Оглавление

Введение

В наше время существует огромное количество задач, связанных с анализом большого количества данных. Все труднее становится осуществлять их решение вручную, поэтому набирает популярность машинное обучение, которое позволяет автоматически находить закономерности в массивах данных с помощью алгоритмов.

Машинное обучение применяется во многих сферах, в том числе и в медицине. Было проведено множество исследований, в которых говорится, что каждое заболевание по-особому изменяет ритм сердца. Следовательно, анализируя электрокардиографический сигнал и находя признаки, специфичные для конкретной болезни, можно строить системы автоматической диагностики, что открывает большой простор для машинного обучения.

Одна из возможных проблем организма, которую можно диагностировать с помощью анализа ЭКГ-сигнала – это высокое содержание сахара в крови, или глюкозы, которое возникает по причине недостаточного воздействия инсулина. Речь идет о таком заболевании, как сахарный диабет.

В течение последних нескольких десятилетий распространение диабета неуклонно растет. По данным Всемирной организацией здравоохранения число людей с этим заболеванием по всему миру возросло с 108 миллионов в 1980 году до 422 миллионов в 2014 году. В Российской Федерации в 2015 году число больных составляет 4,4 миллиона человек. По сравнению с 2014-м годом количество больных сахарным диабетом увеличилось на 5,6%, а за 3 года с 2013 по 2015 годы – на 23%.

Болезнь опасна как ранними осложнениями - диабетической и гипогликемической комой, так и поздними, возникающими при длительном повышенном уровне глюкозы в крови. К таким осложнениям относятся инсульт, болезни сердца, почек, потеря зрения и слуха.

Результаты лечения диабета показали, что более частый контроль глюкозы и инсулина в крови может предотвратить многие из долгосрочных осложнений сахарного диабета. Пациентам рекомендуется проводить самоконтроль сахара в крови несколько раз в день, чтобы корректировать свою диету и применение сахароснижающих препаратов.

Действие самых популярных на сегодняшний день приборов для мониторинга содержания глюкозы в крови основано на измерении электрического тока, возникающего при окислении глюкозы капиллярной крови на поверхности тестовой полоски. Очевидным недостатком такого метода является его инвазивность, т.е. необходимость нарушения кожных покровов, прокола кожи.

Целью данной работы является разработка системы, определяющей уровень сахара в крови безопасным, неинвазивным методом.

1. Техническое задание

1.1. Назначение разработки и область применения

Практически все существующие методы определения уровня глюкозы в крови требуют производить забор крови для анализа. Все они имеют ряд недостатков:

1. Наличие вероятности занесения во внутреннюю среду организма болезнетворных вирусов и бактерий, чужеродных веществ.
2. Болевые и неприятные ощущения во время процедуры.
3. Высокая стоимость расходных материалов, необходимых для анализа.

Это влечет за собой недостаточную частоту измерений и, как следствие, невозможность в должной мере компенсировать проблемы, вызванные заболеванием.

Неинвазивные методы лишены этих недостатков. Такие системы, включающие в себя средства для сохранения результатов и устройства для передачи сигналов, дают возможность также осуществлять контроль показателей пациента и создавать удаленные системы мониторинга. Это может оказаться полезным в качестве средства слежения за состоянием пациентов из медицинского учреждения и оказания своевременной помощи при наступивших критических ситуациях.

Разрабатываемая система реализует неинвазивный метод определения уровня сахара в крови по результатам обработки данных, полученных с электрокардиограммы.

1.2. Технические требования

Разрабатываемая система должна работать по следующему принципу:

1. Для каждого конкретного пациента с удаленного сервера берутся данные, которые представляют собой записи электрокардиографического сигнала и реальные значения уровня глюкозы.
2. По этим данным вырабатываются различные системы признаков, которые подаются на вход классификатора.
3. Выбирается диапазон значений параметров классификатора.
4. Проводится обучение на тестовых данных с различными системами признаков и параметрами.
5. Из всех полученных моделей выбирается несколько лучших, которые отправляются на удаленный сервер.
6. Для получения значения уровня глюкозы на вход классификатора подается ЭКГ-сигнал.

2. Анализ технического задания

2.1. Выбор операционной системы

Перед началом разработки необходимо выбрать операционную систему, для которой будет создаваться система предсказания глюкозы в крови. Самыми популярными ОС сейчас являются Windows и Linux. Поскольку работа системы предполагает получение данных с сервера, их обработку и отправку обратно на сервер, будем рассматривать преимущества и недостатки операционных систем с этой позиции.

- Стоимость.

  Покупка и поддержка работы Windows обойдется намного дороже, чем общедоступная ОС Linux, которая является бесплатной.

- Безопасность.

  Linux имеет более высокий уровень безопасности, нежели Windows. так как является открытой ОС, в которой любая обнаруженная уязвимость закрывается сообществом в считанные дни, из-за чего для нее существует очень небольшое число вредоносных программ. Также Linux имеет возможность более тонкой настройки прав пользователей, что позволяет эффективно ограничивать доступ.

- Стабильность.

  Сбои операционной системы Linux случаются значительно реже, чем Windows. Поэтому в плане стабильности ОС Linux предпочтительней.

- Разработка и применение программного обеспечения

  Для поставленной задачи были выбраны скриптовые языки программирования, выбор основного языка рассматривается в следующем пункте, здесь лишь скажем, что для таких языков нет принципиальной разницы, на какой ОС разрабатывать и использовать программу, подходит как Windows, так и Linux.

Исходя из всего вышеперечисленного, операционной системой для разработки была выбрана Linux.

2.2. Выбор языка программирования

Поскольку в данной работе рассматривается та часть приложения, которая отвечает за предсказание уровня сахара в крови по ЭКГ-сигналу, язык программирования будем выбирать только для нее. Предполагаемый язык должен быть приспособлен под обработку больших массивов данных и иметь инструменты для применения машинного обучения.

Рассмотрим самые распространенные языки, подходящие для заявленной цели - Python, MatLab и R.

Python - язык программирования, ориентированный на повышение производительности разработчика и читаемости кода. Python является универсальным языком и применяется во многих сферах, в том числе для анализа и статистической обработки данных. Но, с другой стороны, этот язык предпочтительней применять в тех случаях, когда помимо решения задач, связанных с анализом данных, необходимо реализовывать оболочку в виде веб-приложения или рабочей базы данных.

MatLab - язык программирования, используемый в одноименном пакете прикладных программ MatLab для инженерных расчетов. Имеет высокую скорость работы, удобный графический интерфейс и является простым в изучении. Но, между тем, имеет и ряд недостатков. Это и неполная поддержка необходимых нам статистических функций, и слабость в части интеграции при разработке большой системы, и дороговизна лицензии.

R - язык программирования, созданный специально для статистической обработки данных. В нем можно найти любые необходимые функции. R является языком с открытым исходным кодом, поэтому в нем всегда своевременно появляются пакеты с новыми методами обработки данных, созданные активным сообществом. Язык позволяет легко визуализировать данные, чтобы они стали выразительнее и понятнее. К недостаткам следует отнести невысокую скорость работы.

Учитывая все плюсы и минусы рассмотренных языков в качестве используемого языка был выбран R. В нем есть современные методы обработки данных и все необходимые функции, а также среда, разработанная специально для анализа и обработки данных, которая позволяет с комфортом решать поставленную задачу.

Рассмотрим библиотеки языка R, которые помогут нам в разработке системы анализа сахара в крови:

- tuneR - библиотека для преобразования .wav файлов с ЭКГ в числовой формат, подходящий для дальнейшей работы.
- ggplot2 - библиотека для наглядного отображения ЭКГ-сигнала, имеющего числовой формат, а также для мониторинга промежуточных вычислений, таких как среднеквадратичная ошибка отдельных блоков ЭКГ-сигнала и других критериев оценки точности.
- extraTrees - библиотека, позволяющая обучить классификатор ExtraTrees для получения уникальной системы признаков для каждого пациента.
- xlsx - библиотека для работы с файлами Microsoft Excel, для осуществления импорта и экспорта.

2.3. Выбор среды разработки

В языке программирования R используется интерфейс командной строки, но для удобства работы доступны различные варианты графического пользовательского интерфейса, которые расширяют возможности языка и упрощают написание кода. В качестве такого интерфейса была выбрана среда разработки RStudio, которая обладает рядом преимуществ:

- Подсветка синтаксиса.
- Автодополнение команд.
- История выполнения команд.
- Инструменты для построения графиков.
- Выполнение всего кода или его части непосредственно из редактора.
- Интерактивный отладчик для выявления и исправления ошибок.
- Возможность работы одновременно с несколькими файлами.

2.4. Обзор существующих систем неинвазивной оценки уровня глюкозы в крови

Используемые для определения глюкозы неинвазивные методы подразделяются на две категории. В первую входят методы, основанные на анализе физических свойств крови и ткани. Они базируются на предположении, что глюкоза – доминантный, постоянно изменяющийся признак, следовательно, он способствует изменению в соответствующих физических параметрах. Эта группа методов включает в себя оптическую когерентную томографию, определение светорассеяния и эмиссии, регистрацию температурных и электрических изменений тканей, методы флуоресцентного анализа.

Во вторую категорию входят методы, определяющие функциональные группы молекул глюкозы. К ним относятся спектроскопия ближнего и среднего инфракрасного диапазона длин волн, инфракрасный фотоакустический анализ, рамановская спектроскопия, метод оптической ротации [4].

Общая проблема этих методов заключается в их недостаточной точности измерения в различных условиях.

Рассмотрим несколько неинвазивных глюкометров, которые на настоящий момент позиционируются как наиболее точные.

Глюкометр GlucoTrack DF-F, разработанный израильской компанией Integrity Applications, проводит измерение уровня глюкозы в крови тремя способами: ультразвуковым, электромагнитным и термальным, что значительно повышает точность измерений. Осуществляется замер с помощью датчика-клипсы, которая крепится на мочке уха, далее

результат отображается на специальном портативном устройстве. Все накопленные данные могут быть перенесены на ПК. Одновременно устройство может быть откалибровано под 3 пользователей, но датчик должен быть у каждого свой. Каждый месяц устройству необходимо проходить перекалибровку, а раз в пол года необходимо производить замену датчика-клипсы.

К самым значительным недостаткам можно отнести:

- Отсутствие возможности измерения гипогликемии. Диапазон измерений 3,9-28 mmol/l, следовательно, прибор подходит только для людей с сахарным диабетом II типа.
- Высокая стоимость. По состоянию на 2016 год устройство и один датчик стоят 1900$ (~115 тыс. руб.), отдельно датчик потом можно приобрести за 130$ (~7800 руб.).
- Необходимость заказывать устройство из города Вильнюс в Литве.

Внешний вид устройства показан на рис. 1.

Рисунок 1 – Внешний вид GlucoTrack DF-F

Рассмотрим еще одну неинвазивную систему мониторинга глюкозы – Dexcom G5 Platinum. Он состоит из трех компонентов:

1. Ресивер (пульт управления с дисплеем, где постоянно отображается график изменения глюкозы), который покупается один раз.
2. Сенсор (инвазивный датчик, который крепится на тело с помощью клейкого основания), который рекомендуется менять раз в 7 дней.
3. Трансмиттер (передатчик, который крепится к сенсору и непрерывно передает данные о сахаре на ресивер), который служит до 6 месяцев.

Система измеряет уровень глюкозы в межклеточной жидкости подкожно-жирового слоя, не имея прямого контакта с кровью. Поэтому полученный уровень глюкозы в крови показывает значения, которые были актуальными около 15 минут назад. Измерение сахара происходит каждые 5 минут. Результаты отображаются на экране в виде графика. Тонкий сенсор размером с человеческий волос обеспечивает безболезненное введение и ношение. Прибор способен измерять значения глюкозы в диапазоне 2-22 mmol/l, при слишком низких и высоких значениях предусмотрено звуковое оповещение.

К недостаткам относятся:

1. Высокая стоимость. Покупка стартового набора обойдется в 1500$ (~90 тыс. руб.). Раз в 6 месяцев необходимо покупать новый трансмиттер за 500$ (~30 тыс. руб.), а также примерно раз в месяц новый набор сенсоров за 330$ (~20 тыс. руб.).
2. Частота калибровки. Прибор требует калибровки с помощью инвазивного глюкометра минимум 2 раза в день.
3. Необходимость заказывать сам прибор и все расходные материалы из США.

Внешний вид показан на рис. 2.

Рисунок 2 – Внешний вид Dexcom G5 Platinum

Еще одна популярная система измерения глюкозы в крови называется FreeStyle Libre.

Прибор состоит из двух компонентов:

1. Круглый сенсор 35 мм в диаметре и 5 мм в высоту, который крепится на кожу с помощью клейкого основания.
2. Ридер (пульт), который нужно подносить к сенсору для того, чтобы считать его показания.

Каждую минуту происходит измерение сахара с помощью сенсора, данные сохраняются в его памяти. При поднесении ридера к сенсеру, эти данные сканируются и становятся доступными для последующего анализа. На пульте с дисплеем можно видеть текущее значение сахара, динамику его движения и график.

Преимущества рассматриваемой системы:

1. Отсутствует необходимость калибровки с помощью обычного инвазивного глюкометра
2. Стоимость. По сравнению с другими рассматриваемыми системами оценки уровня глюкозы, данная обладает наименьшей стоимостью. Стартовый набор имеет стоимость в 170 евро (~10 тыс. руб).

Недостатки:

1. Несмотря на невысокую стоимость стартового комплекта, ежемесячные траты на сенсоры будут составлять 120 евро (~7.5 тыс. руб).

Внешний вид показан на рис. 3.

Рисунок 3 – Внешний вид FreeStyle Libre

Таким образом, все существующие системы имеют один главный недостаток - высокую стоимость расходных материалов. Также к недостаткам относятся низкий диапазон измерения глюкозы, частота калибровки инвазивным глюкометром и сложность приобретения прибора.

2.5. Выбор подхода к решению задачи регрессионного анализа

Задача регрессионного анализа состоит из 3 этапов:

1. Предварительная обработка сигнала.
2. Формирование системы признаков.
3. Принятие решения.

Этап предварительной обработки сигнала заключается в фильтрации исходного сигнала: удалении из него посторонних шумов и определении границ информативности.

Этап формирования системы признаков нужен для создания описания исходного сигнала. Здесь сигнал разбивается на множество сегментов, к каждому из которых применяется U-преобразование, в результате чего формируется спектральное представление каждого сегмента. На основе этого формируется описание с помощью закрытых групп.

На этапе принятия решения есть возможность использовать разные алгоритмы для регрессионного анализа: метод опорных векторов, метод k ближайших соседей, градиентный бустинг, случайные деревья. Из всего перечисленного для решения нашей задачи не использовался только алгоритм, реализующий случайные деревья, поэтому на этапе принятия решения будет использоваться именно он.

Алгоритм заключается в использовании ансамбля решающих деревьев, которые строятся независимо друг от друга. Суть алгоритма построения каждого такого дерева заключается в следующем:

1. Из обучающей выборки генерируется подвыборка с повторением, т.е. некоторые объекты в ней могут повторяться, а некоторые в нее не войдут вообще.
2. Строится решающее дерево, но на основе не всех признаков, а опять-таки, некоторых случайно выбранных.
3. Строительство дерева продолжается до полного исчерпания подвыбрки.

Окончательное принятие решения по каждому объекту проводится путем вычисления среднего от всех полученных ответов.

Случайный выбор объектов для каждого дерева, а также случайный выбор признаков позволяют добиться низкой корреляции между деревьями, что значительно повышает качество работы алгоритма.

3. Разработка структуры системы неинвазивной оценки уровня глюкозы

3.1. Разработка общей структуры системы

Систему оценки уровня глюкозы в крови на основе ЭКГ - сигнала можно представить как систему распознавания образов, которая включает три этапа обработки данных: предварительная обработка, вычисление признаков и принятие решения [1]. Для реализации первых двух этапов будет использоваться теория активного восприятия [2], для третьего – классификатор extraTrees языка R, реализующий алгоритм композиции случайных деревьев. Обобщенная структура системы представлена на рис. 4.

Рисунок 4 - Обобщенная структурная схема

На вход системы подается звуковой сигнал в формате .wav, который перед этапом формирования исходного описания (предварительной обработки) преобразовывается в числовую и графическую формы. После прохождения трех этапов обработки получаем предсказанное числовое значение уровня глюкозы.

Расширенная структурная схема, описывающая обучение и предсказание представлена на рис. 5.

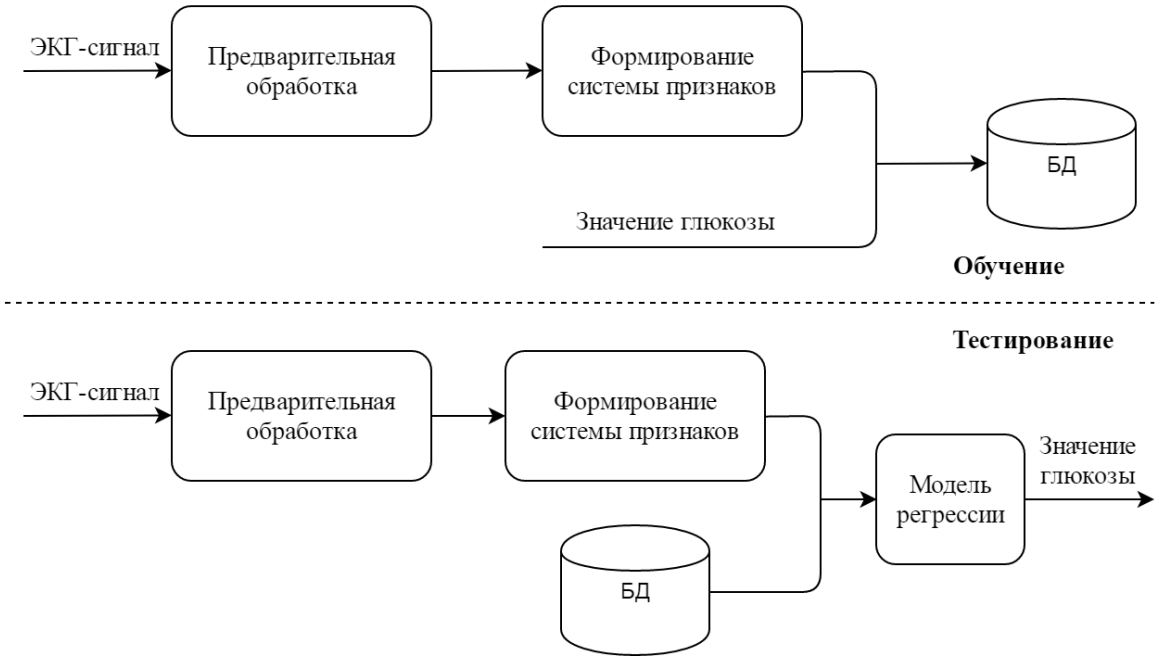

Рисунок 5 - Расширенная структурная схема

При обучении модели после предварительной обработки ЭКГ-сигнала сопоставляются вычисленные по нему признаки и значение глюкозы, полученное пациентом с помощью инвазивного глюкометра. В базу данных (во внешний rdata-файл) сохраняется это сопоставление для последующей работы.

При тестировании модели мы берем базу данных, полученную на этапе обучения, и вместе с вычисленными признаками для нового сигнала после предварительной обработки подаем на вход классификатора, который выдает значение глюкозы.

3.2. Разработка алгоритма предварительной обработки данных

Этап предварительной обработки заключается в выполнении *Q*-преобразования, которое заключается в применении к сегментам исходного сигнала операции сложения:

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

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.

3.3. Разработка алгоритма формирования системы признаков

Формирование признакового описания исходного сигнала заключается в применении к сигналу *g* множества фильтров Уолша системы Хармута:

$$\mu(k, c(t)) = \sum_{i=0}^{M-1} F_k(i) * g(\frac{(t-1)*M+1}{t*M}) \quad (2)$$

где $\mu(k, c(t))$– результат применения множества фильтров Уолша системы Хармута к сигналу *g*;

$k = \overline{0, M-1}$, $t = \overline{0, |c|-1}$, $c = \{1, P, 2 \cdot P, 3 \cdot P, ..., N - T \cdot P\}$ – множество значений смещений по сигналу *g*;

$|c|$ – мощность множества *c*;

P – величина смещения по сигналу *g* ($1 \le P \le M$);

M – число используемых фильтров.

Таким образом, признаковое описание сигнала представляет собой матрицу размером $M \times |c|$, причём каждая строка признакового описания представляет собой результат U-преобразования сегмента сигнала.

Последовательное применение к сигналу Q-преобразования и системы фильтров реализуют U-преобразование, являющееся базовым в теории активного восприятия.

Используемые фильтры представлены на рис. 6.

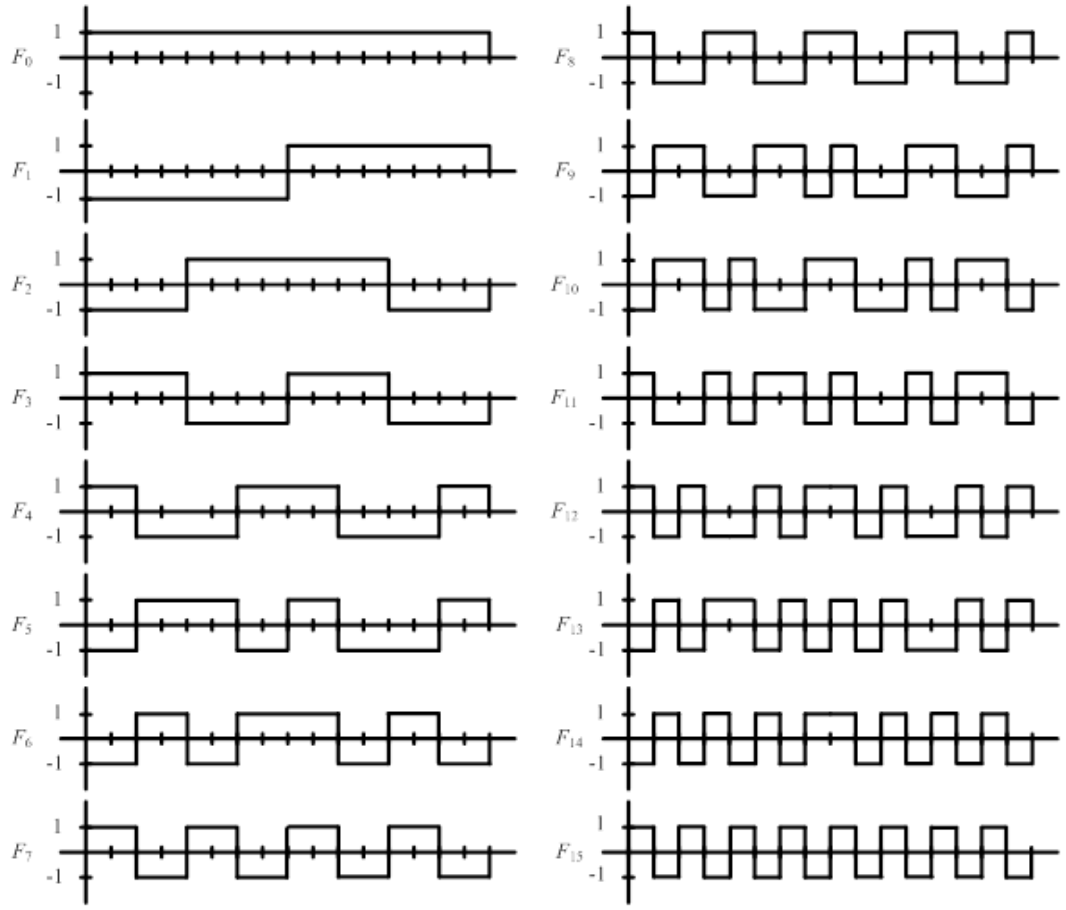

Рисунок 6 – Система фильтров

U-преобразование имеет минимально возможную вычислительную сложность, поскольку при его реализации используются простейшие операции – сложение и вычитание. Стандартные преобразования, требуют реализации свертки, а на уровне весовых коэффициентов – операции арифметического умножения.

Этап формирования системы признаков также включает в себя алгебру групп. Обнаруженные зависимости допускают своё использование на этапах принятия решения и понимания анализируемого сигнала.

Пусть каждому фильтру $F_i \in \{F_i\} \equiv \boldsymbol{F}$ соответствует бинарный оператор $V_i \in \{V_i\} \equiv \boldsymbol{V}$; тогда компоненте $\mu_i \neq 0$ вектора μ допустимо поставить в соответствие оператор V_i либо $\overline{V_i}$ в зависимости от знака компоненты. В результате вектору μ ставится в соответствие

подмножество операторов из $\{V_i\}$, имеющих аналогичную фильтрам конструкцию, но разное значение элементов матрицы ($+1 \leftrightarrow 1$; $-1 \leftrightarrow 0$). Задавая на множестве $\{V_i\}$ операции теоретико-множественного умножения и сложения, имеем алгебру описания сигнала в одномерных булевых функциях. С учётом инверсий всего существует 15 операторов, которые могут использоваться при формировании признакового описания, так как оператор V_0 принимает только прямое значение [3].

На множестве операторов формируется алгебра групп (этап синтеза) анализируемого сигнала [5]:

1. семейство алгебраических структур (названных полными группами) $\{P_{ni}\}$ вида $P_{ni} = \{V_i, V_j, V_k\}$ мощности 35;
2. семейство алгебраических структур (названных замкнутыми группами) $\{P_{si}\}$ вида $P_{si} = \{V_i, V_j, V_k, V_r\}$ мощности 105, где каждая группа образована из пары определенным образом связанных полных групп.

Схематическое представление алгебры групп представлено на рис. 7.

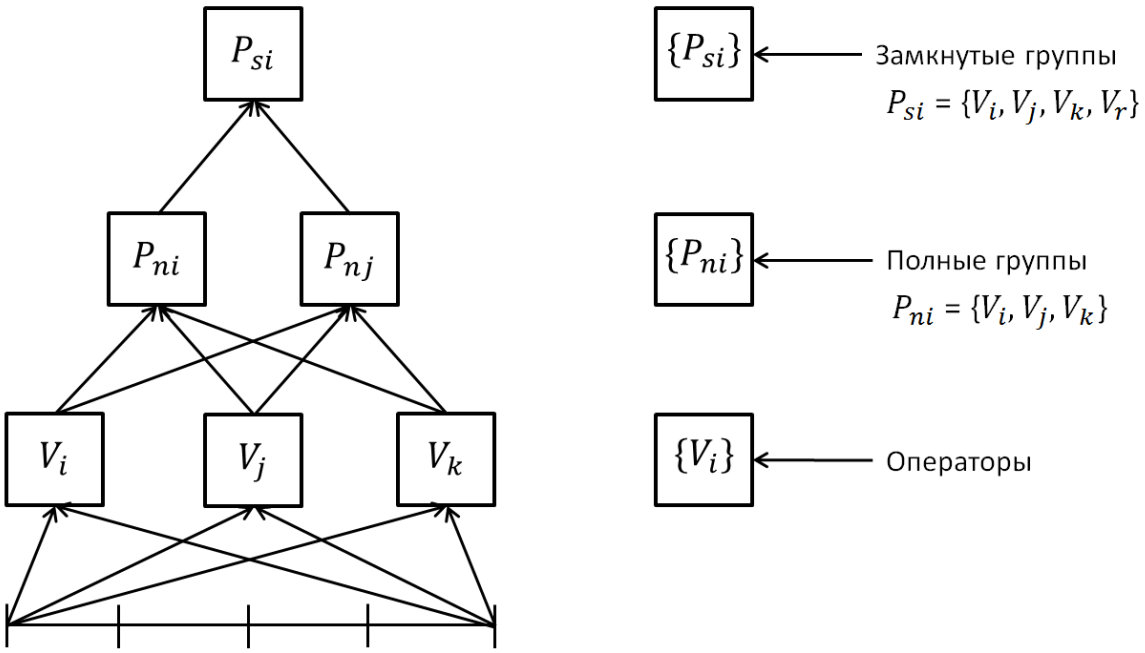

Рисунок 7 – Алгебра групп

Среди полных групп выделяют полные группы на операции сложения и на операции умножения, среди замкнутых групп – замкнутые группы и замкнутые множества.

Две группы (полные или замкнутые) называются несовместными, если в их состав входят операторы с одинаковыми номерами, но с разными знаками.

С помощью замкнутых и полных групп выполняется спектрально-корреляционный анализ [3]. Полные группы позволяют выявить корреляционные связи между операторами, замкнутые группы - между полными группами. Если множество операторов – алфавит, то

множества групп – более сложные грамматические описания наблюдаемого сигнала: полная группа – слово, замкнутая группа – словосочетание.

Используя спектральное представление сигнала μ, формируется множество операторов описывающий данный сигнал, а затем множества полных и замкнутых групп:

$$V = GV[\mu],\ P_{na} = GP_{na}[\mu, V],\ P_{nm} = GP_{nm}[\mu, V],$$
$$P_s = GP_s[\mu, V, P_{na}, P_{nm}],\ P_c = GP_c[\mu, V, P_{na}, P_{nm}], \quad (3)$$

где GV – оператор вычисления по спектральному представлению сигнала признакового описания V на основе операторов;

GP_{na} (GP_{nm}) - на основе полных групп на операции сложения P_{na} (умножения, P_{nm});

GP_c (GP_s) – на основе замкнутых групп P_s (замкнутых множеств, P_c).

Матрицы вероятностей переходов - это модель признакового описания, в которой учитываются связи между соседними сегментами сигнала. Метод вычисления признакового описания в таком случае состоит в формировании матрицы вероятностей переходов между описаниями соседних сегментов.

Предлагаются следующие системы признаков, основанные на матрицах вероятностей переходов:

1. Система признаков PVI, описывающая вероятности переходов между значениями операторов (оператор может принимать три возможных значения: прямое, инверсное и равное нулю), вычисленными по соседним сегментам сигнала, без учёта связей между различными операторами, размерность пространства признаков – 3 × 3 × 15 (рис. 8).
2. Система признаков PVD, описывающая вероятности переходов между значениями операторов, вычисленными по соседним сегментам сигнала, с учётом связи между операторами, размерность пространства признаков – 45 × 45 (рис. 9).
3. Система признаков PVS, описывающая вероятности переходов между описаниями сегментов, представленных в виде полных групп, размерность пространства признаков – 140×140; при использовании полных групп допустимо использовать только несколько максимальных по сумме отсчётов, находящихся под их образами, групп (граф переходов для полных групп подобен графу переходов для операторов, рис. 9).
4. Система признаков, описывающая вероятности переходов между описаниями сегментов, представленных в виде замкнутых групп, размерность пространства признаков – 840×840; при использовании замкнутых групп допустимо использовать группы, сумма отсчётов, под образом которых максимальна.

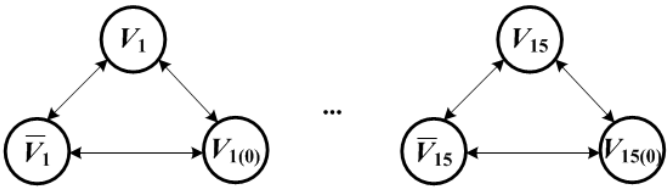

Рисунок 8 - Графы переходов между операторами (без учёта связей между операторами)

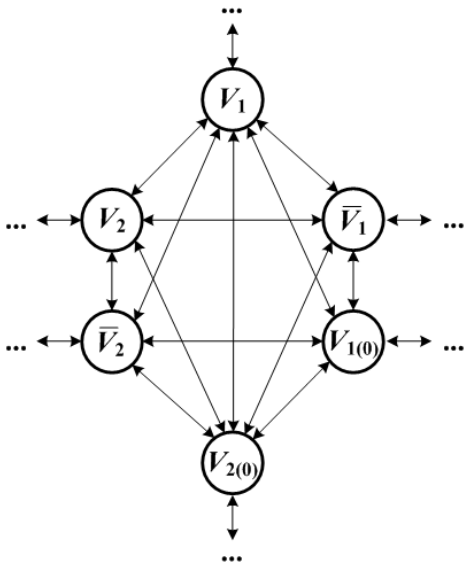

Рисунок 9 - Графы переходов между операторами (с учётом связей между операторами)

Алгоритм формирования системы признаков на основе полных (замкнутых) групп, Di – описание i-го сегмента сигнала в виде полных (замкнутых) групп:

$$\begin{gathered}\forall i=\overline{1,(N-1)}\\ \forall k=\overline{1,|D_i|}\\ \forall l=\overline{1,|D_{i+1}|}\\ MPG_{D_i}[D_i[k],D_{i+1}[l]]=MPG_{D_i}[D_i[k],D_{i+1}[l]]+1.\end{gathered} \tag{4}$$

Далее будут использоваться следующие обозначения PP_{na} (PP_{nm}) – матрица вероятностей переходов между полными группами на операции сложения (умножения), PP_s – на основе замкнутых групп, PP_{ca} – на основе замкнутых множеств на операции сложения, PP_{cm} – на основе замкнутых множеств на операции умножения.

При использовании в качестве систем признаков матриц вероятностей переходов, между описаниями сегментов можно учитывать вероятности переходов не только между i и $(i + 1)$ сегментом сигнала, но и учитывать связи между большим числом сегментов. Для i-го сегмента возможен учёт не только $(i + 1)$, $(i + 2)$ и дальнейших сегментов, но и $(i – 1)$, $(i – 2)$ сегментов, т.е. не только «будущего», но и «прошлого».

Пример формирования признакового описания сигнала в виде матрицы вероятностей переходов размером 140×140 элементов для полных групп показан на рис. 10. При вычислении значений матрицы рассматривались связи между описаниями только пары соседних сегментов.

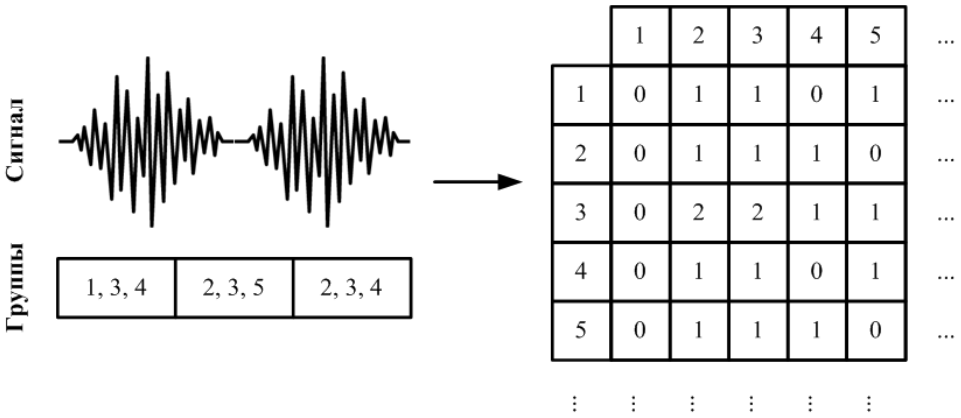

Рисунок 10 - Матрица вероятностей переходов между группами

Пример формирования признакового описания сигнала в виде 15 независимых матриц вероятностей переходов для операторов на показан рис. 11. При вычислении значений матрицы рассматривались связи между описания только пары соседних сегментов и учитываются вероятности переходов только для одного оператора.

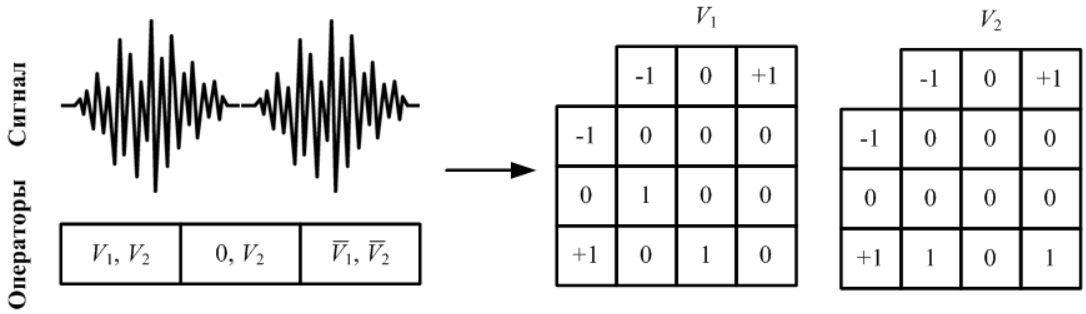

Рисунок 11 - Матрицы вероятностей переходов между операторами

Пример формирования признакового описания сигнала в виде матрицы вероятностей переходов для операторов размером 45×45 показан на рис. 12. При вычислении значений матрицы рассматривались связи между описания только пары соседних сегментов. В описаниях сегментов приведены значения только операторов V_1 и V_2.

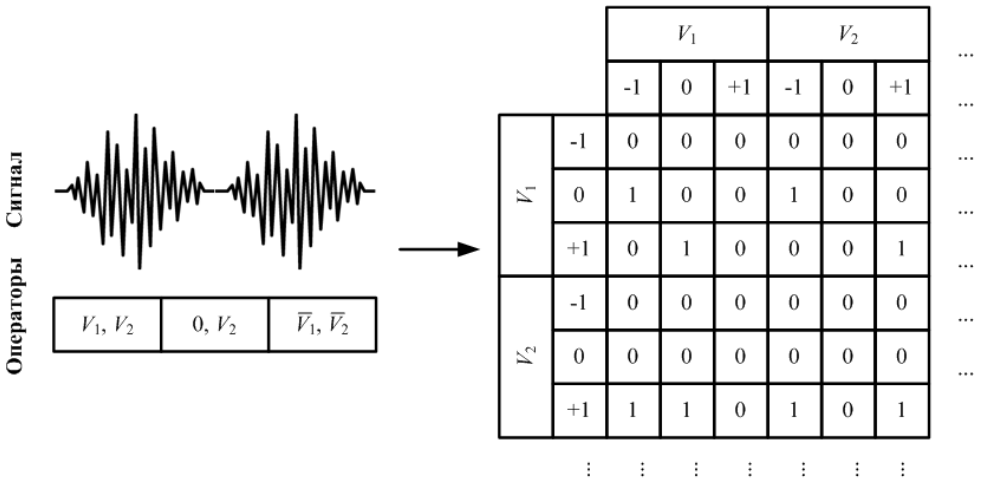

Рисунок 12 - Матрица вероятностей переходов между операторами

3.4. Разработка алгоритма принятия решения

Этап принятия решения основан на алгоритме композиции случайных деревьев принятия решения. Реализует этот алгоритм классификатор extraTrees языка R.

Композиция - это объединение нескольких алгоритмов в один. Идея заключается в том, чтобы обучить алгоритмы, а затем усреднить полученные от них ответы. Чтобы построить композицию, нужно сначала обучить N базовых алгоритмов, причем их нельзя обучать на всей обучающей выборке, так как в этом случае они получаются одинаковыми.

Один из способов сделать базовые алгоритмы различными - использовать рандомизацию - обучать базовые алгоритмы на разных подвыборках обучающей выборки. В нашем случае рандомизация достигается двумя способами: путем генерации случайного подмножества обучающей выборки и случайного подмножества признаков из всех существующих.

При этом производится выборка случайных двух третей наблюдений для обучения, а оставшаяся треть используется для оценки результата. Эти операции проделываем сотни раз. Результирующая модель будет получена результатом «голосования» набора полученных при моделировании деревьев.

Исходя из вышеописанного, составим алгоритм принятия решения:

1. Получаем N случайных подвыборок.
2. Каждая получившаяся подвыборка используется как обучающая выборка для построения соответствующего решающего дерева. Причем:
    - Дерево строится, пока в каждом листе окажется не более определенного числа объектов. Чем меньше объектов в каждом листе, тем получаются более сложные и переобученные решающие деревья с низким смещением.
    - Процесс построения дерева рандомизирован: на этапе выбора оптимального признака, по которому будет происходить разбиение, он ищется не среди всего множества признаков, а среди случайного подмножества.
    - Случайное подмножество выбирается заново каждый раз, когда необходимо разбить очередную вершину.
3. Построенные деревья объединяются в композицию: для нашей задачи регрессии берется усредненное значение результатов работы всех алгоритмов.

Главные преимущества композиции случайных деревьев заключаются в следующем:

1. Они не переобучаются при росте числа базовых алгоритмов.
2. При отсутствии корреляции между отдельно взятыми деревьями алгоритм может дать идеальный результат, так как разброс ответов всей композиции равен разбросу отдельного дерева деленного на число деревьев.

4. Разработка программных средств

Система, для которой разрабатывается модуль предсказания уровня глюкозы, работает по следующему принципу: пациент записывает ЭКГ с помощью смартфона, данные передаются на сервер, на котором происходит предсказание, затем результат передается пациенту и его лечащему врачу. Врач, в зависимости от результата, может связаться с пациентом для каких-либо рекомендаций. Схема работы системы представлена на рис. 13.

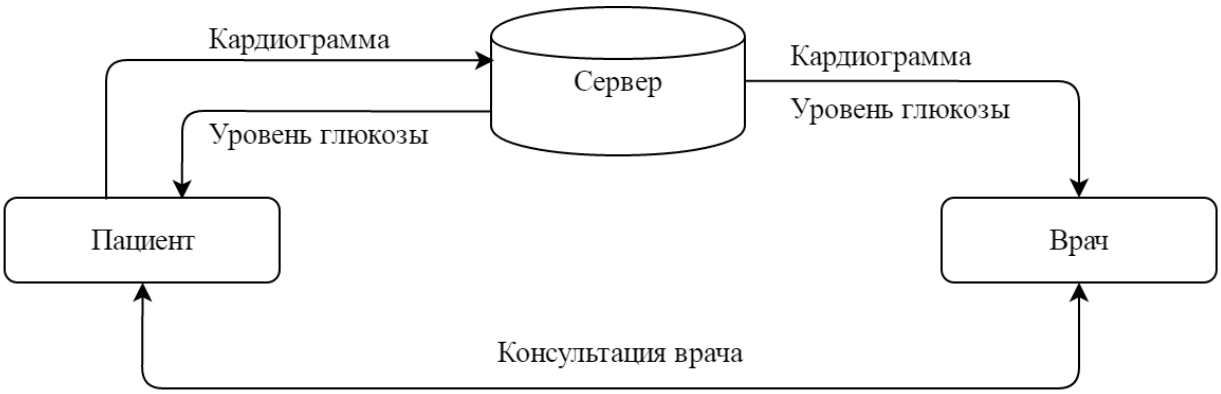

Рисунок 13 - Схема работы системы

Сервер, на котором происходят все вычисления, представляет собой удаленный компьютер с ОС Linux и не имеет специального интерфейса. У пациента для снятия ЭКГ и получения результатов есть специальное iOS-приложение, интерфейс представлен на рис. 14.

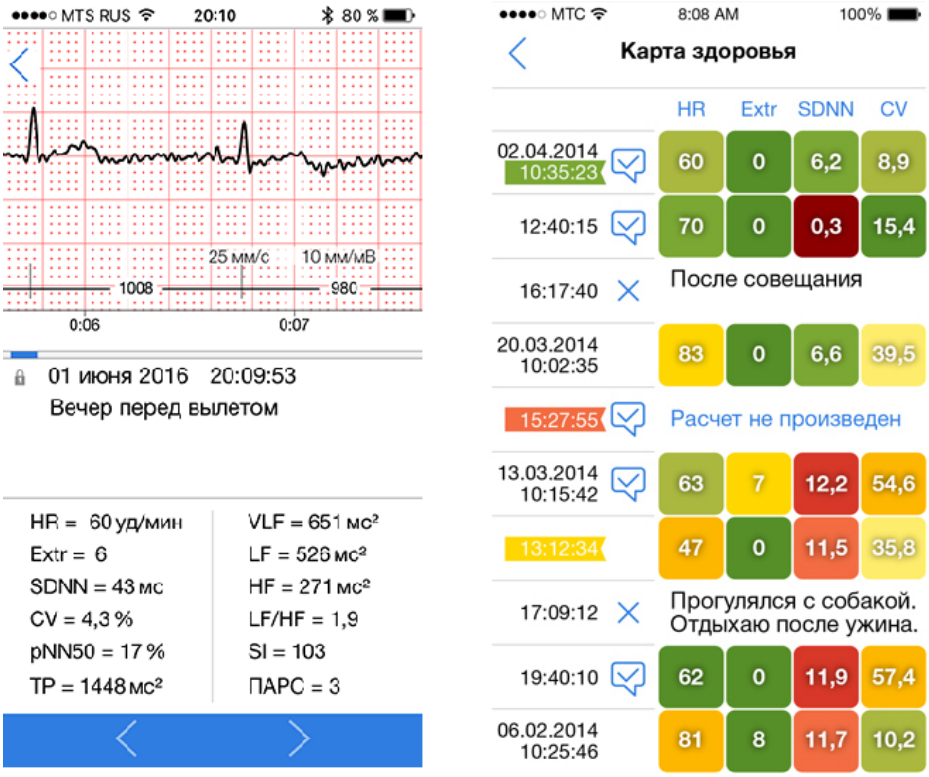

Рисунок 14 - Вид заполненного профиля

Приложение сохраняет результаты всех снятых ЭКГ и они всегда доступны для просмотра.

Помимо пациента доступ к результату измерений может получить его лечащий врач. С помощью iPad- или Windows-приложения он может отслеживать всех прикрепленных к нему пациентов (рис. 15).

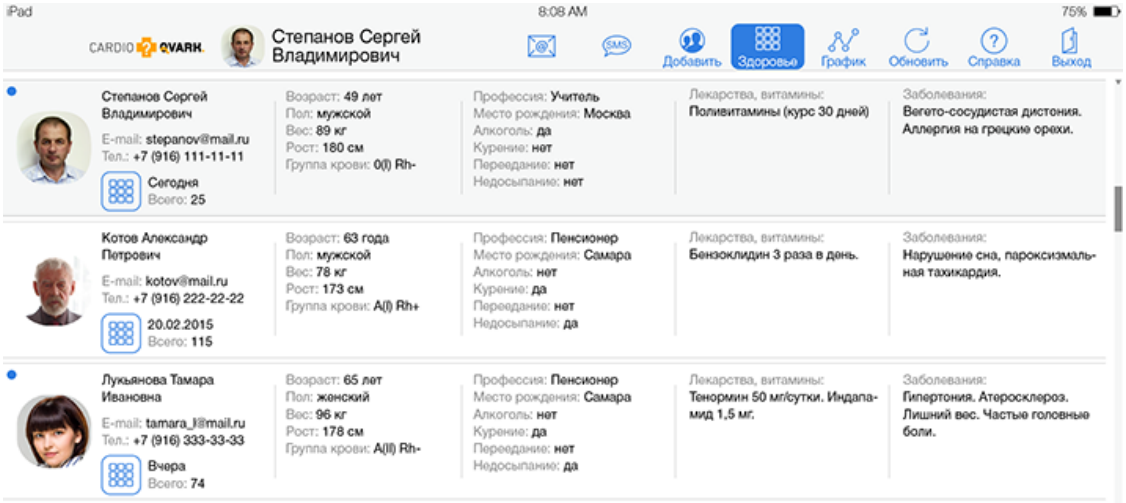

Рисунок 15 – Список пациентов

Рассмотрим основные методы, реализованные в модуле оценки уровня глюкозы, на примере пациента 1003.

ЭКГ-сигнал, полученный от пациента, представлен в виде звукового wav-файла. Для визуализации и последующего анализа используется следующий код:

```
#запись в массив названий всех файлов с ЭКГ-сигналами
lf <- list.files(path = paste("H:\\gdata\\1003\\", sep = "" ),
       pattern = '*.wav', all.files = FALSE, full.names = T)
#обработка каждого файла
for (i in 1:length(lf))
{
  sndObj <- readWave(lf[i])
  s1 <- sndObj@left
  m1 <- max(s1)
  m2 <- min(s1)
  fname <- basename(lf[i])
  fname <- paste(fname, '.png', sep = '')
  png(fname, height = 4, width = 8, units = 'in', res=300)
  plot(s1, type='l', col='black', xlab='Time (ms)',
       ylab='Amplitude', xlim = c(0,10000),ylim=c(m2, m1))
  dev.off()
}
```

С помощью полученных графиков мы можем визуально оценивать качество записанного сигнала. Если какое-то определенное измерение вызывает подозрение на плохое качество, используем следующий код, чтобы вывести график средних значений сигнала и его среднеквадратичных отклонений, поделенных на зоны.

```
lf <- readWave('H:/gdata/1003/filt/5.5_35957_1K.wav')
s1 <- lf@left
Sr_arr = array(dim = 60)
Ot_arr = array(dim = 60)
slen <- 1000 #шаг, с которым сигнал разбивается на зоны
for (i in 1:59)
{
  print(paste((i-1)*slen+1, (i*slen)))
  Sr <- mean(s1[((i-1)*slen+1):(i*slen)])
  Ot <- sd(s1[((i-1)*slen+1):(i*slen)])
  Sr_arr[i] <- Sr
  Ot_arr[i] <- Ot
}
plot(Sr_arr, main = "Средние значения 1003_5.5_35957_1K", type='l',
col='black')
plot(Ot_arr, main = "Среднеквадратичные отклонения 1003_5.5_35957_1K",
type='l', col='black')
```

Для обучения модели и для последующего предсказания результатов используется библиотека extraTrees языка R [6]. Полный исходный код, который подготавливает исходные данные, выполняет обучение модели и предсказание новых данных, представлен в Приложении А. Рассмотрим основные переменные и функции.

Библиотека extraTrees имеет 4 задаваемых параметра. Экспериментальным путем было выявлено, что изменение трех из них влияет на качество предсказания, поэтому один делаем константным.

- Ntree – количество деревьев; 900.
- Nodesize - размер листьев дерева; 1, 2.
- Mtry - количество признаков, испытанных в каждом узле; от 5 до 75 с шагом 5.
- NumRandomCuts - число случайных отрезков для каждой (случайным образом выбранной) функции; от 10 до 45 с шагом 5.

Изменение параметров реализовано следующим способом:

```
mtry_ <- seq(5, 75, 5)
numRandomCuts_ <- seq(10, 45, 5)
nodesize_ <- c(1, 2)
```

Для создания комбинаций из этих параметров используем функцию expand.grid() [7]:

```
mparams  <- expand.grid(nodesize_, mtry_, numRandomCuts_)
myr <- mparams
```

Для обучения модели у нас есть 660 видов признаковых описаний, которые тоже представляют комбинации из заданных параметров. Для получения 660 имен файлов используем следующие параметры:

```
mtype <- c('_1003_104_filt_ids.rdata') #общее начало названий файлов
siglen <- c('1e+05', '125000', '150000') #обрабатываемый сигнал
slen <- seq(from = 128, to = 1152, by = 256) #длины сегментов сигнала
```

Не все системы признаков унифицированы, поэтому для добавления каждой из них в название используется свой метод:

```
featn <- c('cls1d', 'oper2d', 'oper3d', 'poper2d', 'poper2di')
ddd <- expand.grid(siglen, sl, featn, 0, mtype)

featn_ <- c('full1d')
fgn    <- c('fmul', 'fsum')
ddd2 <- expand.grid(siglen, sl, featn_, fgn, mtype)

featn_ <- c('full2d')
fgn    <- c('fmul', 'fsum')
ddd3 <- expand.grid(siglen, sl, featn_, fgn, mtype)

featn_ <- c('pfull2d')
fgn    <- c('fmul', 'fsum')
ddd4 <- expand.grid(siglen, sl, featn_, fgn, mtype)
```

В итоге получаем переменную, хранящую в себе 660 названий файлов для обучения моделей:

```
fnames <- rbind(ddd, ddd2, ddd3, ddd4)
```

При вызове основной функции getresults() мы передаем в нее поочередно все комбинации параметров:

```
list(getresults(X, Y, myr[i1, ]),  myr[i1, 1], myr[i1, 2], myr[i1, 3],
myr[i1, 4],  fn1, fnames[i,])
где i = 1:660, i1 = 1:dim(myr)[1]
```

Результаты, полученные после обучения моделей по каждой из комбинаций, заносятся в массив reszz, который после окончания обучения сохраняется в отдельный файл для последующего анализа результатов.

В функции getresults() есть следующие основные переменные:

- bi – массив, в котором хранятся индексы элементов обучающей выборки.
- Y1 (train_L), X1 (train_T) – значения глюкозы и признаки обучающей выборки
- Y2 (test_L), X2 (test_T) – значения глюкозы и признаки тестовой выборки
- pred – результаты предсказания для тестовой выбоки
- errarr – массив, хранящий следующие значения: ошибку предсказания, среднеквадратичное отклонение обучающей и тестовой выборок, зоны Кларка и коэффициент корреляции Спирмена.

Так как функция каждый раз выбирает разные объекты для обучающей выборки (из нескольких одинаковых может быть выбран любой, индекс будет различен), то необходимо в итоговый массив reszz записывать не только результат предсказания (массив errarr), но и индексы обучающей выборки. Поэтому функция возвращает список, содержащий два массива:

```
return(list(errarr, bi))
```

После окончания обучения необходимо выбрать модель с самыми лучшими результатами. Для этого используем следующий код:

```
mres <- reszz
```

```
for (i in 1:length(mres))
{
  for (j in 1:length(mres[[i]]))
    if (mres[[i]][[j]][[1]][[1]][4] > 60 &&
        && mres[[i]][[j]][[1]][[1]][9] > 0.6 &&
        && abs(mres[[i]][[j]][[1]][[1]][3] -
        - mres[[i]][[j]][[1]][[1]][2]) < 2.2 )
    {
      print(i)
      print(j)
      d1 <- mres[[i]][[j]][[1]][[1]]
      print(paste('A =', floor(d1[4]), 'B =', floor(d1[5]), 'C =',
             floor(d1[6]), 'D =', floor(d1[7]), 'СПИР =', d1[9],
             'РАЗН =', d1[3]-d1[2]))
      print(mres[[i]][[j]][[1]][[1]])
      print(mres[[i]][[j]][[6]])
    }
}
```

Здесь мы указываем необходимые значения, которым должны удовлетворять модели:

```
mres[[i]][[j]][[1]][[1]][4] > 60 – зона A
mres[[i]][[j]][[1]][[1]][9] > 0.6 – коэффициент корреляции Спирмена
abs(mres[[i]][[j]][[1]][[1]][3] – mres[[i]][[j]][[1]][[1]][2]) < 2.2 –
разница между среднеквадратичными отклонениями обучающей и тестовой выборками
```

5. Тестирование системы

5.1. Описание набора данных

Все исходные данные разбиваются на 3 части:

1. Обучающая выборка.
2. Выборка для тестирования полученной модели.
3. Итоговая тестовая выборка.

Обучающая выборка строится, исходя из следующего принципа: в первых двух частях исходных данных мы вычисляем количество одинаковых объектов каждого вида и для обучения берем пропорциональное этому число.

Приведем поясняющую таблицу.

Общее число объектов	Число объектов для обучения
От 1 до 4	1
От 5 до 10	2
От 11 до 14	3
От 15 до 20	4
От 21 до 25	5

Таблица 1 – Принцип построения обучающей выборки

Проиллюстрируем на графике (рис. 16).

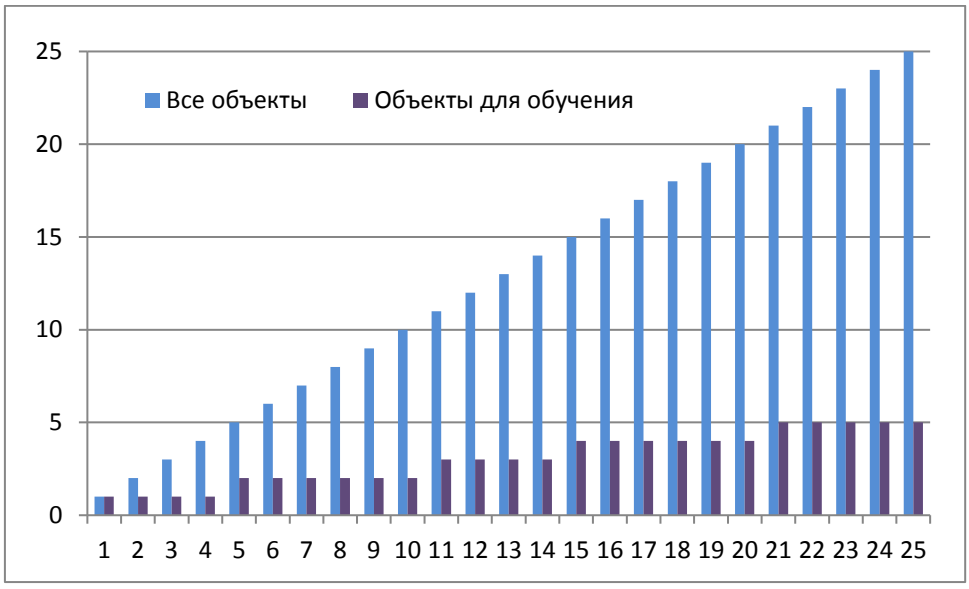

Рисунок 16 – Принцип построения обучающей выборки

Таким образом, чем чаще встречается объект определенного вида, тем больше он принимает участия в обучении. Другими словами, чем чаще у человека встречается

определенный уровень содержания в крови глюкозы, тем чаще он встречается обучающей выборке.

Итоговое тестирование полученной модели проводится по третьей части данных, по независимой выборке. В ней могут встречаться значения глюкозы, на которых не проводилось обучение.

5.2. Описание методики тестирования

Для проверки полученных результатов используется несколько методов: диаграмма Кларка, коэффициент корреляции Спирмена и среднеквадратичное отклонение. Рассмотрим каждый из них.

5.2.1. Диаграмма Кларка

Диаграмма Кларка используется для оценки клинической значимости значений разности между значениями глюкозы, полученными точным лабораторным методом и значениями, полученными с помощью экспериментального прибора. Диаграмма разделена на зоны, каждая из которых соответствует степени тяжести ошибок измерения.

Вид диаграммы представлен на рис.17.

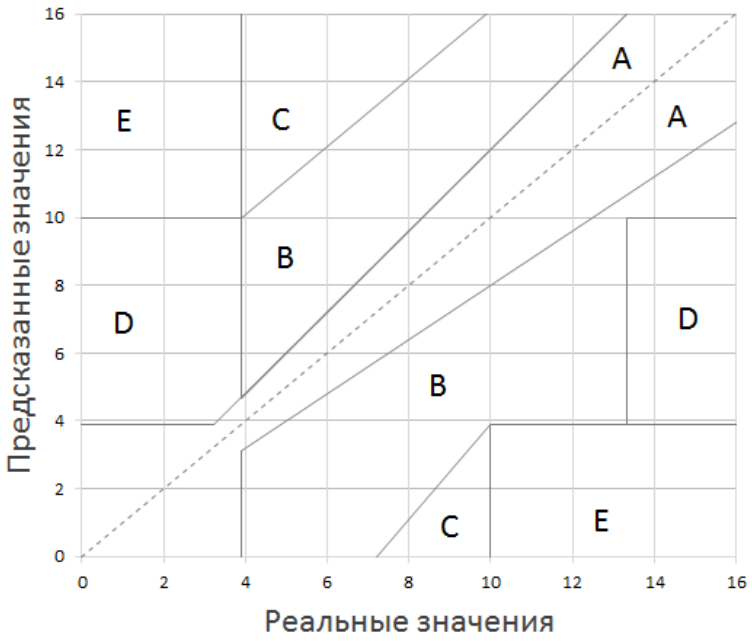

Рисунок 17 – Диаграмма Кларка

Рассмотрим каждую из зон:

- A - зона отсутствия ошибок. Она представляет собой значения глюкозы, которые отклоняются от эталонного не больше, чем на 20%. Значения, попадающие в этот диапазон, приведут к клинически правильным решениям относительно лечения.

- B - зона несущественных ошибок. Она представляет значения, которые отклоняются от эталонных больше, чем на 20%, но приведут к безвредному лечению или отсутствию его на основе наших предположений.
- C - зона существенных ошибок. Значения приведут к переходу границы приемлемых уровней глюкозы в крови; такое лечение может привести к фактическому падению глюкозы в крови до уровня ниже 70 мг/дл или повышению его выше 180 мг/дл, следовательно, это нанесет вред больному.
- D - зона опасных ошибок. Она представляет собой опасный отказ для выявления и лечения ошибок. Фактические значения глюкозы находятся вне целевого диапазона, но значения для пациента генерируется в пределах целевого диапазона.
- E - зона жизненно опасных ошибок. Она является зоной ошибочного лечения. Генерируемые значения в пределах этой зоны противоположны эталонным значениям, и, следовательно, соответствующие решения в отношении лечения будет противоположным необходимым.

Таким образом, значения, содержащиеся в зонах A и B являются клинически приемлемыми, в то время как значения в зонах C, D, и E являются потенциально опасными и, следовательно, являются клинически значимыми ошибками.

Пример диаграммы Кларка в работе представлен на рис.18.

Рисунок 18 – Пример работы диаграммы Кларка

Здесь, как мы видим, большая часть предсказанных значений попадает в зоны A и B, то есть в зону отсутствия ошибок и зону несущественных ошибок, но при этом два значения

находятся в зоне C, а значит говорят о существенной ошибке в предсказании, которая может нанести вред больному.

Если посмотреть на значения этих двух точек (табл. 2), то мы видим, что значения отклоняются от правильных более, чем в два раза.

Значение глюкометра	Значение алгоритма	Разность значений	Зона
6,9	14,3	7,4	C
4,3	14,3	10,0	C

Таблица 2 – Значения глюкозы, попавшие в зону C

5.2.2. Коэффициент корреляции Спирмена

Корреляция – это статистическая взаимосвязь нескольких случайных величин, либо величин, которые можно с некоторой допустимой степенью точности считать таковыми. Математической мерой корреляции двух случайных величин служит коэффициент корреляции. Это величина в диапазоне от -1 до +1, которая характеризует степень связи величин. Значение +1 говорит о том, что при увеличении одной переменной увеличивается значение другой переменной, при -1 наоборот.

Значения коэффициента корреляции можно интерпретировать по шкале Чеддока, которая характеризует показатели тесноты связи между двумя величинами. Шкала показана в табл. 3.

Величина коэффициента корреляции	0.1 – 0.3	0.3 – 0.5	0.5 – 0.7	0.7 – 0.9	0.9 – 1.0
Характеристика силы связи	Слабая	Умеренная	Заметная	Высокая	Весьма высокая
		Средняя		Сильная	

Таблица 3 – Шкала Чеддока

Есть несколько видов коэффициентов корреляции. Рассмотрим их и выберем наиболее подходящий для нашей задачи.

1. Коэффициент Фехнера.

   Зависит только от знаков отклонений величин от своих средних значений и не учитывает их величину, он характеризует не столько тесноту связи, сколько ее наличие и направление.

2. Метод наименьших квадратов.

   Метод чувствителен к выбросам. Требует нормального распределения. При попытке описать изучаемое явление с помощью математического уравнения, прогноз будет точен для небольшого периода времени и уравнение регрессии следует пересчитывать по мере поступления новой информации.

3. Регрессионный анализ.

   Выбросы увеличивают стандартную ошибку коэффициента, снижают значение R-квадрат.

4. Коэффициент корреляции Пирсона.

   Подходит для оценки взаимосвязи между нормальными распределениями переменных. Не очень устойчив к выбросам - при их наличии можно ошибочно сделать вывод о наличии корреляции между переменными.

5. Коэффициент ранговой корреляции Спирмена.

   Непараметрический аналог коэффициента корреляции Пирсона. Подходит, если распределение исследуемых переменных отличается от нормального, или возможны выбросы. Позволяет выявлять не только линейные связи, а любые, которые могут быть описаны монотонной функцией.

6. Коэффициент Кендалла.

   Он дает несколько более строгую оценку связи, нежели коэффициент Спирмена. Считается более информативным. Обычно коэффициент Кендалла меньше коэффициента Спирмена.

7. Критерий Ширахатэ.

   Является аналогом критерия значимости ранговой корреляции Спирмена, но более эффективен для малых выборок.

Для того чтобы выбрать коэффициент корреляции, нужно определиться с некоторыми моментами:

1. Имеются ли выбросы.
2. Нормальное ли распределение величин.

Поскольку результат работы алгоритма существенно зависит от того, как был снят ЭКГ-сигнал (не шевелился ли пациент, плотно ли прижал палец к чехлу и т.д.), будем считать, что в работе алгоритма возможны выбросы.

Теперь проверим распределение на нормальность. Для этого воспользуемся показателями асимметрии и эксцесса.

Асимметрия (A) - это мера несимметричности графика плотности реального распределения в сравнении с нормальным распределением. Эксцесс (E) - мера вытянутости графика плотности реального распределения в сравнении с нормальным распределением.

По численным значениям асимметрии и эксцесса можно приближенно оценить нормальность распределения результатов испытаний.

A и E рассчитываем по формулам 5 и 6

$$A = \frac{1}{ns^3}\sum_{i=1}^{n}(x_i - \bar{x})^3 \quad (5)$$

$$E = \frac{1}{ns^4}\sum_{i=1}^{n}(x_i - \bar{x})^4 - 3 \quad (6)$$

Или при помощи статистических функций в Microsoft Excel СКОС (для А) и ЭКСЦЕСС (для Е).

Далее рассчитаем дисперсию А и Е с помощью формул 7 и 8.

$$D(A) = \frac{6(n-1)}{(n+1)(n+3)} \quad (7)$$

$$D(E) = \frac{24(n-2)(n-3)n}{(n-1)^2(n+3)(n+5)} \quad (8)$$

Где n – число испытаний, в нашем случае число измерений глюкозы.

Если формулы 9 и 10 верны, то распределение считается нормальным.

$$|A| \le 3\sqrt{D(A)} \quad (9)$$

$$|E| \le 5\sqrt{D(E)} \quad (10)$$

Теперь произведем расчеты для всех пациентов. Результаты занесем в табл. 4.

	Пациенты		
	1003	1430	1696
n	104	370	211
A	1.187	0.306	0.242
E	2.559	-0.264	0.531
D(A)	0.055	0.016	0.028
D(E)	0.208	0.063	0.108
|A|	0.704	0.379	0.499
|E|	2.279	1.255	1.643
Распределение	Ненормальное	Нормальное	Нормальное

Таблица 4 – Результаты проверки типа распределения

Видим, что у значений глюкозы пациента 1003 распределение отлично от нормального, у 1430 нормальное, но близко к границе, у 1696 нормальное. Для того чтобы оценка результатов была одинаковая для всех пациентов мы выберем коэффициент корреляции, в котором возможно ненормальное распределение значений, а также возможны выбросы. Этим условиям удовлетворяет коэффициент корреляции Спирмена.

5.3. Результаты вычислительного эксперимента

Рассмотрим результаты работы алгоритма по трем пациентам. По каждому представим проценты зон A-E диаграммы Кларка, графическое представление диаграммы, значения коэффициента корреляции Спирмена, разницу среднеквадратического отклонения значений, полученных от инвазивного глюкометра и значений, полученных с помощью алгоритма, график соответствия реальных и предсказанных значений глюкозы. Для каждого пациента рассмотрим результат, полученный на основе трех систем признаков, PVI, PVD и PVS, представленных в пункте 3.3, и произведем сравнение результатов. В скобках около каждой системы признаков указаны параметра классификатора, которые дали лучший результат.

5.3.1. Пациент 1003

- Система признаков PVI (Nodesize = 2, Mtry = 75, NumRandomCuts = 45)

Обобщенные результаты представлены в табл. 5.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
97.33%	2.67%	0.00%	0.00%	0.00%	0.807	0.132

Таблица 5 – Результаты работы алгоритма пациента 1003, PVI

График соответствия реальных и предсказанных значений глюкозы показан на рис. 19, диаграмма ошибок Кларка - на рис. 20.

Рисунок 19 – Реальные и предсказанные значения глюкозы пациента 1003, PVI

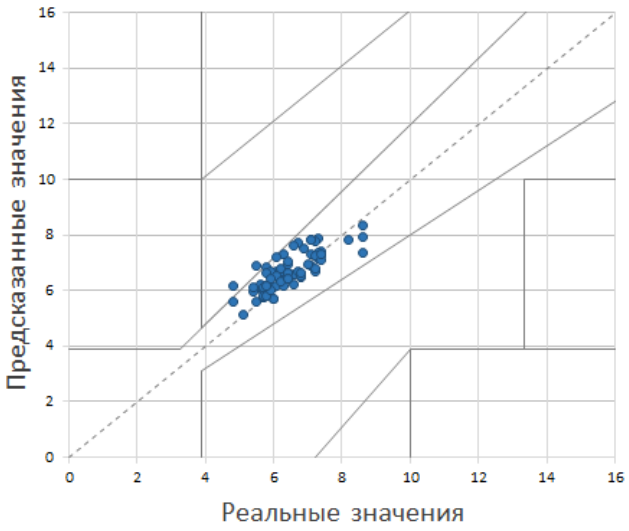

Рисунок 20 – Диаграмма ошибок Кларка пациента 1003, PVI

- Система признаков PVD (Nodesize = 1, Mtry = 55, NumRandomCuts = 15)

  Обобщенные результаты представлены в табл. 6.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
96.00%	4.00%	0.00%	0.00%	0.00%	0.716	0.296

Таблица 6 – Результаты работы алгоритма пациента 1003, PVD

График соответствия реальных и предсказанных значений глюкозы показан на рис.21, диаграмма ошибок Кларка - на рис.22.

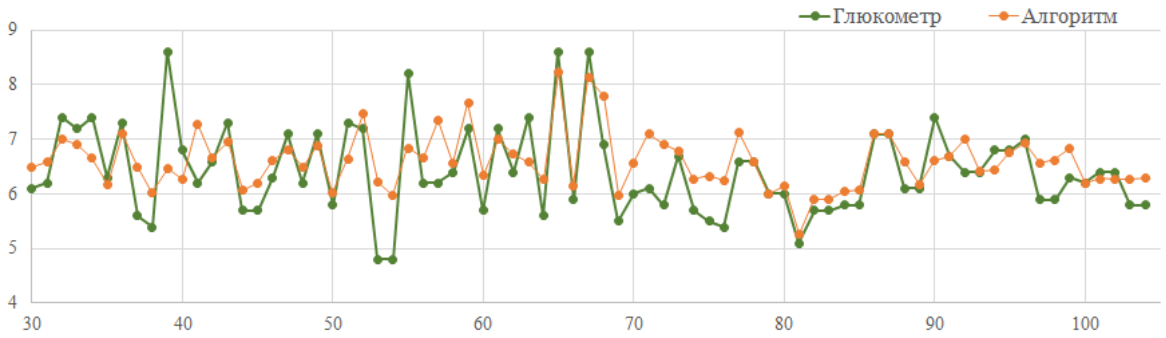

Рисунок 21 – Реальные и предсказанные значения глюкозы пациента 1003, PVD

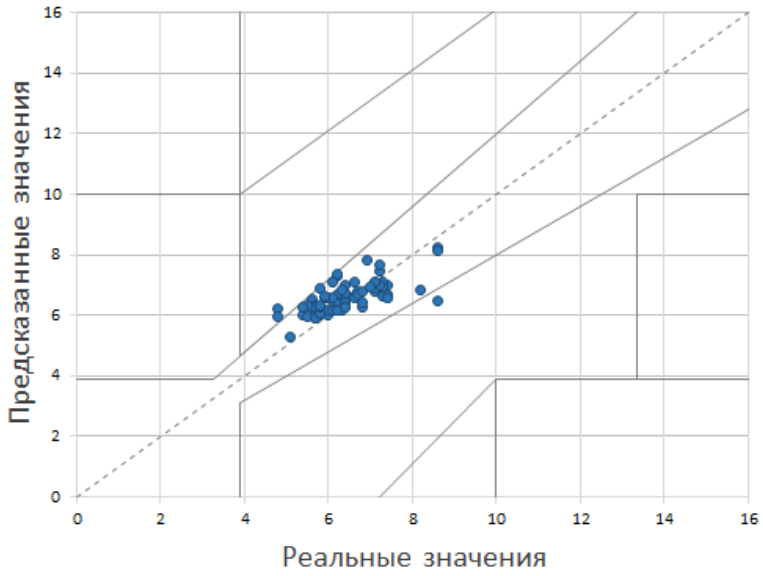

Рисунок 22 – Диаграмма ошибок Кларка пациента 1003, PVD

- Система признаков PVS (Nodesize = 1, Mtry = 65, NumRandomCuts = 30)

  Обобщенные результаты представлены в табл. 7.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
97.33%	2.67%	0.00%	0.00%	0.00%	0.741	0.329

Таблица 7 – Результаты работы алгоритма пациента 1003, PVS

График соответствия реальных и предсказанных значений глюкозы показан на рис.23, диаграмма ошибок Кларка - на рис.24.

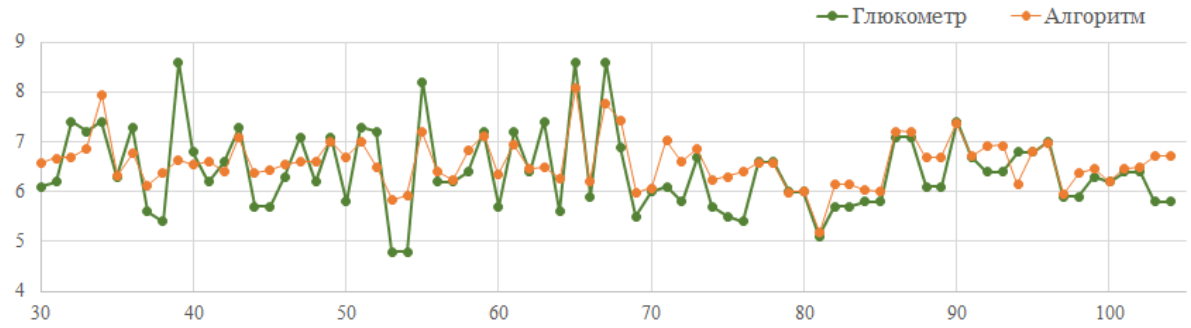

Рисунок 23 – Реальные и предсказанные значения глюкозы пациента 1003, PVS

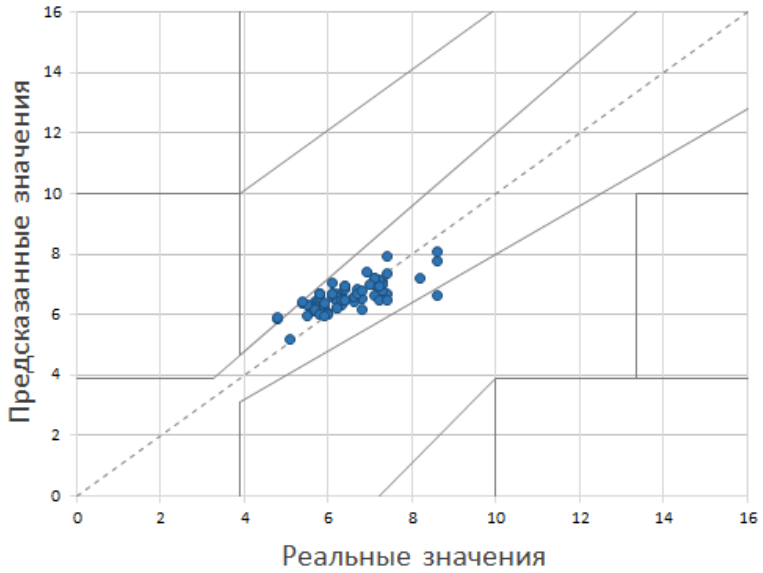

Рисунок 24 – Диаграмма ошибок Кларка пациента 1003, PVS

Сравнение результатов по трем системам признаков пациента 1003 приведены в табл. 8.

Спирмен	$SD_{гл}$-$SD_{ал}$	Зона A	Зона B	Зона C	Зона D
PVI					
0.807	0.132	97.33%	2.67%	0.00%	0.00%
PVD					
0.716	0.296	96.00%	4.00%	0.00%	0.00%
PVS					
0.741	0.329	97.33%	2.67%	0.00%	0.00%

Таблица 8 – Сравнение систем признаков пациента 1003

5.3.2. Пациент 1430

- Система признаков PVI (Nodesize = 1, Mtry = 15, NumRandomCuts = 35)

Обобщенные результаты представлены в табл. 9.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
64.98%	33.18%	1.84%	0.00%	0.00%	0.64	1.456

Таблица 9 – Результаты работы алгоритма пациента 1430, PVI

Графики соответствия реальных и предсказанных значений глюкозы показаны на рис. 25, диаграмма ошибок Кларка - на рис. 26.

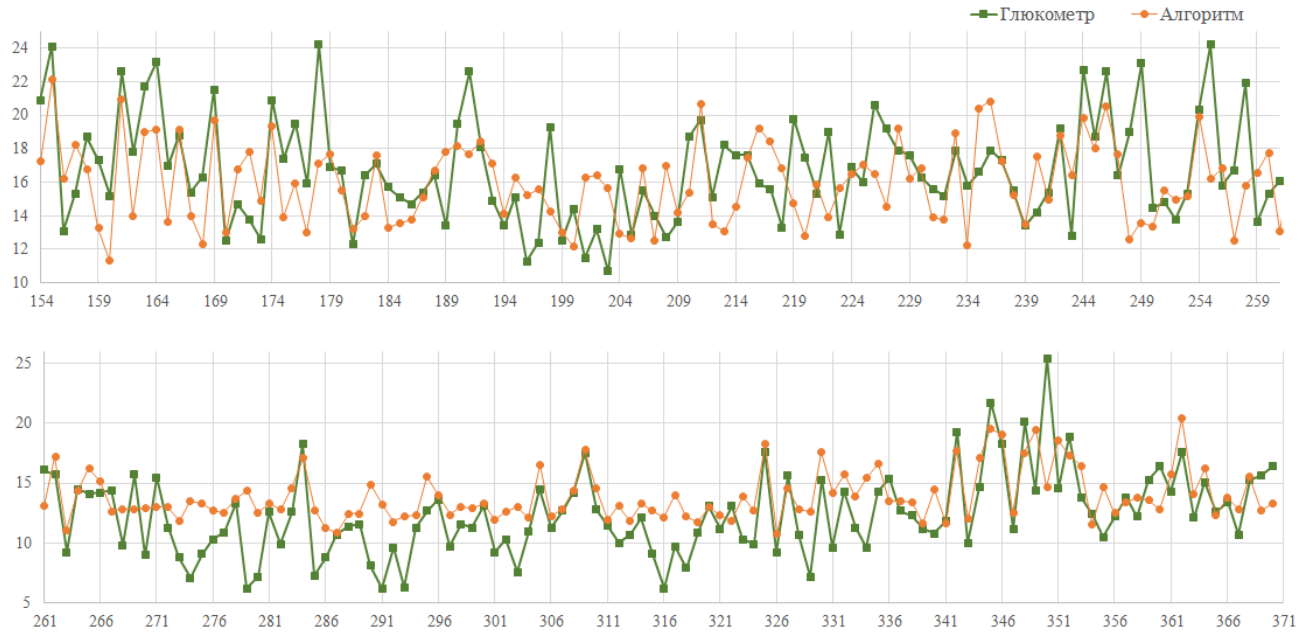

Рисунок 25 – Реальные и предсказанные значения глюкозы пациента 1430, PVI

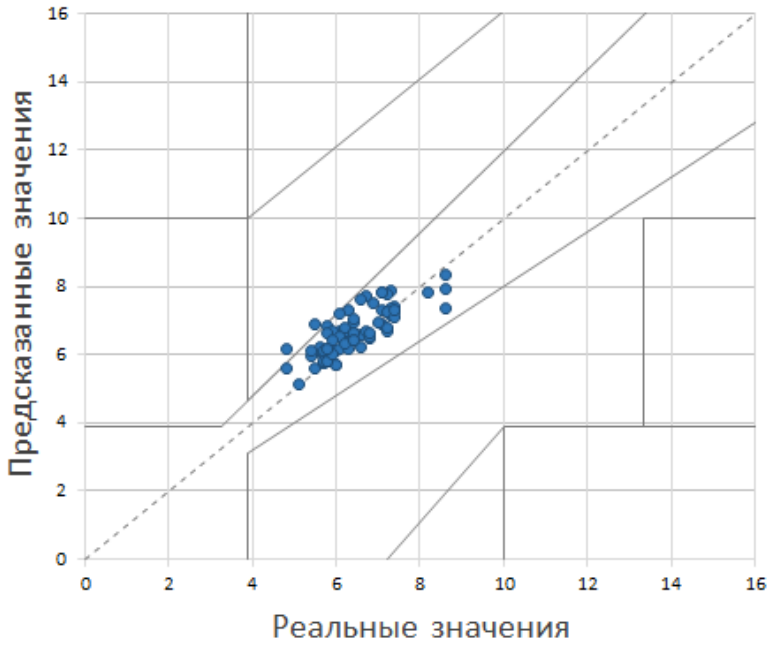

Рисунок 26 – Диаграмма ошибок Кларка пациента 1430, PVI

- Система признаков PVD (Nodesize = 2, Mtry = 35, NumRandomCuts = 15)

  Обобщенные результаты представлены в табл. 10.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
65.44%	31.80%	2.76%	0.00%	0.00%	0.60	1.409

Таблица 10 – Результаты работы алгоритма пациента 1430, PVD

Графики соответствия реальных и предсказанных значений глюкозы показаны на рис. 27, диаграмма ошибок Кларка - на рис. 28.

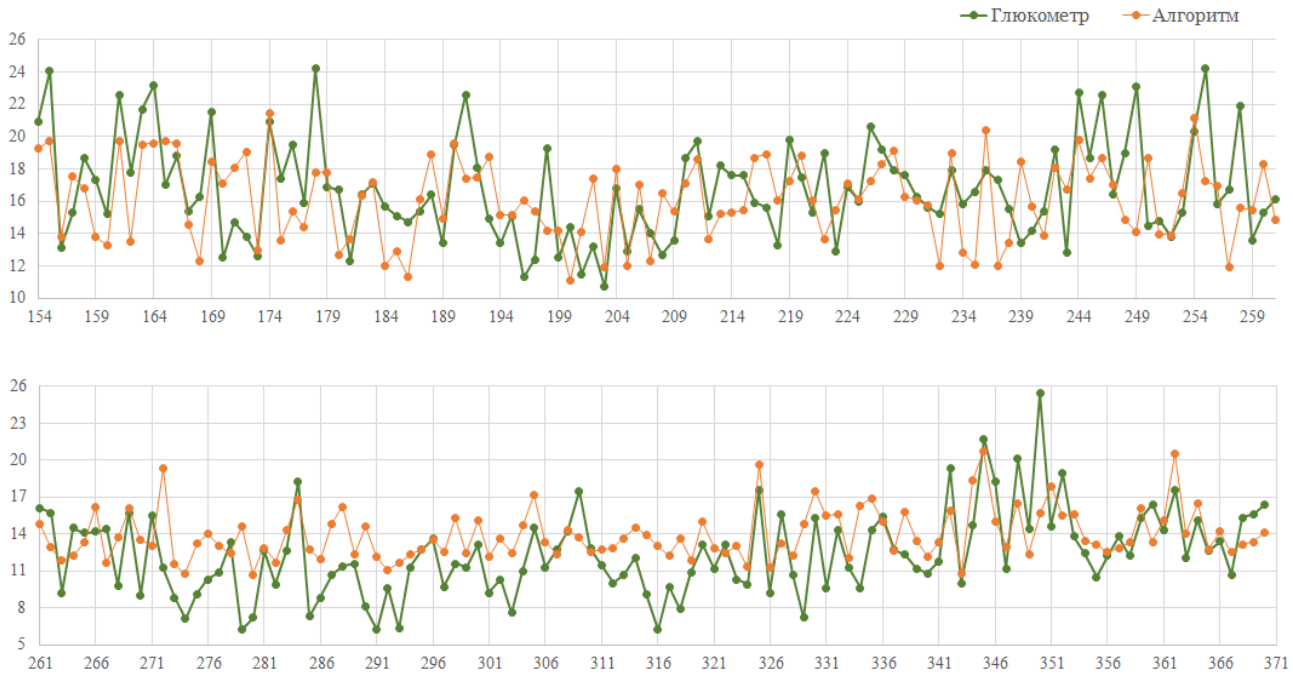

Рисунок 27 – Реальные и предсказанные значения глюкозы пациента 1430, PVD

Рисунок 28 – Диаграмма ошибок Кларка пациента 1430, PVD

- Система признаков PVS (Nodesize = 1, Mtry = 20, NumRandomCuts = 25)

  Обобщенные результаты представлены в табл. 11.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
65.44%	32.72%	1.84%	0.00%	0.00%	0.62	1.328

Таблица 11 – Результаты работы алгоритма пациента 1430, PVS

Графики соответствия реальных и предсказанных значений глюкозы показаны на рис. 29, диаграмма ошибок Кларка - на рис. 30.

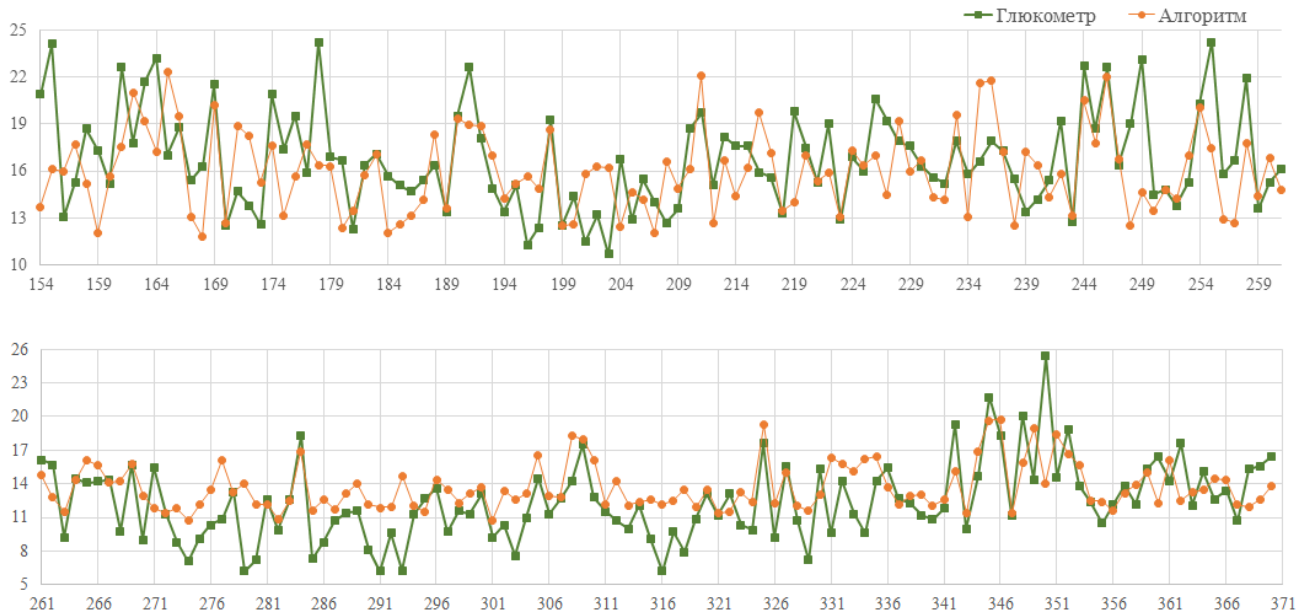

Рисунок 29 – Реальные и предсказанные значения глюкозы пациента 1430, PVS

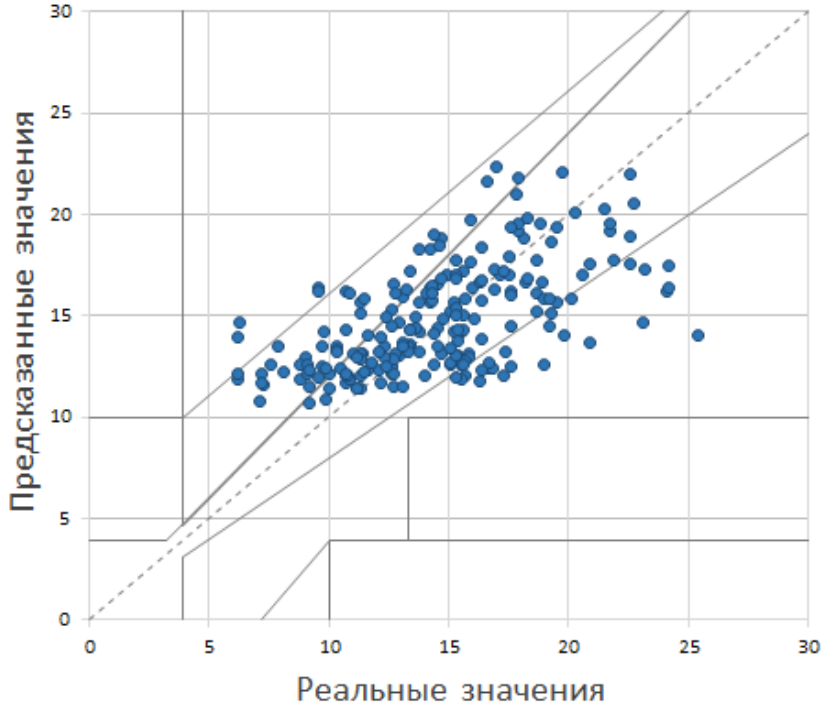

Рисунок 30 – Диаграмма ошибок Кларка пациента 1430, PVS

Сравнение результатов по трем системам признаков пациента 1430 приведены в табл. 12.

Спирмен	$SD_{гл}$-$SD_{ал}$	Зона A	Зона B	Зона C	Зона D
PVI					
0.64	1.456	64.98%	33.18%	1.84%	0.00%
PVD					
0.60	1.409	65.44%	31.80%	2.76%	0.00%
PVS					
0.62	1.328	65.44%	32.72%	1.84%	0.00%

Таблица 12 – Сравнение систем признаков пациента 1430

5.3.3. Пациент 1696

- Система признаков PVI (Nodesize = 2, Mtry = 35, NumRandomCuts = 10)

Обобщенные результаты представлены в табл. 13.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
91.67%	8.33%	0.00%	0.00%	0.00%	0.50	1.159

Таблица 13 – Результаты работы алгоритма пациента 1696, PVI

График соответствия реальных и предсказанных значений глюкозы показан на рис. 31, диаграмма ошибок Кларка - на рис. 32.

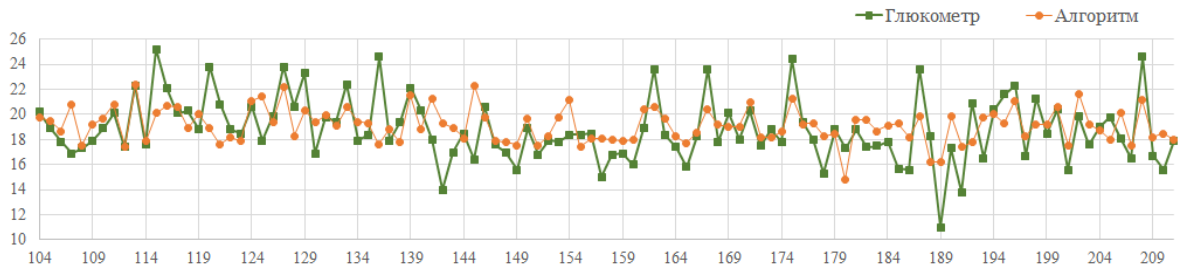

Рисунок 31 – Реальные и предсказанные значения глюкозы пациента 1696, PVI

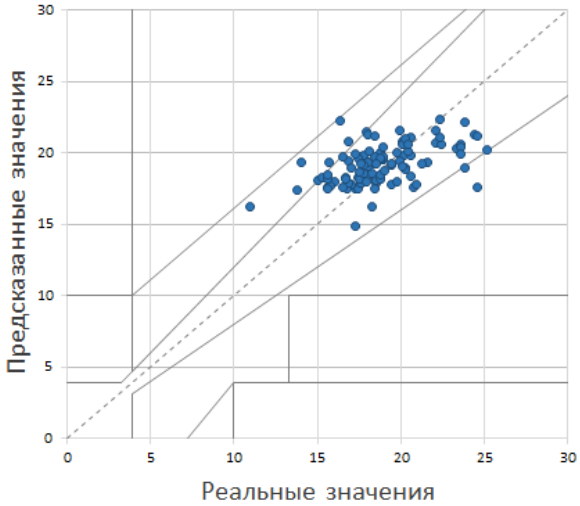

Рисунок 32 – Диаграмма ошибок Кларка пациента 1696, PVI

- Система признаков PVD (Nodesize = 2, Mtry = 65, NumRandomCuts = 20)

  Обобщенные результаты представлены в табл. 14.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
90.74%	7.41%	1.85%	0.00%	0.00%	0.47	0.972

Таблица 14 – Результаты работы алгоритма пациента 1696, PVD

График соответствия реальных и предсказанных значений глюкозы показан на рис. 33, диаграмма ошибок Кларка - на рис. 34.

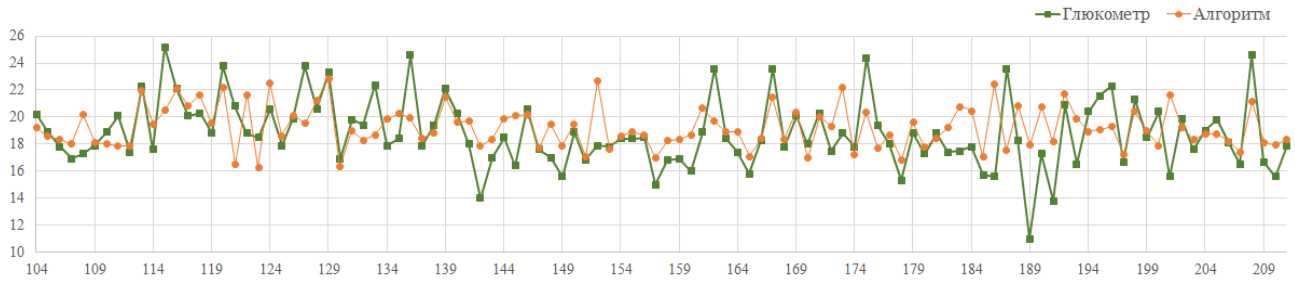

Рисунок 33 – Реальные и предсказанные значения глюкозы пациента 1696, PVD

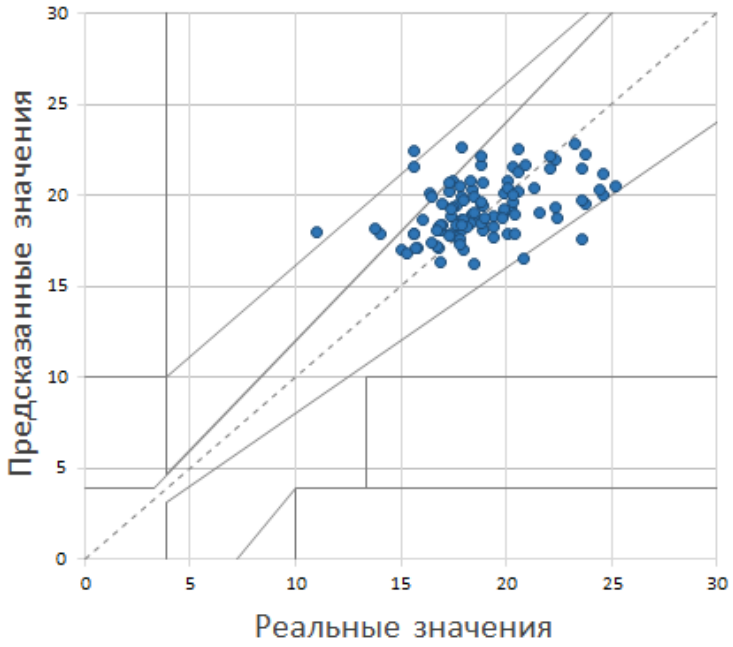

Рисунок 34 – Диаграмма ошибок Кларка пациента 1696, PVD

- Система признаков PVS (Nodesize = 2, Mtry = 50, NumRandomCuts = 30)

  Обобщенные результаты представлены в табл. 15.

Зона A	Зона B	Зона C	Зона D	Зона E	Спирмен	$SD_{гл}$-$SD_{ал}$
90.74%	7.41%	1.85%	0.00%	0.00%	0.46	1.789

Таблица 15 – Результаты работы алгоритма пациента 1696, PVS

График соответствия реальных и предсказанных значений глюкозы показан на рис. 35, диаграмма ошибок Кларка - на рис. 36.

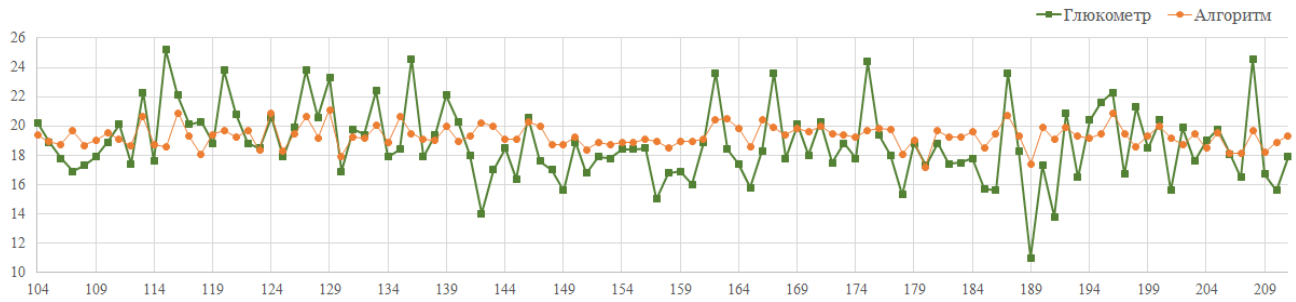

Рисунок 35 – Реальные и предсказанные значения глюкозы пациента 1696, PVS

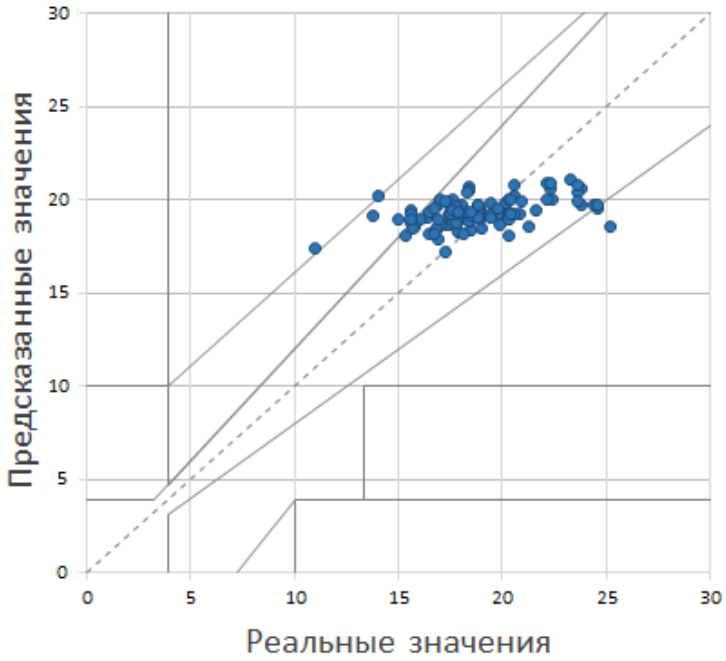

Рисунок 36 – Диаграмма ошибок Кларка пациента 1696, PVS

Сравнение результатов по трем системам признаков приведены в табл. 16.

Спирмен	$SD_{гл}$-$SD_{ал}$	Зона A	Зона B	Зона C	Зона D
PVI					
0.50	1.159	91.67%	8.33%	0.00%	0.00%
PVD					
0.47	0.972	90.74%	7.41%	1.85%	0.00%
PVS					
0.46	1.789	90.74%	7.41%	1.85%	0.00%

Таблица 16 – Сравнение систем признаков пациента 1430

Сведем лучшие из получившихся результатов в табл. 17.

Спирмен	$SD_{гл}$-$SD_{ал}$	Зона A	Зона B	Зона C	Зона D
Пациент 1003					
0.81	0.132	97.33%	2.67%	0.00%	0.00%
Пациент 1430					
0.62	1.328	65.44%	32.72%	1.84%	0.00%
Пациент 1696					
0.50	1.159	91.67%	8.33%	0.00%	0.00%

Таблица 17 – Обобщенные результаты

Сравним результаты, полученные с помощью классификатора extraTrees (в основе которого лежит алгоритм композиции случайных деревьев) с результатами, полученными ранее с помощью классификаторов xgboost (градиентный бустинг) и svr (метод опорных векторов).

Лучшие результаты, полученные ранее, представлены в табл. 18.

Спирмен	$SD_{гл}$-$SD_{ал}$	Зона A	Зона B	Зона C	Зона D
Пациент 1003					
0.72	-0.104	92.47%	7.53%	0.00%	0.00%
Пациент 1430					
0.46	0.156	59.35%	29.67%	7.72%	3.25%
Пациент 1696					
0.52	-0.131	83.49%	16.51%	0.00%	0.00%

Таблица 18 – Результаты, полученные ранее

Как мы видим, результаты, полученные в ходе данной работы, лучше. Рассмотрим их отдельно по каждому пациенту.

- 1003

  Коэффициент корреляции Спирмена увеличился на 9%, результаты предсказаний в зоне B уменьшились на 5%, соответственно в зоне A они увеличились на 5%.

- 1430

  Коэффициент корреляции Спирмена увеличился на 18%, зона D (опасных ошибок предсказания) полностью исчезла, зона C сократилась на 5.8%, зона A увеличилась на 5.3%.

- 1696

  Коэффициент корреляции Спирмена остался почти без изменений (+ 0,02%), зона B сократилась на 8%, зона A соответственно увеличилась на 8%.

Заключение

В результате выполнения выпускной квалификационной работы была спроектирована и программно реализована система предсказания уровня глюкозы в крови по ЭКГ-сигналу. Для определения численного значения глюкозы использовался метод композиции случайных деревьев.

Созданная программная система предназначена для определения уровня глюкозы в крови человека по готовой электрокардиограмме. Тестирование системы подтвердило её работоспособность и возможность использования для решения поставленной задачи. Дальнейшее развитие системы будет направлено на улучшение результатов предсказания.

Список литературы

1. Perez-Meana H. (ed.). Advances in Audio and Speech Signal Processing: Technologies and Applications: Technologies and Applications. – Igi Global, 2007, Chapter XIII, page 374.
2. Физические интерпретации элементов алгебры изображения - Утробин В.А. - Успехи физических наук, Т. 174, №10, 2004 г.
3. «Определение уровня глюкозы в крови человека на основе электрокардиографического сигнала» - В.Е. Гай, С.С. Бобко, Н.А. Домнина - Сборник трудов ИСТ-2017, 2017 г.
4. «Динамика развития методов контроля гликемии от инвазивных к неинвазивным. Актуальные перспективы» - Бабенко А.Ю., Кононова Ю.А., Циберкин А.И., Ходзицкий М.К., Гринева Е.Н - Эндокринологический научный центр (Москва), 2016 г
5. «Элементы теории активного восприятия изображений» - Утробин В. А. – Труды НГТУ им Алексеева, 2010 г.
6. «R в действии. Анализ и визуализация данных на языке R.» - Кабаков Р. – «ДМК Пресс», Москва, 2013 г.
7. «Статистический анализ и визуализация данных с помощью R» - Мастицкий С.Э., Шитиков В.К. – «ДМК Пресс», Москва, 2015 г.

Приложение А

```
getresults <- function(X, Y, mparams)
{
 library(ega)
 library(extraTrees)

 z <- as.numeric(names(table(Y)))

bi <- c()
for (i in 1:length(z))
{
  ind <- which(Y == z[i])
  if (length(ind) == 1)
  {
    bi <- c(bi, ind)
  }
  else
  {
    bi <- c(bi, sample(ind, 1))
  }
}

 Y1 <- Y[bi]
 X1 <- X[bi,]

 Y2 <- Y[-bi]
 X2 <- X[-bi,]

 mparams <- mparams[1:3]

 errarr <- matrix(0, 18, 1)

 mypred <- c()
 myreal <- c()

 train_T <- X1
 test_T  <- X2

 train_L <- Y1
 test_L  <- Y2

 mmodel <- extraTrees(train_T, train_L, nodesize=mparams[1], mtry=mparams[2], ntree = 900,
          numRandomCuts = mparams[3], numThreads=5)
 pred <- predict(mmodel, test_T)

 mypred <- c(mypred, pred)
 myreal <- c(myreal, test_L)
 test_L1 = 18*myreal
 pred1 = 18*mypred

 zones <- getClarkeZones(test_L1, pred1)
```

```
  cc <- table(zones)/length(zones)*100
  err <- sum(abs(mypred - myreal)) / length(myreal)
  sdpred <- sd(mypred)
  sdtest <- sd(myreal)

  errarr[1] <- err
  errarr[2] <- sdpred
  errarr[3] <- sdtest
  errarr[4:8] <- cc[1:5]
  errarr[9] <- cor(mypred, myreal, method = "spearman")

  ####################
  pred    <- predict(mmodel, train_T)
  mypred  <- pred
  myreal  <- train_L
  test_L1 <- 18*myreal
  pred1   <- 18*mypred
  zones   <- getClarkeZones(test_L1, pred1)
  cc      <- table(zones)/length(zones)*100
  err     <- sum(abs(mypred - myreal)) / length(myreal)
  sdpred  <- sd(mypred)
  sdtest  <- sd(myreal)
  errarr[10]    <- err
  errarr[11]    <- sdpred
  errarr[12]    <- sdtest
  errarr[13:17] <- cc[1:5]
  errarr[18]    <- cor(mypred, myreal, method = "spearman")

  print(paste('ОШИБ =', round(errarr[1], 3), 'РАЗН =', round(errarr[3]-errarr[2], 3), 'СПИР =',
            round(errarr[9], 3), 'A =', round(errarr[4], 3), 'B =', round(errarr[5], 3), 'C =',
            round(errarr[6], 3), 'D =', round(errarr[7], 3), 'E =', round(errarr[8], 3)))

  writeLines(c(paste(round(errarr[1], 3), '\t', round(errarr[3]-errarr[2], 3), '\t', round(errarr[9], 3),
            '\t', round(errarr[4], 3), '\t', round(errarr[5], 3), '\t', round(errarr[6], 3), '\t',
            round(errarr[7], 3), '\t', round(errarr[8], 3))), fileConn, sep = "\n")

  return(list(errarr, bi))
}

# extraTrees

library(foreach)
library(doParallel)

writeLines(c('Номер', 'Файл', 'nodesize', 'mtry', 'numRandomCuts', 'Ошибка', 'Разность',
            'Спирмен', 'A','B', 'C', 'D', 'E'), fileConn, sep = "\t")
writeLines('\n', fileConn)

mtry_ <- seq(5, 75, 5)
numRandomCuts_ <- seq(10, 45, 5)
nodesize_ <- c(1, 2)
```

```
mparams  <- expand.grid(nodesize_, mtry_, numRandomCuts_)

myr <- mparams

#####

mtype <- c('_1003_104_filt_ids.rdata') # filt, nofilt +

siglen <- c('1e+05', '125000', '150000') # +

slen <- seq(from = 128, to = 1152, by = 256)

slen_ <- c()
shft <- c()

for (i in slen)
{
  for (j in 1:4)
  {
    slen_ <- c(slen_, i)
    shft <- c(shft, (i*j/4))
  }
}
slen <- slen_

sl <- c()
for (i in 1:length(slen))
{
  sl <- c(sl, paste(shft[i], slen[i], sep = '_'))
}

featn <- c('cls1d', 'oper2d', 'oper3d', 'poper2d', 'poper2di')
ddd <- expand.grid(siglen, sl, featn, 0, mtype)

featn_ <- c('full1d')
fgn    <- c('fmul', 'fsum')
ddd2 <- expand.grid(siglen, sl, featn_, fgn, mtype)

featn_ <- c('full2d')
fgn    <- c('fmul', 'fsum')
ddd3 <- expand.grid(siglen, sl, featn_, fgn, mtype)

featn_ <- c('pfull2d')
fgn    <- c('fmul', 'fsum')
ddd4 <- expand.grid(siglen, sl, featn_, fgn, mtype)

fnames <- rbind(ddd, ddd2, ddd3, ddd4)

#############

reszz <- 0
pref <- 'E:/1003/'
```

```
z <- gsub(':', '_', as.character(Sys.time()))
z <- gsub('-', '_', z)
z <- gsub(' ', '_', z)

reszz <- foreach(i = 1:660, .packages = 'foreach') %do%
{
 if (fnames[i, 4] == 0)
 {
  fn1 <- paste(pref, paste(fnames[i, 5], fnames[i, 3], fnames[i, 2], fnames[i, 1], '.rdata', sep =
'_'),
       sep = '')
 } else {

  if (fnames[i, 4] == 'fsum')
  {
   fn1 <- paste(pref, paste(fnames[i, 5], fnames[i, 3], fnames[i, 2], fnames[i, 1], 'fsum.rdata',
sep
       = '_'), sep = '')
  }

  if (fnames[i, 4] == 'fmul')
  {
   fn1 <- paste(pref, paste(fnames[i, 5], fnames[i, 3], fnames[i, 2], fnames[i, 1], 'fmul.rdata',
sep
       = '_'), sep = '')
  }
 }
 print(Sys.time())
 print(i)
 print(fn1)
 load(fn1)

 zmy <- foreach(i1 = 1:dim(myr)[1]) %do%
 {
  writeLines(c(i, fn1), fileConn, sep = "\t")
  writeLines(c(deparse(myr[i1, 1]), deparse(myr[i1, 2]), deparse(myr[i1, 3])), fileConn, sep =
       "\t")
  print(paste('nodesize =', myr[i1, 1], 'mtry =', myr[i1, 2], 'numRandomCuts =', myr[i1, 3]))
  list(getresults(X, Y, myr[i1, ]),  myr[i1, 1], myr[i1, 2], myr[i1, 3], myr[i1, 4],  fn1, fnames[i,])
 }
}
save(reszz, file = paste(z, mtype[1], mtype[2], 'one_signal_et.rdata', sep = ''))
```