Оглавление

Введение

В современном мире развивающихся технологий мы сталкиваемся с проблемой роста количества информации, которую необходимо обрабатывать. В отдельных задачах массивы входных данных настолько велики, что классические методы обработки оказываются крайне неэффективны, так как требуют колоссальных вычислительных мощностей и тщательного контроля результата. Причём сложность может возрастать едва ли не экспоненциально с ростом как количества, так и разрядности входных значений. Для решения этой проблемы и обеспечения наукоёмких сфер информационных технологий эффективными методами анализа и обработки данных в настоящее время всё активнее используют машинное обучение.

Особенностью данного класса методов является отсутствие прямого решения задачи, вместо которого используется обучение в процессе применения к решению множества сходных задач. Для построения таких методов используются средства теории вероятностей, математической статистики, численных методов, теории графов, методов оптимизации и различные техники работы с данными в цифровой форме.

Машинное обучение получает все более широкое распространение. С его помощью уже на сегодняшний день планируют транспортное движение в мегаполисах, определяют различные социальный и экономические факторы, проводят сложные научные вычисления. Даже в медицинской сфере с его помощью проводят анализ экспериментальных данных, белков ДНК и многое другое.

Одной из важных медицинских задач является облегчения контроля уровня глюкозы в крови, столь необходимого больным диабетом. Количество случаев заболевания сахарным диабетом из года в год неуклонно растет. Согласно статистике ВОЗ количество больных возросло с 108 миллионов в 1980 году до 422 миллионов в 2014 году. А с 2014 года по 2016 количество увеличилось еще на 23%.

Сахарный диабет опасен, и при отсутствии контроля может вызывать различный осложнения, как ранние, такие как потеря сознания и кома, так и поздние в виде потери зрения и многократного повышения вероятности тромбоза и атеросклероза.

Врачебная практика и статистика показывают, что чем регулярнее осуществляется контроль уровня глюкозы в крови, тем меньше риск наступления осложнений сахарного диабета. В связи с этим пациенты вынуждены проводить его самостоятельно, ежедневно и по несколько раз, и на основе полученной информации корректировать свою диету, нагрузки и при необходимости принимать инсулин и другие снижающие сахар препараты.

На сегодняшний день нет достаточно точных неинвазивных методов. Большинство приборов для измерения уровня глюкозы требуют нарушения кожных покровов для забора

крови, что крайне некомфортно при регулярных измерениях. Приборы же, позволяющие определять уровень глюкозы без проколов в свою очередь требую дорогостоящих расходных материалов.

1. Техническое задание

1.1. Назначение разработки и область применения

В работе была поставлена цель разработать программную систему распознавания сигналов на основе гистограмм полных и замкнутых групп.

В результате планируется получить программный комплекс, который принимая данные ЭКГ будет предсказывать текущий уровень глюкозы в крови и будет применим в следующих областях:

1) Индивидуальный контроль уровня глюкозы, проводимый самими пациентами вне больниц;

2) Измерение уровня глюкозы у детей ввиду отсутствия необходимости совершения проколов кожи;

3) Возможность самообучения вслед за изменением индивидуальных особенностей пациента;

4) Возможность использования системы предсказаний на домашнем компьютере.

Разрабатываемая система предназначена для использования на домашних стационарных и портативных компьютерах при наличии данных ЭКГ.

1.2. Технические требования

Определим требования разработки, предъявляемые к ЭВМ:

1) Компьютер под управлением Microsoft Windows версии не ниже Vista, и соответствующий минимальным требованиям операционной системы;

2) Клавиатура, мышь;

3) Доступ к сети Интернет.

Рассмотрим, необходимые функционал для системы определения уровня глюкозы в крови по данным ЭКГ:

1) В начале работы система должна, приняв данные ЭКГ, составить систему признаков согласно теории активного восприятия, и подготовить их для дальнейшего использования;

2) Система должна используя выбранный признак произвести обучение на основе подготовленных данных, состоящих из строки показаний признака и соответствующего значения уровня глюкозы;

3) Произведя обучение система должна сохранить полученную нейросеть для дальнейшего использования;

4) При необходимости предсказания текущего уровня глюкозы система должна принять данные ЭКГ, подготовить систему признаков и сохранить их для дальнейшего использования;

5) использовав подготовленные данные и обученную нейросеть произвести предсказания уровня глюкозы и вывести результат в удобной для пользователя форме.

2. Анализ технического задания.

2.1 Выбор операционной системы

Важным этапом является выбор операционной системы. В дальнейшем это решение повлияет как на удобство разработки самого программного комплекса, выбора перечня необходимых для разработки инструментов, так и на скорость, стабильность его работы и комфорт работы конечного пользователя с программой.

Рассмотрим самые распространённые операционные системы, а именно Linux, Mac OS и Windows:

1) Linux. Общее название для всех Unix-подобных систем основанных на ядре Linux Kernel. Впервые была опубликована в 1991 году и являлась на тот момент первой операционной системой с полностью открытым исходным кодом. Поскольку Linux Kernel лишь основа операционной системы существует большое количество дистрибутивов на его основа, отличающиеся как по надежности и безопасности, так и по входящим в их состав компонентам. Распространяются они как правило бесплатно и большинство из них имеет обширные сообщества в которых можно найти ответы на все возникающие в процессе работы вопросы. Благодаря этому, а также гибкости настроек самой системы популярность Linux со временем только растет.

2) Mac OS. Операционная система на основе Unix разработанная компанией Apple, впервые вышла на рынок в 1984 году в составе компьютера Macintosh под названием System Software и позднее была переименована в Mac OS. В отличии от Linux, несмотря на общую основу не является системой с открытым исходным кодом.

3) Windows. Наиболее распространённая на сегодняшний день ОС, производится и поддерживается компанией Microsoft с 1985 года. Является первой операционной системой, ориентированной на графический интерфейс и считается наиболее привычной и дружелюбной к пользователю.

Рассмотрим их преимущества, особенности и недостатки:

1) Стабильность работы.

Наиболее стабильной операционной системой является Linux. Возможность тонкой настройки установленных пакетов, контроль за работающими программами, простота и больше количество документации благоприятно сказывается на времени безотказной работы. На втором месте находится Mac OS, однако регулярно встречаются зависания отдельных приложений. На последнем месте находятся операционные системы Windows, встречаются как зависания

отдельных приложений, так и ошибки системы требующие перезагрузки для исправления. Однако при грамотном администрировании вероятность отказом можно свести в минимум.

Ни одна из рассмотренных операционных систем не имеет критических проблем со стабильностью работы и каждая из них пригодна для каждодневного использования.

2) Безопасность.

При оценке такого параметра как безопасность прежде всего имеется в виду защищенность от внешних воздействий и неправомерного доступа к данным. Операционная система Windows проигрывает как Mac OS, так и Linux. Однако связано это с большей распространённостью операционных систем от Microsoft, а, следовательно, и вредоносного ПО предназначенного для них. Mac OS традиционно считается достаточно защищенной, однако отсутствие доступа к исходному коду не позволяет своевременно обнаруживать уязвимости, что отрицательно сказывается на безопасности. Linux же является наиболее защищенной ОС, однако требующий сложной настройки для реализации потенциала защищенности.

Оценка безопасности хоть и является важным параметром операционной системы, однако не критично для поставленных перед нами задач.

3) Прикладное программное обеспечение.

По распространённости и доступности прикладного программного обеспечения все три рассматриваемые операционные системы примерно равны. Каждая из них достаточно распространена чтобы иметь широкий выбор различных как офисных, так и прочих программ. Однако Mac OS все же проигрывает Linux и Windows, так как они имеют более обширные библиотеки программ благодаря распространённости и доступности.

4)Специальное программное обеспечение

Все представленные операционные системы имеют свои инструменты разработки. Однако наибольшее их число в операционных системах Windows. Так же наибольшее число новых версий программ выходим именно для этой операционной системы и лишь в последствии появляются в других. Остальные рассматриваемые системы имеют возможность портирования ПО либо специализированные версии, однако повышаются накладные расходы на разработку, в виде решении проблем совместимости, отсутствии библиотек и понижение производительности в режиме совместимости, а также неактуальности версий специального ПО.

Данный параметр ОС наиболее важен при разработке, так как именно доступность, простота и производительность инструментов важна при написании программного комплекса данной работы.

Для разработки программной системы определения уровня глюкозы по данным ЭКГ была выбрана операционная система Windows. Решающим фактором стала её распространённость, что уменьшает расходы на портирование и наличие широкого диапазона актуальных средств разработки. Данная операционная система наиболее полно соответствует требованиям разработки этого проекта.

2.2 Выбор языка программирования

Важным этапом любой разработки является выбор языка программирования. В рамках работы была поставлена задача разработать программный комплекс, состоящий из двух различных по логике работы частей. Первая часть комплекса предназначена для определения система признаков согласно теории активного восприятия из исходной электрокардиограммы, вторая же часть предназначена для создания, обучения и использования нейросетей на основе полученных в первой части признаков. Соответственно будет разумно выбрать два различных подхода к этим задачам и, возможно, выбрать различные языки программирования.

Определим язык для создания системы признаков по данным ЭКГ. Наиболее подходящими к задачам поставленным перед первой частью программного комплекса являются такие языки как C#, Python и R.

Рассмотрим каждый из них:

1) C# является объектно-ориентированным языком программирования. Он был разрабатывался с 1998 до 2001 года группой инженеров компании Microsoft для платформы .Net Framework. Относится с C-подобным языкам программирования и наиболее близок к Java и C++. Поддерживает все основные парадигмы ООП. Также является языком прикладного уровня и потому зависит от возможностей прикладной среды. На данный момент актуальной является версия 7.0.

2) Python представляет собой высокоуровневый язык общего назначения, задуманный как язык повышающий производительность разработчика и улучшающий читаемость кода за счет минималистичного синтаксиса. На данный момент активно развивается, а новые версии выходят с промежутком примерно в два с половиной года. Python портирован почти на все известные платформы.

3) R – язык предназначенный для статистической обработки данных и работы с графикой и широко используется как статистическое программное обеспеченье для анализа данных. Огромное количество разнообразных библиотек, предназначенных для работы с различными типами баз данных и методов их обработки, способствуют все более широкому распространению этого языка, который уже де-факто является стандартом для статистических программ.

С учетом специфики поставленной задачи наиболее подходящим будет язык R. Решающим фактором для такого решения стало наличие специальных библиотек для работы со звуковыми файлами [1].

Рассмотрим варианты наиболее подходящих языков программирования для создания, обучения и использования нейросетей. Для сравнения были выбраны три различных решения: это классические C++ и MATLAB, и новый Azure ML.

Опишем каждый из них более подробно:

1) C++ является языком программирования общего назначения со статической типизацией. Является одним из самых популярных языков программирования и был создан в 1980 году как дополнение к языку C. На данный момент язык продолжает развивается и отвечает современным требованиям. В последних обновлениях в него были добавлены некоторые особенности метапрограммирования. Программы, написанные на этом языке, отличаются высокой скоростью работы и эффективностью.

2) MATLAB представляет из себя как пакет программ, так и язык используемый в них, который является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования. Все программы написанные на MATLAB представляют из себя либо функции, либо скрипты. Первые имею раздельное пространство для промежуточных элементов и переменных, у вторых же оно общее. Особенностью языка так же является широкие возможности обработки матриц.

3) Azure ML — это облачная служба прогнозной аналитики, которая позволяет быстро создавать и развертывать прогнозные модели в качестве решений аналитики. Служба машинного обучения Azure содержит все необходимое для создания полных решений прогнозной аналитики в облаке — от большой библиотеки алгоритмов до студии для создания моделей и удобных функций развертывания моделей в виде веб-служб. Обладает возможностью быстро создавать, тестировать и вводить в эксплуатацию прогнозные модели, управлять ими, а также включать элементы исходного кода на R и Python. Сервис машинного обучения Azure Machine Learning в настоящее время находится в предварительном публичном тестировании и доступном каждому. Так же есть возможность предоставлять доступ к использованию обученных моделей нейросетей через веб интерфейс или с использованием ML API Service.

Из рассмотренных вариантов наиболее удобным инструментом для реализации поставленных задач является Azure Machine Learning. Ключевыми достоинствами является возможность использования обученных моделей через API, а также простота и скорость разработки нейросетей [2].

2.3 Выбор среды разработки

После выбора основных языков для реализации поставленных задач необходимо определится со средой разработки. Это важное решение, так как от него зависит удобство программирования, а соответственно и скорость реализации проекта. Поскольку Azure Machine Learning является облачным решением то нет необходимости в использовании локальной среды разработки, весь необходимый для реализации функционал доступен через веб-интерфейс. Присутствуют модули для инициализации модели, обучения, проверки производительности и валидности и оценки полученной нейросети.

Azure ML предлагает множество алгоритмов для классификации задач, включая Multiclass и Two-Class Decision Forests, Decision Jungles, Logistic Regression, Neural Networks, а так же Two-Class Averages Perceptrons, Bayes Point Machine, Boosted Decision Trees и Support Vector Machines (SVM). Кластеризация использует вариацию стандартного K-Means подхода. Регрессии включают Bayesian Linear, Boosted Decision Trees, Decision Forests, конечно Linear Regression, Neural Network Regression, Ordinal и Poisson Regression [2]. Так же есть возможность применять статистические функции, такие как вычисление отклонений и оценки корреляции. Предоставляется функционал для визуализации данных, построения графиков и диаграмм.

Для языка R выбор сред разработки достаточно обширен, начиная от непосредственной среды выполнения, поставляемой вместе с языком, так и средства сторонних разработчиков, графические и консольные варианты [1].

2.4 Выбор подхода к решению задачи регрессионного анализа

Для корректного и эффективного решений задачи определения уровня глюкозы необходимо определить способ обработки исходного сигнала ЭКГ. Данная задача является задачей регрессионного анализа и решается с использованием методов машинного обучения.

Для большинства существующих методов выделяют следующие проблемы:

1) Формирование исходного описания сигнала вызывает проблемы ввиду необходимости заранее априорно знать свойства анализируемого сигнала что в большинстве случаев затруднительно.

2) Проблематика формирования системы признаков в конечное множество, обеспечивающее единственное решение задачи классификации на этапе распознавания при этом являющиеся необходимыми и достаточными. Данный этап позволяет сократить размерность входного описания.

3) Принятие решений в условиях неопределенности так же является проблемой. Этап принятия решения заключается в сравнении с эталонным признаковым описанием анализируемого сигнала. Априорно предполагается что эталону соответствует компактное множество точек в системе признаков, однако наличие шумов и искажений приводят к появлению перекрытий класса. Из этого следует что проблема принятия решения замыкается на проблеме формирования системы признаков, позволяющей сформировать компактный эталон.

С помощью применения теории активного восприятия возможно решить описанные проблемы [3].

Алгоритм, основанный на применении теории активного восприятия включается в себя следующие этапы:

1) Формирование исходного описания сигнала и его предварительно обработки;

2) Формирования системы признаков сигнала;

3) Классификация сигнала на основе системы признаков.

Рассмотрим последний этап более внимательно и выберем классификатор:

SVM- support vector machine или метод опорных векторов. Применим для общей регрессии и классификации, и плотности оценки. SVM основан на максимальных пределах линейных дискриминантов, и этим похож на модель с вероятностным подходом, с тем исключением, что здесь не учитываются зависимости между атрибутами объектов модели. В отличии от классических подходов, страдающих от искажения и зашумливания данных, svm больше направлен на принципы минимизации структурных рисков (SRM), за счет отсутствия оптимизации выбранных параметров и статистических показателей [3]. Это позволяет

алгоритму получать неплохие эмпирические показатели. Пример работы разделяющих линейных классификаторов представлен на рисунке 1.

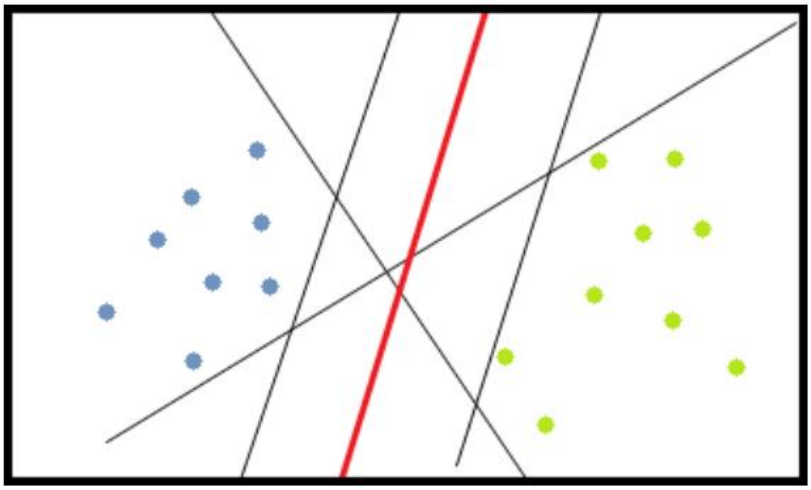

Рисунок 1. Пример разделяющих линейных классификаторов

Идеей SVM является поиск оптимальной гиперплоскости, то есть такого линейного классификатора, который обеспечит максимальную дистанцию между собой и ближайшими примерами объектов из каждого класса. Главным его недостатком является отсутствия общего подхода к автоматическому выбору ядра, и построению спрямляющего подпространства в целом, в случае линейной неразделимости классов.

Алгоритм ближайшего соседа или *k*-nearest neighbor algorithm (kNN). Алгоритм выделяет среди всех элементов *k* известных объектов (*k*-ближайших соседей), похожих на новый неизвестный ранее элемент. На основе классов ближайших соседей выносится решение касательно нового элемента. Важной задачей данного алгоритма является подбор коэффициента *k* – количество записей, которые будут считаться близкими. Пример работы алгоритма kNN представлен на рисунке 2.

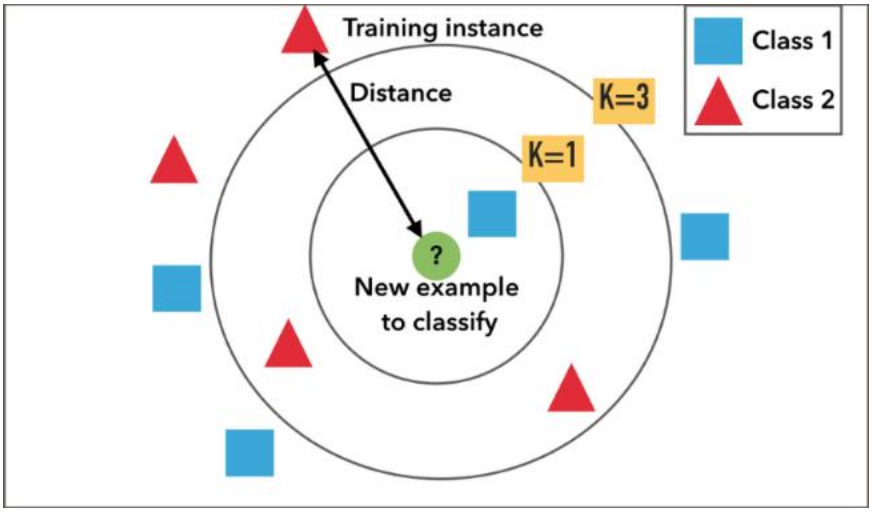

Рисунок 2. Пример работы алгоритма kNN.

Для повышения надёжности классификации объект относится к тому классу, которому принадлежит большинство из его соседей — *k* ближайших к нему объектов обучающей выборки *x_i*. В задачах с двумя классами число соседей берут нечётным, чтобы не возникало ситуаций неоднозначности, когда одинаковое число соседей принадлежат разным классам. Однако в задачах с числом классов 3 и более нечётность уже не эффективна, и ситуации неоднозначности всё равно имеют место. Тогда *i*-му соседу приписывается вес *w_i*, как правило, убывающий с ростом ранга соседа *i*. В таком случае объект относится к тому классу, который набирает больший суммарный вес среди *k* ближайших соседей.

Основными преимуществами knn являются:

1) Простота реализации;

2) Прочность касательно пространства поиска;

3) Небольшое количество параметров для настройки оптимального классификатора.

К недостаткам относят дорогое тестирование каждого нового объекта, что связанно с необходимостью вычислять расстояние до всех известных ранее объектах, количество которых может быть большим, и чувствительность к шумным и незначительным атрибутам.

Нейронные сети так же применяются для решения задач классификации и в том числе для решения задач регрессии. Наиболее часто используемые архитектуры выдают выходные значения в некотором определенном диапазоне (например, на отрезке от 0 до 1 в случае логистической функции активации). Для задач классификации это не создает никаких трудностей. Однако для задач регрессии особую важность имеет масштаб и диапазон существования выходных значений, поскольку на передний план выходят проблемы, связанные с эффектом экстраполяции.

Задачи регрессии методами нейросетевого моделирования можно решать с помощью сетей различных типов: многослойного персептрона, линейной сети, радиальной базисной функции и обобщенной регрессионной сети. Линейная модель по сути ничем не отличается от обычной линейной регрессии, но на языке нейронных сетей представляется сетью без промежуточных слоев, которая в выходном слое содержит только линейные элементы (то есть элементы с линейной функцией активации). Схема строение простой нейронной сети представлена на рисунке 3.

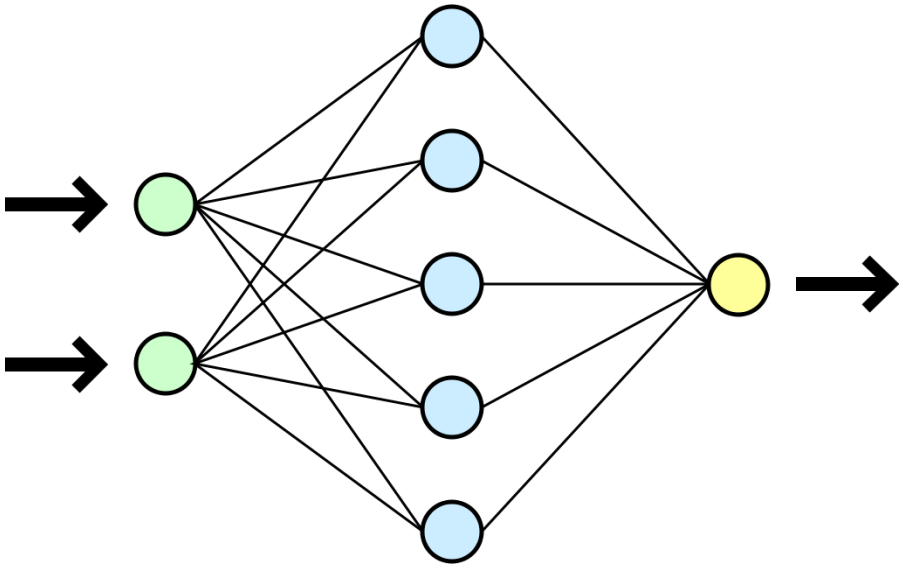

Рисунок 3. Модель простой нейронной сети.

В рамках этой работы именно нейронные сети были выбраны для решения задачи классификации, так как они показывают наибольшую производительность и наиболее подходят для решения класса подобных задач.

3.Разработка структуры системы неинвазивной оценки уровня глюкозы в крови

3.1 Разработка общей структуры системы

Программную систему оценки уровня глюкозы по данным ЭКГ можно представить в виде нескольких основным частей, представленных на рисунке 4.

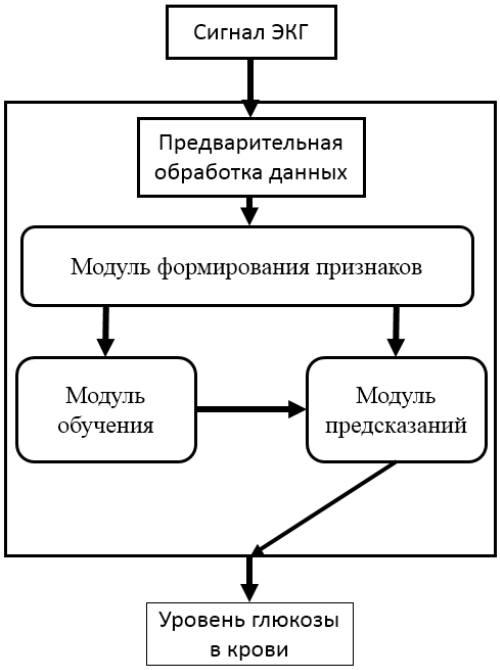

Рисунок 4. Структура программной системы.

Рассмотрим их подробнее:

1) Входной сигнал ЭКГ в формате .wav длительностью 5 минут и частотой дискретизации в 1кГц проходит предварительную обработку и представляется в числовой форме с помощью *Q*-преобразования и далее будут переданы в модуль формирования признаков.

2) Модуль формирования признаков электрокардиограммы на выходе которого будут получены признаки на основе гистограмм полных и замкнутых групп теории активного восприятия. Полные группы позволяют выявить корреляционные связи между операторами. Замкнутые группы позволяют выявить корреляционные связи между полными группами.

Он выполняет следующие действия:

2.1) Считывает участки фиксированной длинны из исходного сигнала ЭКГ;

2.2) Производит *U*-преобразования этих участков;

2.3) Определяет систему признаков для каждого участка;

2.4) Создает массив систем признаков всех записей;

2.5) При необходимости дополняет полученный массив столбцом в каждой строчке которого соответствующий записи уровень глюкозы;

2.6) Преобразовывает массив в пригодный для дальнейшей обработки формат.

Модуль расположен отдельно от других и используется на компьютере пользователя и сохраняет файлы с системами признаков локально, что позволяет более удобно хранить массивы признаков и при необходимости переносить их на другой компьютер без доступа к сети Интернет. К примеру пациент может получить данные ЭКГ в специализированном учреждении, а сохраненные файлы использовать для обучения нейросети уже со своей локальной машины.

1) Модуль обучения нейросети, который принимая на вход признаки произведет обучение и тестирование нейросети, индивидуальной для каждого пациента. Именно с помощью обученной нейросети и будут производится вычисления текущего уровня глюкозы в крови.

Для выполнения этих функций он производит следующие действия:

1.1) Принимает ранее подготовленные в предыдущем модуле файлы содержащие массив признаков и информацию об уровне глюкозы;

1.2) Считывает файл, разделяя его на обучающую и проверочную выборки в соотношении 5:1;

1.3) Самостоятельно производит подбор параметров нейросети, получая конфигурацию с минимальной относительной квадратичной ошибкой;

1.4) Производит обучение нейросети с помощью обучающей выборки;

1.5) Производит проверку обученной нейросети с помощью проверочной выборки

1.6) Определяет уровень корреляции Спирмена, использую предсказанные и фактические данные об уровне глюкозы для оценки качества полученной нейросети;

1.7) Сохраняет обученную нейросеть для дальнейшего использования.

Данный модуль предназначен для обучения модели и доступен через Web-интерфейс или с помощью API, что позволяет использовать его в составе других программных комплексов.

2) Модуль предсказаний, который получая файл признаков на входе будет на выходе выдавать актуальный уровень глюкозы в крови. В отличии от второго модуля он не будет производить обучения, а лишь использовать нейросеть полученную ранее, что существенно уменьшит время предсказаний.

Он способен производит следующие действия:

2.1) Принимает подготовленные первым модулем данные;

2.2) Принимает подготовленную вторым модулем нейросеть;

2.3) Используя полученные признаки и обученную нейросеть производит; предсказания уровня глюкозы;

2.4) Сохраняет полученные значения в файл в удобном для дальнейшей работы формате.

Данный модуль предназначен для непосредственного использования пациентом и обладает возможностью доступа к обученной модели и загрузке файлов для предсказания через Web-интерфейс с практически любого устройства поддерживающего возможность доступа к сети Интернет.

3.2 Алгоритм предварительной обработки данных

Предварительная обработка [4] заключается в применении к исходным сегментам сигнала операции сложения вида:

$$g(t) = \sum_{k=(t-1)*L+1}^{t*L} f(k), t = \overline{1, N} \ ,$$

где $g(t)$ – t-ый отсчёт сигнала g;

g – результат применения Q-преобразования к сигналу f;

L – число отсчётов, входящих в сегмент;

$f(k)$ – k-ый отсчёт сигнала f;

N – число сегментов сигнала.

В результате будет выполнено Q-преобразования исходного сигнала, в результате которого он будет представлен в числовой форме пригодной для дальнейшей обработки.

3.3 Алгоритм формирования системы признаков

Метод, который будет использован для создания признакового описания в виде гистограммы [6] заключается в следующем:

1) отсчёты сигнала s_{dig} разбиваются на множество сегментов $s_{dig} = \{s_{dig,\,k}\}$, $k = \overline{1, N}$, длиной M отсчётов, со смещением в T отсчётов;

2) к каждому сегменту $s_{dig,\,k}$ применяется U-преобразование, в результате формируется спектральное представление каждого сегмента: $\mu_k = U\,[s_{dig,\,k}]$, $\mu = \{\mu_k\}$;

3) по вычисленному спектральному представлению μ_k сегмента $s_{dig,\,k}$ формируется описание с помощью одной или нескольких структур, входящих в алгебру групп. В алгебре групп существуют следующие структуры, которые могут использоваться для создания описания (допустимо использование сочетаний данных структур):

3.1) Полные группы (P_n), известны полные группы на операции умножения (P_{nm}, 140 элементов), полные группы на операции сложения (P_{na}, 140 элементов);

3.2) Замкнутые группы (P_s, 840 элементов), замкнутые множества (P_{ca} – на операции сложения, 840 элементов, P_{cm} – на операции умножения, 840 элементов).

4) для объединения данных, полученных от разных сегментов анализируемого сигнала, вычисляется гистограмма элементов структур, использованных при создании описания сегмента сигнала:

$h_{na} = H\,[P_{na}, \Gamma]$, $h_{nm} = H\,[P_{nm}, \Gamma]$,

$h_c = H\,[P_c, \Gamma]$,

h_{na} – гистограмма полных групп на операции сложения,

h_{nm} – гистограмма полных групп на операции умножения,

h_c – гистограмма замкнутых групп,

H – оператор вычисления гистограммы заданной размерности,

Γ – размерность гистограммы:

$1d$ – одномерная гистограмма.

Схема формирования одномерной гистограммы представленная на рисунке 5 [3].

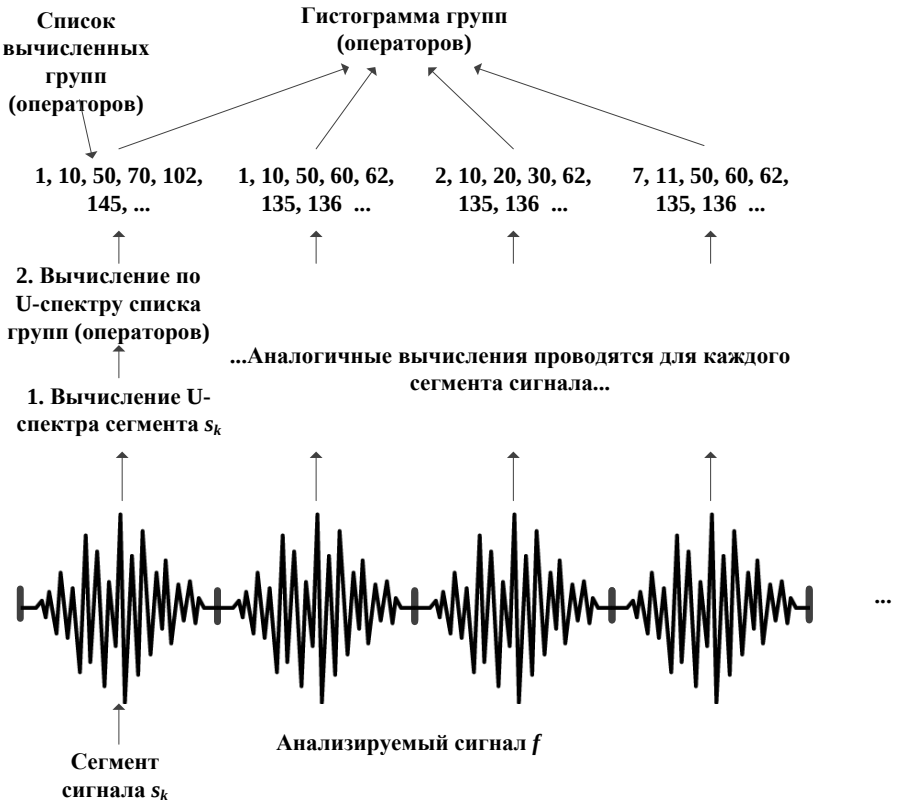

Рисунок 5. Формирование одномерной гистограммы признаков.

Ниже приведен алгоритм [6] формирования гистограмм. Признаковое описание, на основе которого формируется гистограмма h_D обозначается как D. В качестве данного описания могут использоваться операторы, полные и замкнутые группы.

Алгоритм формирования одномерной гистограммы признаков:

$$\forall i = \overline{1, M}$$
$$\forall j = \overline{1, |D_i|}$$
$$h_D(D_i[j]) = h_D(D_i[j]) + 1.$$

3.4 Алгоритм принятия решения

Классификатор является классической регрессионной нейронной сетью, состоящий из входного слоя, размер которого равен размеру используемого признака, динамически определяемого диапазоном параметров скрытого слоя и единственного выхода, являющегося соответствующем уровнем глюкозы.

Искусственная нейронная сеть представляет собой математическую модель, повторяющую своим строением биологические нейронные сети, и состоит из взаимодействующих между собой простых процессоров.

С точки зрения машинного обучения нейронная сеть является частным случаем методов распознавания образов. Такие сети не программируются в классическом понимании этого термина, а обучаются. Технически этот процесс представляет из себя нахождение коэффициентов связей между нейронами, что позволяет выявлять сложные зависимости между входными и выходными данными.

Искусственная нейронная сеть, также как и биологическая, состоит из множества связанных между собой нейронов. Искуственный нейрон состоит из синапсов, которые связывают входы нейрона с его ядром. Ядро осуществляет обработку входных сигналов и с помощью единственного аксона связано с нейронами следующего слоя. Каждый синапс имеет вес, который определяет, насколько соответствующий вход нейрона влияет на его состояние.

Модель искусственного нейрона представлена на рисунке 6.

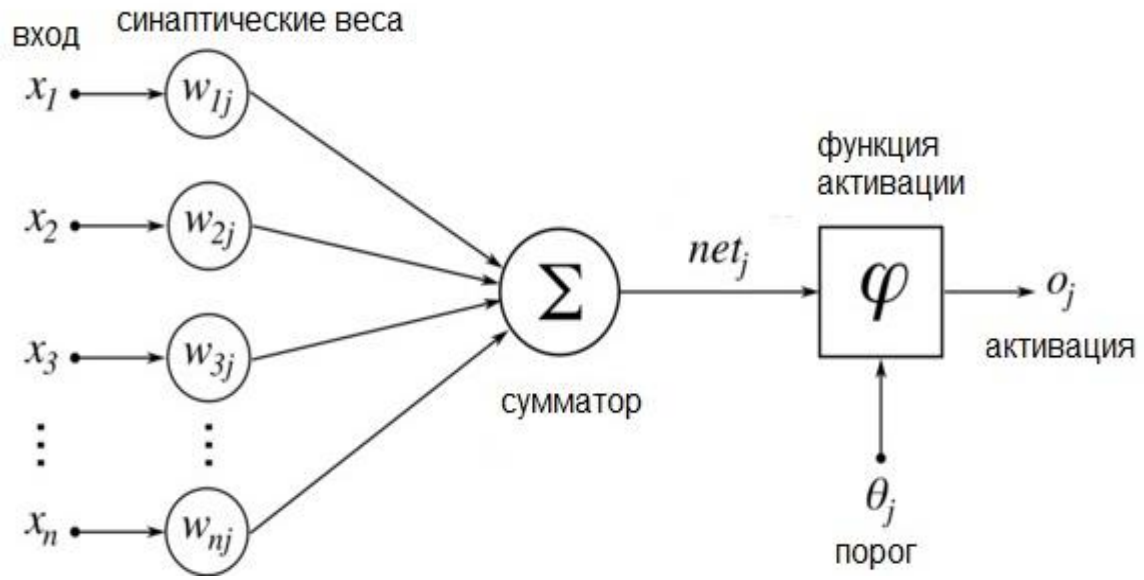

Рисунок 6. Модель искусственного нейрона.

Состояние нейрона определяется по формуле [5] :

$$S = \sum_{i=1}^{n} x_i w_i$$

в которой:

n – число входов нейрона

x_i – значение i-го входа нейрона

w_i – вес i-го синапса.

Затем определяется значение аксона нейрона по формуле:

$Y = f(S)$

где f – некоторая функция, которая называется активационной.

Наиболее часто в качестве активационной функции используется так называемый сигмоид, который имеет следующий вид:

$f(x)=11+e-ax.$

Основное достоинство этой функции в том, что она дифференцируема на всей оси абсцисс и имеет очень простую производную:

$f'(x)=\alpha\, f(x)(1- f(x)).$

При уменьшении параметра α сигмоид становится более пологим, вырождаясь в горизонтальную линию на уровне 0,5 при α =0. При увеличении α сигмоид все больше приближается к функции единичного скачка.

В классификаторе используется модель обучения с учителем, называемая персептроном, в котором на каждой итерации обучения на вход нейросети подается ранее определенный признак, а на выходе точное значение глюкозы, согласно которым корректируются коэффициенты внутри нейронной сети.

Перед передачей параметров входному слою нейросети они нормализуются методом minmax, линейно масштабирующий все численные коэффициенты, приводя их к диапазону от 0 до 1, использующиеся в нейросети.

Поскольку оптимальные параметры нейросети априорно неизвестны, используется дополнительный модуль, производящий предварительные серии тестов обучения сети с различными параметрами, в результате которого будут выбраны значения с минимальной среднеквадратической ошибкой.

4. Разработка программных средств

Рассмотрим реализацию основных модулей системы определения уровня глюкозы по данным ЭКГ. Начнем с части предобработки и формирования признаков. Данная часть выполнена на языке R.

Функция getUTrans() принимает файл в формате .wav, длину одного сегмента, смещение сигнала и множество фильтров, затем разделяет файл на сегменты ,после чего производит *U*-преобразования с помощью 16 фильтров, а результат представляется в виде *Q*-преобразований. На рисунке 7 представлен отрывок реализации этой функции.

```
getUTrans <- function(sig, slen, shft,  flt)
{

  nseg <- (length(sig) - slen + 1) / shft
  if ((nseg - floor(nseg)) > 0)
  {
    nseg <- floor(nseg)
    nseg <- nseg + 1
  }
  if ((slen %% 16) != 0)
  {
    stop('Неверная длина сегмента!!!')
  }

  steps <- c(1)
  for (i in 1:15)
```

Рисунок 7. Отрывок реализации функции getUTrans().

Функция getfeatures() пронимает в себя спектральное разложение сигнала сгенерированное функцией getUTrans() и производит вычисление признаков в виде полных и замкнутых групп. Часть её реализации можно видеть на рисунке 8.

```
getfeatures <- function(featname, udec, fullGrp, clsGrp, oper, gfullm2d, goper2d, goper3d, gfulls2d, gpfullm2d, gpfulls2d)
{
  if (featname == "cls1d")
  {
    feat <- matrix(0, 1, 840)

    for (i in 1:length(udec))
    {
      grp <- getclosedGroup(udec[[i]][[1]], clsGrp, udec[[i]][[2]], oper)

      grp <- grp[1, ]
      feat[grp] <- feat[grp] + 1

    }

    return(feat)
  }

  if (featname == "set1d")
  {
    feat <- matrix(0, 1, 840)
```

Рисунок 8. Отрывок реализации функции getfeatures().

В результате её работы будет получены полные и замкнутые группы, согласно теории активного восприятия, содержащих вычисленные признаки исходного сигнала.

Следующие части, отвечающие за обучение и предсказания, созданы при помощи Azure Machine Learning [2]. Модуль обучения представлен в виде следующего вычислительного графа, показанного на рисунке 9.

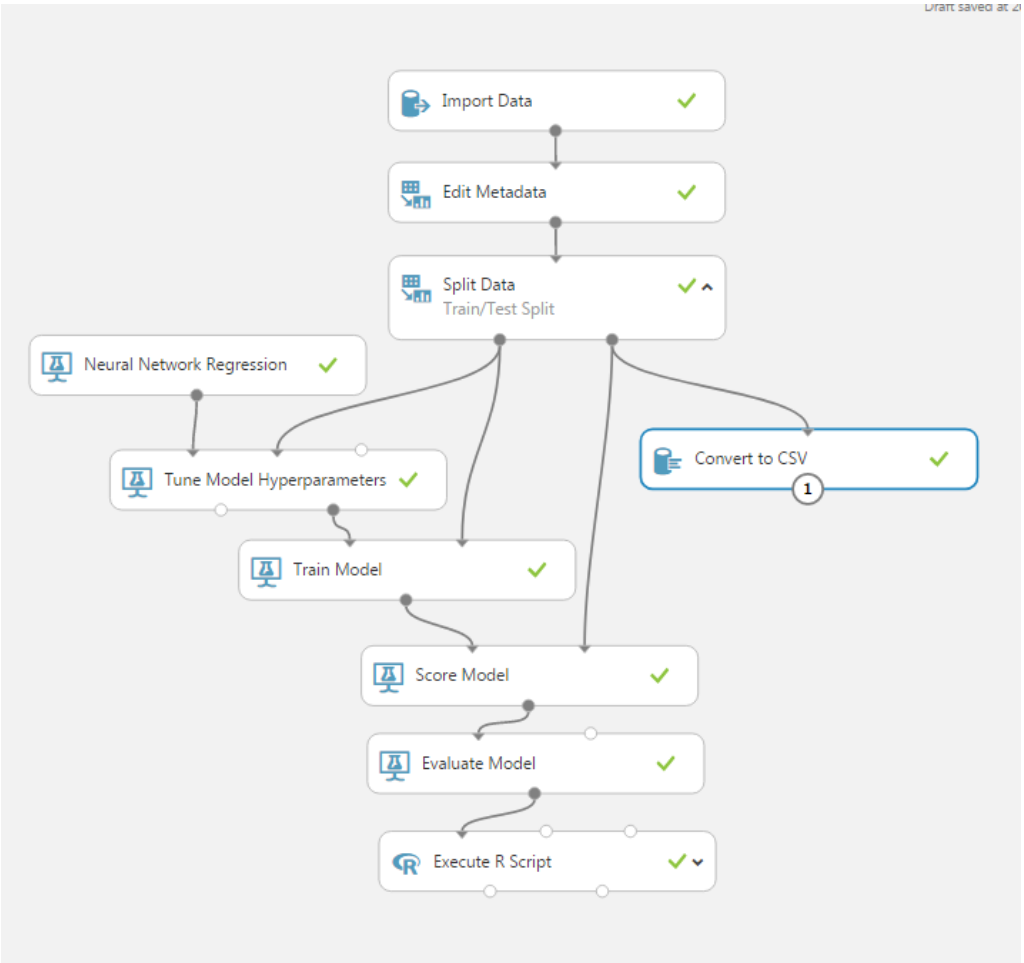

Рисунок 9. Вычислительный граф обучения нейросети.

Вычислительный граф выполнятся следующим образом:

1) Модуль Import Data принимает исходный признак по ссылке в бинарном виде.

2) файл представляется в виде строчек, состоящих из числовых значений признака сигнала и замыкается столбцом, помеченным как Target, представляющим собой соответствующие ему значения уровня глюкозы. Затем исходный массив разделяется на обучающую и проверочную выборки. Проверочная выборка конвертируется в формат CSV и сохраняется для последующих тестов, а обучающая передается далее.

Модуль Neural Network Regression содержит в себе диапазон параметров нейросети, список их можно видеть ниже:

Далее параметры нейросети и данные для обучения передаются в Tune Model Hyperparameters , который проведя серию вычислений, результаты которых предоставлены на рисунке 10.

Learning rate	LossFunction	Number of iterations	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
0.01	CrossEntropy	70	3.535773	4.452414	0.991017	1.007457	-0.007457
0.01	SquaredError	70	3.466473	4.357402	0.971593	0.964919	0.035081
0.01	SquaredError	20	3.458523	4.354155	0.969365	0.963481	0.036519
0.01	CrossEntropy	70	3.459087	4.347513	0.969523	0.960544	0.039456
0.01	SquaredError	70	3.440444	4.344767	0.964298	0.959331	0.040669
0.01	CrossEntropy	20	3.427706	4.331263	0.960727	0.953377	0.046623
0.01	SquaredError	20	3.435618	4.331026	0.962945	0.953272	0.046728
0.01	SquaredError	30	3.416245	4.32445	0.957515	0.95038	0.04962
0.01	CrossEntropy	70	3.454925	4.321345	0.968357	0.949016	0.050984
0.01	CrossEntropy	30	3.40731	4.314776	0.955011	0.946132	0.053868
0.01	CrossEntropy	20	3.4222	4.295043	0.959184	0.937498	0.062502
0.01	SquaredError	70	3.428177	4.294762	0.96086	0.937375	0.062625
0.01	CrossEntropy	30	3.386761	4.290197	0.949251	0.935384	0.064616
0.01	SquaredError	30	3.383044	4.286858	0.94821	0.933929	0.066071
0.01	CrossEntropy	30	3.402806	4.286846	0.953748	0.933923	0.066077
0.01	SquaredError	20	3.39343	4.284245	0.951121	0.93279	0.06721
0.01	SquaredError	30	3.391235	4.280427	0.950505	0.931129	0.068871
0.01	CrossEntropy	20	3.382975	4.277045	0.94819	0.929658	0.070342

Рисунок 10. Определение оптимальных параметров нейросети.

В результате определяются оптимальные параметры нейросети, которы можно наблюдать на рисунке 11.

Neural Network Regressor

Settings

Setting	Value
Is Initialized From String	False
Is Classification	False
Initial Weights Diameter	0.001
Learning Rate	0.01
Loss Function	CrossEntropy
Momentum	0
Neural Network Definition	
Data Normalizer Type	MinMax
Number Of Input Features	3640
Number Of Hidden Nodes	System.Collections.Generic.List`1[System.Int32]
Number Of Iterations	20
Number Of Output Classes	1
Shuffle	True
Allow Unknown Levels	True
Random Number Seed	

Рисунок 11. Оптимальные параметры нейросети.

После параметры и данные для обучения передаются модулю Train Model, производящему контролируемое обучения нейросети.

Затем модуль Score Model приняв от тренера обученную модель проводит её тестирование на данных не участвующих в обучении, а Evaluate Model проводит вычисление итоговых результатов, таких как абсолютная, среднеквадратичная ошибки и коэффициента детерминации.

Дополнительный модуль содержащий код на R представляет эти сведения в более удобной табличной форме. Пример представлен на рисунке 12.

Algorithm	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
TUNE Regression	4.401205	5.14329	1.206546	1.34565	-0.34565

Рисунок 12. Результаты тестирования обученной нейросети, представленные в табличном виде.

Обученная нейросеть может быть сохранена для дальнейшего использования.

Теперь рассмотрим модуль предсказаний.

Он принимает ранее сохраненную обученную нейронную сеть и тестовые данные, на основе которых и будут сделаны предсказания. Сам граф представлен на рисунке 13.

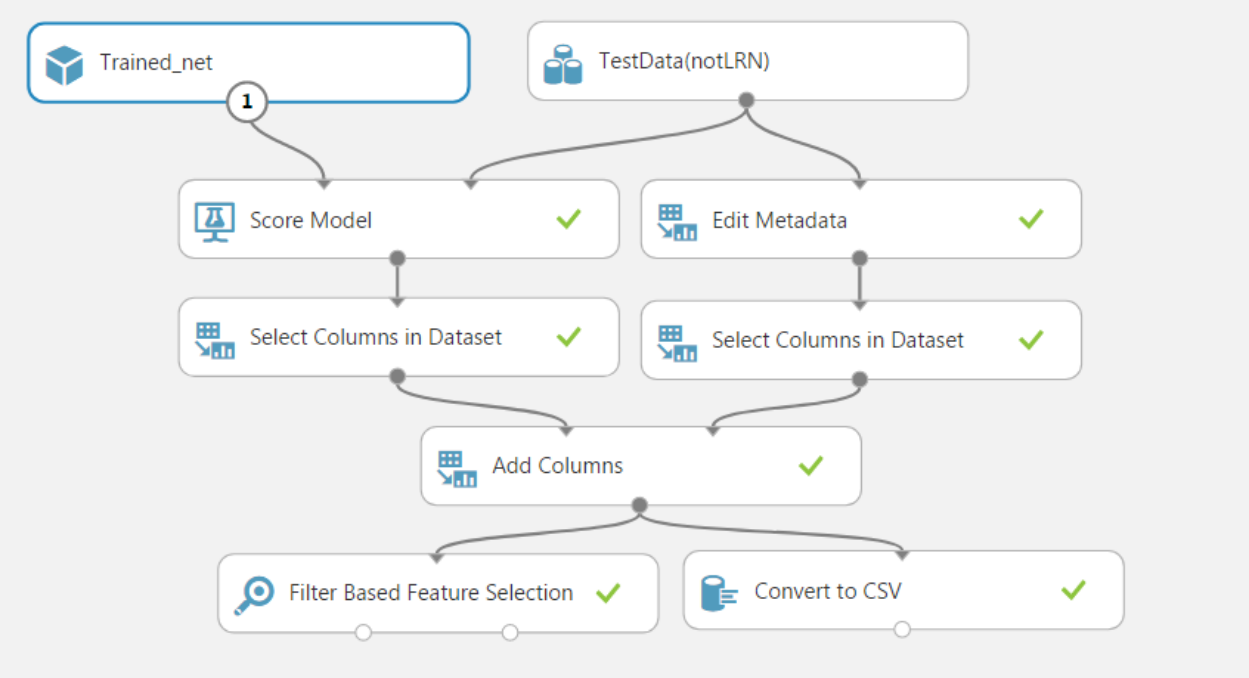

Рисунок 13. Вычислительный граф модуля предсказаний.

Нейросеть инициализируется данным нуждающимися в предсказании, а затем результат сохраняется в таблицу. При наличии в исходных данных уровня глюкозы они отсекаются и используются в дальнейшем для проверки.

Модуль Add Column сводит предсказанные и фактические уровни глюкозы в одну таблицу для дальнейшей работы с ними. Эта таблица может быть сохранена в формате CSV из модуля Convert to CSV.

Модуль Filter Based Feature Selection производит вычисления корреляции Спирмена между полученными и фактическими значениями уровня глюкозы.

5. Тестирование системы

5.1 Описание набора данных

Для каждого пациента исходные данные разбиваются на три неравные части:

1) Обучающая, служащая для тренировки нейросети, это наиболее крупная часть, составляющая около 80% всех измерений.

2) Тестовая служит для вычисления абсолютной и среднеквадратичной ошибок, так же коэффициента детерминации. Данная часть случайно выбирается из обучающей, и в обучении сети не применяется.

3) Проверочная служит для проверки предсказаний нейросети, сравнивая предсказания на её основе с фактическими вычисляется коэффициент корреляции.

5.2 Описание методики тестирования

Для проверки предсказанных значений используется коэффициент корреляции Спирмена, представляющей собой меру линейной связи между случайными величинами.

Коэффициент корреляции Спирмена вычисляется по формуле:

$$p = 1 - \frac{6}{n(n-1)(n+1)} \sum_{i=1}^{n} (R_i - S_i)^2,$$

где R_i - ранг наблюдения x_i в ряду x,

S_i - ранг наблюдения y_i в ряду y.

Коэффициент p принимает значения из отрезка [-1;1].

Равенство $p = 1$ указывает на строгую прямую линейную зависимость, $p = -1$ на обратную. Данная корреляция является ранговой, что означает что для оценки силы связей используется не численное значение, а соответствующие им ранги. Он инвариантен по отношению к любому монотонному преобразованию использующейся шкалы измерений.

В данном случае он позволяет установить корректность работы нейросети и показывает, что между предсказанными и фактическими значениями есть зависимость и предсказанные значения не являются случайными.

Для оценки силы корреляции можно ориентироватся на шкалу Чеддока, характеризующую тесноту связанности между величинами. Она представлена в таблице 1.

Величина коэффициента множественной корреляции	Оценка силы связи
0.1-0.3	Слабая
0.3-0.5	Умеренная
0.5-0.7	Заметная
0.7-0.9	Высокая
0.9-0.99	Весьма высокая

Таблица 1. Шкала Чеддока.

5.3 Результаты вычислительного эксперимента

Рассмотрим результаты по нескольким пациентам. Для каждого эксперимента будет обучена нейросеть, простроен график предсказанных и фактических значений и рассчитан коэффициент корреляции Спирмена. Исходные сигналы представлены в формате .wav с частотой дискретизации 1кГц и длительностью 5 минут.

5.3.1 Пациент 1430

Для обучения был использован признак full1d, являющийся одномерной гистограммой полных групп. Результат предсказаний представлен на графике 1.

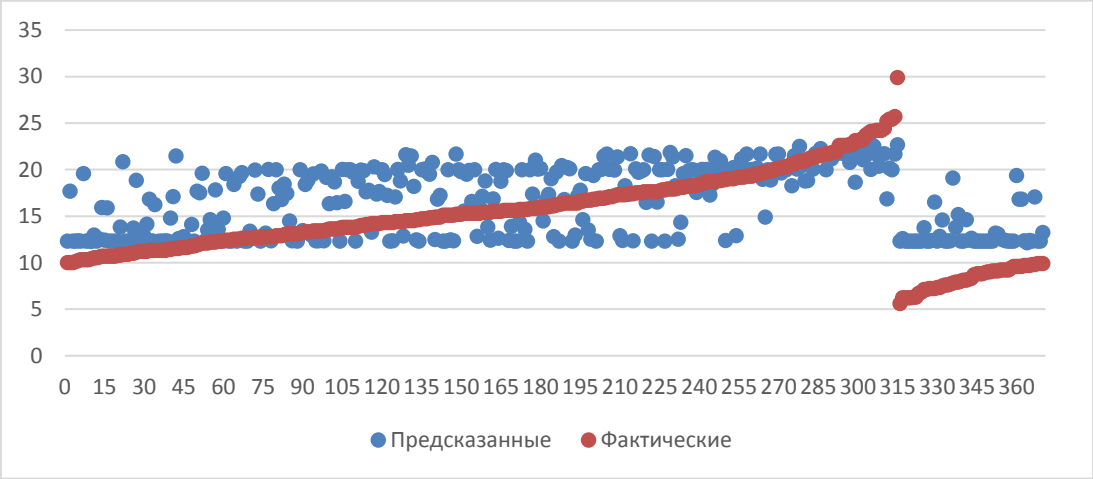

График.1 Фактических и предсказанные уровни глюкозы пациента 1430

На первый взгляд не удается установить зависимости между предсказанными и фактическими уровнями глюкозы, однако рассчитаем корреляцию Спирмена.

Согласно блоку Filter Based Feature Selection корреляция Спирмена составила около 0.7, что согласно шкале Чеддока является пограничным значением между заметной и высокой. Это является неплохим результатом.

5.3.2 Пациент 1024

Для обучения был использован признак full1d, одномерная гистограмма полных групп.

Результат предсказаний представлен на графике 2.

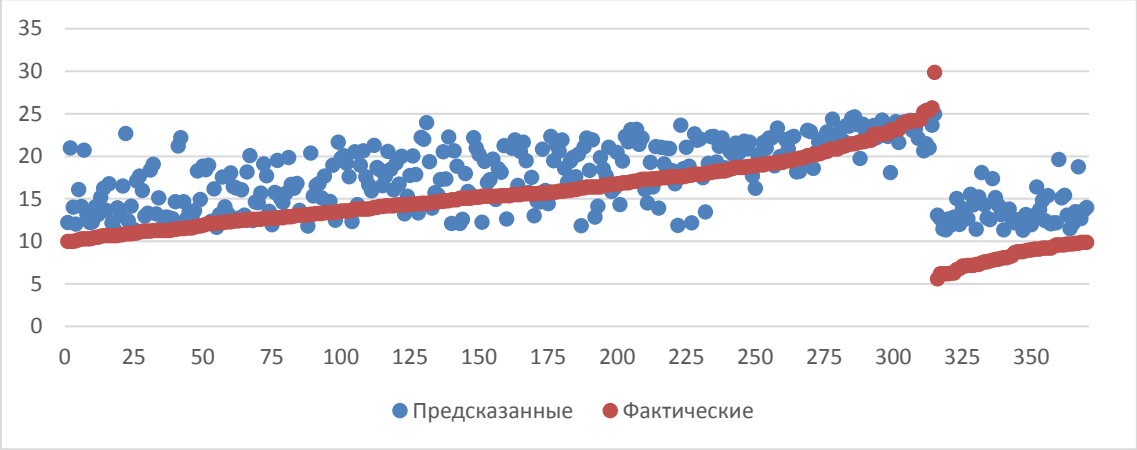

График 2. Фактических и предсказанные уровни глюкозы пациента 1024

На данном графике связь предсказанных и фактических значений более наглядна.

Согласно блоку Filter Based Feature Selection корреляция Спирмена составил 0.73, что согласно шкале Чеддока является высокой степенью.

5.3.3 Пациент 1215

Для обучения был использован признак cls1d, одномерная гистограмма замкнутых групп.

Результат можно наблюдать на графике 3.

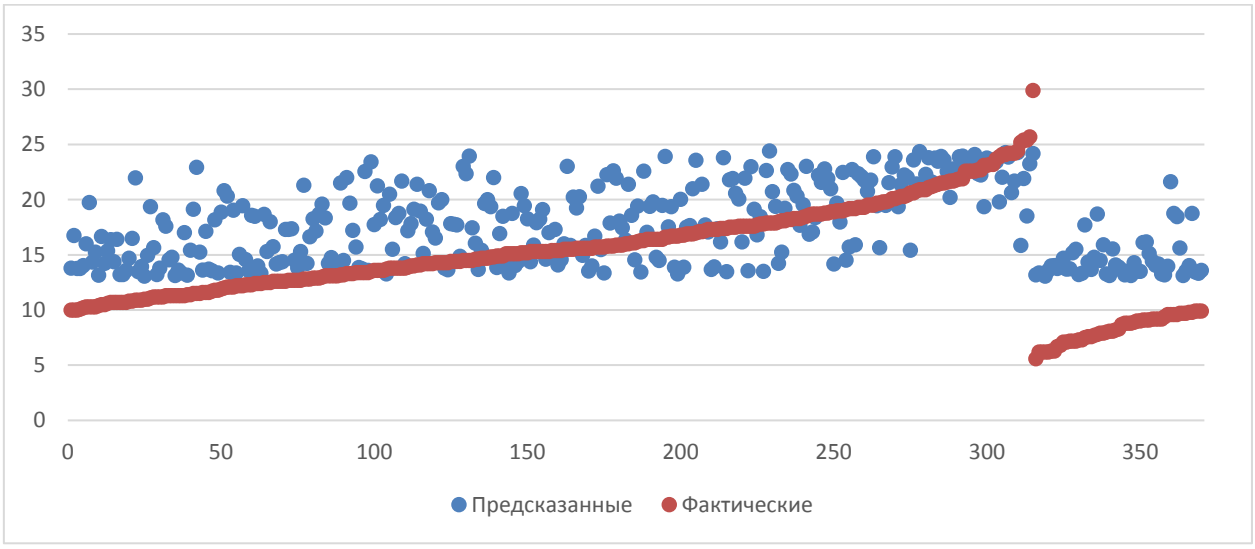

График 3. Фактических и предсказанные уровни глюкозы пациента 1215

Здесь согласно блоку Filter Based Feature корреляция меньше чем в предыдущих экспериментах и составляет 0.64. По шкале Чеддока это соответствует заметной корреляции.

5.3.4 Пациент 1318

Для обучения был использован признак full1d, одномерная гистограмма полных групп. Результат представлен на графике 4.

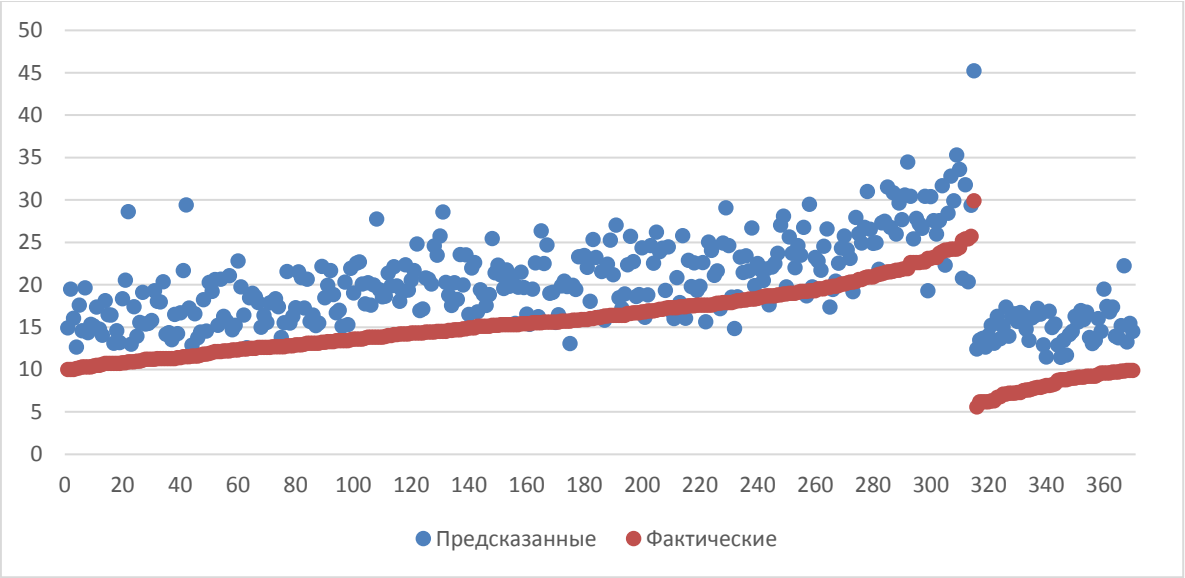

График 4. Фактических и предсказанные уровни глюкозы пациента 1318

В данным случае зависимость предсказанных и фактических значений возможно обнаружить невооруженным глазом.

Здесь корреляция максимальна среди представленных экспериментов и составляет 0.75. Это значительная корреляция и по шкале Чеддока она соответствует высокому уровню.

Заключение

В результате выполнения квалификационной работы была разработана программная система распознавания сигналов на основе гистограмм полных и замкнутых групп.
Данная система применима, в частности, для предсказания уровня глюкозы в крови по данным ЭКГ, благодаря включенной в её состав нейросети, способной решать задачи регрессии, необходимые для выполнения предсказаний.

Тестирование показало достаточно высокий уровень корреляции между предсказанными и фактическими значениями, что доказывает, как корректность параметров нейросети и её обучения, так и общую применимость, и правильность программной реализации теории активного восприятия для генерации системы признаков на её основе, к использованию совместно с нейронными сетями для решения задач регрессии.

Полученная система может быть адаптированная для решения различного спектра задач распознавания сигналов и является масштабируемой, что позволит, при минимальной доработке, использовать её в дальнейшем в других проектах.

Список литературы

1. «Статистический анализ и визуализация данных с помощью R» - Мастицкий С.Э., Шитиков В.К. – «ДМК Пресс», Москва, 2015 г.

2. Документация по машинному обучению Azure – URL: https://docs.microsoft.com/ru-ru/azure/machine-learning/

3. «Определение уровня глюкозы в крови человека на основе электрокардиографического сигнала» - В.Е. Гай, С.С. Бобко, Н.А. Домнина

4. «Элементы теории активного восприятия изображений» - Утробин В. А.

5. Нейронные сети — математический аппарат – URL: https://basegroup.ru/community/articles/math

6. Модели звуковых сигналов с позиций теории активного восприятия. Учебно-методическое пособие по курсу "Моделирование информационных процессов и систем"- Гай В.Е., Утробин В.А., Викулова Е.Н.