3.4.4 Стратегическое улучшение модели

Далее были внесены следующие стратегические изменения в модель:

1) Модификация архитектуры, в исследование использовали различные слои и функции активации, чтобы улучшить способность модели к обобщению.

2) К регуляризации и оптимизации можно отнести включение слоев Dropout которые помогают снизить переобучение, уменьшая зависимость от отдельных нейронов. Также была адаптирована скорость обучения, используя ReduceLROnPlateau для автоматического уменьшения скорости обучения, когда производительность модели перестает улучшаться.

3) Используя стратегию кросс-валидации, тренировка модели на разных подвыборках данных, что помогает улучшить устойчивость и надежность модели.

4) В процессе обучения сохраняется история потерь и точности, что позволяет визуализировать производительность модели во времени. Это критически важно для диагностики проблем, таких как переобучение или недообучение.

После внесенных изменений, получаем следующие результаты, представленные на рисунках 3.29 и 3.30.

Classification Report				
	precision	recall	f1-score	support
0.0	0.50	1.00	0.67	4799
1.0	0.00	0.00	0.00	4783
Accuracy			0.50	9582
Macro avg	0.25	0.50	0.33	9582
Weighted avg	0.25	0.50	0.33	9582

Confusion Matrix

[[4799 0]
[4783 0]]

Рисунок 3.29 - Результаты Classification Report и Confusion Matrix модифицированной модели

Анализируя матрицу ошибок, можно сделать следующие выводы:

1) Классификационный отчет и матрица ошибок показывают, что модель всегда предсказывает один и тот же класс (класс 0), что приводит к высокой точности только для этого класса. Это свидетельствует о существенной проблеме в модели или данных.

2) Модель не смогла правильно классифицировать ни одного примера положительного класса, так как Precision и Recall для класса 1 равны 0.

3) Все положительные предсказания сосредоточены на одном классе, из-за чего модель эффективно не лучше случайного угадывания (Accuracy 50%).

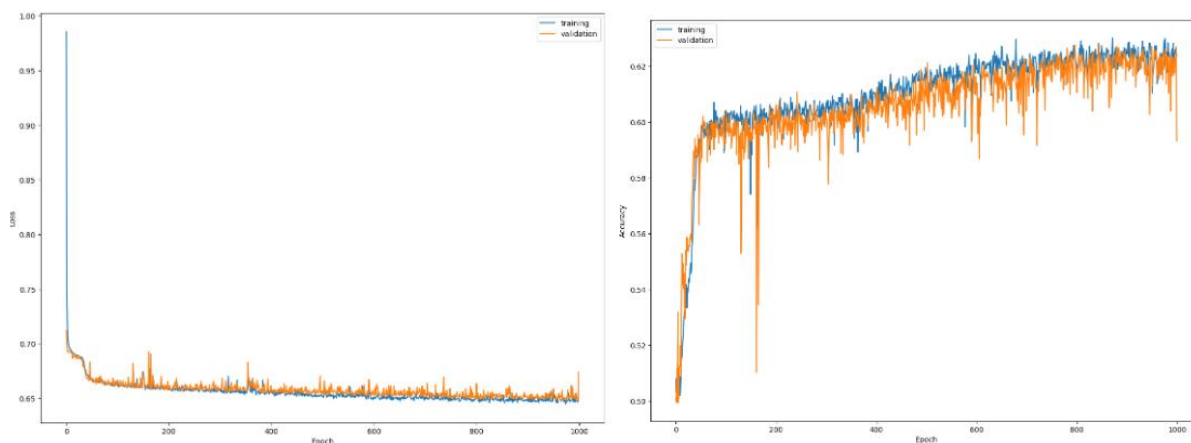


Рисунок 3.30 - График потерь и точности модифицированной модели

Проведя анализ графиков точности и потерь рисунка 3.30 можно сделать следующие выводы:

1) На основе графика потерь видно, что тренировочная и валидационная потери сходятся. Это показывает стабильность процесса обучения без значительного переобучения или недообучения. После начального спада потери стабилизируются, что является хорошим признаком.

2) На основе графика точности можно сделать вывод, что тренировочная и валидационная точности также сходятся. Однако они сходятся на довольно низком уровне около 50-55%. Это указывает на то, что модель не лучше случайного угадывания.

Для проверки реализована матрицу ошибок, которая представлена на рисунке 3.31. Данная матрица помогает визуализировать, где именно модель ошибается.

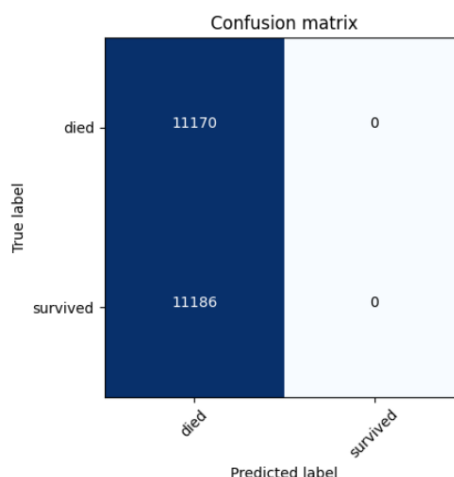


Рисунок 3.31 - Матрица ошибок модифицированной модели

В данном случае модель предсказывает, что все пациенты умерли ('died'), не предсказывая ни одного случая выживания ('survived'). Это приводит к следующим результатам:

- Ассигасу (точность) - показывает общую долю правильных предсказаний от всех предсказаний. Несмотря на то, что она может показаться

высокой из-за большого количества правильно классифицированных случаев (т.е., правильно предсказанные смерти), это заблуждение, так как модель просто всегда выбирает наиболее часто встречающийся класс.

- Precision (точность предсказания для класса) - для класса 'survived' равна 0, потому что не было ни одного, верно, предсказанного случая выживания. Точность для класса 'died' равна общему количеству, верно, предсказанных смертей, деленному на общее количество предсказанных смертей, что может быть 100%, так как все предсказанные случаи - смерти.

- Recall (полнота) - для класса 'survived' также равен 0, так как нет, верно, предсказанных случаев выживания. Полнота для 'died' равна 100%, так как все реальные случаи смертей были предсказаны как смерти.

- F1-score - Среднегармоническое точности и полноты. Для класса 'survived' это 0, так как и точность, и полнота равны 0.

На гистограмме (рисунок 3.32) показаны вероятности, с которыми модель классифицировала обучающий набор данных X_train. Гистограмма разделена на два распределения:

- Синий цвет (высота баров в левой и правой частях графика). Это вероятности для правильно классифицированных примеров. Высокие столбцы около 0 и 1 указывают на то, что модель была уверена в большинстве своих правильных предсказаний.

- Оранжевый цвет. Это вероятности для неправильно классифицированных примеров. Это распределение, скорее всего, представлено в виде более равномерного распределения по всему диапазону вероятностей, что указывает на меньшую уверенность модели в этих случаях.

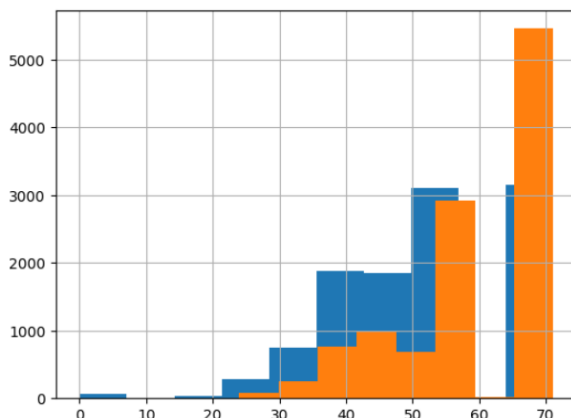


Рисунок 3.32 - Гистограмма вероятности классификации обучающей выборки

Анализируя гистограмму вероятности, можно делать следующие выводы:

- высокая уверенность модели в правильно классифицированных случаях может быть хорошим знаком, если только это не следствие переобучения (overfitting);

- наличие ошибок классификации при средних вероятностях (например, в диапазоне 40%-60%) может указывать на необходимость дальнейшей

настройки модели, чтобы улучшить её уверенность и точность в более сложных для классификации случаях.

Далее вычисляем важность признаков, используя подход к обучению и оценке модели нейронной сети с помощью KerasClassifier для получения важности признаков через метод перестановок (PermutationImportance). Этот подход позволяет оценить, как изменение порядка значений каждого признака влияет на точность модели, что может выявить наиболее значимые признаки для классификации. В итоге получаем результаты, представленные на рисунках 3.33 и 3.34.

Classification Report				
	precision	recall	f1-score	support
0.0	0.65	0.47	0.54	4799
1.0	0.58	0.74	0.65	4783
Accuracy			0.60	9582
Macro avg	0.61	0.60	0.60	9582
Weighted avg	0.61	0.60	0.60	9582

Confusion Matrix

[[2248 2551]
[1237 3546]]

Рисунок 3.33 - Матрица ошибок модели нейронной сети с помощью KerasClassifier

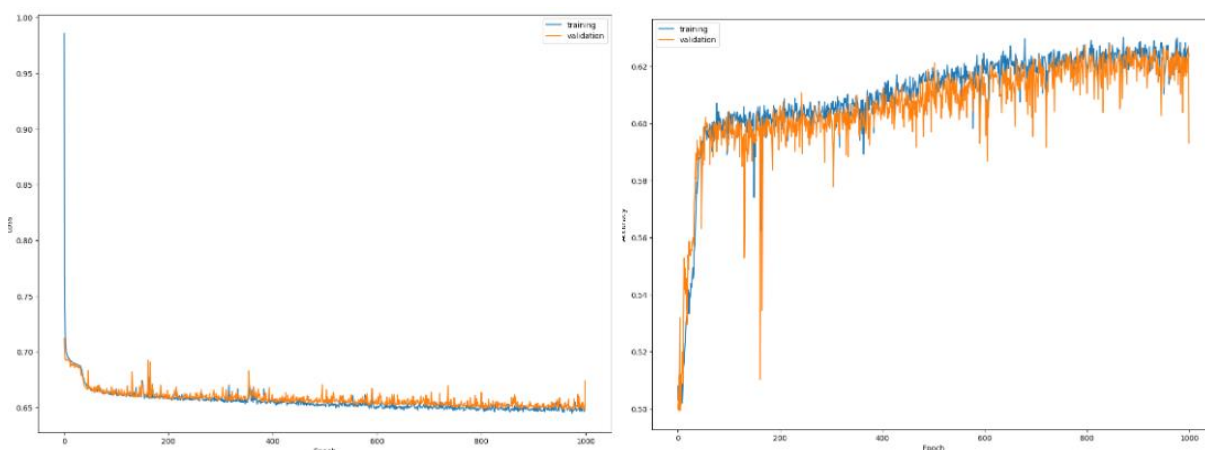


Рисунок 3.34 - График потерь и точности модели нейронной сети с помощью KerasClassifier

График потерь рисунок 3.34 показывает, что после начального скачка потери снижаются и стабилизируются на низком уровне, что является хорошим знаком. Тем не менее, есть резкий всплеск потерь на валидации около 200 эпохи, что может указывать на начало переобучения или некоторые аномалии в данных или процессе обучения.

График точности рисунок 3.34 показывает, что точность на обучающем наборе данных стабилизируется около 0.62, в то время как точность на тестовом наборе остается около 0.50. Это расхождение может свидетельствовать о переобучении, хотя разница в точности между обучением и тестированием не очень велика.

В целом, представленная модель показывает приемлемую производительность, но есть потенциал для улучшения, особенно в уменьшении переобучения и улучшении баланса между классами.

3.4.5 Создание модели с использованием *Grid Search*

Оптимизация гиперпараметров модели на основе применения метода *Grid Search* имеет ряд преимуществ:

1) Выбор активационной функции и инициализатора весов. *SELU* помогает в автоматической нормализации выходных данных слоёв, сохраняя среднее значение 0 и стандартное отклонение 1, что уменьшает проблемы с исчезающими или взрывающимися градиентами;

2) Оптимизация оптимизатора. Исследуется несколько оптимизаторов, среди которых *Nadam* показал наилучшие результаты. *Nadam* — это вариант оптимизатора *Adam* с использованием адаптивной оценки моментов;

3) Оптимизации скорости обучения. Скорость обучения является одним из ключевых параметров в процессе обучения нейронных сетей. Для выбора оптимальной скорости обучения использовались различные методы, включая *GridSearchCV*, который показал, что значение 0.0005 дает лучшие результаты.

4) Оптимизации количества эпох и размера батча (пакета). Количество эпох и размер батча также были оптимизированы с помощью *GridSearchCV*. Оптимальное количество эпох и размер батча могут варьироваться в зависимости от модели и данных.

5) Построение модели. Функция `create_model` строит модель с заданными параметрами. Она использует последовательную модель с множеством скрытых слоёв, где каждый слой сопровождается слоем *Dropout* для снижения переобучения. В конце добавляется выходной слой с сигмоидной функцией активации для бинарной классификации.

В итоге процесс использует комбинированный подход, который может включать как одновременную, так и последовательную оптимизацию различных параметров для нахождения наилучших условий работы модели.

GridSearchCV используется для систематического поиска по сетке гиперпараметров, что позволяет автоматически настроить модель для достижения наилучшей производительности. Это включает в себя определение лучших значений для числа скрытых слоёв, количества нейронов в слоях, скорости обучения и других параметров, заданных в конфигурации модели.

Модель обучается на данных, используя все комбинации параметров из сетки. При этом для каждой комбинации параметров проводится 1000 эпох обучения с размером батча 100 и валидацией на 20% данных.

После обучения строятся графики точности и функции потерь (рисунок 3.35) на тренировочных и валидационных данных. Это позволяет визуально оценить, как модель справлялась с обучением и валидацией на каждом этапе эпох.

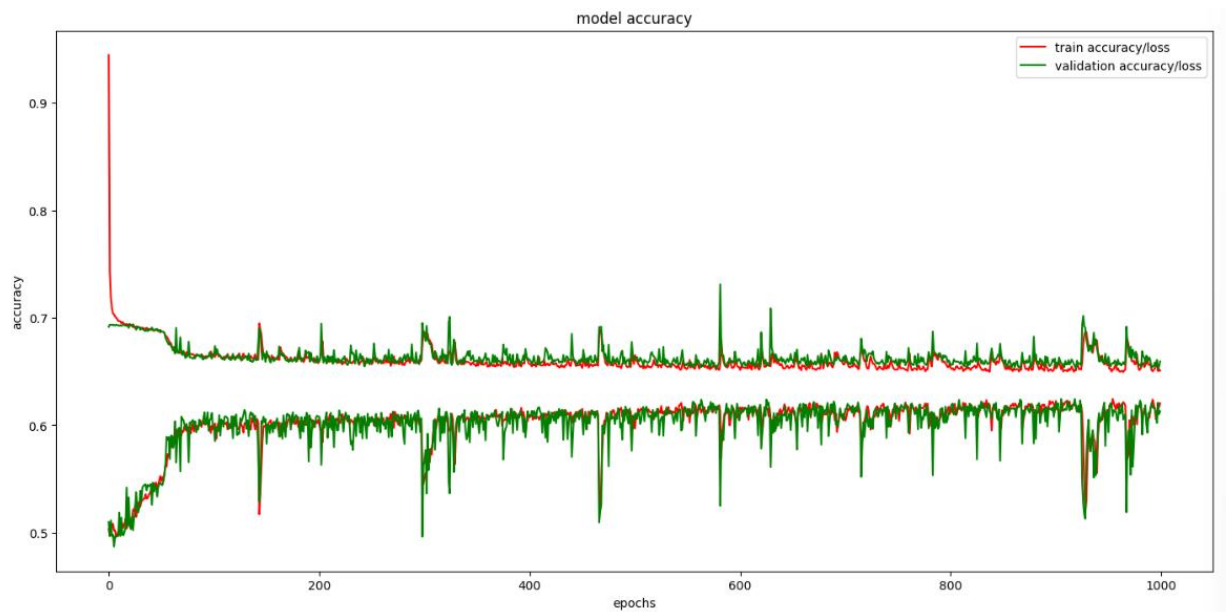


Рисунок 3.35 - Графики точности и функции потерь модели с оптимизированными гиперпараметрами

По рисунку 3.35 график точности и функции потерь можно сделать следующие выводы:

1) На начальном этапе (особенно в первые десятки эпох) точность тренировочных данных резко увеличивается, в то время как потери уменьшаются. Это типично для начальной фазы обучения, когда веса модели адаптируются к данным.

2) После начального всплеска, как точность, так и потери стабилизируются. Точность тренировочных данных остаётся относительно стабильной около значения в 0.7, в то время как точность валидации колеблется, но также находится примерно на том же уровне.

3) Заметно, что разрыв между тренировочной и валидационной точностью относительно мал, что может указывать на отсутствие значительного переобучения. Вместе с тем, стабильность потерь и точности на валидации указывает на то, что модель достигла своего предела в улучшении производительности на данных этой задачи.

4) Колебания в потерях и точности валидации говорят о чувствительности модели к изменениям в данных валидации. Это может быть связано с неоднородностью данных или их меньшей репрезентативностью по сравнению с тренировочными данными.

5) Несмотря на стабильность, наблюдается небольшая разница между тренировочной точностью и валидационной точностью, особенно заметная в конце обучения. Это может быть признаком начала переобучения, хотя и не очень значительного.

3.5 Оценка точности реализации алгоритма ансамблирования архитектур нейронных сетей LSTM и RNN для задач поддержки клинических решений

3.5.1 Исследование CNN для бинарной классификации в контексте медицинской диагностики

В ходе подготовки данных выборки были разделены на тренировочные (X_{train} , y_{train}) и тестовые (X_{test} , y_{test}) группы. Входные данные были преобразованы в формат, подходящий для CNN. Целевые переменные y_{train} и y_{test} были подготовлены с использованием метода one-hot encoding, что обеспечивает их готовность к процессу классификации. Архитектура модели CNN включает в себя два слоя свертки, за каждым из которых следует слой субдискретизации (pooling), который сокращает размерность данных, сохраняя при этом ключевые характеристики. Для обработки данных после слоев свертки применяется слой Flatten, после чего следуют полносвязные слои с промежуточными слоями Dropout для предотвращения переобучения. Выходной слой с активацией softmax классифицирует результаты на два класса.

Тренировка модели происходит на данных тренировочной выборки с использованием функции потерь категориальной перекрестной энтропии и оптимизатора Adam. В процессе тренировки проводится также валидация модели на тестовой выборке. По завершении обучения оцениваются потери и точность модели на тестовых данных. Диагностика модели включает визуализацию потерь и точности на этапах тренировки и валидации, что позволяет оценить качество обучения и способность модели к обобщению на новых данных. Кроме того, составляется отчет о классификации и строится матрица ошибок для детальной оценки точности предсказаний модели по каждому классу. Классификационный отчет CNN в таблице 3.8, матрица ошибок формулы (3.1).

Таблица 3.8 - Классификационный отчет CNN

	precision	recall	f1-score	Number of samples
False	0.66	0.82	0.73	4782
True	0.77	0.59	0.66	4800
macro avg	0.72	0.70	0.70	9582
weighted avg	0.72	0.70	0.70	9582

Матрица ошибок CNN (3.1):

$$\begin{bmatrix} [3926 & 856] \\ [1988 & 2812] \end{bmatrix} \quad (3.1)$$

График потерь, представленный на рисунке 3.36, показывает стабильное и последовательное уменьшение потерь с каждой эпохой. Это указывает на то,

что модель хорошо обучается и адаптируется к тренировочным данным, минимизируя ошибки предсказания.

На валидационных данных (оранжевый цвет видно, что потери также уменьшаются, но в определенных точках видны резкие пики, что может быть признаком переобучения или нестабильности в обучении на валидационных данных. Эти всплески могут указывать на то, что модель чувствительна к определенным особенностям или выбросам в валидационных данных.

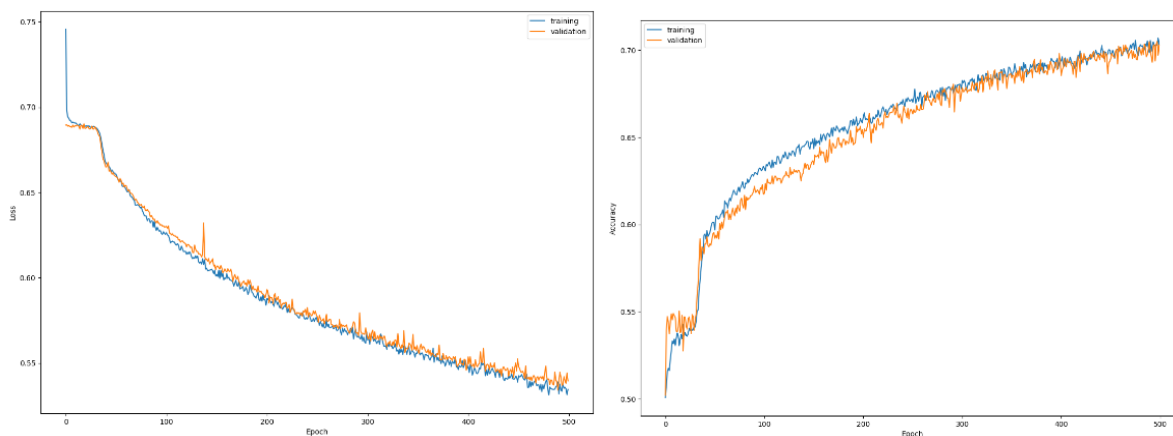


Рисунок 3.36 - Графики потерь и точности

На графике точности, представленном на рисунке 3.36 наблюдается, что тренировочные данные (синий цвет) постепенно увеличивают точность, что говорит о том, что модель все лучше распознает классы по мере обучения. Точность на валидационных данных (оранжевый цвет) также увеличивается, однако, как и на графике потерь, присутствуют колебания. Это может означать, что модель переобучается или что на валидационных данных присутствуют элементы, на которые модель реагирует нестабильно.

Общий вывод по графикам для модели CNN демонстрируют улучшение производительности модели с течением времени на тренировочных данных. Однако нестабильность на валидационных данных требует дополнительного анализа и возможной корректировки модели, чтобы уменьшить переобучение и повысить обобщающую способность модели.

3.5.2 Исследование LSTM для бинарной классификации в контексте медицинской диагностики

В качестве функции потерь применяется категориальная перекрестная энтропия, а выходной слой активируется функцией softmax, что является стандартным подходом для задач классификации. Данные для обучения (X_{train}) и тестирования (X_{test}) адаптированы под требования модели LSTM, преобразованы в последовательности по 38 временных шагов каждая, что позволяет LSTM эффективно обрабатывать временные зависимости в данных. Настройка слоя LSTM установлено 50 единиц скрытых нейронов, которые позволяют улавливать ключевые зависимости между последовательными временными точками данных.

Для дополнительного изучения нелинейных зависимостей в данных применяются множественные полносвязные слои с активацией ReLU и слои Dropout, которые помогают противодействовать переобучению путем случайного исключения части нейронов во время обучения. Модель компилируется с использованием оптимизатора Adam с заданной скоростью обучения 0.001. Обучение проводится в течение 500 эпох с размером партии 512, включая процедуры валидации для мониторинга и улучшения обобщающей способности модели. По завершении тренировочного процесса, оценка производительности на тестовой выборке выявляет ключевые метрики, такие как точность, полнота и F1-мера, что позволяет оценить эффективность модели в задачах классификации.

График потерь представлен на рисунке 3.36. Потери на обучающем и валидационном наборах уменьшаются по мере обучения, что указывает на эффективное обучение модели. Всплески потерь на валидации свидетельствуют о моментах нестабильности в обобщающей способности модели.

Классификационный отчет LSTM в таблице 3.9, матрица ошибок в формуле (3.2)

Таблица 3.9 - Классификационный отчет по модели LSTM

	precision	recall	f1-score	Number of samples
False	0.67	0.74	0.70	4782
True	0.71	0.64	0.67	4800
macro avg	0.69	0.69	0.69	9582
weighted avg	0.69	0.69	0.69	9582

Матрица ошибок LSTM (3.2):

$$\begin{bmatrix} [3521 & 1261] \\ [1725 & 3075] \end{bmatrix} \quad (3.2)$$

График точности, представленный на рисунке 3.37 демонстрирует что и на обучающем, так и на валидационном наборах увеличивается со временем, стабилизируясь около 65%. Этот постепенный рост и стабилизация указывают на то, что модель извлекает пользу из долгосрочного обучения без значительного переобучения, так как точность валидации тесно следует за точностью обучения.

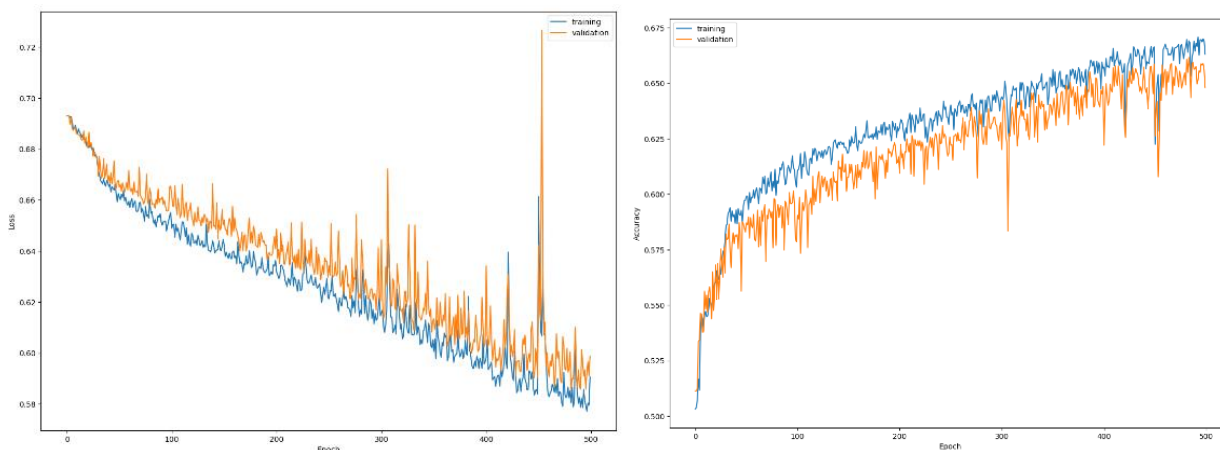


Рисунок 3.37 - Графики потерь и точности модели LSTM

В целом, модель LSTM показывает разумное обучение и способность к обобщению на задаче классификации, с общей точностью 65% на валидационном наборе, что указывает на её способность эффективно различать два класса, хотя и с возможностью улучшения, особенно с точки зрения снижения ошибок классификации, как показывают результаты матрицы ошибок.

3.5.3 Исследование RNN для бинарной классификации в контексте медицинской диагностики

Модель рекуррентной нейронной сети (RNN), реализованная в данном исследовании, базируется на слое SimpleRNN с параметром `cell_size`, установленным в 50. Это означает, что на каждом временном шаге данных, каждый из которых представляет собой вектор заданной размерности, входная информация обрабатывается 50 рекуррентными нейронами. Слой SimpleRNN эффективно обрабатывает последовательные данные, поддерживая временные зависимости между последовательными точками данных, что критично для анализа временных рядов или последовательностей данных.

В архитектуру модели также интегрированы несколько полносвязных слоев (Dense), которые последовательно сокращают размерность данных с 64 до 2 нейронов, используя функцию активации ReLU для промежуточных слоёв и softmax для выходного слоя, что позволяет классифицировать результаты на две категории. Оптимизация процесса обучения достигается с помощью алгоритма Adam с параметром скорости обучения 0.001, что обеспечивает эффективное и динамичное обновление весов в сети посредством минимизации категориальной перекрёстной энтропии, часто применяемой в задачах многоклассовой классификации. Процесс обучения и валидации модели демонстрируется через графики потерь и точности, что позволяет визуально оценить успех обучения и способность модели генерализовать знания на новых данных.

График потерь на рисунке 3.38 показывает, что потери уменьшаются с каждой эпохой как на тренировочных, так и на валидационных данных. На тренировочных данных потери плавно уменьшаются, что свидетельствует о

хорошей сходимости модели. На валидационных данных потери также уменьшаются, но с некоторыми колебаниями, что может указывать на переобучение.

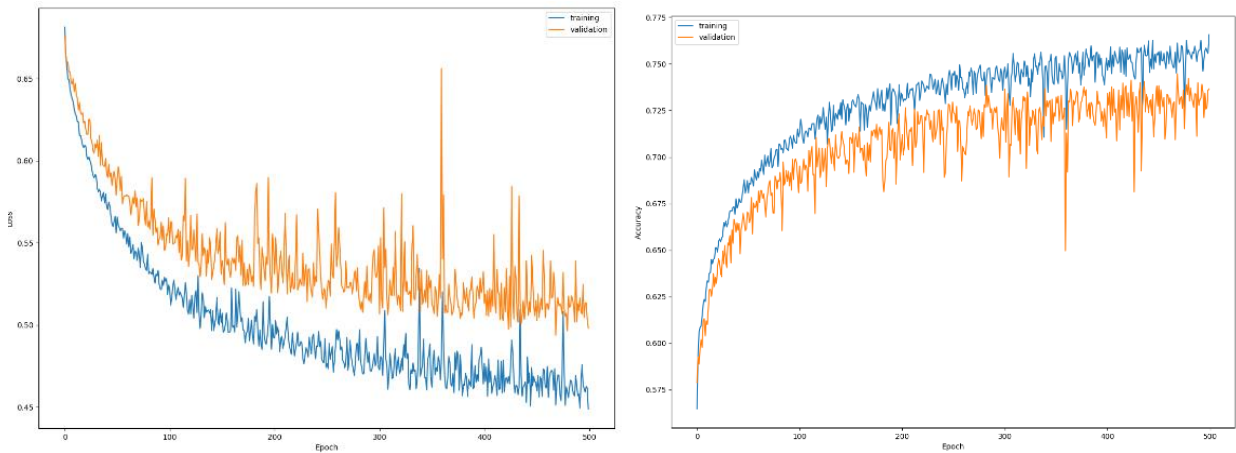


Рисунок 3.38 - Графики потерь и точности модели RNN

График точности, представленный на рисунке 3.38 отражает точность на тренировочных данных постепенно увеличивается, что хорошо для обучения модели. Однако точность на валидационных данных, хотя и увеличивается, показывает некоторую нестабильность, которая может быть признаком переобучения или недостаточной обобщающей способности модели.

Классификационный отчет модели RNN представлен в таблице 3.10.

Таблица 3.10 - Классификационный отчет модели RNN

	precision	recall	f1-score	Number of samples
False	0.70	0.81	0.75	4782
True	0.78	0.66	0.71	4800
macro avg	0.74	0.73	0.73	9582
weighted avg	0.74	0.73	0.73	9582

Матрица ошибок RNN (3.3):

$$\begin{bmatrix} [3884 & 898] \\ [1646 & 3154] \end{bmatrix} \quad (3.3)$$

В результате классификации модель достигла общей точности 74% на тестовом наборе данных. По матрице ошибок видно, что модель лучше распознает класс False (истинно отрицательные результаты), чем класс True (истинно положительные результаты). Это может указывать на то, что модель лучше распознает отсутствие характеристик, специфичных для одного из классов. Отчет о классификации дает более детальное представление о точности, полноте и F1-оценке для каждого класса, подчеркивая, как модель работает с каждым из классов в отдельности.

3.5.4 Исследование эффективности различных ансамблевых методов классификации для прогнозирования наличия диабета

В рамках экспериментального исследования применяется комплексный подход, включающий следующие ключевые этапы:

1) Пропущенные значения в признаках данных заполняются медианными значениями для каждого признака, чтобы минимизировать влияние отсутствующих данных на результаты анализа.

2) Данные разделяются на обучающую и тестовую выборки, где тестовая выборка составляет 30% от общего объема данных. Используется фиксированный параметр `random_state` для гарантии воспроизводимости результатов.

3) Используются разнообразные алгоритмы, включая Decision Tree, AdaBoost, Random Forest, Extra Trees и Gradient Boosting, для оценки их способности к классификации.

4) Проводится пятикратная кросс-валидация для каждого алгоритма на обучающей выборке, чтобы оценить устойчивость и точность моделей (рисунок 3.39).

5) Рассчитываются средние значения точности и стандартные отклонения по результатам кросс-валидации, что позволяет оценить эффективность каждого алгоритма.

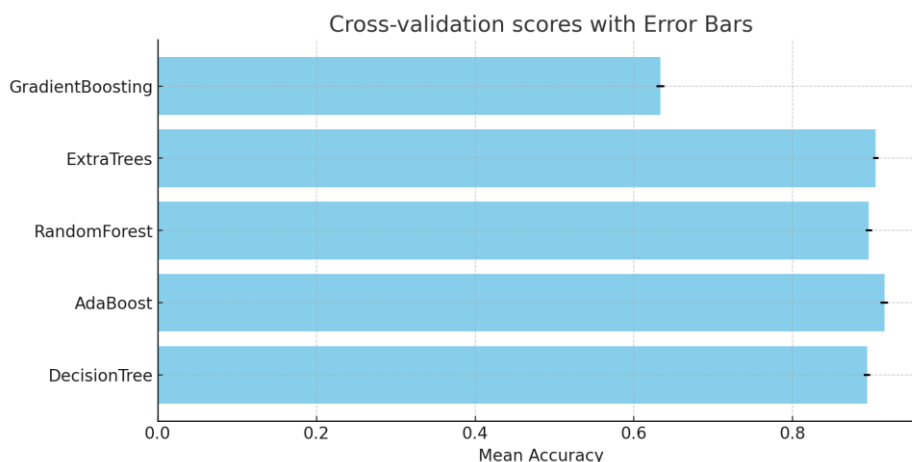


Рисунок 3.39 - Результаты кросс-валидации алгоритмов машинного обучения

Таблица результатов кросс-валидации представлена на рисунке 3.40.

	CrossValMeans	CrossValerrors	Algorithm
0	0.893585	0.004231	DecisionTree
1	0.915459	0.004543	AdaBoost
2	0.896046	0.004023	RandomForest
3	0.904589	0.003482	ExtraTrees
4	0.633432	0.004944	GradientBoosting

Рисунок 3.40 - Таблица результатов кросс-валидации

Таблица результатов кросс-валидации показывает, что алгоритм AdaBoost демонстрирует наивысшую среднюю точность (0.915459) среди рассмотренных алгоритмов, что делает его предпочтительным выбором для дальнейшего использования в проекте. Напротив, Gradient Boosting показал значительно ниже среднюю точность (0.633432), что может указывать на его недостаточную адаптацию к данным или переобучение.

3.5.5 Комплексный подход к оптимизации и оценке алгоритмов

На данном промежутке исследования реализуется комплексный подход к оптимизации и оценке ансамблевых алгоритмов машинного обучения.

Первым этапом экспериментального исследования выступает оценка различных моделей. Используя начальные данные, провели оценку производительности различных алгоритмов с помощью кросс-валидации, измеряя такие метрики, как минимальная, средняя и максимальная точность. Это помогло выявить потенциал каждой модели в контексте вашего исследования.

Первым этапом экспериментального исследования является использование методики GridSearchCV. Данный инструмент используется из библиотеки scikit-learn, предназначенный для автоматизированного подбора параметров моделей машинного обучения. GridSearchCV проводит исчерпывающий поиск по заданной сетке гиперпараметров модели и оценивает каждую комбинацию с помощью кросс-валидации для нахождения наилучшей конфигурации.

В целом использование техники Grid Search позволяет систематически рассматривать множество комбинаций параметров, заданных в формате сетки(grid). Такой подход помогает определить какие параметры дают наилучший результат в соответствии с заданной метрикой производительности. Grid Search автоматически применяет кросс-валидацию, тестируя каждую комбинацию параметров на различных подмножествах данных для оценки их эффективности.

В экспериментальном исследовании был использован класс EstimatorSelectionHelper, который основан на методе GridSearchCV и предназначен для определения наилучших параметров различных ансамблевых алгоритмов. Определяется сетка параметров для каждого классификатора, таких как глубина дерева для DecisionTree и RandomForest, скорость обучения и количество оценщиков для AdaBoost, и т.д. GridSearchCV систематически обучает модели с каждым набором параметров на основе кросс-валидации, и возвращает модели с наилучшими параметрами в зависимости от выбранной метрики (в данном случае, точности).

Выведенная таблица (рисунок 3.41) показывает сводку результатов выполнения GridSearchCV для нескольких классификаторов с использованием класса EstimatorSelectionHelper. В таблице перечислены классификаторы (в столбце estimator) вместе с их минимальными, средними, максимальными оценками и стандартными отклонениями, а также значениями параметров, при которых были достигнуты эти оценки.

	estimator	min_score	mean_score	max_score	std_score	learning_rate	max_depth	max_features	n_estimators
41	GradientBoostingClassifier	0.910952	0.914678	0.917058	0.00266821	NaN	9	auto	500
37	GradientBoostingClassifier	0.874507	0.876104	0.877888	0.00138689	NaN	7	auto	500
39	GradientBoostingClassifier	0.866804	0.870875	0.874319	0.0030996	NaN	9	sqrt	500
40	GradientBoostingClassifier	0.861732	0.866491	0.873943	0.00533642	NaN	9	auto	200
36	GradientBoostingClassifier	0.799361	0.809569	0.81796	0.00770064	NaN	7	auto	200
38	GradientBoostingClassifier	0.797577	0.802774	0.812324	0.00676151	NaN	9	sqrt	200
33	GradientBoostingClassifier	0.787808	0.793037	0.803119	0.00713072	NaN	5	auto	500
35	GradientBoostingClassifier	0.792316	0.79504	0.798892	0.00280016	NaN	7	sqrt	500
32	GradientBoostingClassifier	0.714071	0.723402	0.739339	0.011324	NaN	5	auto	200
34	GradientBoostingClassifier	0.724497	0.730603	0.739245	0.00628198	NaN	7	sqrt	200
31	GradientBoostingClassifier	0.70928	0.715386	0.71858	0.00431883	NaN	5	sqrt	500
29	GradientBoostingClassifier	0.677156	0.687927	0.694345	0.00766274	NaN	3	auto	500
30	GradientBoostingClassifier	0.66269	0.671301	0.681007	0.00751782	NaN	5	sqrt	200
17	RandomForestClassifier	0.664475	0.669798	0.67894	0.00649393	NaN	9	NaN	500
16	RandomForestClassifier	0.661845	0.668671	0.678753	0.00727648	NaN	9	NaN	200
28	GradientBoostingClassifier	0.646158	0.651919	0.659215	0.00543959	NaN	3	auto	200
27	GradientBoostingClassifier	0.631881	0.643653	0.652452	0.0086575	NaN	3	sqrt	500
24	ExtraTreesClassifier	0.639019	0.643966	0.652076	0.00578036	NaN	9	NaN	200
25	ExtraTreesClassifier	0.641368	0.645062	0.651418	0.00451417	NaN	9	NaN	500
15	RandomForestClassifier	0.630378	0.638111	0.649258	0.00807682	NaN	7	NaN	500
14	RandomForestClassifier	0.634135	0.639959	0.649164	0.00658507	NaN	7	NaN	200
3	DecisionTreeClassifier	0.63282	0.640522	0.645501	0.00552365	NaN	9	NaN	NaN
26	GradientBoostingClassifier	0.626245	0.630315	0.636295	0.00431974	NaN	3	sqrt	200
23	ExtraTreesClassifier	0.611497	0.619701	0.633853	0.0100495	NaN	7	NaN	500
22	ExtraTreesClassifier	0.60774	0.615662	0.628687	0.0092818	NaN	7	NaN	200
13	RandomForestClassifier	0.609243	0.616601	0.628311	0.00837079	NaN	5	NaN	500
12	RandomForestClassifier	0.605298	0.615787	0.627841	0.00926975	NaN	5	NaN	200
2	DecisionTreeClassifier	0.602574	0.609807	0.618354	0.00650916	NaN	7	NaN	NaN
9	AdaBoostClassifier	0.602386	0.60702	0.614785	0.00552489	0.1	NaN	NaN	500
10	RandomForestClassifier	0.583693	0.596374	0.611028	0.011246	NaN	3	NaN	200
21	ExtraTreesClassifier	0.590269	0.597564	0.610652	0.00927483	NaN	5	NaN	500

Рисунок 3.41 - Сводка результатов выполнения GridSearchCV

GradientBoostingClassifier занимает верхние строки таблицы, показывая наилучшие результаты максимальной точности (max_score). Это говорит о том, что этот классификатор с определёнными настройками гиперпараметров (max_depth, max_features, n_estimators) показал лучшие результаты среди протестированных моделей.

ExtraTreesClassifier и RandomForestClassifier также присутствуют в таблице, но их результаты располагаются ниже, чем результаты GradientBoostingClassifier.

DecisionTreeClassifier и AdaBoostClassifier замыкают таблицу с их лучшими результатами.

Сводка результатов, отсортированная по максимальной точности, позволяет идентифицировать наиболее перспективные модели и их параметры для дальнейшего углублённого анализа или непосредственного использования в предсказательном моделировании.

Сортировка по `max_score` полезна, когда интерес представляет пиковая производительность модели, но при выборе финальной модели также важно учитывать среднюю точность (`mean_score`) и стабильность модели (насколько мала `std_score`), чтобы гарантировать, что модель будет хорошо работать в различных условиях и на разных данных.

На третьем этапе экспериментального исследования происходит финальная оценка ансамблевых алгоритмов машинного обучения, которые были оптимизированы на предыдущем этапе с использованием Grid Search. Данный этап критичен, поскольку он предоставляет полное понимание производительности моделей на данных, которые они не видели во время обучения, тем самым проверяя их обобщающую способность и практическую применимость.

Каждая модель, настроенная с лучшими параметрами, обучается на полном обучающем наборе данных. Далее модели тестируются на отдельном тестовом наборе данных, чтобы оценить их производительность в условиях, максимально приближенных к реальным. Следующим шагом будем измерение ключевых метрик качества квалификации. После чего происходит анализ полученных результатов, путем сравнения метрик между моделями для выбора лучшей. Так же анализируются отчеты классификации и матрицы ошибок для глубокого понимания поведения моделей.

В данном экспериментальном исследовании была получена таблица с результатами оценки различных классификаторов машинного обучения, представленная на рисунке 3.42. Для каждой модели были рассчитаны метрики точности на обучающем и тестовом наборах, а также метрики `precision`, `recall` и `F1 score` на тестовом наборе данных.

	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
Model					
DecisionTree	0.658347	0.641202	0.684453	0.526458	0.595148
AdaBoost	0.590222	0.585055	0.623501	0.433333	0.511308
RandomForest	0.688853	0.671467	0.696948	0.608958	0.649989
ExtraTrees	0.658481	0.643394	0.671887	0.563125	0.612717
GradientBoosting	0.958356	0.916301	0.970132	0.859375	0.911401

Рисунок 3.42- Результаты оценки различных классификаторов машинного обучения

По рисунку 3.42 видно, что модель GradientBoostingClassifier показала значительно лучшую производительность по всем метрикам по сравнению с другими моделями, включая впечатляющую точность на тестовом наборе данных (Test Accuracy) и высокие показатели precision и recall.

Вывод результатов classification_report и confusion_matrix для каждой модели дополнительно подтверждает эти результаты, демонстрируя подробные показатели для каждого класса (False и True), а также общую точность (accuracy) моделей. Подробное описание каждой из моделей представлены в приложение М.

Визуализация, представленная на рисунке 3.43, это график сравнения различных метрик качества для выбранных алгоритмов машинного обучения.

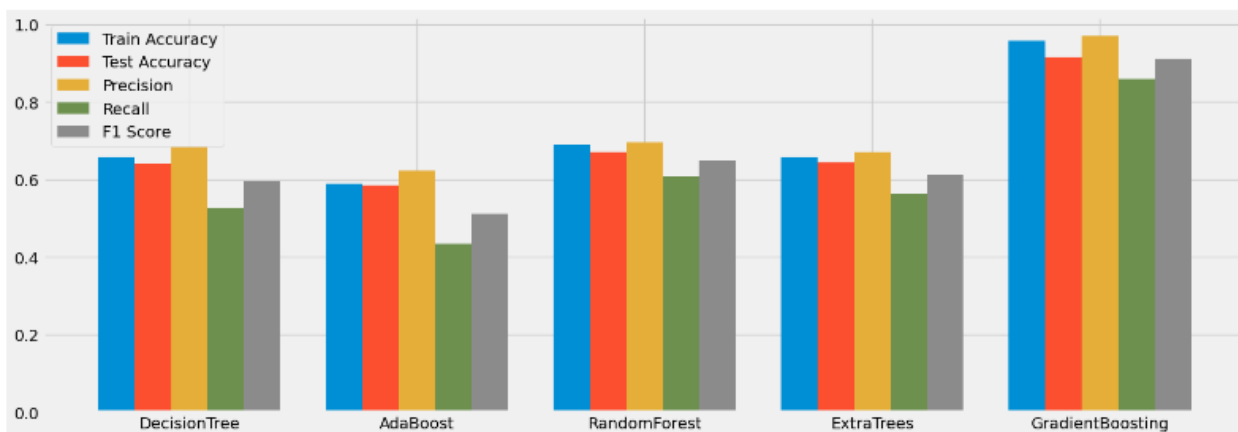


Рисунок 3.43 - График сравнения различных метрик качества для выбранных алгоритмов машинного обучения

График в стиле fivethirtyeight с использованием столбчатых диаграмм отображает пять различных метрик: точность на обучающем наборе данных (Train Accuracy), точность на тестовом наборе данных (Test Accuracy), precision, recall и F1 score для каждой из моделей: DecisionTree, AdaBoost, RandomForest, ExtraTrees и GradientBoosting.

3.5.6 Ансамблевое обучение LSTM и RNN моделей

Ансамблевое обучение с использованием метода Deep Learning Ensemble позволяет объединить предсказательную силу нескольких нейронных сетей, каждая из которых обучается на различных порциях данных. В данном случае ансамбль состоит из двух моделей: одна основана на рекуррентной нейронной сети LSTM, а другая - на RNN. Для интеграции моделей в ансамбль используется класс KerasMember, который позволяет задать конкретные модели, а также данные, используемые для их обучения и валидации.

Для координации работы ансамбля применяется класс DirichletEnsemble, который использует метод Дирихле для определения оптимальных весов каждой модели, учитывая их индивидуальные показатели производительности, измеряемые через ROC AUC. Метод describe()

предоставляет детальную информацию о распределении весов и эффективности каждой модели в ансамбле. После определения оптимальных весов предсказания моделей комбинируются с использованием взвешенного голосования для окончательного прогноза. Эффективность ансамбля оценивается с помощью стандартных метрик классификации, таких как матрица ошибок и отчет по классификации, что позволяет оценить точность и надежность полученных результатов.

График потерь (рисунок 3.44) показывает уменьшение значения функции потерь как для обучающего, так и для валидационного набора данных по мере продвижения эпох обучения. Стабильное уменьшение потерь на обучающем наборе и их колебания на валидационном наборе могут указывать на переобучение или недообучение в зависимости от характера изменений.

График точности отражает изменение точности модели на обучающем и валидационном наборах. Устойчивое увеличение точности на обучающем наборе и его волатильность на валидационном могут служить индикатором адекватности выбранной модели и её настроек.

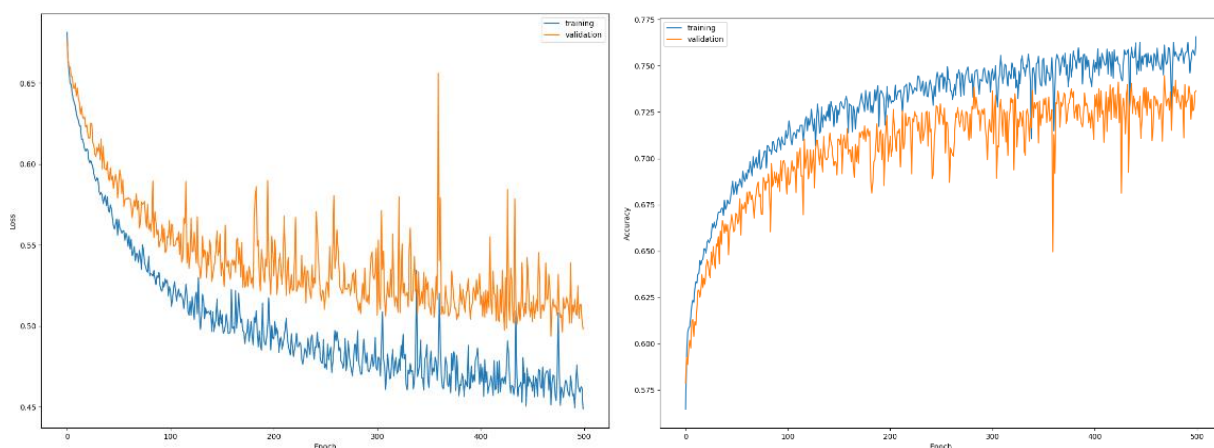


Рисунок 3.44 - График потерь и точности ансамблевого обучения LSTM и RNN моделей

Исходя из представленных выше результатов можно сделать вывод, что RNN превзошла все другие модели, основанные на глубоких нейронных сетях, которые были построены. По сравнению с другими моделями, RNN обладает самыми высокими средними показателями точности, отзывчивости и f1-балла, превышающими 0,74. В модели RNN используется четыре скрытых слоя, чтобы сохранить простоту и повысить надежность при переналадке. В скрытых слоях используют функцию активации ELU наряду со стандартными инициализаторами ядра. Для прогнозирования используется функция активации сигмовидной формы на выходном уровне. Этот подход соответствует научной методологии, но при этом остается доступным в понимании.

3.6 Архитектура системы поддержки принятия клинических решений

СППКР предназначена для помощи медицинским специалистам в процессе диагностики, выбора методов лечения и управления пациентами. Такие системы интегрируют данные пациента, медицинские знания, алгоритмы анализа и рекомендации, предоставляя клиницистам обоснованную информацию для принятия решений. СППКР позволяют повысить эффективность медицинской диагностики, выдавать индивидуализированные рекомендации по лечению, предупреждать о потенциальных рисках и осложнениях, стандартизировать медицинскую помощь на основе доказательной медицины и др. СППКР становится незаменимым помощником, который, обеспечивая поддержку на всех этапах клинического процесса, улучшает качество и безопасность медицинской помощи, повышает удовлетворенность пациентов и способствует развитию доказательной медицины.

Для создания структурированной, гибкой и надежной основы, обеспечивающей эффективную работу всех компонентов СППКР необходимо построение модели ее архитектуры. Архитектура СППКР позволяет интегрировать данные, процессы и алгоритмы в единое целое.

Хорошо продуманная архитектура позволяет адаптировать СППКР к изменениям и расширять её возможности. Это особенно важно, так как требования к системам могут изменяться с появлением новых технологий, увеличением объемов данных или изменением клинических протоколов. Гибкая архитектура позволяет быстро вносить изменения и добавлять новые функции.

Предложенные в рамках диссертации алгоритмы поддержки принятия клинических решений являются основными ключевыми компонентами модели архитектуры СППКР.

Модель архитектуры СППКР, интегрирующая алгоритмы поддержки принятия клинических решений, демонстрирует ключевые уровни, модули и их взаимодействия для обеспечения эффективной обработки медицинских данных и поддержки принятия решений (рисунок 3.45).

Модель архитектуры СППКР включает несколько основных слоев:

- слой безопасности;
- слой бизнес-логики;
- слой представления;
- слой сбора данных;
- слой обработки данных;
- слой обучения моделей (model training layer);
- модуль интерпретации результатов.

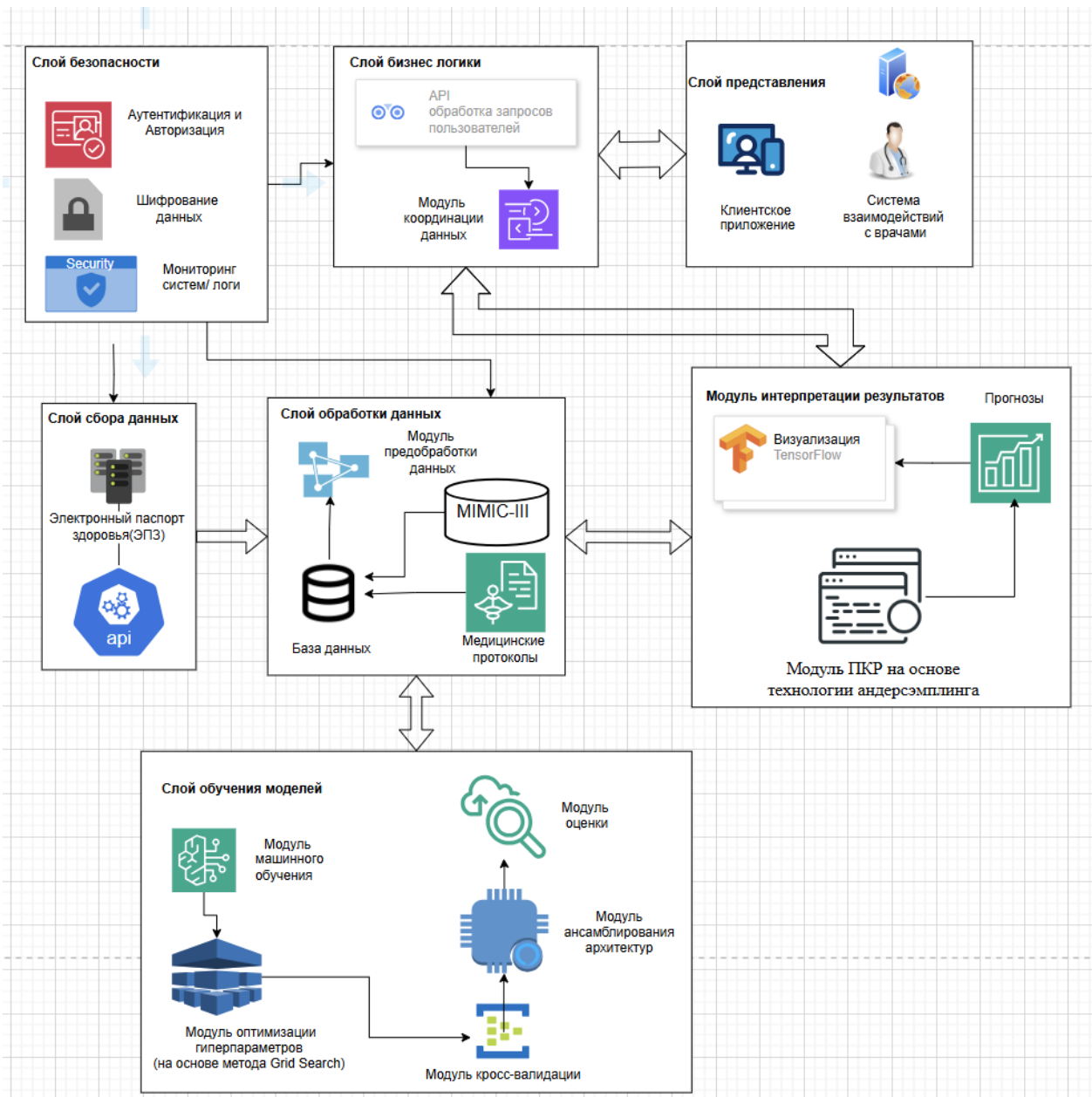


Рисунок 3.45 – Архитектура СППКР

Слой безопасности предназначен для обеспечения защиты медицинских данных и общего мониторинга системы. Он включает модули аутентификации и авторизации, шифрование данных, мониторинг систем и логирование. Нарушение безопасности может привести к утечке медицинских данных, что не только нарушает законы о защите, но и угрожает приватности пациентов. Шифрование данных гарантирует защиту данных как при передаче, так и при хранении, обеспечивая их целостность и конфиденциальность. Мониторинг систем и логирование, позволяет отслеживать производительность системы и логировать действия для аудита и безопасности, что способствует оперативному обнаружению и устранению потенциальных угроз. В рамках данного исследования использовались только обезличенные медицинские данные. Вся остальная медицинская информация проходила шифрование методом md5 (шифрование в хэш-функции). Так же

при построение алгоритмов и обработке данных использовался принцип минимизации данных. Собирается и храниться необходимое количество данных только для конкретного исследования. Как видно из рисунка 3.45 слой безопасности взаимодействует непосредственно со слоями сбор и обработка данных, а так же слоем бизнес логики.

Слой бизнес-логики обеспечивает обработку пользовательских запросов и координацию потоков данных в системе. Основные компоненты включают API для обработки пользовательских запросов и модуль координации данных. REST API, обеспечивающий взаимодействие между интерфейсом пользователя и внутренними компонентами системы, что позволяет эффективно управлять запросами и доступом к данным.

Слой представления отвечает за пользовательский интерфейс, с которым непосредственно работают пользователи СППКР. Как видно из рисунка 3.45 данный слой включает 2 компонента: клиентское приложение и система взаимодействия с врачами. В СППКР предусмотрено применение нескольких оконных приложений, тем самым исключаются некоторые проблемы с безопасностью домена. Интерфейс слоя представления взаимодействует с бизнес-логикой для получения данных, что обеспечивает пользователю доступ к результатам и возможность их интерпретации.

Слой сбора данных выполняет функции интеграции и извлечения данных из различных источников. Основными компонентами данного слоя являются электронный паспорт здоровья (ЭПЗ) и API внешних источников. ЭПЗ является источником медицинских данных, которые используются для дальнейшего анализа и принятия решений. API предусмотрено на данном этапе для обеспечения подключения к внешним системам и извлечение данных, необходимых для анализа и предсказаний. Собранные данные передаются на слой обработки данных для выполнения предобработки и последующего анализа.

Слой обработки данных отвечает за предобработку, нормализацию и хранение данных. Основные компоненты включают модуль предобработки данных, саму БД и медицинские протоколы. Модуль предобработки осуществляет очистку данных, нормализацию, удаление выбросов и заполнение пропусков, что улучшает качество данных перед их анализом. В нем заложены описанные ранее принципы обработки данных EDA. БД хранит очищенные и обработанные данные, а также промежуточных результатов для последующего использования. На данном так же подразумевается сохранение моделей обучения и оптимальных гиперпараметров для тех или иных данных. Медицинские протоколы в данном случае подразумевают как сами медицинские рекомендации/протоколы, так же хранение МКБ и возможность масштабирования для других систем классификации. Подготовленные данные затем передаются на слой обучения моделей для дальнейшего анализа.

Слой обучения моделей отвечает за создание и оптимизацию моделей машинного обучения. Данный слой включает следующие модули: модуль машинного обучения, модуль оптимизации гиперпараметров, модуль кросс-

валидации и модуль оценки. Обучение моделей предобработанных данных основано на применении алгоритмов глубокого обучения (CNN, LSTM, RNN), позволяющих реализовать задачи классификации и прогнозирования. В модуле оптимизации осуществляется применение методов, таких как Grid Search, для оптимизации параметров моделей и повышения их точности. В модуле кросс-валидации происходит оценка качества моделей для обеспечения их надежности и предотвращения переобучения. Модуль оценки моделей служит для анализа производительности моделей, включающий проверку их точности и способности к определению верных результатов. После обучения модели используются для предсказания результатов, которые передаются в бизнес-логику и слой представления.

Модуль интерпретации результатов отвечает за визуализацию и объяснение предсказаний моделей. Визуализация TensorFlow, обеспечивает наглядное представление предсказаний моделей, делая результаты понятными и доступными для конечных пользователей. В модуль прогнозы включаются результаты моделей, подаваемые на слой представления для дальнейшего использования медицинским персоналом.

В целом данные проходят через все уровни системы, начиная с их сбора в слое сбора данных, предобработки в слое обработки данных, обучения и анализа в слое обучения моделей. Далее предсказания передаются в слой бизнес-логики и слой представления для визуализации и интерпретации результатов. слой безопасности интегрируется на всех этапах обработки данных, обеспечивая защиту и целостность системы.

Таким образом, архитектура системы представляет собой комплексную иерархическую структуру, в которой каждый слой выполняет критически важные функции для обеспечения эффективного функционирования СППКР. Данная архитектурная схема четко отображает взаимодействие всех уровней и модулей, что позволяет обеспечить целостность, безопасность и надежность процессов обработки данных, их анализа и предоставления результатов для поддержки принятия клинических решений.

3.7 Выводы по третьему разделу

На основе экспериментальных исследований алгоритмов интеллектуальной поддержки принятия клинических решений были сделаны следующие выводы:

1) Gradient Boosting демонстрирует высокую производительность по всем ключевым метрикам точность тренировочной и тестовой выборок, точность (precision), полнота (recall) и F1-метрика. Это указывает на то, что алгоритм хорошо справляется с задачей, минимизируя как ошибки первого, так и второго рода. Gradient Boosting является предпочтительным кандидатом для дальнейшего использования и реализации в системе поддержки принятия решений. Это обусловлено его высокой точностью, обобщающей способностью и относительной устойчивостью к изменениям в данных.

2) Random Forest и Extra Trees также показывают сильные результаты, но несколько уступают Gradient Boosting, особенно в тестовой точности и F1-метрике. Это может свидетельствовать о том, что Gradient Boosting лучше адаптирован к особенностям данных. Decision Tree и AdaBoost имеют сравнительно низкую производительность, особенно AdaBoost, который показывает заметное ухудшение показателей на тестовой выборке по сравнению с тренировочной, что может быть признаком переобучения.

3) Сравнение тренировочной и тестовой точности для Gradient Boosting показывает меньшую разницу по сравнению с другими моделями, что говорит о хорошей обобщающей способности модели.

Учитывая полученные экспериментальным путём данные, можно сделать вывод о том, что ансамблевые методы, особенно Gradient Boosting, могут быть более эффективными для этой задачи бинарной классификации. Важно отметить, что андерсэмплинг, по-видимому, не оказал существенного влияния на результаты. Это может говорить о том, что первоначальный дисбаланс классов не имел сильного воздействия на производительность моделей, или что алгоритмы достаточно устойчивы к несбалансированным данным.

Для улучшения моделей можно рассмотреть возможность дальнейшей настройки гиперпараметров, особенно для моделей с высокой разницей между тренировочной и тестовой точностью, таких как AdaBoost. А также провести дополнительные исследования влияния различных признаков на модель и возможностей feature engineering для улучшения результатов.

ЗАКЛЮЧЕНИЕ

Процесс поддержки принятия клинических решений является систематизированным подходом, направленный на помощь врачам и медицинскому персоналу в принятии оптимальных решений на основе анализа данных о пациентах. Системы ППКР используются для улучшения качества лечения, снижения ошибок и повышения эффективности медицинской помощи.

В данной диссертационной работе проведен всесторонний анализ и разработка интеллектуальной системы поддержки принятия клинических решений, направленной на оптимизацию диагностических процессов в области эндокринологии и диабетологии.

В первом разделе исследованы существующие подходы к диагностике, выявлены ключевые проблемы и преимущества применения систем поддержки принятия клинических решений.

Второй раздел сосредоточился на алгоритмах и моделях, поддерживающих принятие клинических решений. Разработанные концептуальная модель и алгоритмы, продемонстрировали свою практическую значимость. Ансамблирование различных архитектур нейронных сетей показало свою способность повышать точность диагностических прогнозов, что имеет ключевое значение для улучшения качества медицинского обслуживания.

Экспериментальное исследование продемонстрировало превосходство предложенных методов по сравнению с существующими. Алгоритм LSTM показал точность 65%, что на 15% выше аналогичных моделей, таких как Decision Tree. Модель RNN достигла точности 74%, что на 9% лучше по сравнению с другими рекуррентными сетями. Ансамблевая модель, объединяющая RNN и LSTM, достигла наивысшей точности в 78%, что на 10-12% выше, чем у алгоритмов AdaBoost и RandomForest.

Оценка моделей по метрикам Precision, Recall и F1 Score также показала превосходство предложенных подходов. Например, алгоритм LSTM достиг F1 Score 0.67 для положительного класса, что на 12% выше по сравнению с классическими методами. Ансамблевое обучение улучшило общую точность и надежность диагностики.

Таким образом, результаты работы подтверждают актуальность и значимость разработки интеллектуальных систем в медицине. Внедрение предложенных решений может привести к более точной и своевременной диагностике, что, в свою очередь, будет способствовать повышению качества медицинского обслуживания и улучшению здоровья пациентов.

Исследование соответствует и часто превосходит современные мировые стандарты в области медицинской информатики и биоинформатики. Результаты демонстрируют значительное улучшение точности диагностики сахарного диабета и могут быть применимы для дальнейших исследований в диагностике других заболеваний.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Постановление от 25 декабря 2012 года № 2050 «О государственной программе развития образования Республики Казахстан на 2011–2020 годы». Республики Казахстан, 2024.
2. Республиканский центр электронного здравоохранения. Электронное здравоохранение [Electronic resource] // <https://rcez.kz/e-health>. URL: <https://rcez.kz/e-health> (accessed: 11.11.2024).
3. Правительство Республики Казахстан. Постановление от 15 декабря 2017 года № 827 «Об утверждении Государственной программы цифрового Казахстана» [Electronic resource]. URL: <https://adilet.zan.kz/rus/docs/P1700000827> (accessed: 11.11.2024).
4. Министерство здравоохранения Республики Казахстан. Нацпроект «Здоровая нация»: профилактика, ранняя диагностика и улучшение качества жизни граждан Республики Казахстан [Electronic resource]. URL: <https://www.gov.kz/memleket/entities/dsm/press/news/details/38036?lang=ru&ysclid=m3cnd7obno753184279> (accessed: 11.11.2024).
5. Bagheri A.B. et al. Potential applications of artificial intelligence and machine learning on diagnosis, treatment, and outcome prediction to address health care disparities of chronic limb-threatening ischemia // *Semin Vasc Surg.* 2023. Vol. 36, № 3. P. 454–459.
6. Mutz J., Lewis C. 383. Mental Health Diagnosis and Polygenic Risk Scores Predict Accelerated Metabolomic Ageing: A Machine Learning Study // *Biol Psychiatry.* 2023. Vol. 93, № 9. P. S249.
7. Gupta R. et al. New era of artificial intelligence and machine learning-based detection, diagnosis, and therapeutics in Parkinson’s disease // *Ageing Res Rev.* 2023. Vol. 90. P. 102013.
8. Ngan H.-L. et al. Machine learning facilitates the application of mass spectrometry-based metabolomics to clinical analysis: A review of early diagnosis of high mortality rate cancers // *TrAC Trends in Analytical Chemistry.* 2023. Vol. 168. P. 117333.
9. Bazarbekov I. et al. A review of artificial intelligence methods for Alzheimer’s disease diagnosis: Insights from neuroimaging to sensor data analysis // *Biomed Signal Process Control.* 2024. Vol. 92. P. 106023.
10. Dunenova G. et al. The Performance and Clinical Applicability of HER2 Digital Image Analysis in Breast Cancer: A Systematic Review // *Cancers (Basel).* 2024. Vol. 16, № 15. P. 2761.
11. Tyulepberdinova G. et al. Development and research of a remote patient monitoring system // *International Journal of Innovative Research and Scientific Studies.* 2024. Vol. 7, № 2. P. 317–329.
12. Н. П. Сапарходжаев, Г. К. Балбаев, А.К. Мукашева. Разработка информационной системы на основе технологий BigData для диагностики и лечения диабета // *Вестник АУЭС.* 2018. Vol. 4, № 6. P. 43.
13. Н.П. Сапарходжаев, А.К. Мукашева. Анализ системы для диагностики сахарного диабета на основе технологии BigData // *Труды Международных Сатпаевских чтений «Инновационные решения традиционных проблем: инженерия и технологии».* 2018. P. 1254–1256.
14. A. Mukasheva et al. Prevalence of diabetes in the republic of Kazakhstan based on regression analysis methods // *International Conference on Research in E-Learning & Distance Education, Social Sciences, Economics and Management.* 2019.
15. Elhadary M. et al. Revolutionizing chronic lymphocytic leukemia diagnosis: A deep dive into the diverse applications of machine learning // *Blood Rev.* 2023. Vol. 62. P. 101134.
16. Asatryan B., Bleijendaal H., Wilde A.A.M. Toward advanced diagnosis and management of inherited arrhythmia syndromes: Harnessing the capabilities of artificial intelligence and machine learning // *Heart Rhythm.* 2023. Vol. 20, № 10. P. 1399–1407.

17. Qian L. et al. Breast cancer diagnosis using evolving deep convolutional neural network based on hybrid extreme learning machine technique and improved chimp optimization algorithm // *Biomed Signal Process Control*. 2024. Vol. 87. P. 105492.
18. Eldin Rashed A.E., Elmorsy A.M., Mansour Atwa A.E. Comparative evaluation of automated machine learning techniques for breast cancer diagnosis // *Biomed Signal Process Control*. 2023. Vol. 86. P. 105016.
19. Refat M.A.R. et al. A Comparative Analysis of Early Stage Diabetes Prediction using Machine Learning and Deep Learning Approach // *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*. IEEE, 2021. P. 654–659.
20. K. Rajesh, V. Sangeetha. Application of Data Mining Methods and Techniques for Diabetes Diagnosis // *International Journal of Engineering and Innovative Technology (IJEIT)*. 2012. Vol. 2, № 3. P. 224–229.
21. Sarwar M.A. et al. Prediction of Diabetes Using Machine Learning Algorithms in Healthcare // *2018 24th International Conference on Automation and Computing (ICAC)*. IEEE, 2018. P. 1–6.
22. Кузнецов Е.В. et al. Эндокринные заболевания как медико-социальная проблема современности // *Современные проблемы науки и образования*. 2017. № 4.
23. Zimmet P. et al. Diabetes mellitus statistics on prevalence and mortality: facts and fallacies // *Nat Rev Endocrinol*. 2016. Vol. 12, № 10. P. 616–622.
24. ВОЗ. Информация о странах с диабетом. 2016.
25. Бітіс Г. Факты о диабете в Казахстане // *Factcheck.kz*. 2022.
26. Lama L. et al. Machine learning for prediction of diabetes risk in middle-aged Swedish people // *Heliyon*. 2021. Vol. 7, № 7. P. e07419.
27. Увалиева И.М. et al. Методы и модели диагностики клинико-гематологических синдромов для электронного паспорта здоровья. Монография. Усть-Каменогорск: ТОО «КАСУ», 2024. 107 p.
28. P Xue et al. WHO global strategy on digital health and its implications to China. 2022. Vol. 56, № 2. P. 218–221.
29. WHO. Digital health [Electronic resource] // https://www.who.int/health-topics/digital-health#tab=tab_1. 2020.
30. ITU. ITU - The role of the Global Strategy on Digital Health 2020 [Electronic resource] // <https://www.itu.int/net4/wsis/forum/2023/Agenda/Session/334>. 2020.
31. WHO. Health Policy Watch - Digital Health - Big WHO Ambitions But Progress Lags [Electronic resource] // <https://healthpolicy-watch.news/digital-health-big-who-goals-slow-progress/>.
32. World Health Organization 2021. The WHO Global strategy on digital health 2020-2025. Geneva, 2021. P. 1–60.
33. WHO. Crowell Health Solutions Blog - WHO Announces Global Initiative Focused on Digital Health [Electronic resource] // <https://www.crowellhealthsolutionsblog.com/2023/08/who-announces-global-initiative-focused-on-digital-health>.
34. Wang C. et al. Artificial Intelligence Algorithm with ICD Coding Technology Guided by Embedded Electronic Medical Record System in Medical Record Information Management // *Microprocess Microsyst*. 2023. P. 104962.
35. Li Q. et al. Automating and improving cardiovascular disease prediction using Machine learning and EMR data features from a regional healthcare system // *Int J Med Inform*. 2022. Vol. 163. P. 104786.
36. Barter L., Cooper C.L. The impact of electronic medical record system implementation on HCV screening and continuum of care: a systematic review // *Ann Hepatol*. 2021. Vol. 24. P. 100322.
37. Esmati E. et al. Short Term Impact Of An Inpatient Electronic Medical Record Pathway On Heart Failure Outcomes // *J Card Fail*. 2024. Vol. 30, № 1. P. 156.

38. Xie J. et al. Learning an expandable EMR-based medical knowledge network to enhance clinical diagnosis // *Artif Intell Med*. 2020. Vol. 107. P. 101927.
39. Vrca Botica M. et al. How to improve opportunistic screening by using EMRs and other data. The prevalence of undetected diabetes mellitus in target population in Croatia // *Public Health*. 2017. Vol. 145. P. 30–38.
40. Sauer C.M. et al. Leveraging electronic health records for data science: common pitfalls and how to avoid them // *Lancet Digit Health*. 2022. Vol. 4, № 12. P. e893–e898.
41. Naydanov C., Palchunov D., Sazonova P. Development of automated methods for the critical condition risk prevention, based on the analysis of the knowledge obtained from patient medical records // 2015 International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON). IEEE, 2015. P. 33–38.
42. Kulich A. A. Компьютерная система моделирования когнитивных карт: подходы и методы // *obzory. obzory*. 2010. P. 2–16.
43. Strunkin D. YU, Abdrahmanov E. F. Patient’s individual survival prediction system based on fuzzy neural // *Vrach i informacionnye tekhnologii*. 2012. Vol. 5.
44. A.A. Litvin, O.YU. Rebrova. Интеллектуальный анализ данных, логистическо регрессии // *problemy zdorov’ya i ekologii*. 2016. P. 10–17.
45. Schekina E. System analysis for creation of program complexes of medical Support systems (literature report) // *Journal of New Medical Technologies. eJournal*. 2017. Vol. 2. P. 1–1.
46. Zhou L. et al. Machine learning on big data: Opportunities and challenges // *Neurocomputing*. 2017. Vol. 237. P. 350–361.
47. Piskunova T.A. Применение интеллектуального анализа данных для создания системы решающих правил // XIII Vserossijskaya nauchno-prakticheskaya konferenciya «Tekhnologii Microsoft v teorii i praktike programmirovaniya».
48. Berestneva O.G. Создание подсистемы принятия решений в медицинских информационных системах // *Izvestiya Tomskogo politekhnicheskogo universiteta*. 2010. Vol. 317, № 5. P. 194–197.
49. E.V. Sadykova. INFORMATION TECHNOLOGY OF MEDICAL EXPERT DECISION MAKING SUPPORT SYSTEMS // *kratkie soobshcheniya*. 2012. Vol. 5. P. 89–91.
50. Gusev S.D, Gusev N.S, Bochanova E.N. Information support for the provision of high-quality medical care by using medical information systems // *medical information systems*. 2016.
51. Ebrahimi M. et al. Comprehensive analysis of machine learning models for prediction of sub-clinical mastitis: Deep Learning and Gradient-Boosted Trees outperform other models // *Comput Biol Med. Pergamon*, 2019. Vol. 114. P. 103456.
52. Liu T., Fan W., Wu C. A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset // *Artif Intell Med. Elsevier*, 2019. Vol. 101. P. 101723.
53. Spänig S. et al. The virtual doctor: An interactive clinical-decision-support system based on deep learning for non-invasive prediction of diabetes // *Artif Intell Med. Elsevier*, 2019. Vol. 100. P. 101706.
54. Ghiasi M.M., Zendejboudi S. Application of decision tree-based ensemble learning in the classification of breast cancer // *Comput Biol Med. Pergamon*, 2021. Vol. 128. P. 104089.
55. Barchitta M. et al. A machine learning approach to predict healthcare-associated infections at intensive care unit admission: findings from the SPIN-UTI project // *Journal of Hospital Infection*. 2021. Vol. 112. P. 77–86.
56. Ross E.G. et al. The use of machine learning for the identification of peripheral artery disease and future mortality risk // *J Vasc Surg*. 2016. Vol. 64, № 5. P. 1515-1522.e3.
57. Gonzales Martinez R., van Dongen D.-M. Deep learning algorithms for the early detection of breast cancer: A comparative study with traditional machine learning // *Inform Med Unlocked*. 2023. Vol. 41. P. 101317.

58. Alballa N., Al-Turaiki I. Machine learning approaches in COVID-19 diagnosis, mortality, and severity risk prediction: A review // *Inform Med Unlocked*. 2021. Vol. 24. P. 100564.
59. Cobre A. de F. et al. Diagnosis and prediction of COVID-19 severity: can biochemical tests and machine learning be used as prognostic indicators? // *Comput Biol Med*. 2021. Vol. 134. P. 104531.
60. Albagmi F.M. et al. Prediction of generalized anxiety levels during the Covid-19 pandemic: A machine learning-based modeling approach // *Inform Med Unlocked*. 2022. Vol. 28. P. 100854.
61. Costantini G. et al. Deep learning and machine learning-based voice analysis for the detection of COVID-19: A proposal and comparison of architectures // *Knowl Based Syst*. 2022. Vol. 253. P. 109539.
62. Gao J. et al. MedML: Fusing medical knowledge and machine learning models for early pediatric COVID-19 hospitalization and severity prediction // *iScience*. 2022. Vol. 25, № 9. P. 104970.
63. Hasan M. et al. Pre-hospital prediction of adverse outcomes in patients with suspected COVID-19: Development, application and comparison of machine learning and deep learning methods // *Comput Biol Med*. 2022. Vol. 151. P. 106024.
64. Moradi Khaniabadi P. et al. Two-step machine learning to diagnose and predict involvement of lungs in COVID-19 and pneumonia using CT radiomics // *Comput Biol Med*. 2022. Vol. 150. P. 106165.
65. Verma H., Mandal S., Gupta A. Temporal deep learning architecture for prediction of COVID-19 cases in India // *Expert Syst Appl*. 2022. Vol. 195. P. 116611.
66. Xu L., Magar R., Barati Farimani A. Forecasting COVID-19 new cases using deep learning methods // *Comput Biol Med*. 2022. Vol. 144. P. 105342.
67. Khanna V.V. et al. A machine learning and explainable artificial intelligence triage-prediction system for COVID-19 // *Decision Analytics Journal*. 2023. Vol. 7. P. 100246.
68. Marshall R.J. The use of classification and regression trees in clinical epidemiology // *J Clin Epidemiol*. 2001. Vol. 54, № 6. P. 603–609.
69. Grey M. et al. Clinical and psychosocial factors associated with achievement of treatment goals in adolescents with diabetes mellitus // *Journal of Adolescent Health*. 2001. Vol. 28, № 5. P. 377–385.
70. Ergün U. et al. Classification of carotid artery stenosis of patients with diabetes by neural network and logistic regression // *Comput Biol Med*. 2004. Vol. 34, № 5. P. 389–405.
71. Marshall R.J. The use of classification and regression trees in clinical epidemiology // *J Clin Epidemiol*. 2001. Vol. 54, № 6. P. 603–609.
72. Polat K., Güneş S., Arslan A. A cascade learning system for classification of diabetes disease: Generalized Discriminant Analysis and Least Square Support Vector Machine // *Expert Syst Appl*. 2008. Vol. 34, № 1. P. 482–487.
73. Dogantekin E. et al. An intelligent diagnosis system for diabetes on Linear Discriminant Analysis and Adaptive Network Based Fuzzy Inference System: LDA-ANFIS // *Digit Signal Process*. 2010. Vol. 20, № 4. P. 1248–1255.
74. Park J., Edington D.W. A sequential neural network model for diabetes prediction // *Artif Intell Med*. 2001. Vol. 23, № 3. P. 277–293.
75. Rabhi S. et al. Temporal deep learning framework for retinopathy prediction in patients with type 1 diabetes // *Artif Intell Med*. 2022. Vol. 133. P. 102408.
76. Chee L.Z. et al. Gait acceleration-based diabetes detection using hybrid deep learning // *Biomed Signal Process Control*. 2024. Vol. 92. P. 105998.
77. Katiyar N., Thakur H.K., Ghatak A. Recent advancements using machine learning & deep learning approaches for diabetes detection: a systematic review // *e-Prime - Advances in Electrical Engineering, Electronics and Energy*. 2024. Vol. 9. P. 100661.

78. Horie S. et al. Blue Widefield Images of Scanning Laser Ophthalmoscope Can Detect Retinal Ischemic Areas in Eyes With Diabetic Retinopathy // *Asia-Pacific Journal of Ophthalmology*. 2021. Vol. 10, № 5. P. 478–485.
79. Jacoba C.M.P. et al. Performance of Automated Machine Learning for Diabetic Retinopathy Image Classification from Multi-field Handheld Retinal Images // *Ophthalmol Retina*. 2023. Vol. 7, № 8. P. 703–712.
80. Evans E.I. et al. Health literacy of patients using continuous glucose monitoring // *Journal of the American Pharmacists Association*. 2024. Vol. 64, № 4. P. 102109.
81. Zhang M., Flores K.B., Tran H.T. Deep learning and regression approaches to forecasting blood glucose levels for type 1 diabetes // *Biomed Signal Process Control*. 2021. Vol. 69. P. 102923.
82. Dogan A. et al. A utility-based machine learning-driven personalized lifestyle recommendation for cardiovascular disease prevention // *J Biomed Inform*. 2023. Vol. 141. P. 104342.
83. Levy-Loboda T. et al. Personalized insulin dose manipulation attack and its detection using interval-based temporal patterns and machine learning algorithms // *J Biomed Inform*. 2022. Vol. 132. P. 104129.
84. Pinhas-Hamiel O., Zeitler P. The global spread of type 2 diabetes mellitus in children and adolescents // *J Pediatr*. 2005. Vol. 146, № 5. P. 693–700.
85. Shah S. et al. The Impact of Guideline Integration into Electronic Medical Records on Outcomes for Patients with Diabetes: A Systematic Review // *Am J Med*. 2021. Vol. 134, № 8. P. 952-962.e4.
86. Tsur N. et al. Gestational diabetes and risk of future diabetes in a multi-ethnic population // *J Diabetes Complications*. 2024. Vol. 38, № 4. P. 108720.
87. Kokkorakis M. et al. Effective questionnaire-based prediction models for type 2 diabetes across several ethnicities: a model development and validation study // *EClinicalMedicine*. 2023. Vol. 64. P. 102235.
88. Fereydooni A. et al. Racial, ethnic, and socioeconomic inequities in amputation risk for patients with peripheral artery disease and diabetes // *Semin Vasc Surg*. 2023. Vol. 36, № 1. P. 9–18.
89. Kamel Rahimi A. et al. Machine learning models for diabetes management in acute care using electronic medical records: A systematic review // *Int J Med Inform*. 2022. Vol. 162. P. 104758.
90. Hernández-Ávila J. et al. Extracting And Using Data From Electronic Medical Records (Emr) To Monitor Quality Of Care And Prescription Patterns For Diabetes Prevention And Control In Outpatient Clinics Of Low And Mid Resources Countries: The Case Of Colima, Mexico // *Value in Health*. 2015. Vol. 18, № 7. P. A811.
91. Dibato J., Montvida O., Sanjoy P.K. Racial disparity in the co-occurrence of depression and type 2 diabetes mellitus. An electronic medical record study involving African American and White Caucasian adults from the US // *J Affect Disord*. 2023. Vol. 330. P. 173–179.
92. Terrin N. et al. External validity of predictive models: a comparison of logistic regression, classification trees, and neural networks // *J Clin Epidemiol*. 2003. Vol. 56, № 8. P. 721–729.
93. Villikudathil A.T., Mc Guigan D.H., English A. Exploring metformin monotherapy response in Type-2 diabetes: Computational insights through clinical, genomic, and proteomic markers using machine learning algorithms // *Comput Biol Med*. 2024. Vol. 171. P. 108106.
94. Walle A.D. et al. Intention to use wearable health devices and its predictors among diabetes mellitus patients in Amhara region referral hospitals, Ethiopia: Using modified UTAUT-2 model // *Inform Med Unlocked*. 2023. Vol. 36. P. 101157.

95. Liu S. et al. LiverRisk score: An accurate, cost-effective tool to predict fibrosis, liver-related, and diabetes-related mortality in the general population // *Med.* 2024. Vol. 5, № 6. P. 570-582.e4.
96. Lama L. et al. Machine learning for prediction of diabetes risk in middle-aged Swedish people *Heliyon.* 2021. Vol. 7, № 7.
97. Rajesh K., Sangeetha V. Application of data mining methods and techniques for diabetes diagnosis // *Int. J. Eng. Innov. Technol. (IJEIT).* 2012.
98. Sarwar M.A. et al. Prediction of diabetes using machine learning algorithms in healthcare // *24th International Conference on Automation and Computing (ICAC) Newcastle Upon Tyne, United Kingdom .* 2018. P. 1–6.
99. Shan F. et al. Effects of data smoothing and recurrent neural network (RNN) algorithms for real-time forecasting of tunnel boring machine (TBM) performance // *Journal of Rock Mechanics and Geotechnical Engineering.* 2023.
100. Rafique Q. et al. Reviewing methods of deep learning for diagnosing COVID-19, its variants and synergistic medicine combinations // *Comput Biol Med.* 2023. Vol. 163. P. 107191.
101. Bibault J.E., Giraud P., Burgun A. Big Data and machine learning in radiation oncology: State of the art and future prospects // *Cancer Lett. Elsevier,* 2016. Vol. 382, № 1. P. 110–117.
102. Jian Zheng et al. Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network // *2017 51st Annual Conference on Information Sciences and Systems (CISS). IEEE,* 2017. P. 1–6.
103. Mokina E.E et al. USING DATA MINING METHOD IN THE MAKING OF MEDICAL DIAGNOSTIC DECISIONS // *tekhnicheskie nauki.* 2016. Vol. 5. P. 269–274.
104. Sharma D., Kumar R., Jain A. Breast cancer prediction based on neural networks and extra tree classifier using feature ensemble learning // *Measurement: Sensors.* 2022. Vol. 24. P. 100560.
105. Uddin M.N., Halder R.K. An ensemble method based multilayer dynamic system to predict cardiovascular disease using machine learning approach // *Inform Med Unlocked.* 2021. Vol. 24. P. 100584.
106. Ehwerhemuepha L. et al. A super learner ensemble of 14 statistical learning models for predicting COVID-19 severity among patients with cardiovascular conditions // *Intell Based Med.* 2021. Vol. 5. P. 100030.
107. Elhazmi A. et al. Machine learning decision tree algorithm role for predicting mortality in critically ill adult COVID-19 patients admitted to the ICU // *J Infect Public Health.* 2022. Vol. 15, № 7. P. 826–834.
108. Tripoliti E.E., Fotiadis D.I., Manis G. Modifications of the construction and voting mechanisms of the Random Forests Algorithm // *Data Knowl Eng.* 2013. Vol. 87. P. 41–65.
109. Ruder S. An overview of gradient descent optimization algorithms. 2017. P. 1–14.
110. Kaur H., Kumari V. Predictive modelling and analytics for diabetes using a machine learning approach // *Applied Computing and Informatics.* 2022. Vol. 18, № 1/2. P. 90–100.
111. Refat M.A.R. et al. A Comparative Analysis of Early Stage Diabetes Prediction using Machine Learning and Deep Learning Approach // *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC). IEEE,* 2021. P. 654–659.
112. Nadeem M.W. et al. A fusion-based machine learning approach for the prediction of the onset of diabetes // *In: Healthcare.* Vol. 9. P. 1393.
113. Flores A.M. et al. Leveraging Machine Learning and Artificial Intelligence to Improve Peripheral Artery Disease Detection, Treatment, and Outcomes // *Circ Res.* 2021. Vol. 128, № 12. P. 1833–1850.
114. Sisodia D., Sisodia D.S. Prediction of Diabetes using Classification Algorithms // *Procedia Comput Sci.* 2018. Vol. 132. P. 1578–1585.

115. Rajesh K., Sangeetha V. Application of Data Mining Methods and Techniques for Diabetes Diagnosis // *International Journal of Engineering and Innovative Technology (IJEIT)*. 2012. Vol. 2, № 3. P. 224–229.
116. Kelarev A. et al. Empirical investigation of consensus clustering for large ECG data sets // 2012 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS). IEEE, 2012. P. 1–4.
117. Ganie S.M., Malik M.B. An ensemble Machine Learning approach for predicting Type-II diabetes mellitus based on lifestyle indicators // *Healthcare Analytics*. 2022. Vol. 2. P. 100092.
118. Hassan Md.M. et al. A Comparative Study, Prediction and Development of Chronic Kidney Disease Using Machine Learning on Patients Clinical Records // *Human-Centric Intelligent Systems*. 2023. Vol. 3, № 2. P. 92–104.
119. Khanam J.J., Foo S.Y. A comparison of machine learning algorithms for diabetes prediction // *ICT Express*. 2021. Vol. 7, № 4. P. 432–439.
120. Ahmed U. et al. Prediction of Diabetes Empowered With Fused Machine Learning // *IEEE Access*. 2022. Vol. 10. P. 8529–8538.
121. Laila U. e et al. An Ensemble Approach to Predict Early-Stage Diabetes Risk Using Machine Learning: An Empirical Study // *Sensors*. 2022. Vol. 22, № 14. P. 5247.
122. Ejjiyi C.J. et al. A robust predictive diagnosis model for diabetes mellitus using Shapley-incorporated machine learning algorithms // *Healthcare Analytics*. 2023. Vol. 3. P. 100166.
123. Gupta H. et al. Comparative performance analysis of quantum machine learning with deep learning for diabetes prediction // *Complex & Intelligent Systems*. 2022. Vol. 8, № 4. P. 3073–3087.
124. Sivaranjani S. et al. Diabetes Prediction using Machine Learning Algorithms with Feature Selection and Dimensionality Reduction // 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2021. P. 141–146.
125. Tupasela A., Di Nucci E. Concordance as evidence in the Watson for Oncology decision-support system // *AI Soc*. 2020. Vol. 35, № 4. P. 811–818.
126. Feldman M.J., Octo Barnett G. An approach to evaluating the accuracy of DXplain // *Comput Methods Programs Biomed*. 1991. Vol. 35, № 4. P. 261–266.
127. Vardell E., Moore M. Isabel, a Clinical Decision Support System // *Med Ref Serv Q*. 2011. Vol. 30, № 2. P. 158–166.
128. Wolfram D.A. An appraisal of INTERNIST-I // *Artif Intell Med*. 1995. Vol. 7, № 2. P. 93–116.
129. Haverkamp H.T., Fosse S.O., Schuster P. Accuracy and usability of single-lead ECG from smartphones - A clinical study // *Indian Pacing Electrophysiol J*. 2019. Vol. 19, № 4. P. 145–149.
130. Prorok J.C. et al. The quality, breadth, and timeliness of content updating vary substantially for 10 online medical texts: an analytic survey // *J Clin Epidemiol*. 2012. Vol. 65, № 12. P. 1289–1295.
131. Carson E.R. Decision support systems in diabetes: a systems perspective // *Comput Methods Programs Biomed*. 1998. Vol. 56, № 2. P. 77–91.
132. Lehmann E.D. et al. AIDA: an interactive diabetes advisor // *Comput Methods Programs Biomed*. 1994. Vol. 41, № 3–4. P. 183–203.
133. Frontoni E. et al. A Decision Support System for Diabetes Chronic Care Models Based on General Practitioner Engagement and EHR Data Sharing // *IEEE J Transl Eng Health Med*. 2020. Vol. 8. P. 1–12.
134. Petach-Tikva. Use of Insulin Adjustment Device DreaMed Advisor Pro During Routine Clinical Use for Subjects With Diabetes Type 1 [Electronic resource] // <https://clinicaltrials.gov/study/NCT04271228>.

135. Sands D.Z. Beyond the EHR: How Digital Health Tools Foster Participatory Health and Self-Care for Patients with Diabetes // *American Journal of Medicine Open*. 2023. Vol. 10. P. 100043.
136. Wikipedia. Диагноз // <https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D0%BD%D0%BE%D0%B7>.
137. Milo T., Somech A. Automating Exploratory Data Analysis via Machine Learning: An Overview // *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2020. P. 2617–2622.
138. Dhany H.W., Sutarman S., Izhari F. Exploratory Data Analysis (EDA) methods for healthcare classification // *Journal of Intelligent Decision Support System (IDSS)*. 2023. Vol. 6, № 4. P. 209–215.
139. Фирюлина М.А., Каширина И.Л. Описание процесса прогнозирования проблемных состояний с применением ансамблевых методов машинного обучения // *Инженерный вестник Дона*. 2022. Vol. 4, № 88. P. 34–46.
140. Gusev A.V., Gavrilenko G.G., Gavrilov D.V. Development of a machine learning model to interpret the results of laboratory diagnostics in order to identify suspected diseases // *Laboratornaya sluzhba*. 2022. Vol. 11, № 2. P. 9.
141. Swana E.F., Doorsamy W., Bokoro P. Tomek Link and SMOTE Approaches for Machine Fault Classification with an Imbalanced Dataset // *Sensors*. 2022. Vol. 22, № 9. P. 3246.
142. Lalu Ganda Rady Putra, Khairani Marzuki, Hairani Hairani. Correlation-based feature selection and Smote-Tomek Link to improve the performance of machine learning methods on cancer disease prediction // *Engineering and Applied Science Research*. 2023. Vol. 50, № 6. P. 577–583.
143. Sarker I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions // *SN Comput Sci*. 2021. Vol. 2, № 6. P. 420.
144. Yu H. et al. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives // *Neurocomputing*. 2021. Vol. 444. P. 92–110.
145. S. K. Zhou et al. A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies With Progress Highlights, and Future Promises // *Proceedings of the IEEE*. 2021. Vol. 109, № 5. P. 820–838.
146. Tran K.A. et al. Deep learning in cancer diagnosis, prognosis and treatment selection // *Genome Med*. 2021. Vol. 13, № 1. P. 152.
147. Wang Y., He Y., Zhu Z. Study on fast speed fractional order gradient descent method and its application in neural networks // *Neurocomputing*. 2022. Vol. 489. P. 366–376.
148. Liquan Z., Cheng S. Designing fuzzy inference system based on improved gradient descent method // *Journal of Systems Engineering and Electronics*. 2006. Vol. 17, № 4. P. 853–857.
149. Yu G., Zhao Y., Wei Z. A descent nonlinear conjugate gradient method for large-scale unconstrained optimization // *Appl Math Comput*. 2007. Vol. 187, № 2. P. 636–643.
150. Bouwmans T. et al. Deep neural network concepts for background subtraction: A systematic review and comparative evaluation // *Neural Networks*. 2019. Vol. 117. P. 8–66.
151. Mall P.K. et al. A comprehensive review of deep neural networks for medical image processing: Recent developments and future opportunities // *Healthcare Analytics*. 2023. Vol. 4. P. 100216.
152. Montavon G., Samek W., Müller K.-R. Methods for interpreting and understanding deep neural networks // *Digit Signal Process*. 2018. Vol. 73. P. 1–15.
153. Liao G., Zhang L. Solving flows of dynamical systems by deep neural networks and a novel deep learning algorithm // *Math Comput Simul*. 2022. Vol. 202. P. 331–342.
154. Sammut C., Webb G.I. *Encyclopedia of Machine Learning and Data Mining* / ed. Sammut C., Webb G.I. Boston, MA: Springer US, 2017. Vol. 2.
155. Al-Shehari T., Alsowail R.A. An Insider Data Leakage Detection Using One-Hot Encoding, Synthetic Minority Oversampling and Machine Learning Techniques // *Entropy*. 2021. Vol. 23, № 10. P. 1258.

156. Gu B., Sung Y. Enhanced Reinforcement Learning Method Combining One-Hot Encoding-Based Vectors for CNN-Based Alternative High-Level Decisions // *Applied Sciences*. 2021. Vol. 11, № 3. P. 1291.
157. Muning Wen et al. Large sequence models for sequential decision-making: a survey // *Front Comput Sci*. 2023. Vol. 17.
158. Jan B. et al. Deep learning in big data Analytics: A comparative study // *Computers & Electrical Engineering*. 2019. Vol. 75. P. 275–287.
159. Смышляев Г. Е., Красикова Е. М. Математические модели биологических нейронов // *Моделирование нелинейных процессов и систем. Материалы*. 2024. P. 232.
160. Soe Y.N. et al. Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture // *Sensors*. 2020. Vol. 20, № 16. P. 4372.
161. Guohao Li et al. SGAS: Sequential Greedy Architecture Search // *Proceedings of the IEEE. CVF conference on computer vision and pattern recognition*. 2020. P. 1620–1630.
162. Пересыпкина И. Г. Эмпирическое исследование влияния Dropout и Batch Normalization на глубокое обучение // *Международная научно-техническая конференция молодых ученых БГТУ им. ВГ Шухова, посвященная 300-летию Российской академии наук*. 2022. P. 224.
163. Арбузова А. А. Диагностика пневмонии по рентгеновским снимкам с помощью сверточных нейронных сетей // *Модели, системы, сети в экономике, технике, природе и обществе*. 2021. Vol. 2. P. 107–114.
164. Сенькович Д.С., Жвакина А.В. Нейросетевая система поддержки принятия банковских решений при выдаче кредитов // *BIG DATA AND ADVANCED ANALYTICS*. 2020. Vol. 6, № 1. P. 358–366.
165. S. Liu, Sergey Kabanikhin, S. V. Strijhak. Regularization of linear machine learning problems. 2024.
166. Bondarenko V. A., Popov D. I. Research and development of algorithms for the formation of an effective ensemble of convolutional neural networks for image classification // *Программные системы и вычислительные методы*. 2024. № 1. P. 48–67.
167. Иващенко А. В., Кривошеев А. В. Модель ансамблирования интеллектуальных компонентов системы компьютерного зрения на основе рекуррентной нейронной сети // *НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ПОВОЛЖЬЯ*. 2022. P. 164–167.
168. Волошин Т. А., Зайцев К. С., Дунаев М. Е. Применение адаптивных ансамблей методов машинного обучения к задаче прогнозирования временных рядов // *International Journal of Open Information Technologies*. 2023. P. 57–63.
169. Алабугин С. К., Соколов А. Н. Обнаружение вторжений в автоматизированных системах управления технологическими процессами с использованием ансамбля моделей рекуррентной и двунаправленной генеративно-состязательной нейронных сетей // *Вестник УрФО. Безопасность в информационной сфере*. 2021. Vol. 3, № 41. P. 38–48.
170. Хлыбов А. А. МЕТОД И Алгоритм анализа процессов управления в многоуровневых и распределенных системах в условиях неустранимой неопределенности на основе статистической модели распределения дирихле // *Труды НГТУ им. ПЕ Алексева*. 2022. Vol. 3, № 138. P. 44–53.
171. Jankowski M. Ensemble Methods for Improving Classification of Data Produced by Latent Dirichlet Allocation // *Computer Science and Mathematical Modelling*. 2019. Vol. 0, № 8/2018. P. 17–28.
172. Землянский С. А., Лызин И. А., Аксёнов С. Анализ текстов в медицинских исследованиях // *Модели и методы повышения эффективности инновационных*. 2022. P. 49.
173. PhysioNet Credentialed Health Data License 1.5.0.
174. Syed M. et al. Application of Machine Learning in Intensive Care Unit (ICU) Settings Using MIMIC Dataset: Systematic Review // *Informatics*. 2021. Vol. 8, № 1. P. 16.

ПРИЛОЖЕНИЕ А. СПРАВКА ОБ УЧАСТИИ В ПРОЕКТЕ ГРАНТОВОГО ФИНАНСИРОВАНИЯ



EKTU
1958

Қазақстан Республикасы, ШҚО
070000, Өскемен қаласы, Серікбаев көшесі, 19
тел: 70-46-30, 70-20-10
E-mail: kento@edu.ektu.kz
«НАЛЫК БАНК» АҚ ЕЕ Өскемен қаласындағы филиалы
ЖСК K2296017151000000043
БСК HSBKZKX, РНН 181800000624, Код (КБел) 16
БСН 010440002379

Республика Казахстан, ВКО
070000, г. Усть-Каменогорск, ул. Серікбаева, 19
тел: 70-46-30, 70-20-10
E-mail: kento@edu.ektu.kz
ДБ АО «НАЛЫК БАНК» филиал в г. Усть-Каменогорск
И/К K2296017151000000043, Б/К HSBKZKX,
РНН 181800000624, Код (КБел) 16
БИН 010440002379

The Republic of Kazakhstan, East Kazakhstan
070000, Ust-Kamenogorsk city, 19 Serikbayev Street
tel.: 0 (7232)70-46-30, 0 (7232)70-20-10
E-mail: kento@edu.ektu.kz
Branch of JSC «NALYK BANK» Ust-Kamenogorsk city
BC K2296017151000000043, BIC HSBKZKARNT
181800000624, Code 16,
BIN 010440002379

0409 2024 №15-22-12/1580

Справка

Дана Исмухамедовой Айгерим Мэлсатовне, докторанту ОП «8D06101– Информационные системы», о том, что являлась старшим научным сотрудником научного проекта AP19679525 «Программный комплекс диагностики клинико-гематологических синдромов для электронного паспорта здоровья», выполняемого в рамках бюджетной программы «Грантовое финансирование научных исследований», договор №321/23-25 от 03.08.2023.

Член правления –
проректор по науке и инновациям



Ж. Конурбаева

ПРИЛОЖЕНИЕ Б. СПРАВКА ОБ УЧАСТИИ В ПРОЕКТЕ «ЖАС ГАЛЫМ»



«Д. СЕРИКБАЕВ атындағы ШЫҒЫС ҚАЗАҚСТАН ТЕХНИКАЛЫҚ УНИВЕРСИТЕТІ»
коммерциялық емес акционерлік қоғамы
«ВОСТОЧНО-КАЗАХСТАНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени Д. СЕРИКБАЕВА»
некоммерческое акционерное общество
«D. SERIKBAEV EAST KAZAKHSTAN TECHNICAL UNIVERSITY»
non-profit joint-stock company

Қазақстан Республикасы, ШҚР
070000, Өскемен қаласы, Серікбаев көшесі, 19
тел: 70-46-38, 70-20-10
E-mail: korse@edu.ektu.kz
«НАЛЫК БАНК» АҚ ЕБ Өскемен қаласындағы филиалы
ЖСК K2296017151000000043
БСК HSBKZKX, РНН 181800000624, Код (КБс) 16
БИН 010440002379

Республика Казахстан, ВКО
070000, г. Усть-Каменогорск, ул. Серікбаева, 19
тел: 70-46-38, 70-20-10
E-mail: korse@edu.ektu.kz
ДБ АО «НАЛЫК БАНК» филиал в г. Усть-Каменогорск
ИИК K2296017151000000043, БСК HSBKZKX,
РНН 181800000624, Код (КБс) 16
БИН 010440002379

The Republic of Kazakhstan, East Kazakhstan
070000, Ust-Kamenogorsk city, 19 Serikbayev Street
tel.: 8 (7232)70-46-38, 8 (7232)70-20-10
E-mail: korse@edu.ektu.kz
Branch of JSC «NALYK BANK» Ust-Kamenogorsk city
IC K2296017151000000043, BIC SABKZKARNT
181800000624, Code 16,
BIN 010440002379

0408 2024 №15-22-12/1589

Справка

Дана Исмухамедовой Айгерим Мэлсатовне, докторанту ОП «8D06101– Информационные системы», о том, что является руководителем научного проекта AP22683316 «Применение алгоритмов машинного обучения для систем поддержки принятия врачебных решений», выполняемого в рамках конкурса на грантовое финансирование исследований молодых ученых по проекту «Жас галым» на 2024-2026 годы, договор №128/ЖГ 5-24-26 от 26.06.2024.

Член правления –
проректор по науке и инновациям



Ж. Конурбаева

**ПРИЛОЖЕНИЕ В. АВТОРСКОЕ СВИДЕТЕЛЬСТВО «АЛГОРИТМ
ПОДДЕРЖКИ КЛИНИЧЕСКИХ РЕШЕНИЙ НА ОСНОВЕ
ТЕХНОЛОГИИ АНДЕРСЭМПЛИНГА»**

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ  РЕСПУБЛИКА КАЗАХСТАН

СВИДЕТЕЛЬСТВО
О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР
ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ
№ 49449 от «4» сентября 2024 года

Фамилия, имя, отчество, (если оно указано в документе, удостоверяющем личность) автора (ов):
ИСМУХАМЕДОВА АЙГЕРИМ МЭЛСАТОВНА, УВАЛШЕВА ИНДИРА МАХМУТОВНА

Вид объекта авторского права: **программа для ЭВМ**

Название объекта: **Алгоритм поддержки клинических решений на основе технологии андерсэмплинга
(выполнено в рамках проекта AP22683316)**

Дата создания объекта: **01.08.2024**





Курсат тудусқалпын <http://www.kazpatent.kz/ru> сайтының
"Авторлық құқық" бөлімінде тексеруге болады. <https://copyright.kazpatent.kz>

Подлинность документа возможно проверить на сайте kazpatent.kz
в разделе «Авторское право» <https://copyright.kazpatent.kz>

Подписано ЭЦП Е. Оспанов

**ПРИЛОЖЕНИЕ Г. АВТОРСКОЕ СВИДЕТЕЛЬСТВО
«ПРОГРАММНЫЙ МОДУЛЬ ДИАГНОСТИРОВАНИЯ КЛИНИКО-
ГЕМАТОЛОГИЧЕСКИХ СИНДРОМОВ»**

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

СВИДЕТЕЛЬСТВО
О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР
ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ

№ 4737 от «1» августа 2019 года

Фамилия, имя, отчество, (если оно указано в документе, удостоверяющем личность) автора (ов):
УВАЛIEВА ИИДПРА МАХМУТОВНА, БЕЛЫГИНОВА САУЛЕ АСКЕРБЕКОВНА, ИСМУХАМЕДОВА
АЙГЕРИМ МЭЛСАТОВНА

Вид объекта авторского права: программа для ЭВМ

Название объекта: Программный модуль диагностирования клинико-гематологических синдромов

Дата создания объекта: 08.07.2019





Құжат тегін ұсыналыны <http://www.kazpatent.kz/ru> сайтының
"Авторлық құқық" бөлімінде тексеруге болады <https://copyright.kazpatent.kz>

Подлинность документа возможно проверить на сайте kazpatent.kz
в разделе «Авторское право» <https://copyright.kazpatent.kz>

Подписано ЭЦП

Оспанов Е.К.

**ПРИЛОЖЕНИЕ Д. АВТОРСКОЕ СВИДЕТЕЛЬСТВО
«ДИАГНОСТИКА КЛИНИКО-ГЕМАТОЛОГИЧЕСКИХ СИНДРОМОВ
НА ОСНОВЕ МОРФОЛОГИЧЕСКОЙ КЛАССИФИКАЦИИ»**

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ  РЕСПУБЛИКА КАЗАХСТАН

СВИДЕТЕЛЬСТВО
**О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР
ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ**
№ 41784 от «5» января 2024 года

Фамилия, имя, отчество, (если оно указано в документе, удостоверяющем личность) автора (ов):
**УВАЛИЕВА ИНДИРА МАХМУТОВНА, БОРОЗЕНЕЦ ДАВИД РАФАЭЛЕВИЧ, БЕЛЬГИНОВА САУЛЕ
АСКЕРБЕКОВНА, ИСМУХАМЕДОВА АЙГЕРИМ МЭЛСАТОВНА**

Вид объекта авторского права: **база данных**

Название объекта: **База данных дифференциального диагностирования клинико-гематологических
синдромов на основе алгоритма морфологической классификации (выполнено в рамках проекта
АР19679525)**

Дата создания объекта: **03.01.2024**





Құжат түпнұсқасын <http://www.kazpatent.kz/ru> сайтының
"Авторлық құқық" бөлімінде тексеруге болады <https://copyright.kazpatent.kz>
Подлинность документа возможно проверить на сайте kazpatent.kz
в разделе «Авторское право» <https://copyright.kazpatent.kz>

Подписано ЭЦП **Е. Оспанов**

ПРИЛОЖЕНИЕ Е. АКТ ВНЕДРЕНИЯ В ПРОИЗВОДСТВО С ТОО «ЮВЕНТАМЕД»



Өскемен қаласы, Добролюбова көшесі, 39/2
РК, ВКО, г. Усть-Каменогорск, ул. Добролюбова 39/2
БИН: 970340000645

АКТ ВНЕДРЕНИЯ

результаты диссертационного исследования Исмухамедовой А.М. в производство

Мы, представители ТОО «ЮвентаМед», настоящим актом подтверждаем, что результаты диссертационного исследования Исмухамедовой А.М. на тему "Алгоритмическое обеспечение интеллектуальной системы поддержки принятия клинических решений" были частично внедрены в виде модуля обработки данных, включающего алгоритм поддержки клинических решений на основе технологии андерсэмплинга.

В рамках оптимизации и повышения эффективности работы, был внедрен модуль обработки данных, разработанный Исмухамедовой А.М. Этот модуль предназначен для предварительной обработки и анализа медицинских данных пациентов, что позволяет улучшить точность диагностики и сократить время на принятие клинических решений.

Внедрение данного модуля является первым этапом работы по созданию полной интеллектуальной системы поддержки принятия клинических решений. В дальнейшем планируется расширение функционала модуля с добавлением дополнительных алгоритмов и методов анализа данных, что позволит еще более повысить эффективность диагностических и лечебных процессов.

Результаты внедрения и эксплуатации модуля подтвердили его работоспособность и потенциал для дальнейшего использования и развития.

Директор ТОО «ЮвентаМед»



Баландина Г.Г.

ПРИЛОЖЕНИЕ Ж. АКТ ВНЕДРЕНИЯ В УЧЕБНЫЙ ПРОЦЕСС КАСУ

УТВЕРЖДАЮ

Проректор по академическим
вопросам и информатизации

Казахстанско-Американского
свободного университета, доктор

PhD

Сарсембаева Г.Ж.

2024 г.



АКТ ВНЕДРЕНИЯ (ИСПОЛЬЗОВАНИЯ)

результатов научно-исследовательской работы, выполненной в рамках диссертационного исследования «Алгоритмическое обеспечение интеллектуальной системы поддержки принятия клинических решений».

в учебный процесс

Мы, нижеподписавшиеся, Бордияну И.В. – заведующий кафедрой «Бизнеса», доктор PhD; Мензюк Г.А. – к. ю. н., доцент, декан факультета "Бизнеса, права и педагогики"; Исмухамедова А.М. – докторант, старший преподаватель; Увалиева И.М. – профессор школы цифровых технологий и искусственного интеллекта, доктор PhD, ассоциированный профессор, научный руководитель внедряемых результатов, составили настоящий АКТ ВНЕДРЕНИЯ (ИСПОЛЬЗОВАНИЯ) результатов научно-исследовательской работы, выполненных в рамках диссертационного исследования «Алгоритмическое обеспечение интеллектуальной системы поддержки принятия клинических решений».

Основные результаты работы: Целью диссертационного исследования является интеграция и адаптация алгоритмов интеллектуальной системы поддержки принятия клинических решений (СППР) в рамках дисциплины "IT Менеджмент", с акцентом на стратегическое планирование, управление IT-проектами и этические аспекты на примере области здравоохранения. В результате достижения цели были получены следующие результаты и новизна: разработаны стратегии управления IT-проектами; созданы методы управления большими данными, включая их сбор, хранение и обработку, на основе работы с медицинскими данными в контексте СППР; разработаны инновационные кейсы для внедрения новых технологий в медицинских учреждениях; выполнен анализ этических вопросов, связанных с использованием алгоритмов и искусственного интеллекта в здравоохранении, с рекомендациями для IT-менеджеров по соблюдению этических норм; созданы практические кейсы для студентов по разработке IT-систем в здравоохранении, основанные на реальных примерах;

выполнен анализ и разработаны рекомендации по роли IT-менеджера в условиях внедрения интеллектуальных систем в здравоохранении.

Указанная работа внедрена (использована) в учебный процесс на 2024-2025 учебный год в следующих курсах лекционных и практических (лабораторных) занятий образовательной программы 7М04105 "IT Менеджмент":

- «Design and Implementation of Software Systems»;
- «Digital Business Modelling».

Наименование объекта и предмета внедрения (использования) результатов научно-исследовательской работы: объект внедрения – алгоритмы интеллектуальных систем поддержки принятия решений в различных отраслях и IT-инфраструктурах; предмет внедрения – методы управления IT-проектами, обработки и анализа данных, а также стратегии внедрения и использования алгоритмических систем в организациях и учреждениях.

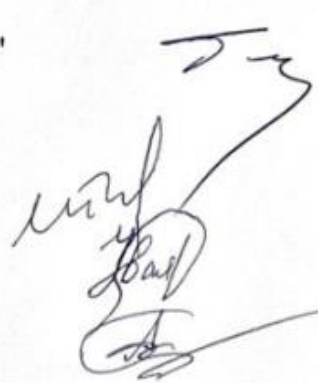
Эффект от внедрения (использования) результатов внедрения: научно-технические результаты по проблеме оптимизации и автоматизации процессов принятия решений в различных отраслях, включая здравоохранение, IT и бизнес, являются основой для повышения эффективности и точности этих процессов. Эффективность применения интеллектуальных систем поддержки принятия решений была продемонстрирована на примере внедрения алгоритмических методов управления IT-проектами и анализа данных. Приобретение практического опыта в области разработки и внедрения таких систем позволяет значительно улучшить качество подготовки специалистов по программам «IT Менеджмент» и смежным направлениям, обеспечивая их навыками работы с передовыми технологиями и инструментами управления.

Декан Факультет
"Бизнеса, права и педагогики"

Заведующий кафедры
«Бизнеса»

Научный руководитель

Докторант



Мензюк Г.А.

Бордияну И.В.

Увалиева И.М.

Исмухамедова А.М.

ПРИЛОЖЕНИЕ II. АКТ ВНЕДРЕНИЯ В УЧЕБНЫЙ ПРОЦЕСС ВКТУ Д.СЕРИКБАЕВА

УТВЕРЖДАЮ

Член правления - проректор по академическим вопросам
НАО ВКТУ им. Д.Серикбаева



А. Х. Машекенова

2024 г.

АКТ ВНЕДРЕНИЯ (ИСПОЛЬЗОВАНИЯ)

результатов научно-исследовательской работы, выполненной в рамках диссертационного исследования «Алгоритмическое обеспечение интеллектуальной системы поддержки принятия клинических решений» докторанта ОП 8D06101 - Информационные системы (по отраслям) Исмухамедовой Айгерим Мэлсатовны

в учебный процесс

Мы, нижеподписавшиеся, Хасенова З.Т. – декан школы цифровых технологий и искусственного интеллекта, доктор PhD; Тезекпаева Ш.Т. – руководитель ОП «Математическое и компьютерное моделирование», магистр технических наук; Увалиева И.М. – профессор школы цифровых технологий и искусственного интеллекта, доктор PhD, ассоциированный профессор, составили настоящий АКТ ВНЕДРЕНИЯ (ИСПОЛЬЗОВАНИЯ) результатов научно-исследовательской работы, выполненных в рамках диссертационного исследования «Алгоритмическое обеспечение интеллектуальной системы поддержки принятия клинических решений».

Основные результаты работы: Цель исследования заключается в разработке алгоритмов интеллектуальной поддержки принятия клинических решений в эндокринологии и диабетологии. В результате достижения цели были получены следующие результаты и новизна: разработана концептуальная модель процесса поддержки принятия клинических решений, основанная на методике исследовательского анализа данных эндокринологии и диабетологии; разработан алгоритм поддержки клинических решений на основе технологии андерсэмплинга; создан алгоритм прогнозирования диабета на основе модели глубокой нейронные сети с оптимизационными гиперпараметрами; предложен алгоритм ассемблирования архитектур нейронных сетей LSTM и RNN для задач поддержки клинических решений. Научная новизна диссертационного исследования заключается в том, что впервые для повышения эффективности процессов поддержки принятия клинических решений в эндокринологии и диабетологии

предложен комплекс алгоритмов, интегрирующий технологию андерсэмплинга и ассемблирования архитектур нейронных сетей LSTM и RNN.

Результаты диссертационного исследования были внедрены (использованы) в учебном процессе в 2023-2024 учебном году в следующих курсах лекционных и практических (лабораторных) занятий ОП «Математическое и компьютерное моделирование»:

- «Моделирование биологических процессов и систем»;
- «Основы нейронных сетей».

Наименование объекта и предмета внедрения (использования) результатов научно-исследовательской работы: *объект внедрения* – модели процесса поддержки принятия клинических решений; *предмет внедрения* – алгоритма поддержки клинических решений; модели глубокой нейронные сети с оптимизированными гиперпараметрами.

Эффект от внедрения (использования) результатов внедрения: научно-технические результаты по поддержке принятия клинических решений является основой для их рационального и эффективного использования. Данное исследование вписывается в широкомасштабный проект по цифровизации медицинской отрасли в Республике Казахстан (Kazakhstan 2050, eHealth, Digital Kazakhstan, электронный паспорт здоровья, реализация концепции Smart City) и касается реализации системы поддержки принятия клинических решений на базе искусственного интеллекта в рамках национальной системы здравоохранения. Приобретение практического опыта решения данной проблемы позволяет значительно улучшить качество подготовки специалистов по ОП «Математическое и компьютерное моделирование».

Декан ШЦТиИИ

Руководитель ОП «МиКМ»

Профессор ШЦТиИИ



З.Т.Хасенова

Ш.Т.Тезекпаева

И.М.Увалиева

ПРИЛОЖЕНИЕ К. ПОКАЗАТЕЛЬ ИНДЕКС ХИРША



Scopus Preview

Поиск авторов

Источники



Создать учетную запись

Войти

Explore this author profile on Scopus Preview

View limited highlights of a Scopus-generated author profile with Scopus Preview. To view the complete profile, check access through your organization. [Learn more about Scopus profiles.](#)

Проверить доступ

Ismukhamedova, Aigerim

D. Serikbayev East Kazakhstan Technical University, Ust-Kamenogorsk, Kazakhstan 57191621865 <https://orcid.org/0000-0002-4931-337X>

Смотреть больше

49

Цитирования из 38 документов

12

Документы

5

h-индекс [Просмотр h-диаграммы](#)

[Просмотреть все параметры >](#)

[Редактировать профиль](#) [Подробнее](#)

12 документы

Импакт

Цитирование из 38 документов

0 Препринты

11 соавторов

0 тем

0 выданных грантов

Примечание.

Пользователи Scopus Preview могут просматривать только последние 10 документов автора, и большинство других функций им недоступно. У вас есть доступ через учреждение? Воспользуйтесь доступом своего учреждения, чтобы просматривать все документы и пользоваться всеми функциями.

12 документы

[Экспортировать все](#)

[Сохранить все в список](#)

Сортировать по [Дата \(самые новые\)](#)

[> Просмотреть список в формате результатов поиска](#)

[> Просмотр приставочных ссылок](#)

[🔔 Настроить оповещение о документах](#)

Article

Experimental study of a medical data analysis model based on comparative performance of classification algorithms

0

Цитирования

Ismukhamedova, A., Uvaliyeva, I., Rakhmetullina, Z.

Indonesian Journal of Electrical Engineering and Computer Science, 2024, 36(1), страницы 672–684

[Просмотреть реферат](#) [Связанные документы](#)

Должность автора

[Проверить доступ through your organization to view author position.](#)

Article • [Открытый доступ](#)

Research and implementation of the medical text analysis algorithm for predicting mortality

0

Цитирования

First author • %

ПРИЛОЖЕНИЕ Л. СОГЛАШЕНИЯ И ЛИЦЕНЗИЙ НА ИСПОЛЬЗОВАНИЕ ДАННЫХ

PhysioNet Credentialed Health Data License 1.5.0

The PhysioNet Credentialed Health Data License

Version 1.5.0

Copyright (c) 2024 MIT Laboratory for Computational Physiology

The MIT Laboratory for Computational Physiology (MIT-LCP) wishes to make data available for research and educational purposes to qualified requestors, but only if the data are used and protected in accordance with the terms and conditions stated in this License.

It is hereby agreed between the data requestor, hereinafter referred to as the "LICENSEE", and MIT-LCP, that:

1. The LICENSEE will not attempt to identify any individual or institution referenced in PhysioNet restricted data.
2. The LICENSEE will exercise all reasonable and prudent care to avoid disclosure of the identity of any individual or institution referenced in PhysioNet restricted data in any publication or other communication.
3. The LICENSEE will not share access to PhysioNet restricted data with anyone else.
4. The LICENSEE will exercise all reasonable and prudent care to maintain the physical and electronic security of PhysioNet restricted data.
5. If the LICENSEE finds information within PhysioNet restricted data that he or she believes might permit identification of any individual or institution, the LICENSEE will report the location of this information promptly by email to PHI-report@physionet.org, citing the location of the specific information in question.
6. The LICENSEE will use the data for the sole purpose of lawful use in scientific research and no other.
7. The LICENSEE will be responsible for ensuring that he or she maintains up to date certification in human research subject protection and HIPAA regulations.
8. The LICENSEE agrees to contribute code associated with publications arising from this data to a repository that is open to the research community.
9. This agreement may be terminated by either party at any time, but the LICENSEE's obligations with respect to PhysioNet data shall continue after termination.

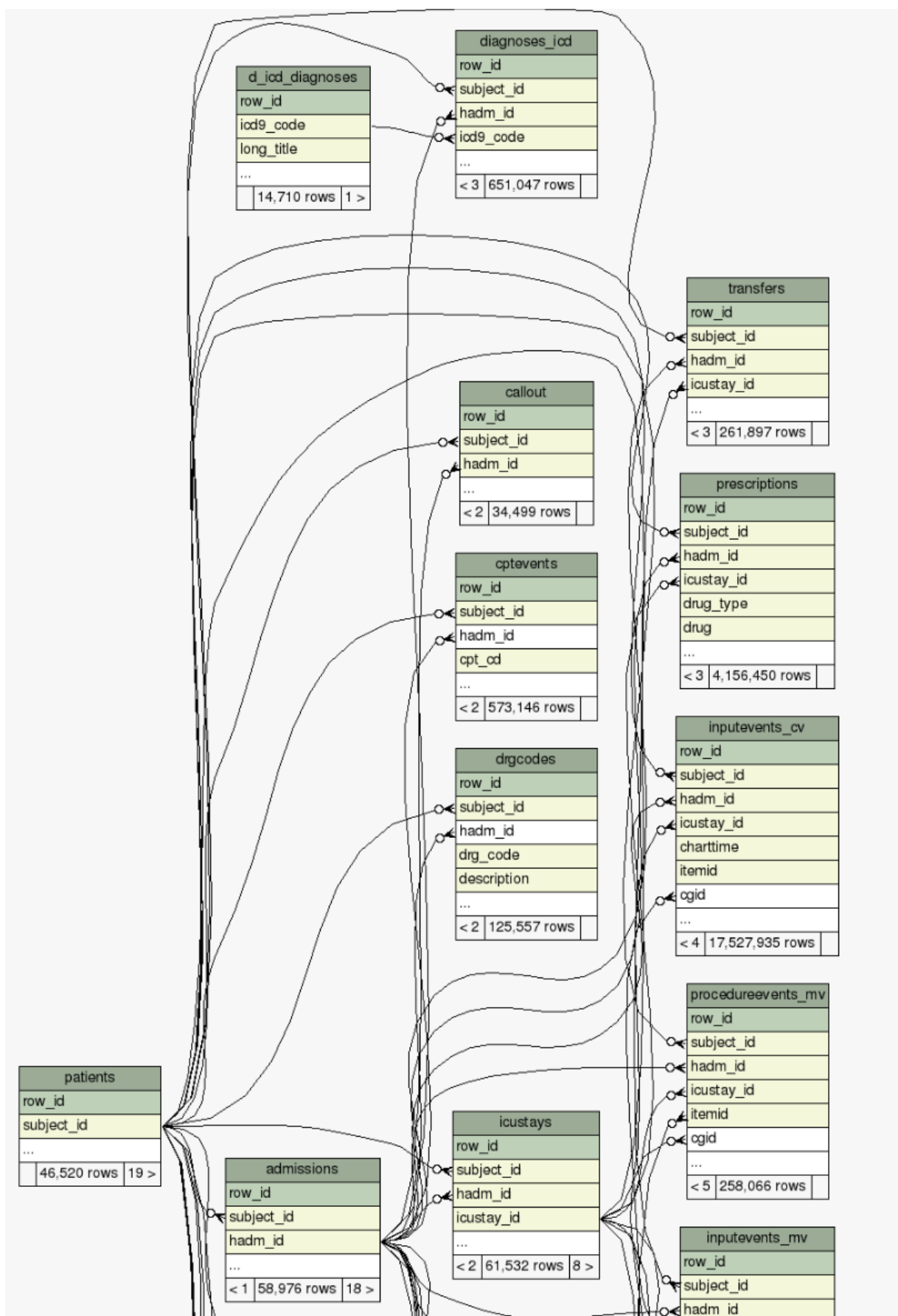
THE DATA ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DATA OR THE USE OR OTHER DEALINGS IN THE DATA.

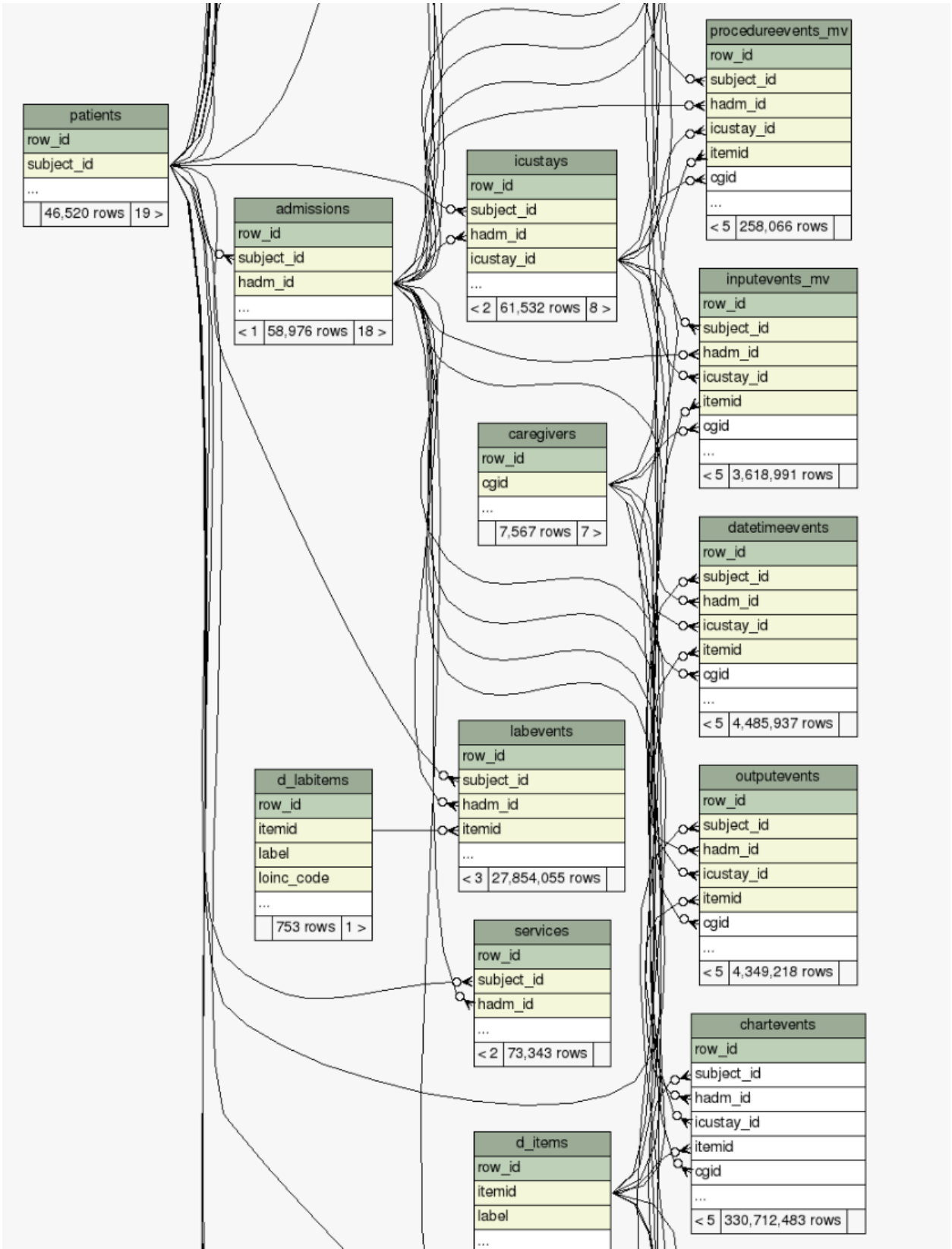
PhysioNet Credentialed Health Data Use Agreement 1.5.0

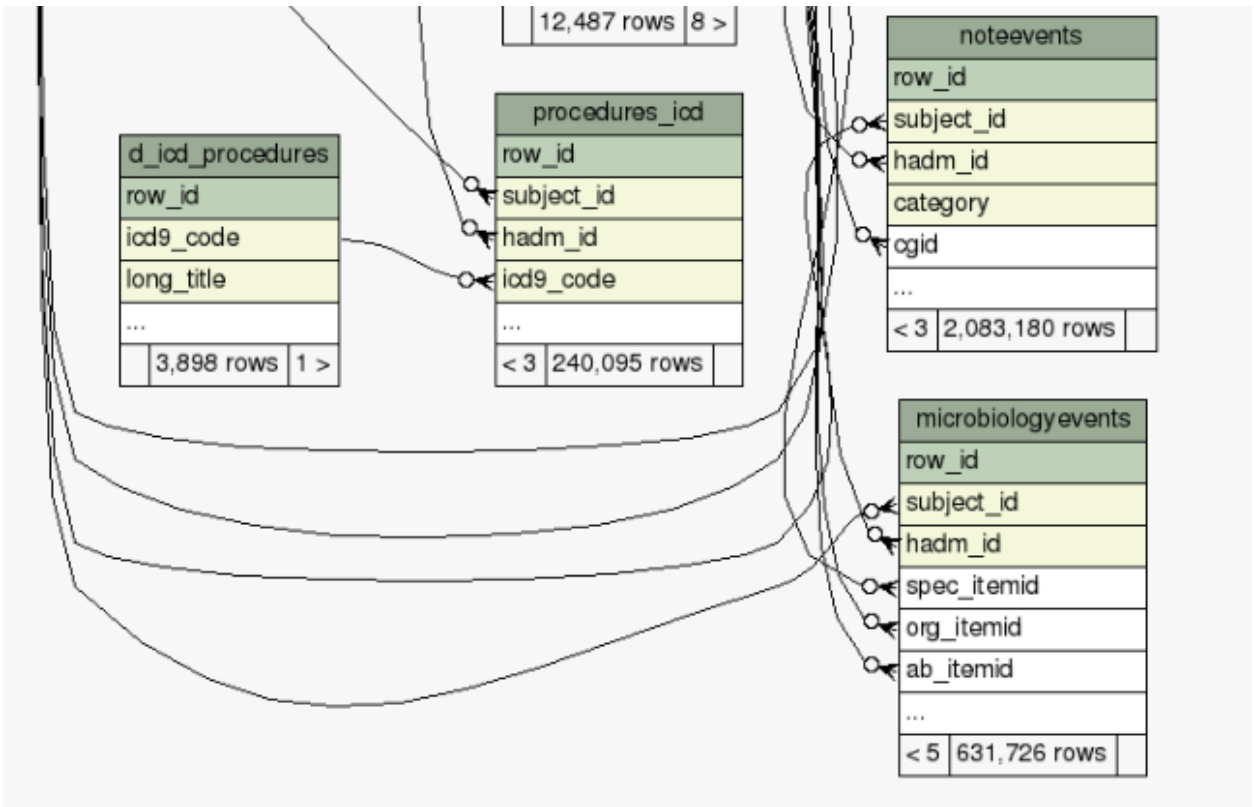
If I am granted access to the database:

1. I will not attempt to identify any individual or institution referenced in PhysioNet restricted data.
2. I will exercise all reasonable and prudent care to avoid disclosure of the identity of any individual or institution referenced in PhysioNet restricted data in any publication or other communication.
3. I will not share access to PhysioNet restricted data with anyone else.
4. I will exercise all reasonable and prudent care to maintain the physical and electronic security of PhysioNet restricted data.
5. If I find information within PhysioNet restricted data that I believe might permit identification of any individual or institution, I will report the location of this information promptly by email to PHI-report@physionet.org, citing the location of the specific information in question.
6. I have requested access to PhysioNet restricted data for the sole purpose of lawful use in scientific research, and I will use my privilege of access, if it is granted, for this purpose and no other.
7. I have completed a training program in human research subject protections and HIPAA regulations, and I am submitting proof of having done so.
8. I will indicate the general purpose for which I intend to use the database in my application.
9. If I openly disseminate my results, I will also contribute the code used to produce those results to a repository that is open to the research community.
10. This agreement may be terminated by either party at any time, but my obligations with respect to PhysioNet data shall continue after termination.

ПРИЛОЖЕНИЕ М. ТАБЛИЦА СВЯЗЕЙ БАЗЫ ДАННЫХ MIMIC III







ПРИЛОЖЕНИЕ Н. ИСХОДНЫЙ КОД

```
import numpy
import pandas as pd
def normalise_icd9_code(code) :
    return code[:3] + '.' + code[3:]

def combine_icd9_codes(groupframe) :
    groupframe =
groupframe.sort_values(by='SEQ_NUM')
    icd9_codes = ';'.join([normalise_icd9_code(x) for x
in list(groupframe.ICD9_CODE)])
    return pd.Series({'ICD9_CODE' : icd9_codes })

df_icd9_codes =
pd.read_csv('data/DIAGNOSES_ICD.csv').dropna()
df_icd9_codes =
df_icd9_codes.groupby(['SUBJECT_ID',
'HADM_ID']).apply(combine_icd9_codes)
df_icd9_codes =
pd.DataFrame(df_icd9_codes).reset_index()
df_icd9_codes.head()

def group_text_reports(groupframe) :
    #Combine main report and addenda
    groupframe = groupframe.sort_values(by=['text',
'chartdate'])
    concat_text = " ".join(groupframe['text']).strip()
    return pd.Series({'text' : concat_text})

df_notes_discharge =
pd.read_csv('data/noteevents.csv')
columns_to_keep = ['subject_id', 'HADM_ID',
'chartdate', 'text', 'TEXT']
df_notes_discharge['text'] =
df_notes_discharge['text'].replace({'Report' : 0,
'Addendum' : 1})
df_notes_discharge_combined =
df_notes_discharge.groupby(['subject_id']).apply(gro
up_text_reports)
df_notes_discharge_combined =
pd.DataFrame(df_notes_discharge_combined).reset_i
ndex()
df_notes_discharge_combined.head()
# Load the diagnosis data and the patients data
df_diag = pd.read_csv('data/diagnose_merged.csv')
df_pat = pd.read_csv('data/patients.csv')
df_pat.head(5)
id_lst = list(diabete_patients['SUBJECT_ID'])
diag_lst = []
descrip_lst = []
seq_lst = []
hadm_lst = []
for i in id_lst:
    d = df_diag[df_diag['SUBJECT_ID']==i]
    this_diag = list(d['ICD9_CODE'])
    this_des = list(d['LONG_TITLE'])
    this_seq = list(d['SEQ_NUM'])
    this_hadm = list(d['HADM_ID'])
    diag_lst.append(this_diag)
    descrip_lst.append(this_des)
    seq_lst.append(this_seq)

hadm_lst.append(this_hadm)

diabete_patients['DIAGNOSES'] = descrip_lst
diabete_patients['ICD9_CODE'] = diag_lst
diabete_patients['SEQ_NUM'] = seq_lst
diabete_patients['HADM_ID'] = hadm_lst
diabete_patients.head()
## EDA on the Diabetes Dataset
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('fivethirtyeight')
plt.figure(figsize=(17,8))
sns.countplot(diabete_patients["GENDER"],
palette='Set1')
plt.title(' Distribution of Diabetic Patients by Gender')
plt.show()
print(f"Total number of patients:
{len(diabete_patients)}")
print(f"Male:
{len(diabete_patients[diabete_patients['GENDER']
== 'M'])}")
print(f"Female:
{len(diabete_patients[diabete_patients['GENDER']
== 'F'])}")
plt.figure(figsize=(17,8))
slices = [5898, 4420]
activities = ['Male', 'Female']
cols = ['c', 'b']
plt.pie(slices, labels = activities, colors = cols,
startangle = 90, shadow = True, explode = (0, 0.1))
plt.show()
plt.figure(figsize=(17,15))
plt.title("Most Common Titles")
df_diabete['SHORT_TITLE'].value_counts().plot(kin
d='barh');
print("Percentage of Diabetes diagnosis:
'+str(len(df_diabete)/len(df_diag))")
print("Unique number of diabetes patients: '+
str(len(df_diabete['SUBJECT_ID'].unique()))")
plt.figure(figsize=(17,15))
plt.title("Diabetic and Non-Diabetic")
df_diag['Diabetes'].value_counts().plot(kind='bar');
df_diag.head()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

# Import libraries

# data analysis
import numpy as np
import pandas as pd

# data visualization
```

```

%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

# data preparation for modeling
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import FunctionTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer

# model optimization
from sklearn.model_selection import cross_val_predict, cross_val_score, cross_validate
from sklearn.model_selection import StratifiedKFold, StratifiedShuffleSplit
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV, learning_curve
from sklearn.feature_selection import SelectFromModel
from scipy.stats import randint
import itertools
from sklearn.metrics import confusion_matrix, roc_curve
from sklearn.metrics import precision_score, recall_score, f1_score

# Artificial Neural Network
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import plot_model
from tensorflow.keras.layers import Input, Dense, Dropout, AlphaDropout
from tensorflow.keras.optimizers import SGD, RMSprop, Adamax, Adagrad, Adam, Nadam
import eli5
from eli5.sklearn import PermutationImportance

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import *

from sklearn.metrics import classification_report

import matplotlib.pyplot as plt

from itertools import chain

from tensorflow.keras import utils
from tensorflow.keras.layers import Dense, Activation, Dropout, Conv2D, MaxPool2D, Flatten
from tensorflow.keras.layers import SimpleRNN, LSTM, GRU
import globalvars
import hyperas

from hyperas import optim
from hyperas.distributions import choice, uniform
from hyperopt import Trials, STATUS_OK, tpe

from tensorflow import keras
import kerastuner
from kerastuner import RandomSearch
df_adm = pd.read_csv('data/ADMISSIONS.csv')
df_diab = pd.read_csv('data/df_diag_DIABETIC_NODIABETIC.csv')

new_df = pd.merge(df_diab, df_adm, how='left', left_on=['SUBJECT_ID', 'HADM_ID'], right_on=['SUBJECT_ID', 'HADM_ID'])
new_df['ADMITTIME'] = pd.to_datetime(new_df['ADMITTIME'])
new_df['DISCHTIME'] = pd.to_datetime(new_df['DISCHTIME'])
new_df['time_admitted'] = (new_df.DISCHTIME - new_df.ADMITTIME).astype('timedelta64[h]')
cols_to_drop = ['Unnamed: 0', 'SUBJECT_ID', 'HADM_ID', 'SEQ_NUM', 'ICD9_CODE', 'ROW_ID', 'EDREGTIME', 'EDOUTTIME', 'RELIGION', 'DEATHTIME', 'ADMITTIME', 'DISCHTIME', 'LANGUAGE']
new_df.drop(cols_to_drop, axis=1, inplace=True)
new_df.head()
new_df.dropna(inplace=True)
print('Number of Patients with Diabetes : ', new_df[new_df['Diabetes']==False].shape)
print('Number of Patients without Diabetes : ', new_df[new_df['Diabetes']==True].shape)
df_diabetes = new_df[new_df['Diabetes']== True]
df_nondiabetes = new_df[new_df['Diabetes']== False]
df_nondiabetes.dropna(inplace=True)
df_nondiabetes = df_nondiabetes.iloc[:df_diabetes.shape[0], :]
finalDF = df_diabetes.append(df_nondiabetes, ignore_index=True)
print('Number of Final balanced Dataframe : ', finalDF.shape)
# Shuffle and reset index
finalDF = finalDF.sample(frac=1).reset_index(drop=True)
finalDF.head()
X = finalDF.drop(['Diabetes'], axis=1)
Y = finalDF['Diabetes']
X_get_dummy = finalDF[['ADMISSION_TYPE', 'INSURANCE', 'ADMISSION_LOCATION', 'DISCHARGE_LOCATION', 'MARITAL_STATUS']]
X_dummies = pd.get_dummies(X_get_dummy, columns=['ADMISSION_TYPE', 'INSURANCE', 'ADMISSION_LOCATION', 'DISCHARGE_LOCATION', 'MARITAL_STATUS'], drop_first=True, sparse=True)
X_num = finalDF[['HOSPITAL_EXPIRE_FLAG', 'HAS_CHARTEVENTS_DATA', 'time_admitted']]
X = pd.concat([X_num, X_dummies], axis=1)

```

```

print('Shape of Features : ', X.shape)
print('Shape of Labels : ', Y.shape)
pd.concat([X, Y], axis=1).to_csv('finalDF.csv',
index=False)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y,
test_size=0.30, random_state=30)
X_train = np.asarray(X_train).astype('float32')
X_test = np.asarray(X_test).astype('float32')
y_train = np.asarray(y_train).astype('float32')
y_test = np.asarray(y_test).astype('float32')
model = Sequential()

model.add(Dense(128, input_dim=X_train.shape[1],
activation="relu",
kernel_initializer="glorot_normal"))
model.add(Dense(64, activation="relu",
kernel_initializer="glorot_normal"))
model.add(Dense(32, activation="relu",
kernel_initializer="glorot_normal"))
model.add(Dropout(0.3))
model.add(Dense(8, activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(1, activation="sigmoid"))
model.compile(optimizer="adam",
loss='binary_crossentropy',
metrics=["binary_accuracy"])
print("=====")
print("*Model Diagnostic*")
print("=====")
print("\n")
predict = model.predict(X_test)
predict = np.argmax(predict,axis=1)
print('-----')
print('Classification Report')
print('-----')
print(classification_report(y_test, predict))
print('-----')
print('Confusion Matrix')
print('-----')
print(confusion_matrix(y_test, predict))

plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)
plt.plot(model_result.history["loss"],
label="training")
plt.plot(model_result.history["val_loss"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(model_result.history["binary_accuracy"],
label="training")
plt.plot(model_result.history["val_binary_accuracy"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

```

```

plt.show()
model = Sequential()

model.add(Dense(32, input_dim=X_train.shape[1],
activation="elu", kernel_initializer="he_normal"))
model.add(Dense(64, activation="elu",
kernel_initializer="he_normal"))
model.add(Dense(128, activation="elu",
kernel_initializer="he_normal"))
model.add(Dropout(0.3))

model.add(Dense(256, activation="elu",
kernel_initializer="he_normal"))
model.add(Dense(128, activation="elu",
kernel_initializer="he_normal"))
model.add(Dense(64, activation="elu",
kernel_initializer="he_normal"))
model.add(Dropout(0.3))

model.add(Dense(32, activation="elu",
kernel_initializer="he_normal"))
model.add(Dense(16, activation="elu",
kernel_initializer="he_normal"))
model.add(Dense(8, activation="elu",
kernel_initializer="he_normal"))
model.add(Dropout(0.3))

model.add(Dense(1, activation="sigmoid"))
model.compile(optimizer="adam",
loss='binary_crossentropy',
metrics=["binary_accuracy"])

model.summary()

print("=====")
print("*Model Diagnostic*")
print("=====")
print("\n")
predict = model.predict(X_test)
predict = np.argmax(predict,axis=1)
print('-----')
print('Classification Report')
print('-----')
print(classification_report(y_test, predict))
print('-----')
print('Confusion Matrix')
print('-----')
print(confusion_matrix(y_test, predict))

plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)
plt.plot(model_result.history["loss"],
label="training")
plt.plot(model_result.history["val_loss"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.subplot(1, 2, 2)

```

```

plt.plot(model_result.history["binary_accuracy"],
label="training")
plt.plot(model_result.history["val_binary_accuracy"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

plt.show()
from tensorflow.keras.models import Sequential,
Model
from tensorflow.keras.layers import Dense, Dropout,
BatchNormalization, Input, Multiply
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l1_l2
from sklearn.model_selection import train_test_split
as tts

def c_model():

    x = Sequential()

    x.add(Dense(38, input_dim=X_train.shape[1],
activation="elu"))
    x.add(Dense(128, activation="elu"))
    x.add(Dropout(0.3))
    x.add(Dense(256, activation="elu"))
    x.add(Dropout(0.3))
    x.add(Dense(128, activation="elu"))
    x.add(Dropout(0.3))
    x.add(Dense(64, activation="elu"))
    x.add(Dropout(0.3))
    x.add(Dense(32, activation="elu"))
    x.add(Dense(16, activation="elu"))
    x.add(Dense(8, activation="elu"))
    x.add(Dense(1, activation="sigmoid"))
    adam = Adam(learning_rate=0.001)
    x.compile(optimizer=adam,
loss='binary_crossentropy', metrics=["accuracy"])
    return x

import matplotlib.pyplot as plt
def plotter(history, n):
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('MODEL ACCURACY #%i' %n)
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper right')
    plt.ylim(top=1, bottom=0.01)
    plt.savefig('history_accuracy_{ }.png'.format(n))
    plt.show()

    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('MODEL LOSS #%i' %n)
    plt.ylabel('Loss')
    plt.xlabel('Epoch')

    plt.legend(['Train', 'Test'], loc='upper right')
    #plt.ylim(top=2, bottom=0.01)
    plt.savefig('history_loss_{ }.png'.format(n))
    plt.show()

from tensorflow.keras.callbacks import import
ReduceLRonPlateau, EarlyStopping,
ModelCheckpoint
lrr = ReduceLRonPlateau(monitor = 'accuracy',
patience = 50,
verbose = 1,
factor = 0.5,
min_lr = 1e-8)

es = EarlyStopping(monitor='accuracy',
mode='max',
verbose=0,
patience=750,
restore_best_weights=True)

folds = 5
epochs = 1000
batch_size = X.shape[0]

train_history = []
all_predictions = None
all_scores = []
Xt = np.asarray(X).astype(np.float32)
Yt = np.asarray(Y).astype(np.float32)

Xtest = np.asarray(X_test).astype(np.float32)
Ytest = np.asarray(y_test).astype(np.float32)

for n in range(1, folds+1):

    print(f"Currently training on Fold: {n}")

    xt, xv, yt, yv = tts(Xt, Yt, test_size=0.2,
random_state=1771, shuffle=True, stratify=Yt)
    model = c_model()
    hist = model.fit(xt, yt, validation_data=(xv, yv),
epochs=epochs, batch_size=batch_size,
callbacks=[lrr, es], verbose=0)

    train_history.append(hist)
    plotter(hist, n)

    loss, acc = model.evaluate(xv, yv)
    predicted = model.predict(Xtest)

    loss2, acc2 = model.evaluate(xv, yv)

    if acc < acc2 or (acc==acc2 and loss < loss2):
        predicted = model.predict(Xtest)
        loss, acc = loss2, acc2

    all_scores.append([loss, acc])

    if acc > .77:

```



```

try:
    all_predictions += predicted*acc
except:
    all_predictions = predicted*acc

def create_model(input_shape=X_train.shape[1:],
                 number_hidden=4,
                 neurons_per_hidden=32,
                 hidden_drop_rate= 0.2,
                 hidden_activation = 'selu',
                 hidden_initializer='lecun_normal',
                 output_activation = 'sigmoid',
                 loss='binary_crossentropy',
                 optimizer = Nadam(lr=0.0005),
                 #lr=0.0005,
                 ):

    #create model
    model = Sequential()
    model.add(Input(shape=input_shape)),
    for layer in range(number_hidden):
        model.add(Dense(neurons_per_hidden,
            activation = hidden_activation
            ,kernel_initializer=hidden_initializer))
        model.add(Dropout(hidden_drop_rate))
        model.add(Dense(neurons_per_hidden/2,
            activation = hidden_activation
            ,kernel_initializer=hidden_initializer))
        model.add(Dropout(hidden_drop_rate))
        model.add(Dense(neurons_per_hidden/4,
            activation = hidden_activation
            ,kernel_initializer=hidden_initializer))
        model.add(Dropout(hidden_drop_rate))

    model.add(Dense(1, activation =
output_activation))

    # Compile model
    model.compile(loss=loss,
        #optimizer = Nadam(lr=lr),
        optimizer = Adam(lr=0.0005),
        metrics = ['accuracy'])
    return model

print("=====")
print("*Model Diagnostic*")
print("=====")
print("\n")
predict = dnn_clf.predict(X_test)
predict = np.argmax(predict,axis=1)
print('-----')
print('Classification Report')
print('-----')
print(classification_report(y_test, predict))
print('-----')
print('Confusion Matrix')
print('-----')
print(confusion_matrix(y_test, predict))

plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)

plt.plot(history.history["loss"], label="training")
plt.plot(history.history["val_loss"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history["accuracy"], label="training")
plt.plot(history.history["val_accuracy"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

plt.show()

# calculate probabilities for training dataset X_train
y_train_proba_dnn = dnn_clf.predict(X_train)
proba_df=pd.DataFrame(y_train_proba_dnn,
columns=['probabilities'])*100
proba_df['false_predictions']=(pd.DataFrame(y_train
_pred_dnn)-pd.DataFrame(y_train))[0]
proba_df['probabilities'][proba_df['false_predictions']
==0].hist()
# The high number to the left (around 0) and to the
right (around 1) suggests that the classifier was quite
certain in most of the predictions that turned out to be
right.
proba_df['probabilities'][proba_df['false_predictions']
!=0].hist()
np.random.seed(42)
tf.random.set_seed(42)

# build and train KerasClassifier
dnn_clf_fit=KerasClassifier(build_fn = create_model)
dnn_clf_fit.fit(X_train, y_train, epochs=1000,
batch_size=32)

# calculate feature importance
perm = PermutationImportance(dnn_clf_fit,
random_state=42).fit(X_train,y_train)

print("=====")
print("*Model Diagnostic*")
print("=====")
print("\n")
predict = dnn_clf_fit.predict(X_test)
predict = list(chain.from_iterable(predict))
print('-----')
print('Classification Report')
print('-----')
print(classification_report(y_test, predict))
print('-----')
print('Confusion Matrix')
print('-----')
print(confusion_matrix(y_test, predict))

plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)
plt.plot(history.history["loss"], label="training")

```

```

plt.plot(history.history["val_loss"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history["accuracy"], label="training")
plt.plot(history.history["val_accuracy"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

plt.show()

def create_model(input_shape=X_train.shape[1:],
                number_hidden=4,
                neurons_per_hidden=32,
                hidden_drop_rate= 0.2,
                hidden_activation = 'selu',
                hidden_initializer="lecun_normal",
                output_activation = 'sigmoid',
                loss='binary_crossentropy',
                optimizer = Nadam(lr=0.0005),
                #lr=0.0005,
                ):

    #create model
    model = Sequential()
    model.add(Input(shape=input_shape)),
    for layer in range(number_hidden):
        model.add(Dense(neurons_per_hidden,
activation = hidden_activation
,kernel_initializer=hidden_initializer))
        model.add(Dropout(hidden_drop_rate))
        model.add(Dense(neurons_per_hidden/2,
activation = hidden_activation
,kernel_initializer=hidden_initializer))
        model.add(Dropout(hidden_drop_rate))
        model.add(Dense(neurons_per_hidden/4,
activation = hidden_activation
,kernel_initializer=hidden_initializer))
        model.add(Dropout(hidden_drop_rate))

    model.add(Dense(1, activation =
output_activation))

    # Compile model
    model.compile(loss=loss,
                #optimizer = Nadam(lr=lr),
                optimizer = optimizer,
                metrics = ['accuracy'])
    return model

# summarize history for accuracy
# training data: default is X_train as used before
X_train_gs=X_train.copy()
#X_train_gs=X_train_bf.copy() #best feature dataset

# test data: default is X_train as used before
X_test_gs=X_test.copy()

keras.backend.clear_session()
np.random.seed(42)
tf.random.set_seed(42)

# build classifier
dnn_clf_gs = KerasClassifier(build_fn =
create_model, verbose = 2)

# define parameter grid:
# uncomment the parameters you want to optimize
param_grid = {
    "optimizer": ['Adam', 'Adagrad', 'Adamax'],
    #"lr":[0.1, 0.01,0.001,0.0001],
    #"epochs": [15, 30, 45, 60],
    #"batch_size": [20,30,40],
    #"number_hidden": [1, 2, 3],
    #"neurons_per_hidden": [5, 10, 15],
    #'input_shape': X_train_gs.shape[1:] # keep this
line
}

# build GridSearchCV model with ANN classifier
grid_search_dnn = GridSearchCV(dnn_clf_gs,
param_grid, cv=5, verbose=0,
return_train_score=True)

# fit GridSearchCV model
training_gs = grid_search_dnn.fit(X_train_gs, y_train,
epochs = 1000,
batch_size = 100,
validation_split=0.2,
shuffle=True,
)

plt.figure(figsize = (17,8))
plt.plot(grid_search_dnn.best_estimator_.model.histo
ry.history['accuracy'], color='red')
plt.plot(grid_search_dnn.best_estimator_.model.histo
ry.history['val_accuracy'], color='green')
plt.plot(grid_search_dnn.best_estimator_.model.histo
ry.history['loss'], color='red')
plt.plot(grid_search_dnn.best_estimator_.model.histo
ry.history['val_loss'], color='green')
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epochs')
plt.legend(['train accuracy/loss', 'validation
accuracy/loss'], loc='best')
plt.show()

X_train, X_test, y_train, y_test = train_test_split(X, Y,
test_size=0.30, random_state=30)
y_train_copy, y_test_copy = y_train.copy(),
y_test.copy()

print('X_train Shape : ',X_train.shape)
print('X_test Shape : ',X_test.shape)
train_value=X_train.values.reshape(22356, 38)
test_value=X_test.values.reshape(9582, 38)

print('trainvalue',train_value.shape)

```

```

print('testvalue',test_value.shape)

a=train_value
train=np.reshape(a,(-1,38,1,1))
b=test_value
test=np.reshape(b,(-1,38,1,1))

print('xtrain',train.shape)
print('xtest',test.shape)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,
MaxPool2D

time_steps=15
input_size=1
cell_size=50

# One-hot encoding
ytrain=utils.to_categorical(y_train,num_classes=2)
ytest=utils.to_categorical(y_test,num_classes=2)

train = np.asarray(train).astype('float32')
test = np.asarray(test).astype('float32')
ytrain = np.asarray(ytrain).astype('float32')
ytest = np.asarray(ytest).astype('float32')

model_CNN=Sequential()
model_CNN.add(Convolution2D(input_shape=(38,1,
1),filters=32,
kernel_size=5,strides=1,padding='same',activation='r
elu'))
model_CNN.add(MaxPool2D(pool_size=2,
strides=2, padding='same'))
model_CNN.add(Convolution2D(64,5,strides=1,padd
ing='same',activation='relu'))
model_CNN.add(MaxPool2D(2,2, 'same'))
model_CNN.add(Flatten())
model_CNN.add(Dense(256,activation='relu'))
model_CNN.add(Dropout(0.4))
model_CNN.add(Dense(128,activation='relu'))
model_CNN.add(Dropout(0.4))
model_CNN.add(Dense(32,activation='relu'))
model_CNN.add(Dropout(0.4))
model_CNN.add(Dense(16,activation='relu'))
model_CNN.add(Dropout(0.4))
model_CNN.add(Dense(2,activation='softmax'))

# sgd=SGD(lr=0.3)
adam=Adam(lr=1e-4)
model_CNN.compile(optimizer=adam,loss='categori
cal_crossentropy',metrics=['accuracy'])

model_CNN.summary()
history =
model_CNN.fit(train,ytrain,batch_size=100,epochs=
500, validation_data=[test, ytest])
loss,accuray=model_CNN.evaluate(test,ytest)

print('\Test Loss',loss)
print('Test Accuray',accuray)
print("=====")

print("**Model Diagnostic**")
print("=====")
print('\n')
predict = model_CNN.predict(test)
predict = np.argmax(predict,axis=1)
print('-----')
print('Classification Report')
print('-----')
print(classification_report(y_test_copy, predict))
print('-----')
print('Confusion Matrix')
print('-----')
print(confusion_matrix(y_test_copy, predict))

plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)
plt.plot(history.history["loss"], label="training")
plt.plot(history.history["val_loss"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history["accuracy"], label="training")
plt.plot(history.history["val_accuracy"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

plt.show()

X_train, X_test, y_train, y_test = train_test_split(X, Y,
test_size=0.30, random_state=30)
y_train_copy, y_test_copy = y_train.copy(),
y_test.copy()
print('X_train',X_train.shape)
print('y_train',y_train.shape)

train_value= X_train.values.reshape(22356, 38)
test_value= X_test.values.reshape(9582, 38)

print('trainvalue',train_value.shape)
print('testvalue',test_value.shape)

import numpy as np
a=train_value
train=np.reshape(a,(-1,38,1))
b=test_value
test=np.reshape(b,(-1,38,1))

print('xtrain',train.shape)
print('xtest',test.shape)
import tensorflow.keras as keras
time_steps=15
input_size=1
cell_size=50

#parameters for LSTM
nb_lstm_outputs = 50

```

```

nb_time_steps = 38
nb_input_vector = 1

y_train=utils.to_categorical(y_train,num_classes=2)
y_test=utils.to_categorical(y_test,num_classes=2)

#build model_LSTM

model_LSTM = Sequential()
model_LSTM.add(LSTM(units=nb_lstm_outputs,
input_shape=(nb_time_steps, nb_input_vector)))
model_LSTM.add(Dense(32,activation='relu',
kernel_initializer = 'glorot_normal'))
model_LSTM.add(Dropout(0.2))
model_LSTM.add(Dense(16,activation='relu'))
model_LSTM.add(Dropout(0.2))
model_LSTM.add(Dense(8,activation='relu'))
model_LSTM.add(Dropout(0.2))
model_LSTM.add(Dense(2,activation='softmax'))
adam=Adam(lr=1e-3)
model_LSTM.compile(optimizer=adam,loss='categorical_crossentropy',metrics=['accuracy'])
model_LSTM.summary()
history =
model_LSTM.fit(train,ytrain,batch_size=512,epochs
=500, validation_data=[test, ytest])

loss,accuracy=model_LSTM.evaluate(test,ytest)

print('\Test Loss',loss)
print('Test Accuray',accuracy)
print("=====")
print("*Model Diagnostic*")
print("=====")
print("\n")
predict = model_LSTM.predict_classes(test)
print('-----')
print('Classification Report')
print('-----')
print(classification_report(y_test_copy, predict))
print('-----')
print('Confusion Matrix')
print('-----')
print(confusion_matrix(y_test_copy, predict))

plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)
plt.plot(history.history["loss"], label="training")
plt.plot(history.history["val_loss"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history["accuracy"], label="training")
plt.plot(history.history["val_accuracy"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

plt.show()

#License: MIT

plt.show()

time_steps=38
input_size=1
cell_size=50

model_RNN=Sequential()
model_RNN.add(SimpleRNN(units=cell_size
,input_shape=(time_steps,input_size)))
model_RNN.add(Dense(64, activation="relu"))
model_RNN.add(Dense(32, activation="relu"))
model_RNN.add(Dense(16, activation="relu"))
model_RNN.add(Dense(8, activation="relu"))
model_RNN.add(Dense(2,activation='softmax'))
adam=Adam(lr=1e-3)
model_RNN.compile(optimizer=adam,loss='categorical_crossentropy',metrics=['accuracy'])

model_RNN.summary()
history =
model_RNN.fit(train,ytrain,batch_size=256,epochs=
500, validation_data=[test, ytest])

loss,accuracy=model_RNN.evaluate(test,ytest)

print('\Test Loss',loss)
print('Test Accuray',accuracy)
print("=====")
print("*Model Diagnostic*")
print("=====")
print("\n")
predict = model_RNN.predict_classes(test)
print('-----')
print('Classification Report')
print('-----')
print(classification_report(y_test_copy, predict))
print('-----')
print('Confusion Matrix')
print('-----')
print(confusion_matrix(y_test_copy, predict))

plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)
plt.plot(history.history["loss"], label="training")
plt.plot(history.history["val_loss"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history["accuracy"], label="training")
plt.plot(history.history["val_accuracy"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

plt.show()
#License: MIT

```

```

from deepstack.base import KerasMember

member1 = KerasMember(name="model_LSTM",
keras_model=model_LSTM, train_batches=(train,
ytrain), val_batches=(test, ytest))
member2 = KerasMember(name="model_RNN",
keras_model=model_RNN, train_batches=(train,
ytrain), val_batches=(test, ytest))

from deepstack.ensemble import DirichletEnsemble

wAvgEnsemble = DirichletEnsemble()
wAvgEnsemble.add_members([member1, member2])
wAvgEnsemble.fit()
wAvgEnsemble.describe()

from scipy import stats
models =[model_RNN,model_RNN]

# Predict labels with models
labels = []
for m in models:
    predicts = np.argmax(m.predict(test), axis=1)
    labels.append(predicts)
# Ensemble with voting
labels = np.array(labels)
labels = np.transpose(labels, (1, 0))
labels = stats.mode(labels, axis=1)[0]
labels = np.squeeze(labels)
print("=====")
print("*Model Diagnostic*")
print("=====")
print("\n")
print('-----')
print('Classification Report')
print('-----')
print(classification_report(y_test_copy, labels))
print('-----')
print('Confusion Matrix')
print('-----')
print(confusion_matrix(y_test_copy, labels))

plt.figure(figsize=(30, 10))

plt.subplot(1, 2, 1)
plt.plot(history.history["loss"], label="training")
plt.plot(history.history["val_loss"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(history.history["accuracy"], label="training")
plt.plot(history.history["val_accuracy"],
label="validation")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

plt.show()
# Sort lengths

lns.sort()
# Take 5% as the removal size
rm_size = int(len(lns) / 100) * 5

# Now plot with removal of most/least frequent
sns.distplot(lns[rm_size:-rm_size], kde=False,
axlabel='Document length')
plt.show()

# Remove rows from the dataframe based on
document length, this is not really
#straightforward, so we'll approximate it and find the
document length that is used as a cutoff
min_ln = max(lns[0:rm_size])
max_ln = min(lns[-rm_size:])

noteevents = noteevents[[True if len(str(x)) > min_ln
and len(str(x)) < max_ln else False for x in
noteevents['text']]]
noteevents.head()
# Again a bit of clean-up, let's remove the bottom/top
1% of patients based on the number of
#documents they have.
docs_per_pt = noteevents['subject_id'].value_counts()
docs_per_pt_vals = docs_per_pt.values
docs_per_pt_vals.sort()

rm_size = int(len(docs_per_pt_vals) / 100) * 1
min_ln = max(docs_per_pt_vals[0:rm_size])
max_ln = min(docs_per_pt_vals[-rm_size:])

keep_subject_id = set([k for k, v in
docs_per_pt.items() if v > min_ln and v < max_ln])
noteevents = noteevents[[True if subject_id in
keep_subject_id else False
for subject_id in
noteevents['subject_id'].values]]
noteevents.head()
print(f"Length after cleaning : {len(noteevents)}")
print(f"Length of the original:
{len(noteevents_original)}")
sns.distplot(noteevents['subject_id'].value_counts().v
alues, kde=False, axlabel='Documents per patient')
plt.show()
# Convert to pandas dates
noteevents['chartdate'] =
pd.to_datetime(noteevents['chartdate'])
patients['dob'] = pd.to_datetime(patients['dob'])

# Add a year column
noteevents['create_year'] =
pd.DatetimeIndex(noteevents['chartdate']).year
patients['dob_year'] =
pd.DatetimeIndex(patients['dob']).year
# Joint noteevents with patients
pt_notes = noteevents.merge(patients, on='subject_id',
how='left')
# Remove patients older than 89 and younger than 16
pt_notes = pt_notes[pt_notes['age_year'] >= 16]
pt_notes = pt_notes[pt_notes['age_year'] <= 89]
# It is possible that the cleaning above created some
patients with only one document

```

```

print("Number of patients with only one doc: " +
str(sum(pt_notes['subject_id'].value_counts().values
== 1)))

# Remove it there are any
remove_subject = set([k for k, v in
pt_notes['subject_id'].value_counts().items() if v ==
1])
pt_notes =
pt_notes[~pt_notes.subject_id.isin(remove_subject)]
print("After removal: " +
str(sum(pt_notes['subject_id'].value_counts().values
== 1)))

# We'll do the following to make sure patients do not
have documents that span
#over multiple years, meaning their age would change.
dif_pt = {}
for ind, row in pt_notes.iterrows():
    sid = row['subject_id']
    if sid in dif_pt:
        dif_pt[sid].append(row['age_year'])
    else:
        dif_pt[sid] = [row['age_year']]

ehr_length = []
median_age = []
for v in dif_pt.values():
    mx = max(v)
    mi = min(v)
    median_age.append(np.median(v))
    ehr_length.append(mx - mi)

```