|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***

***НА ТЕМУ:***

**Модуль обработки спектроскопических данных квантового вычислителя**

Студент \_\_ИУ5-85Б\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_А.А.Акулова\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель ВКР **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Ю.Е.Гапанюк\_\_**

(Подпись, дата) (И.О.Фамилия)

Нормоконтролер **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Ю.Н.Кротов \_\_**

(Подпись, дата) (И.О.Фамилия)

2024 г.

# **АННОТАЦИЯ**

Расчетно-пояснительная записка квалификационной работы бакалавра содержит 50 страниц. Учитывая приложения, объем составляет 69 страницу В процессе работы было использовано 17 источников, применено 25 изображений и 8 таблиц.

Объект разработки представляет из себя приложения для проведения кластерного анализа данных.

Целью дипломной работы является разработка приложения “Модуль обработки спектроскопических данных квантового вычислителя” для нахождения наиболее толстой верхней параболы, которая идет от края до края, и визуализации данных.

Пояснительная записка содержит 3 приложения.

# **СОДЕРЖАНИЕ**

[АННОТАЦИЯ 2](#_Toc169450066)

[СОДЕРЖАНИЕ 3](#_Toc169450067)

[ВВЕДЕНИЕ 4](#_Toc169450068)

[1. ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ 5](#_Toc169450069)

[1.1. Постановка задачи проектирования 5](#_Toc169450070)

[1.2 Описание предметной области 6](#_Toc169450071)

[1. 3 Анализ аналогов 7](#_Toc169450072)

[1.4 Выбор критериев качества 7](#_Toc169450073)

[1.5 Выбор программной платформы 9](#_Toc169450074)

[1.6 Технологии 12](#_Toc169450075)

[2 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 16](#_Toc169450076)

[2.1 Проектирование 16](#_Toc169450077)

[2.1.1 Перечень задач, подлежащих реализации 16](#_Toc169450078)

[2.1.2 Алгоритм работы программы 16](#_Toc169450079)

[2.1.3 Функциональная модель 18](#_Toc169450080)

[2.1.4 Граф диалога 28](#_Toc169450081)

[2.1.5 Программная реализация алгоритма 30](#_Toc169450082)

[2.1.6 Функции приложения 33](#_Toc169450083)

[3. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ 42](#_Toc169450084)

[3.1. Проблемы, решенные в процессе разработки 42](#_Toc169450085)

[3.1.1. Проблема фильтрации данных 42](#_Toc169450086)

[3.1.2. Проблема подбора алгоритма кластеризации 44](#_Toc169450087)

[ЗАКЛЮЧЕНИЕ 48](#_Toc169450088)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 49](#_Toc169450089)

[ПРИЛОЖЕНИЕ А ГРАФИЧЕСКАЯ ЧАСТЬ 51](#_Toc169450090)

[ПРИЛОЖЕНИЕ Б ТЕХНИЧЕСКОЕ ЗАДАНИЕ 56](#_Toc169450091)

[ПРИЛОЖЕНИЕ В ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ 62](#_Toc169450092)

# **ВВЕДЕНИЕ**

Интерес общественности к так называемому «квантовому компьютеру» необычно возрос в последнее время. Идея воспользоваться возможностями квантовой механики не нова, она привлекала людей и раньше. Однако только сейчас этой темой занялись усиленно, только сейчас начинают проводить серьезные эксперименты в этой области.

В университете МГТУ им. Н.Э. Баумана есть квантовые компьютеры, которые на данный момент не всегда запускаются. Однако было проведено много исследований, из которых были получены данные кубитов. Можно было заметить, что квантовый компьютер запускается при хороших кубитах, которые образуют выброс в форме параболы.

По этой причине возникает необходимость в разработке приложения, которое способно автоматически находить выбросы в форме параболы, отсеивать лишние данные, а также визуализировать их.

# **ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ**

## **Постановка задачи проектирования**

Для того, чтобы реализовать модуль обработки спектроскопических данных квантового вычислителя, нужен хороший алгоритм для сортировки данных. Предлагаемое решение должно выполнять конкретную задачу – кластеризацию данных чтобы обеспечить практичность и широту его применения [1] [2].

У пользователя должна быть возможность обрабатывать большие и маленькие объемы данных.

Пользователь также должен иметь возможность изменять названия осей, должен отображаться список, сгенерированный на основе загруженного датасета, с возможностью выбора полей оси X, оси Y, колонки данных.

Также необходимо визуализировать введенные данные, чтобы пользователь имел возможность определить наличие или отсутствие параболы, а также дать пользователю возможность понять, с какой именно стороны она находится, чтобы выбрать верхнюю и нижнюю процентиль данных, потому что в дальнейшем будет применяться алгоритм кластеризации отдельно по выбранной пользователем верхней и нижней границе.

Таким образом, разрабатываемый проект предназначен для выполнения следующих задач:

1. Обрабатывать любые наборы данных, как большие, так и маленькие;
2. Находить параболы и отрисовывать 3D графики;
3. Позволять пользователю самому проанализировать необходимую информацию как визуально, так и при помощи сформированного отчета.

## **1.2 Описание предметной области**

Предметной областью данной информационной системы являются данные кубитов квантового компьютера, средства их обработки, а также пользователи, анализирующие выгруженный отчет.

Для реализации системы требуется хороший алгоритм кластеризации, который способен корректно обработать данные. Кластерный анализ или же просто кластеризация – это разделение большой группы объектов на несколько маленьких, где каждая новая группа и является кластером. Каждая новая группа создается по определенному критерию, который задает разработчик. То есть в качестве критерия может выступать любая особенность объекта: вид, форма, размер, категория и т.д. Внутри одного кластера объекты могут как-либо отличаться по разным критериям, однако хотя бы по одному они должны совпадать. Исследователь получает данные кубитов из квантового компьютера, данные преобразуются в датасет. Пользователь загружает полученный от специалиста датасет в модуль обработки спектроскопических данных квантового вычислителя. Алгоритм кластеризации занимается непосредственно обработкой полученных данных, выявление парабол, а также их визуализацией. Отчет представляет из себя информацию о найденной параболе, а также о ее координатах.

Анализом и изучением данных занимается пользователь, его целями являются: убедиться, что была найдена самая толстая верхняя парабола и проанализировать полученный результат. Подробная схема представлена на рисунке А.1

Схема предметной области представлена на рисунке 1.

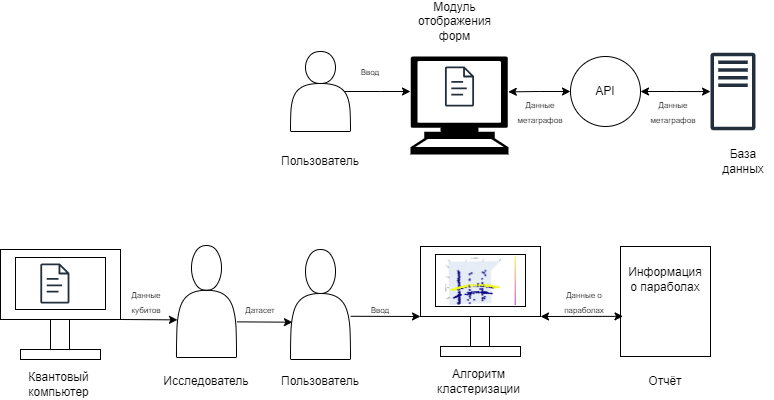


Рисунок – Предметная область приложения

## **1. 3 Анализ аналогов**

Проанализировав поставленную задачу, была предпринята попытка найти аналоги приложения, чтобы проанализировать уже имеющиеся источники и платформы и понять, на сколько необходимо придумывать собственный алгоритм. Однако нигде не было найдено даже отдаленно похожего решения для, которое бы рассматривало эту область, поэтому было решено писать программу и придумывать свой собственный алгоритм кластеризации.

Назначением программы является отыскивать оптимальные параметры запуска, то есть выделять параболы на загруженных датасетатах, а также визуализировать данные. Программа предоставляет необходимый для этих задач интерфейс, внутри которого происходит визуализация и вывод результатов.

## **1.4 Выбор критериев качества**

Для того, чтобы выбрать язык программирования для разрабатываемого модуля обработки спектроскопических данных квантового вычислителя были выбраны критерии качества. Критерии были выбраны на основе требований, предъявляемых к программному продукту. Также была проведена оценка важности критериев [6].

Разобьем по значимости критерии, они находятся в диапазоне от 1 до 3 баллов, где 3 – особо важный критерий, 2 – важный критерий, 1 – самый меньший по важности критерий.

Выбранные критерии представляют собой:

1. Легкость в нахождении ошибок;
2. Наличие библиотек для анализа данных и визуализации;
3. Количество документации на русском языке;
4. Простота изучения;
5. Универсальность – кроссплатформенность.

Оценки важности выбранных критериев качества (таблица 1):

Таблица 1 − Критерии качества и оценки их важности

| **Критерий Кi** | **Расшифровка критерия** | **Значимость (bi)** |
| --- | --- | --- |
| К1 | Легкость в нахождении ошибок | 2 |
| К2 | Наличие библиотек для анализа данных и визуализации | 3 |
| К3 | Количество документации на русском языке | 1 |
| К4 | Простота изучения | 2 |
| К5 | Универсальность | 3 |

Критериям были присвоены весовые коэффициенты путем нормализации бальных оценок, полученных ими, по следующей формуле (1):

где αi – весовой коэффициент критерия;

bi – значимость критерия.

Результаты расчета весовых коэффициентов критериев (таблица 2):

Таблица 2 − Результаты расчета весовых коэффициентов критериев

| **Критерий (Кi)** | **Значимость (bi)** | **Вес(αi)** |
| --- | --- | --- |
| К1 | 2 | 0.18 |
| К2 | 3 | 0.27 |
| К3 | 1 | 0.09 |
| К4 | 2 | 0.18 |
| К5 | 3 | 0.27 |

## **1.5 Выбор программной платформы**

В первую очередь для разработки алгоритма необходимо выбрать правильный язык программирования, который способен полностью реализовать весь функционал поставленного задания. Для это были найдены варианты, а также был произведет сравнительный анализ для выявления наиболее подходящего языка программирования. Для сравнения был выбран метод взвешенной суммы [7].

Для оценки были выбраны следующие языки программирования:

1. Язык R(В1) [3]
2. JavaScript(В2) [4]
3. Python(В3) [5]

Шкала бальных оценок по каждому из вариантов программирования находится в диапазоне от 1 до 5 баллов, где 5 – наивысший балл, 1 – самый маленький балл.

Оценки заносим (таблица 3):

Таблица 3 − Оценки критериев выбора языка программирования

| **№** | **Критерии** | **Варианты** | | |
| --- | --- | --- | --- | --- |
| **B1** | **B2** | **B3** |
| 1 | К1 | 4 | 5 | 5 |
| 2 | К2 | 5 | 3 | 5 |
| 3 | К3 | 3 | 5 | 4 |
| 4 | К4 | 3 | 5 | 5 |
| 5 | К5 | 3 | 3 | 5 |

Далее была проведена относительная нормализация оценок по формуле 2:

где – нормализованное значение оценки j-го варианта по i-му критерию;

– оценки j-го варианта по i-му критерию.

Результаты нормализации представляют собой (таблица 4):

Таблица 4 − Результаты нормализации оценок

| **№** | **Критерии** | **Варианты** | | |
| --- | --- | --- | --- | --- |
| **B1** | **B2** | **B3** |
| 1 | К1 | 0,8 | 1 | 1 |
| 2 | К2 | 1 | 0,6 | 1 |
| 3 | К3 | 0,6 | 1 | 0,8 |
| 4 | К4 | 0,6 | 1 | 1 |
| 5 | К5 | 0,6 | 0,6 | 1 |

При использовании метода взвешенной суммы лучший вариант выбирается согласно формулам 3 и 4:

Результаты расчета взвешенных сумм представляют собой (таблица 5):

Таблица 5 − Результаты расчета взвешенных сумм

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерий (Ki)** | **Вес (αi)** | **Вариант (Вj)** | | |
| **В1** | **В2** | **В3** |
| К1 | 0.18 | 0,8 | 1 | 1 |
| К2 | 0.27 | 1 | 0,6 | 1 |
| К3 | 0.09 | 0,6 | 1 | 0,8 |
| К4 | 0.18 | 0,6 | 1 | 1 |
| К5 | 0.27 | 0,6 | 0,6 | 1 |
| Yj | | 0.74 | 0.77 | 0.97 |

По методу взвешенной суммы лучшим из представленных вариантов оказалась Python так как Yj оказалось максимально. Следовательно, именно на этом языке будет реализовано приложение.

## **1.6 Технологии**

На схеме ниже представлены все используемые в приложении языки программирования и библиотеки (рисунок 2):

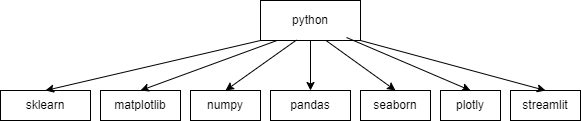
****

Рисунок 2 – Схема основных технологий, применяемых в проекте

Ниже приведено детальное описание всех используемых технологий:

1. sklearn:

Ключевые концепции и функции этой библиотеки включают:

* + Классификацию: относит объекты к определенным классам или же категориям,
  + Регрессия: прогнозирование значений данных на основе среднего значения существующих и имеющихся данных,
  + Кластеризация: автоматическая группировка похожих данных в наборы данных [10].

1. Matlplotlib - используют для визуализации данных любой сложности. Библиотека позволяет строить разные варианты графиков:

* Линейные,
* Диаграммы рассеивания,
* Гистограммы,
* А также комбинировать все графики [9].

1. Numpy - библиотека для работы с многомерными массивами, над которыми можно быстро и эффективно выполнять множество математических, статистических, логических и других операций [11]. Ключевые преимущества:

* Многомерные массивы: NumPy позволяет создавать массивы с любым числом измерений. Это удобно если работаешь с данными, в которых больше одной переменной,
* Быстрые операции: NumPy предоставляет высоко оптимизированные операции над массивами, что делает его значительно быстрее, чем обычные списки в Python,
* Богатый выбор функций: у библиотеки есть много функций для математики: линейной алгебры, статистики, фильтрации данных и других вычислений,
* Интеграция с другими библиотеками: NumPy хорошо интегрируется с другими библиотеками для научных вычислений, это сильно упрощает жизнь, если необходимо создавать сложные вычисления так как используются разнообразные инструменты.

1. Pandas - это библиотека для анализа данных и работы с табличными данными. Она особенно удобна для таких действий, как фильтрация и группировка данных [12].

Основные задачи, для которых используют библиотеку.

* Аналитика данных: для работы с данными требуется анализ и подготовка: необходимо удалить или заполнить пропуски, отфильтровать, отсортировать или каким-то образом изменить данные. Pandas в Python позволяет быстро выполнить все эти действия, а также помогает ещё и автоматизировать их,
* Статистика. Библиотека хороша для статистических методов, которые нужны для работы с данными. Например, расчёт средних значений, их распределение по процентилям и другие.

1. Seaborn - это библиотека для создания статистических графиков на Python. Она построена поверх [matplotlib](https://matplotlib.org/) и тесно интегрирована со структурами данных [pandas](https://pandas.pydata.org/) [13].

Преимущества Seaborn:

* Функция для построения статистических данных временных рядов с гибкой оценкой и представлением неопределенности оценки.,
* Функции для визуализации одномерных и двумерных распределений или для сравнения их между подмножествами данных,
* Функции, которые визуализируют матрицы данных и используют алгоритмы кластеризации для обнаружения структуры в этих матрицах,
* Абстракции высокого уровня для структурирования сеток графиков, которые позволяют легко создавать сложные визуализации,
* Несколько встроенных тем для стилизации графики matplotlib.

1. Plotly − это интерактивная библиотека визуализации. Она позволяет строить трехмерные графики, а также проводить разные манипуляции над ними [14].

Преимущества Plotly:

* Интерактивность. Plotly позволяет создавать графики с возможностью интерактивного взаимодействия, такого как увеличение, выбор данных по клику и прокрутка,
* Широкие возможности. Plotly может помочь построить разные типы графиков и диаграмм для визуализации различных типов данных,
* Хорошую документацию. У Plotly хорошая и структурированная, а также подробная документация,
* Поддержку различных платформ. Plotly может использоваться не только в Jupyter Notebook, но и в веб-приложениях, что делает его очень удобным для разных возможных сценариев пользователей.

1. Streamlit - это [фреймворк](https://blog.skillfactory.ru/glossary/framework/) для языка программирования [Python](https://blog.skillfactory.ru/glossary/python/" \t "_blank). Он содержит набор программных инструментов, которые помогают перенести модель машинного обучения в веб [15].

Преимущества Streamlit:

* Быстрое развертывание. ML-модель или обычную программу можно быстро превратить в веб-приложение и управлять им.
* Использование скриптов.  Когда пользователь взаимодействует с получившимся приложением или разработчик решает заменить что-то в коде, Streamlit сам обновляет нужные части страницы.
* Виджеты и визуализация. В Streamlit есть встроенные виджеты для частого использования, к примеру ползунки или поля для ввода текста. Можно взять уже готовые виджеты и собрать из них работающий интерфейс. Также можно визуализировать график или картинку, вывести результат работы программы в виде схемы или таблицы.

# **2 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ**

## **2.1 Проектирование**

### **Перечень задач, подлежащих реализации**

В данной работе проводятся исследования в области кластерного анализа данных. Для того, чтобы реализовать модуль обработки спектроскопических данных квантового вычислителя, нужен хороший алгоритм для сортировки данных.

Предлагаемое решение должно выполнять конкретную задачу – кластеризацию данных чтобы обеспечить практичность и широту его применения, а именно:

1. Использовать собственный алгоритм кластеризации;
2. Визуализировать данные;
3. Иметь простой ввод в эксплуатацию;
4. Обрабатывать большие и маленькие наборы данных;
5. Программа должна поддерживать интеграцию в другой программный код для поддержки дальнейшего использования в целях анализа данных и машинного обучения.

### **2.1.2 Алгоритм работы программы**

Эксперименты проводились с разными наборами данных, которые были по-своему уникальны. В некоторых датасетах парабола была удачная, проходящая от края до края, в других же она была прерывистая, где-то выброс в виде полоски справа, где-то слева или какое-то среднее значение точек параболы по оси Z было выше, а где-то ниже среднего значения по датасету. Все эти особенности были учтены, чтобы в конечном итоге пользователи вводили необходимые данные и получали нужный график.

В качестве исходных данных поступает датасет, соответствующий изображению кубита. Три колонки X, Y, Z, при этом мы считаем, что задана функция двух переменных . *X* – напряжение, Y – частота, Z – магнитуда. Необходимо найти наиболее толстую верхнюю параболу, которая идет от края до края изображения, и визуализировать данные.

Алгоритм работы программы:

1. Выбираются процентили, по которым будет производиться фильтрация данных. Поскольку парабола представляет из себя выброс по оси Z, разумно ограничить пространство поиска набором наибольших и наименьших значений. Экспериментально было выявлено, что, как правило, парабола находится либо в верхних 95% значений, либо в нижних 5% значений. Соответственно отфильтровываются значения ниже 5-й и выше 95-й процентилей;
2. Производится масштабирование данных и удаление пропусков. Значения по осям X и Y различаются на порядки, что усложняет последующую работу с данными, если их не масштабировать;
3. Параллельно на полученных отфильтрованных и масштабированных данных производятся 2 вида кластеризации:

* Кластеризация DBSCAN по осям X и Y для создания скоплений кластеров, близких по значению Y. Это нужно, чтобы по большому скоплению определять выброс в форме вертикальной "полоски", который часто встречается на краю данных;
* Кластеризация вдоль оси Z. По сути, мы делим данные на 2 равных части по значению Z, то есть делим пополам по 50-й процентили. Это делается для того, чтобы выброс в форме "полоски" впоследствии не повлиял на нахождение точек, принадлежащих параболе, так как выбросы-полоски, как правило, не такие сильные, как выбросы-параболы;

1. Результаты двух кластеризаций объединяются, то есть из пары уникальных меток мы получаем одну;
2. Для каждого получившегося кластера находится по одной характеристической точке. Для нижней процентили это минимум, для верхней - максимум;
3. По характеристическим точкам находим параболы:

* Начинаем отсчет с противоположной стороны от "полоски", то есть от самого большого скопления кластеров;
* Каждая характеристическая точка для выбранного крайнего значения X считается началом кривой;
* Для каждого начала кривой выбирается ближайшая к ней точка со следующим значением X;
* Если для кривой выясняется, что ближайшая точка ближе к другой кривой, то данная кривая отбрасывается;
* Если ближайшая к кривой точка лежит к ней под углом больше 30 градусов, это означает, что кривая недостаточно гладкая, и кривая отбрасывается;
* Если на следующем X не обнаруживается характеристических точек, то отбрасываются все кривые;

### **2.1.3 Функциональная модель**

В данном разделе описывается функциональная модель первый уровень [8]. Модуль обработки спектроскопических данных квантового вычислителя рассматривается с точки зрения пользователя. Использование данной системы позволит найти оптимальные параметры запуска квантового компьютера.

Пользователь загружает свои данные. Далее алгоритм кластеризации предварительно обрабатывает данные и находит нужную параболу. Затем он анализирует получившиеся результаты. В конце алгоритм кластеризации выводит пользователю результаты.

Алгоритм кластеризации можно разделить на несколько важных этапов: загрузка данных, предварительная обработка данных, нахождение параболы, анализ данных, вывод пользовательских результатов. Такая последовательность этапов позволяет составить полноценную качественно выполненную работу алгоритма кластеризации, которая полностью удовлетворяет техническому заданию.

Модуль обработки спектроскопических данных квантового вычислителя помогает пользователям в определении параболы в их наборе данных.

Функции создания модуля обработки спектроскопических данных квантового вычислителя приведены в таблице 6.

Цель: описать процесс создания модуля обработки спектроскопических данных квантового вычислителя.

Точка зрения: пользователь.

Таблица 6 − Описание основных функций

| **Функции** | **Описание** | **Данные, требуемые для выполнения функций** |
| --- | --- | --- |
| 1. Загрузить данные | Пользователь загружает свой набор данных | Входные данные:  Библиотеки  Данные  Управление:  Правила пользования модулем  Механизмы:  Пользователь  Алгоритм кластеризации |
| 1. Предварительно обработать данные | Алгоритм фильтрует и масштабирует данные. После идет 2 кластеризации: методом DBSCAN и по Z. Затем идет объединение двух кластеризаций | Входные данные:  Загруженные данные  Классы и методы библиотек  Управление:  ТЗ  Правила пользования модулем  Механизмы:  Алгоритм кластеризации  Пользователь |
| 1. Найти параболу | Определяются характеристические точки, по ним находят параболу, отмечают ее, убирая лишние кластеры - параболы | Входные данные:  Предварительно обработанные данные  Управление:  ТЗ  Правила пользования  Механизмы:  Алгоритм кластеризации  Пользователь |
| 1. Проанализировать данные | Строятся и выводятся графики по параболам. Выводится отчет о количестве найденных парабол. | Входные данные:  Полностью обработанные данные  Классы и методы библиотек  Управление:  ТЗ  Правила пользования  Механизмы:  Алгоритм кластеризации  Пользователь |
| 1. Вывести пользовательские результаты | Выводятся для пользователя найденные параболы и отчет об их количествах. | Входные данные:  Проведенный анализ данных  Полностью обработанные данные  Данные о параболах  Набор парабол  Управление:  Правила пользования  Механизмы:  Пользователь  Алгоритм кластеризации |

Для изучения предметной области была использована методология SADT. Функциональная модель SADT отображает функциональную структуру объекта, т. е. производимые им действия и связи между этими действиями. Построение модели начинается с контекстной диаграммы, которая представляет всю систему в виде простейшей компоненты – одного блока «Модуль обработки спектроскопических данных квантового вычислителя» и дуг, изображающих все основные связи моделируемой системы с внешним миром. Диаграмма декомпозиции, полученная в результате разбиения контекстной диаграммы на отдельные активности, выявляет полный набор подфункций, каждая из которых представлена как блок, границы которого определены интерфейсными дугами.

Построение модели начинается с контекстной диаграммы, которая представляет всю систему в виде простейшей компоненты - одного блока «Модуль обработки спектроскопических данных квантового вычислителя» и дуг, изображающих все основные связи моделируемой системы с внешним миром (рисунок 3).

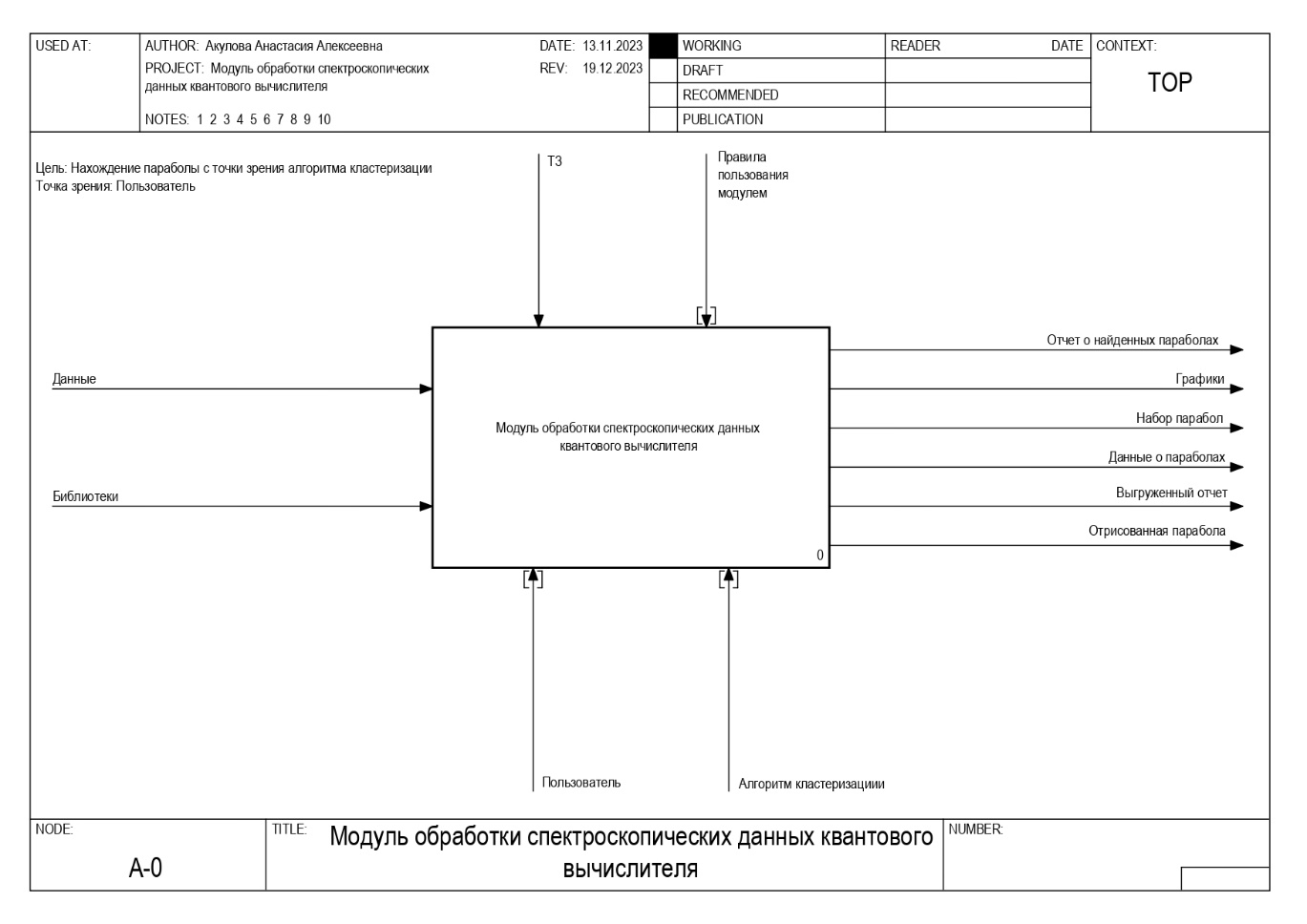


Рисунок 3 – Контекстная диаграмма модуля обработки спектроскопических данных квантового вычислителя.

Функциональная модель второй уровень.

Второй уровень модели – диаграмма декомпозиции, полученная в результате разбиения контекстной диаграммы. Диаграмма декомпозиции выявляет полный набор подфункций, каждая из которых представлена как блок, границы которого определены интерфейсными дугами. В ходе декомпозиции получено 5 блоков (рисунок 4): «Загрузить данные», «Предварительно обработать данные», «Найти параболу», «Проанализировать данные», «Вывести пользовательские результаты».

Изначально в проект вводятся данные, поступившие от пользователя, и библиотеки. После чего алгоритм кластеризации загружает данные пользователя. Далее он предварительно обрабатывает данные и находит нужную параболу. Затем он анализирует получившиеся результаты. В конце алгоритм кластеризации выводит пользовательские результаты.

Второй уровень декомпозиции диаграммы представлен на рисунке 4.

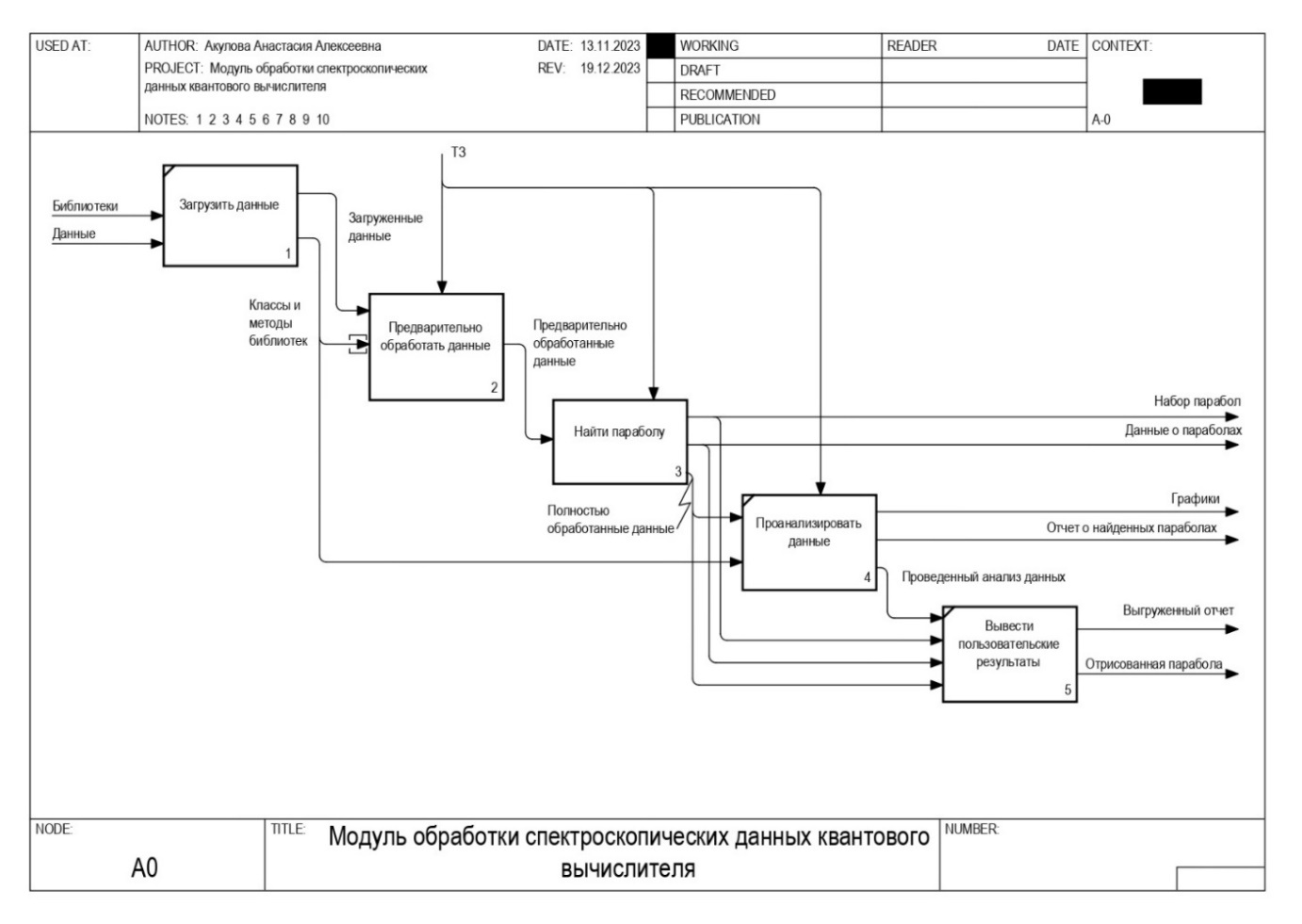


Рисунок 4 – Диаграмма декомпозиции модуля обработки спектроскопических данных квантового вычислителя

Функциональная модель третий уровень.

Третий уровень модели – диаграмма декомпозиции блока «Предварительно обработать данные», а также диаграмма декомпозиции блока «Найти параболу».

Диаграмма декомпозиции блока «Предварительно обработать данные» (рисунок 5) разбита на пять блоков:

1. Отфильтровать данные – алгоритм кластеризации фильтрует пользовательские данные.
2. Отмасштабировать данные – алгоритм кластеризации масштабирует пользовательские данные.
3. Кластеризовать по DBSCAN – алгоритм кластеризации кластеризирует пользовательские данные методом DBSCAN.
4. Кластеризовать по z – алгоритм кластеризации кластеризирует пользовательские данные по z.
5. Объединить кластеризации – алгоритм кластеризации объединяет кластеризации методом DBSCAN и по z.

Диаграмма декомпозиции блока «Найти параболу» (рисунок 6) разбита на четыре блока:

1. Найти характеристические точки – алгоритм кластеризации находит характеристические точки.
2. Найти параболу по характеристическим точкам – алгоритм кластеризации находит параболу по характеристическим точкам.
3. Отметить параболу на данных – алгоритм кластеризации отмечает параболу на данных.
4. Убрать лишние кластеры-параболы – алгоритм кластеризации убирает лишние кластеры-параболы, которые были выявлены в ходе работы.

Третий уровень декомпозиции диаграммы представлен на рисунках 3, 4.

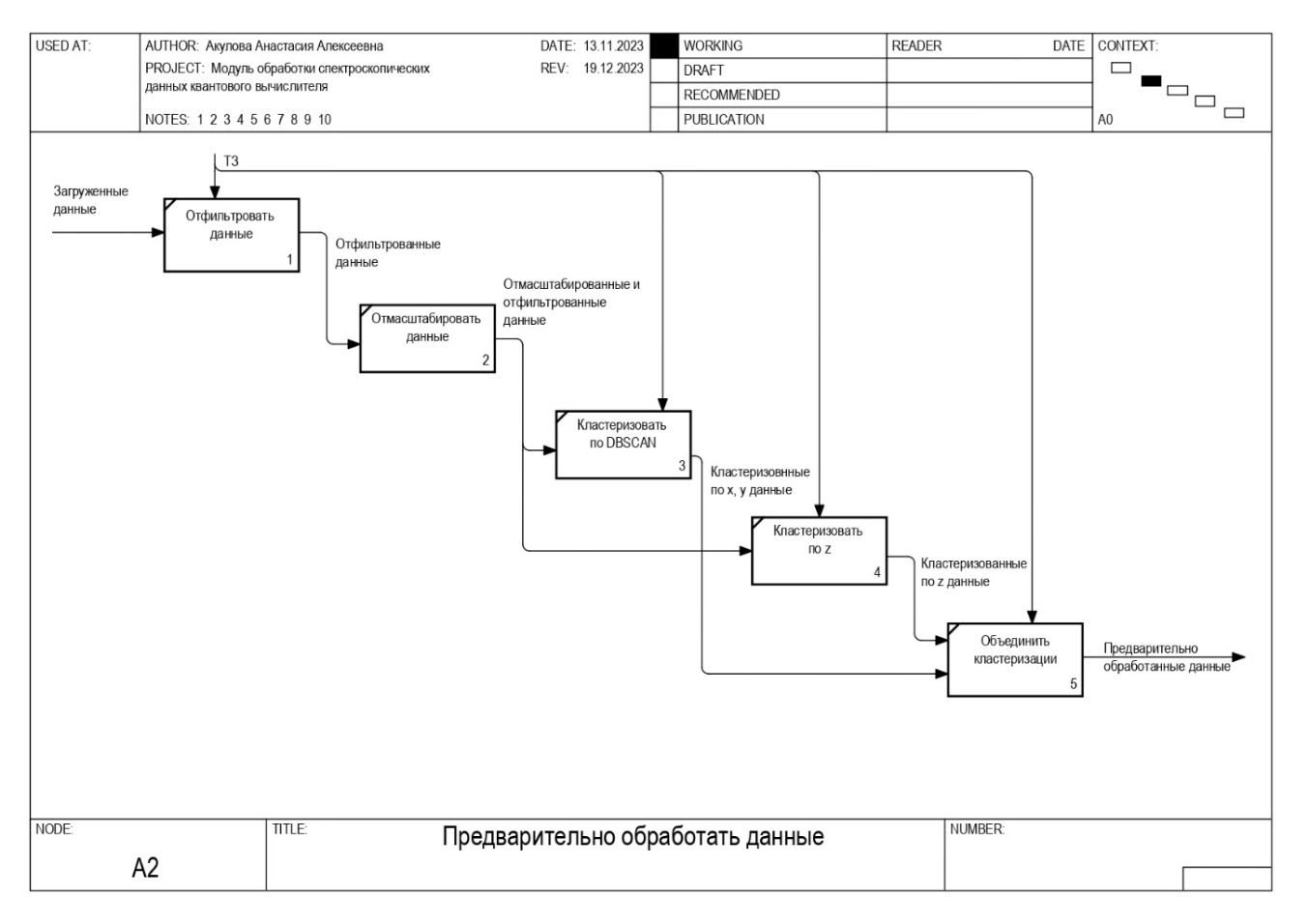


Рисунок 5 – Диаграмма функции «Предварительно обработать данные»

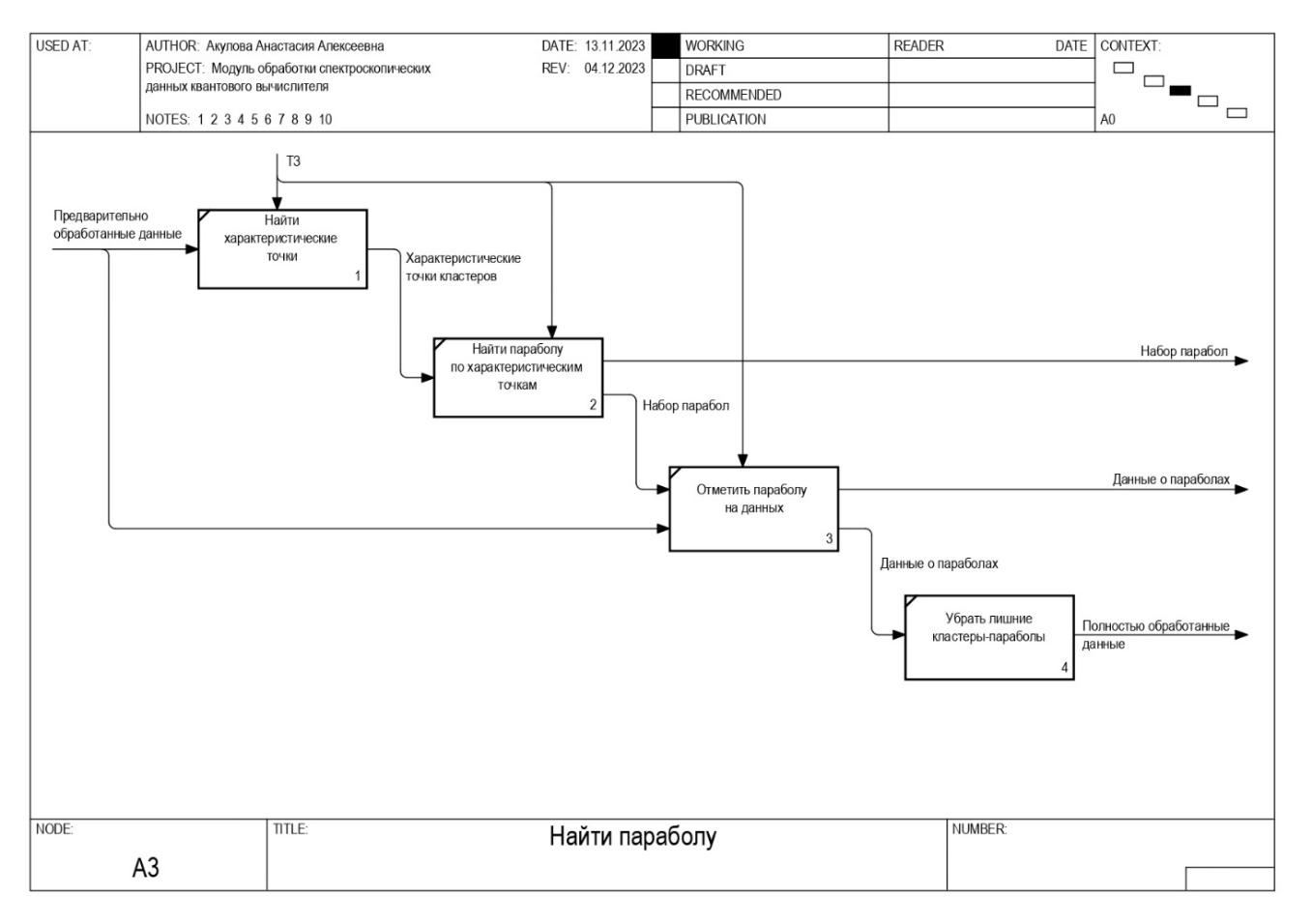


Рисунок 6 – Диаграмма функции «Найти параболу»

Функциональные блоки.

В таблицах 7 и 8 проводится подробное описание всех функций и стрелок представленной модели.

Таблица 7 – Отчёт по функциональным блокам модели

| **Номер блока** | **Название блока** | **Описание функционального блока** |
| --- | --- | --- |
| 0 | Модуль обработки спектроскопических данных квантового вычислителя | Создание модуля обработки спектроскопических данных квантового вычислителя |
| 1 | Загрузить данные | Алгоритм кластеризации получает набор данных от пользователя и загружает его |
| 2 | Предварительно  обработать данные | Алгоритм кластеризации совершает предварительную обработку данных |
| 21 | Отфильтровать данные | Алгоритм кластеризации фильтрует пользовательские данные |
| 22 | Отмасштабировать данные | Алгоритм кластеризации масштабирует пользовательские данные |
| 23 | Кластеризовать по DBSCAN | Алгоритм кластеризации кластеризирует пользовательские данные по z |
| 24 | Кластеризовать по z | Алгоритм кластеризации кластеризирует пользовательские данные по z |
| 25 | Объединить кластеризации | Алгоритм кластеризации объединяет кластеризации методом DBSCAN (по z, y) и по z |
| 3 | Найти параболу | Алгоритм кластеризации находит параболу в пользовательском наборе данных |
| 31 | Найти характеристические точки | Алгоритм кластеризации находит характеристические точки. |
| 32 | Найти параболу по характеристическим точкам | Алгоритм кластеризации находит параболу по характеристическим точкам |
| 33 | Отметить параболу на данных | Алгоритм кластеризации отмечает параболу на данных |
| 34 | Убрать лишние кластеры-параболы | Алгоритм кластеризации убирает лишние кластеры-параболы, которые были выявлены в ходе работы |
| 4 | Проанализировать  данные | Алгоритм кластеризации, анализирует полученные данные, строит графики по параболам, выводится отчет о количестве найденных парабол. |
| 5 | Вывести пользовательские  результаты | Алгоритм кластеризации выводит для пользователя найденные параболы и отчет об их количествах. |

Таблица 8 – Отчёт по стрелкам модели

| **Название стрелки** | **Описание** | **Источник** | **Тип источника** |
| --- | --- | --- | --- |
| Алгоритм кластеризациии | Алгоритм кластеризации | { Border } | Mechanism |
| Библиотеки | Библиотеки | { Border } | Input |
| Выгруженный отчет | Выгруженный для пользователя отчет о найденных параболах | Ввести пользовательские результаты | Output |
| Графики | Графическое отображение найденных парабол | Проанализировать данные | Output |
| Данные | Пользовательский набор данных | { Border } | Input |
| Данные о параболах | Данные о параболах, найденные по характеристическим точкам | Найти параболу | Output |
| Загруженные данные | Загруженные в алгоритм кластеризации данные | Загрузить данные | Output |
| Классы и методы библиотек | Классы и методы используемых библиотек | Загрузить данные | Output |
| Кластеризованные по x,y данные | Кластеризованные алгоритмом DBSCAN данные | Кластеризовать по DBSCAN | Output |
| Кластеризованные по z данные | Кластеризованные по z данные | Кластеризовать по z | Output |
| Набор парабол | Набор парабол по характеристическим точкам | Найти параболу | Output |
| Отмасштабированные и отфильтрованные данные | Данные после фильтрации и масштабирования | Отмасштабировать данные | Output |
| Отрисованная парабола | Отрисованная для пользователя найденная парабола | Ввести пользовательские результаты | Output |
| Отфильтрованные данные | Данные после фильтрации | Отфильтровать данные | Output |
| Отчет о найденных параболах | Отчет о найденных алгоритмом кластеризации параболах | Проанализировать данные | Output |
| Полностью обработанные данные | Полностью обработанные данные | Убрать лишние кластеры | Output |
| Пользователь | Пользователь | { Border } | Mechanism |
| Правила пользования модулем | Правила пользования модулем | { Border } | Control |
| Предварительно обработанные данные | Отмасштабированные, отфильтрованные и предварительно кластеризованные данные | Объединить кластеризации | Output |
| Проведенный анализ данных | Результаты анализа обработанных данных | Проанализированные данные | Output |
| ТЗ | Техническое задание | { Border } | Control |
| Характеристические точки кластеров | Характеристические точки кластеров | Найти характеристические точки | Output |

### **2.1.4 Граф диалога**

Пользовательский интерфейс - комплекс программных и аппаратных средств, предназначенных для обеспечения информационного взаимодействия компьютера и пользователя. Основу такого взаимодействия составляют диалоги. Под диалогом понимается процесс обмена информацией между пользователем и программной системой, осуществляемый через интерактивный терминал и по определенным правилам. Каждый диалог состоит из отдельных процессов ввода-вывода, которые физически обеспечивают связь пользователя и компьютера.

Граф диалога представляет собой ориентированный взвешенный граф, каждой вершине которого сопоставлена конкретная картинка на экране или определенное состояние диалога, характеризующееся набором доступных пользователю действий. Дуги, исходящие из вершин, показывают возможные изменения состояний при выполнении пользователем указанных действий. В качестве весов дуг указывают условия переходов из состояния в состояние и операции, выполняемые во время перехода. Необходимо визуализировать граф диалога (рисунок 7).

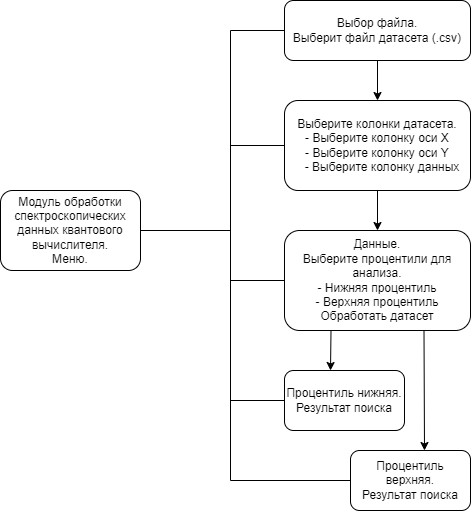


Рисунок 7 – Граф диалога

Данная информационная система предназначена для нахождения наиболее толстой верхней параболы в датасете.

Работа модели процесса начинается с экранной формы выбора датасета. Здесь пользователю предлагается выбрать датасет, который он бы хотел проверить на наличие в нем параболы. На вход может быть загружен только датасет с расширением файла csv. Иной же формат не будет подвержен обработке.

После выхода на поле выбора колонки датасета предлагается нажать на поля выбора осей. При нажатии на поле «Выберете колонку оси X», пользователь выбирает ось X. При нажатии на поле «Выберете колонку оси Y», пользователь выбирает ось Y. При нажатии на поле «Выберете колонку данных», пользователь выбирает ось Z. Выбирать оси можно в любом порядке.

Следующая экранная форма, которая высвечивается да экране, - это форма выбора процентили для анализа данных. Пользователю нажимает на выбор нижней и верхней процентили. Процентили идут в диапазоне от 1 до 100. Необходимо сначала выбрать нижнюю границу и верхнюю. При выборе нижней процентили данных, будет проводиться обработка тех данных, которые находятся ниже заданной границы. При выборе верхней процентили данных, будет проводиться обработка только тех данных, которые находятся выше заданной границы.

После выбора нижней процентили можно нажать на кнопку «Обработать датасет», и начинается предобработка и обработка данных, которые находятся ниже выбранной границы. В конце пользователь может оценить работу программы путем анализа 3D визуализации своего загруженного датасета и выгруженным отчетом.

После выбора верхней процентили можно нажать на кнопку «Обработать датасет», и начинается предобработка и обработка данных, которые находятся выше выбранной границы. В конце пользователь может оценить работу программы путем анализа 3D визуализации своего загруженного датасета и выгруженным отчетом.

### **2.1.5 Программная реализация алгоритма**

В данном разделе описана программная реализация функций приложения. Функции приложения включают в себя:

* Загрузку данных. Загрузка данных в приложение производится внутренними средствами библиотеки Streamlit. Пользователю предлагается выбрать файл CSV-таблицы с данными (рисунок 8), после чего данные извлекаются средствами библиотеки pandas, и из них удаляются пропуски (рисунок 9);

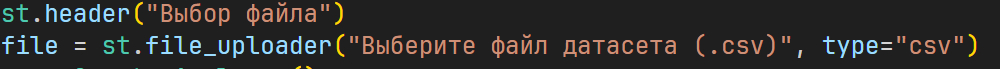


Рисунок 8 – Пример применения Streamlit для отображения формы выбора CSV-файла

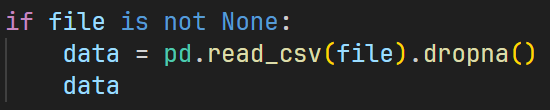


Рисунок 9 – Пример применения pandas для считывания данных из файла и удаления пропусков

* Предварительную обработку данных. После того, как пользователь ввел необходимую информацию о наборе данных, начинается их предварительная обработка, включающая в себя: фильтрацию данных, масштабирование признаков с использованием класса MinMaxScaler библиотеки scikit-learn, кластеризацию DBSCAN с собственной метрикой расстояния, кластеризацию, разделяющую данные на 2 части по оси Z и совмещение наборов меток двух кластеризаций. Фрагмент кода, предназначенный для предварительной обработки данных, представлен на рисунке 10;

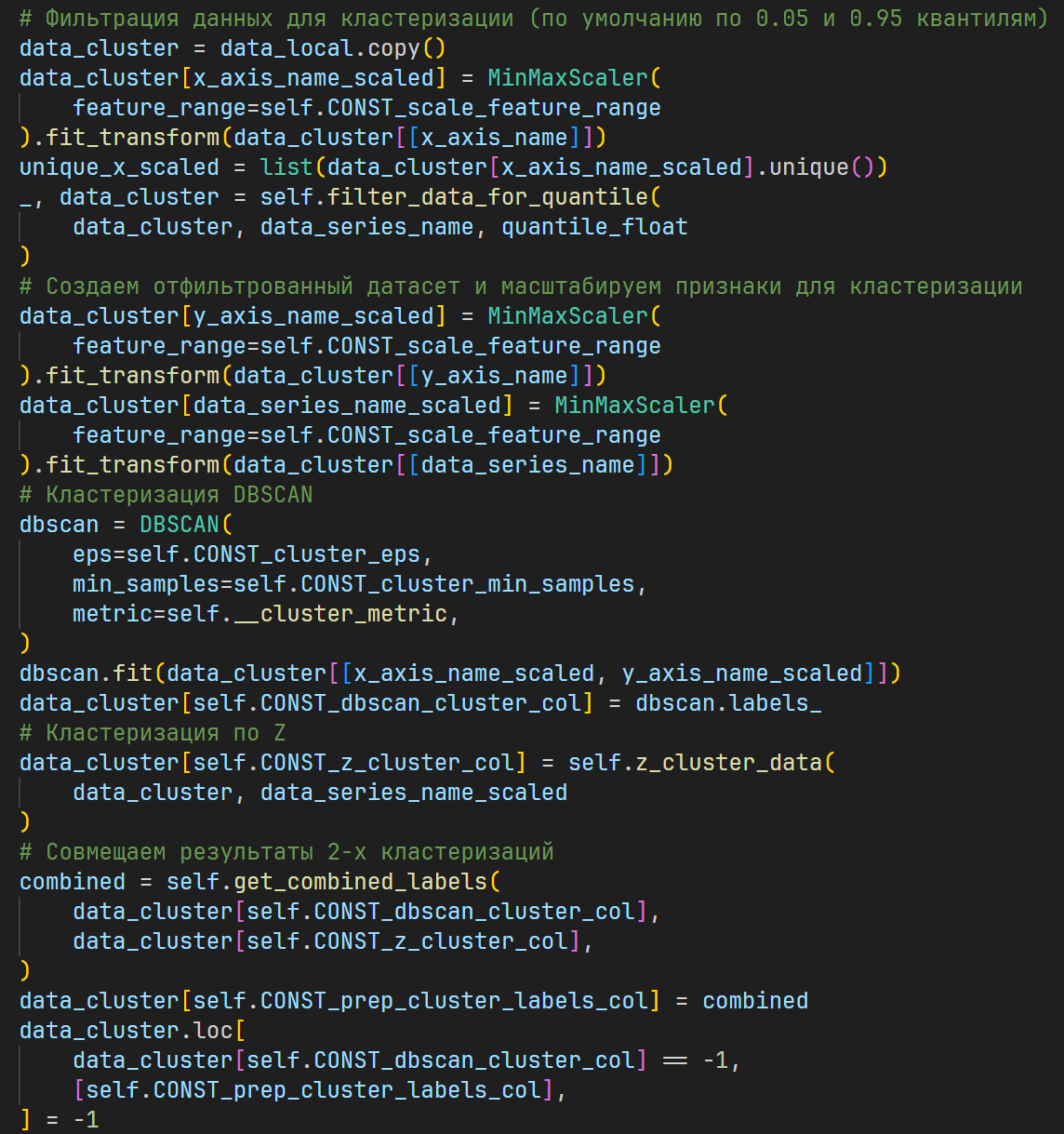


Рисунок 10 – Код предварительной обработки данных

* Нахождение парабол. После прохождения предварительной обработки в данных начинается поиск парабол. Реализация алгоритма поиска парабол состоит из нахождения характеристических точек и непосредственно поиска по этим точкам (рисунок 11).;

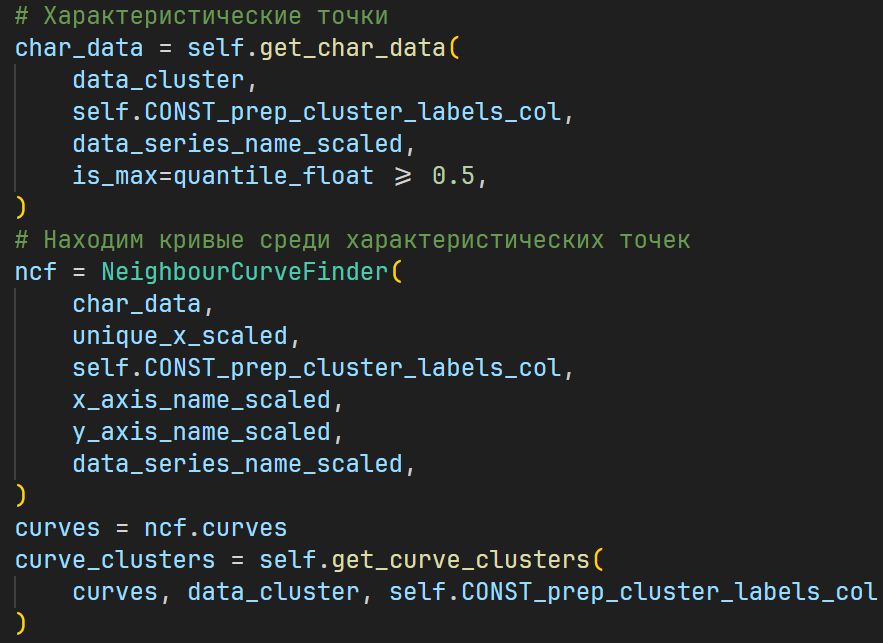


Рисунок 11 – Фрагмент кода, отвечающий за нахождение парабол

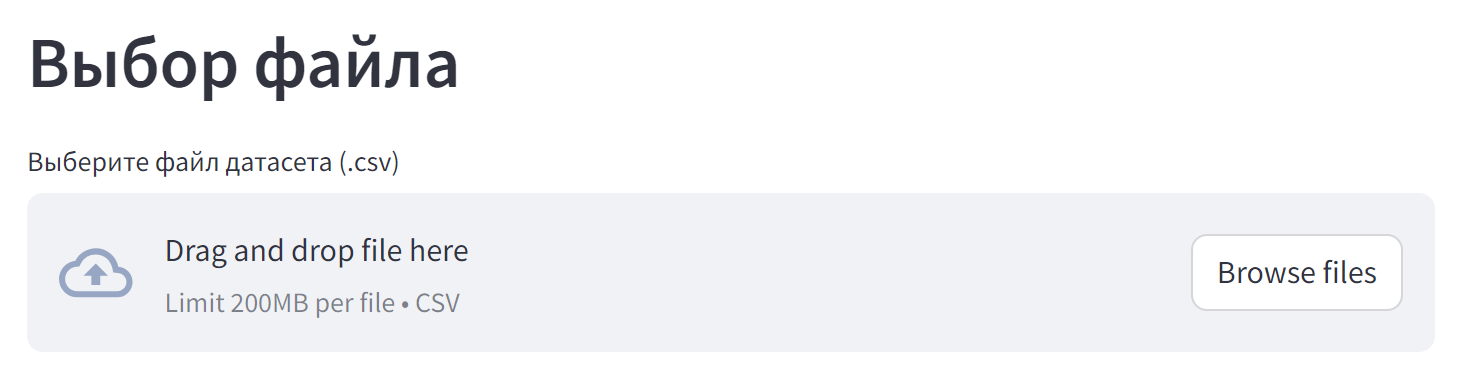
* Визуализацию результатов. Для визуализации данных в проекте используются средства Python-библиотек Matplotlib и seaborn для двухмерной визуализации, а также инструменты библиотеки Plotly для предоставления трехмерного отображения результатов.

### **2.1.6 Функции приложения**

Загрузка и работа модуля обработки спектроскопических данных квантового вычислителя начинается с запуска файла app.py командой streamlit run.

Работа программы начинается с пользователя, который вводит свой датасет (рисунок 12), разберем работу программы на двух из них:

1. С хорошими данными, где будет найдена и отрисована парабола
2. С плохими данными, в которых нельзя найти параболу.

 Рисунок 12 – Кнопка добавления датасета

После загрузки файла отображаются данные датасета (рисунок 13). Сразу после загрузки датасета появляются поля ввода, в которых можно выбрать названия столбцов, данные из которых будут использоваться для осей x, y, z. (рисунок 14).

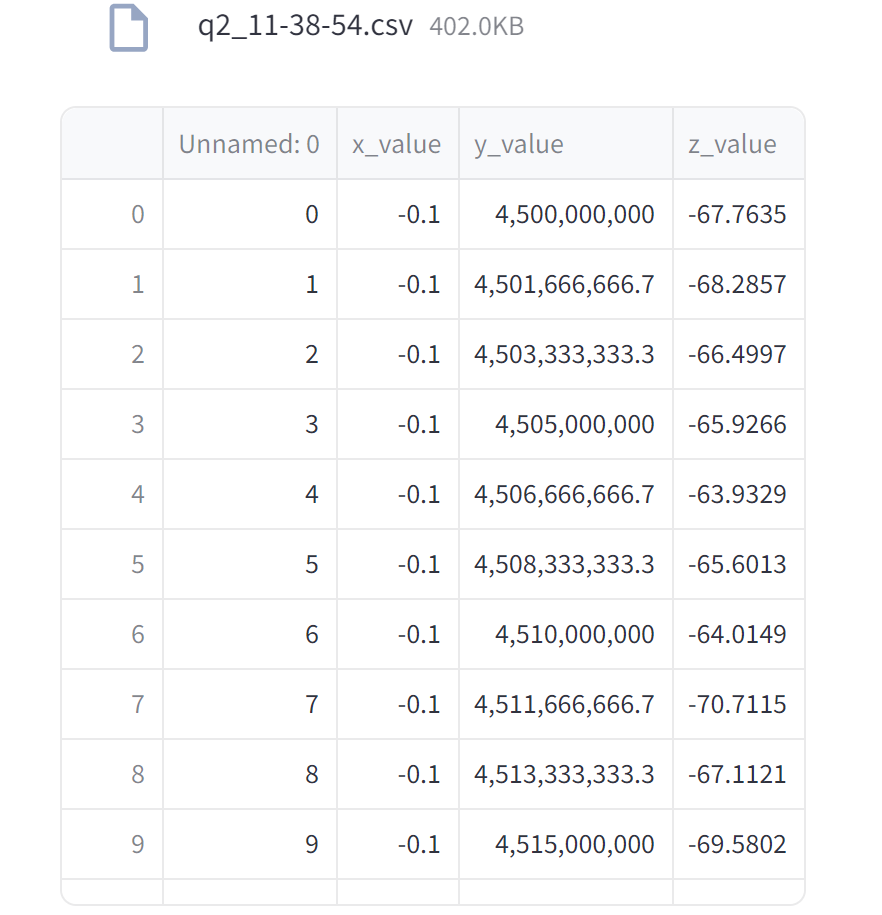


Рисунок 13 – Данные датасета

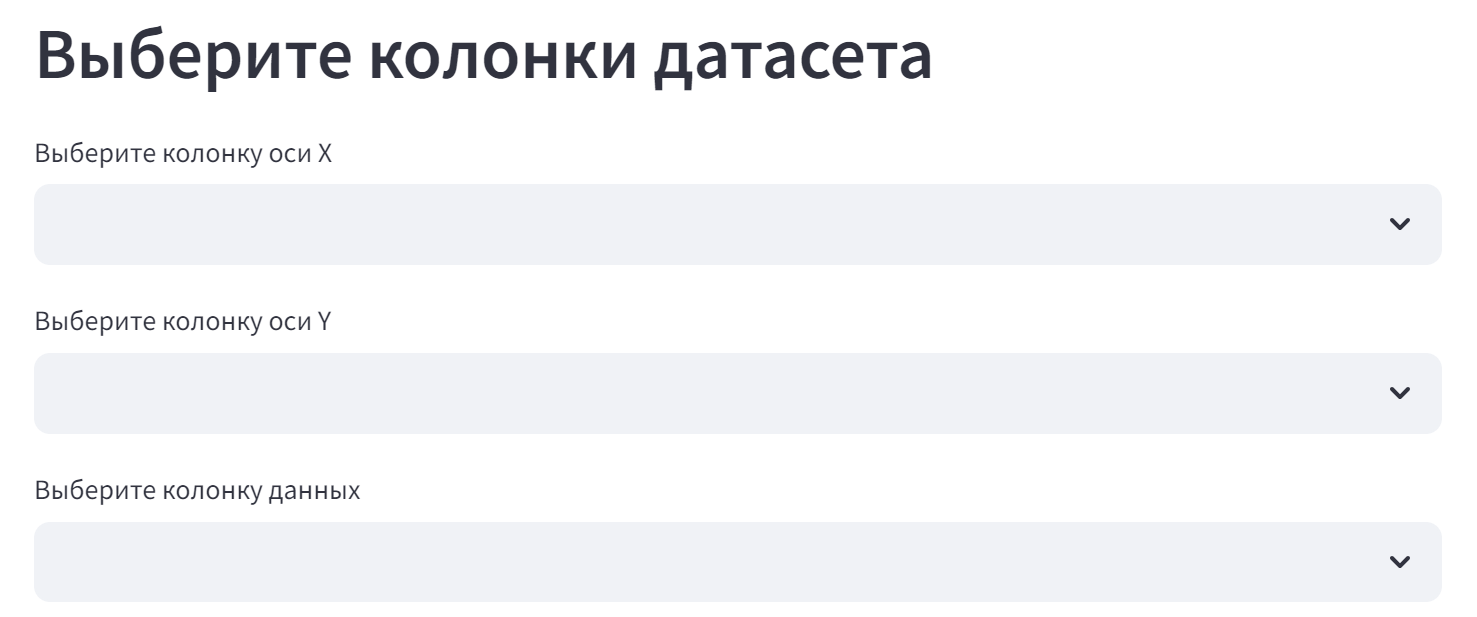


Рисунок 14 – Поля для ввода названий колонок, соответствующих осям

**Датасет с хорошими данными.**

После выбора колонки датасета пользователь увидит тепловую карту и график плотности распределения по Z (рисунок 15). Можно заметить, что данные хорошие и в них четко видно наличие параболы в самом конце графика.

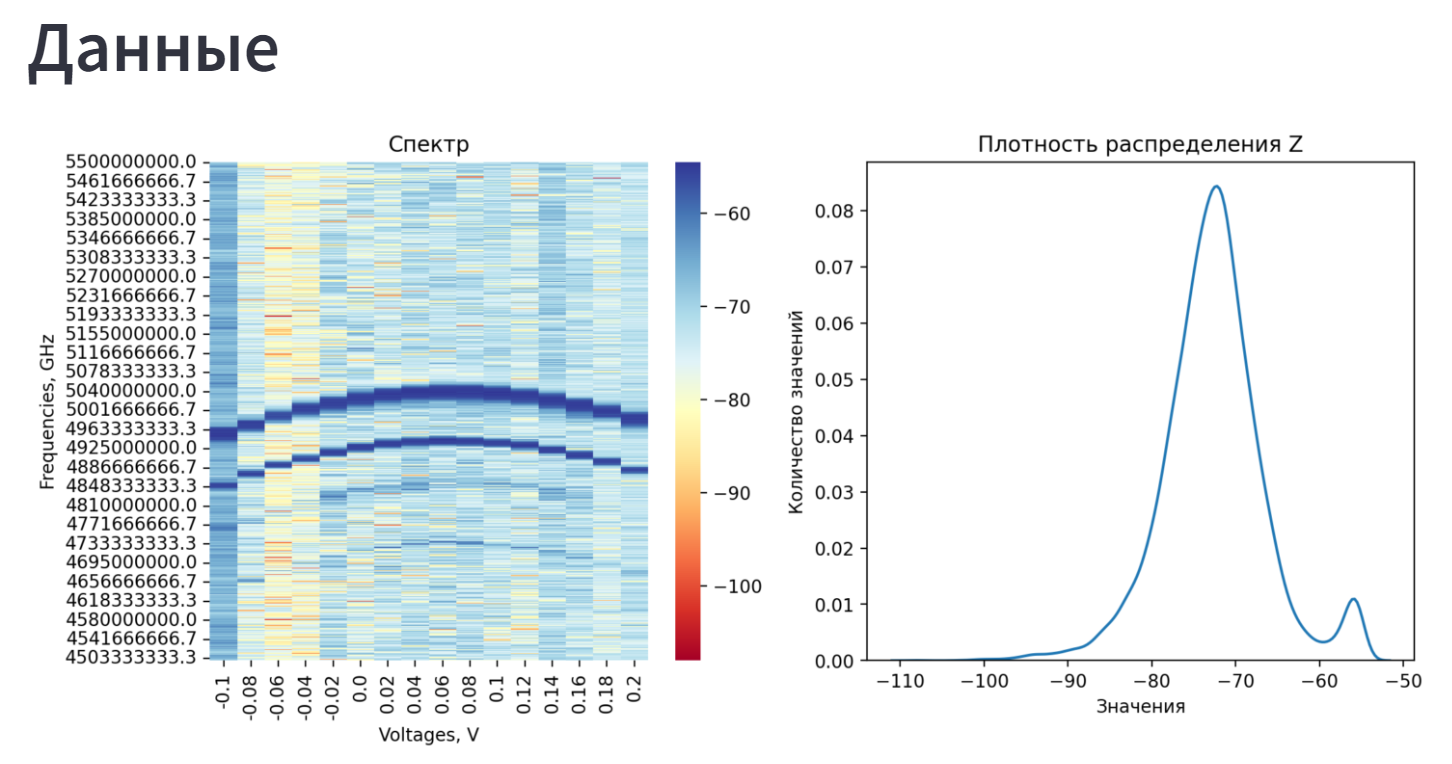


Рисунок 15 – Спектрограмма и график плотности распределения по оси z

Далее программа начнет обработку данных по выбранной пользователем нижней и верхней процентили (рисунок 16), чтобы понять, в какой именно части находится парабола. Изменение процентилей влияет на фильтрацию данных. Экспериментальным путем было выявлено, что чаще всего подходит 5 и 95 процентиль. После произойдет кластеризация, так же отдельно по выбранным процентилям. На рисунке 15 явно видно, что парабола располагается на 95 квантили.



Рисунок 16 – Выбор процентили для анализа

По завершении обработки данных программа выводит на экран результаты обработки для каждой из двух выбранных процентилей. Они включают в себя спектрограмму отфильтрованных данных и график точек после предварительной кластеризации DBSCAN (рисунок 17).

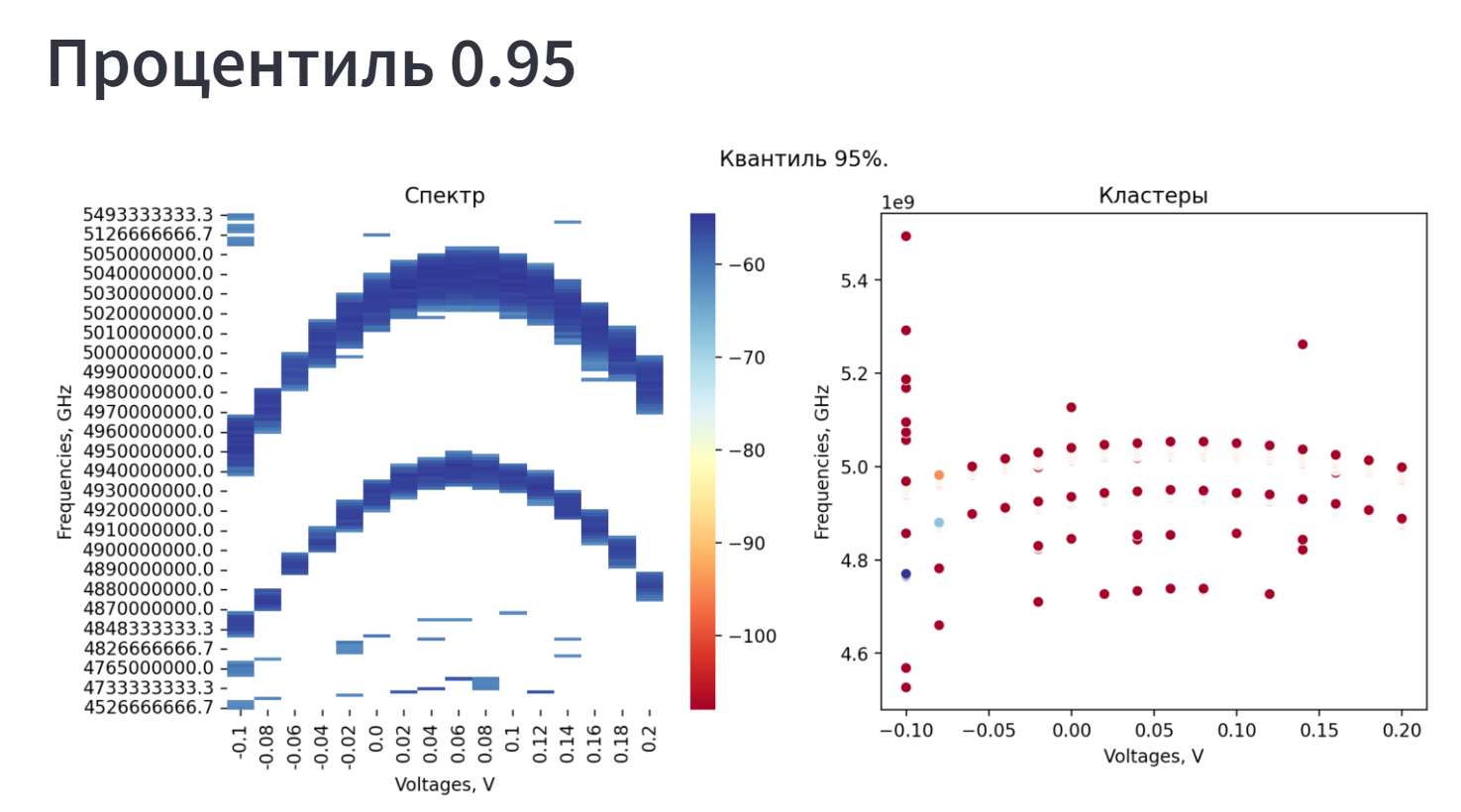


Рисунок 17 – Спектрограмма и кластеризация данных

Затем происходит 3D отрисовка конечных результатов кластеризации (рисунок 18) и выводится текстовый отчет о числе найденных парабол (рисунок 19), чтоб пользователь убедился, что кластеризация прошла успешна, алгоритм смог вычленить параболу из загруженного датасета.

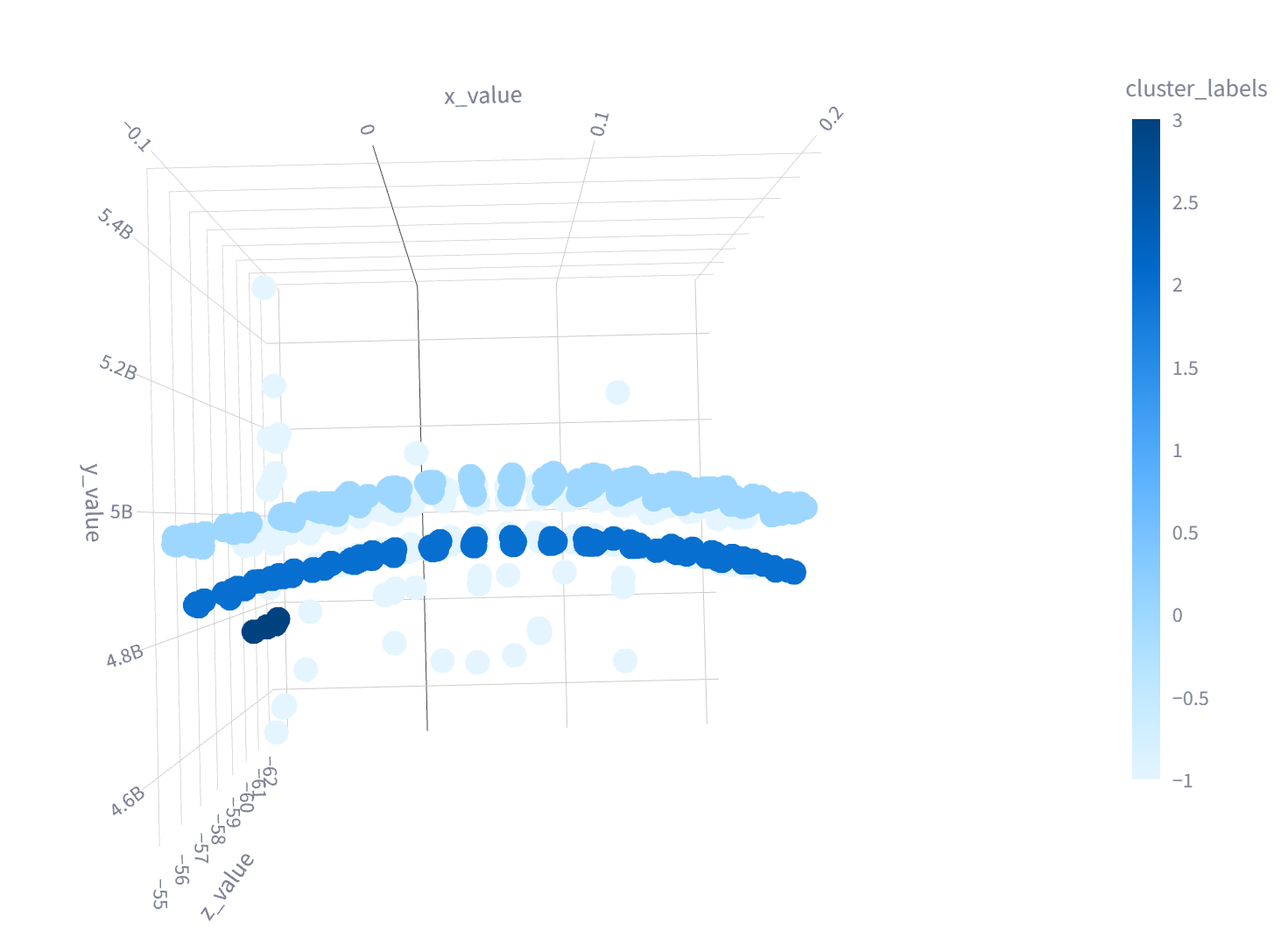


Рисунок 18 – 3D визуализация данных

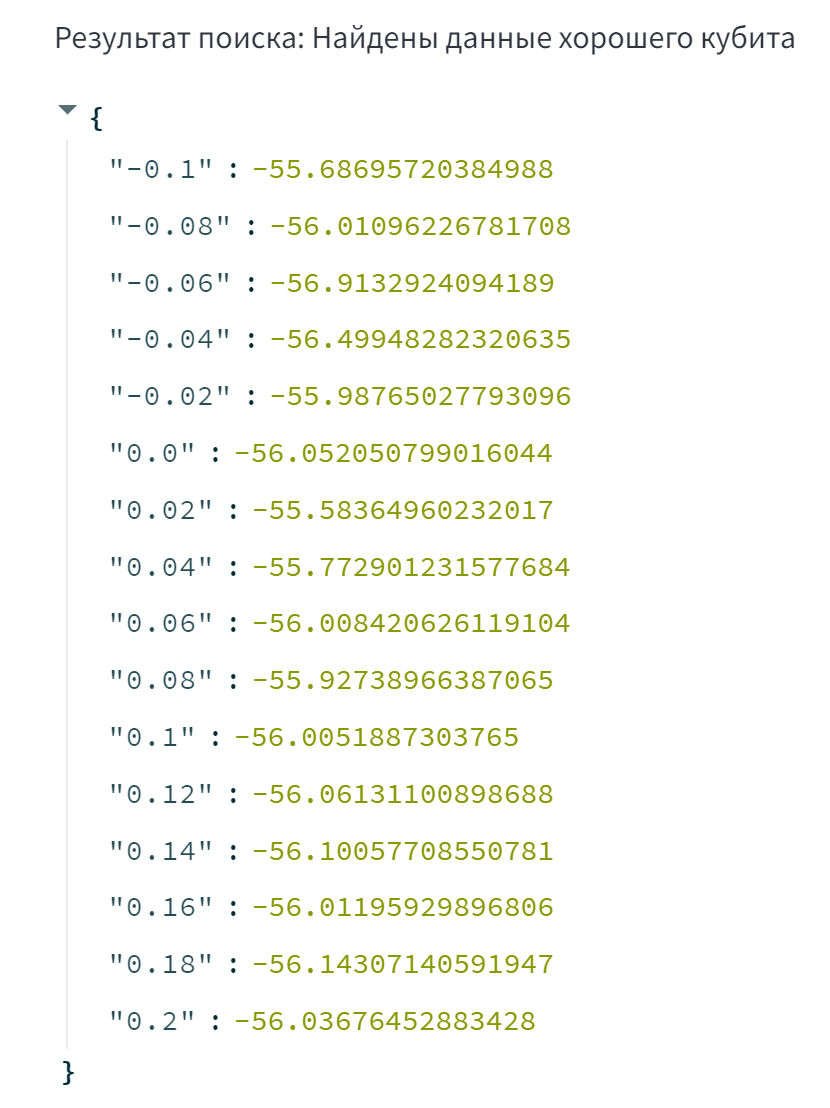


Рисунок 19 – Отчет о найденных параболах

**Работа программы на плохих данных.**

На тепловой карте не получается разглядеть параболу, так же, как и на графике с плотностью распределения по Z (рисунок 20).

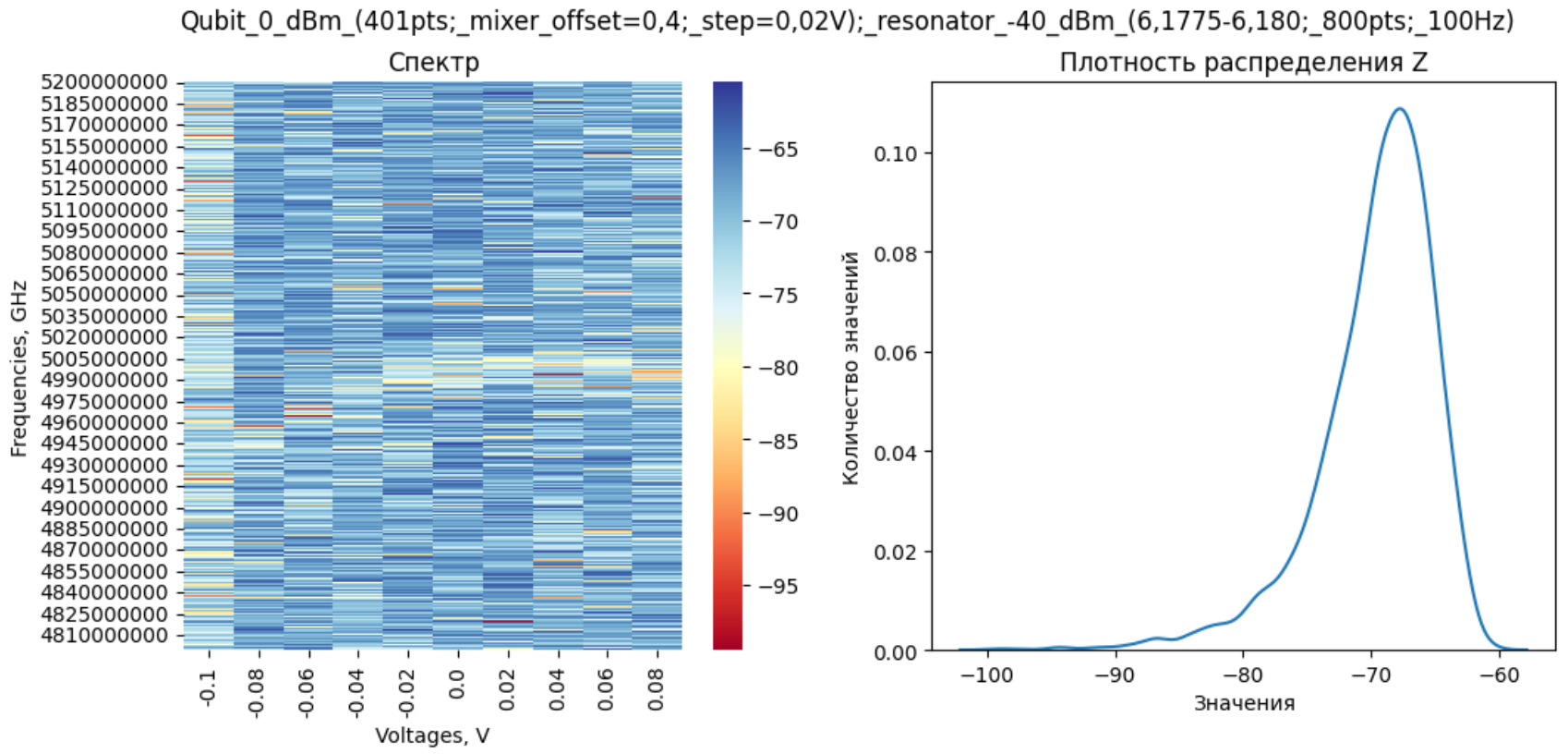


Рисунок 20 – Тепловая карта и график плотности распределения по Z плохих данных

Следовательно, и обработка по выбранной нижней и верхней процентили не может найти параболу, а алгоритм кластеризации показывает набор точек (рисунок 21).

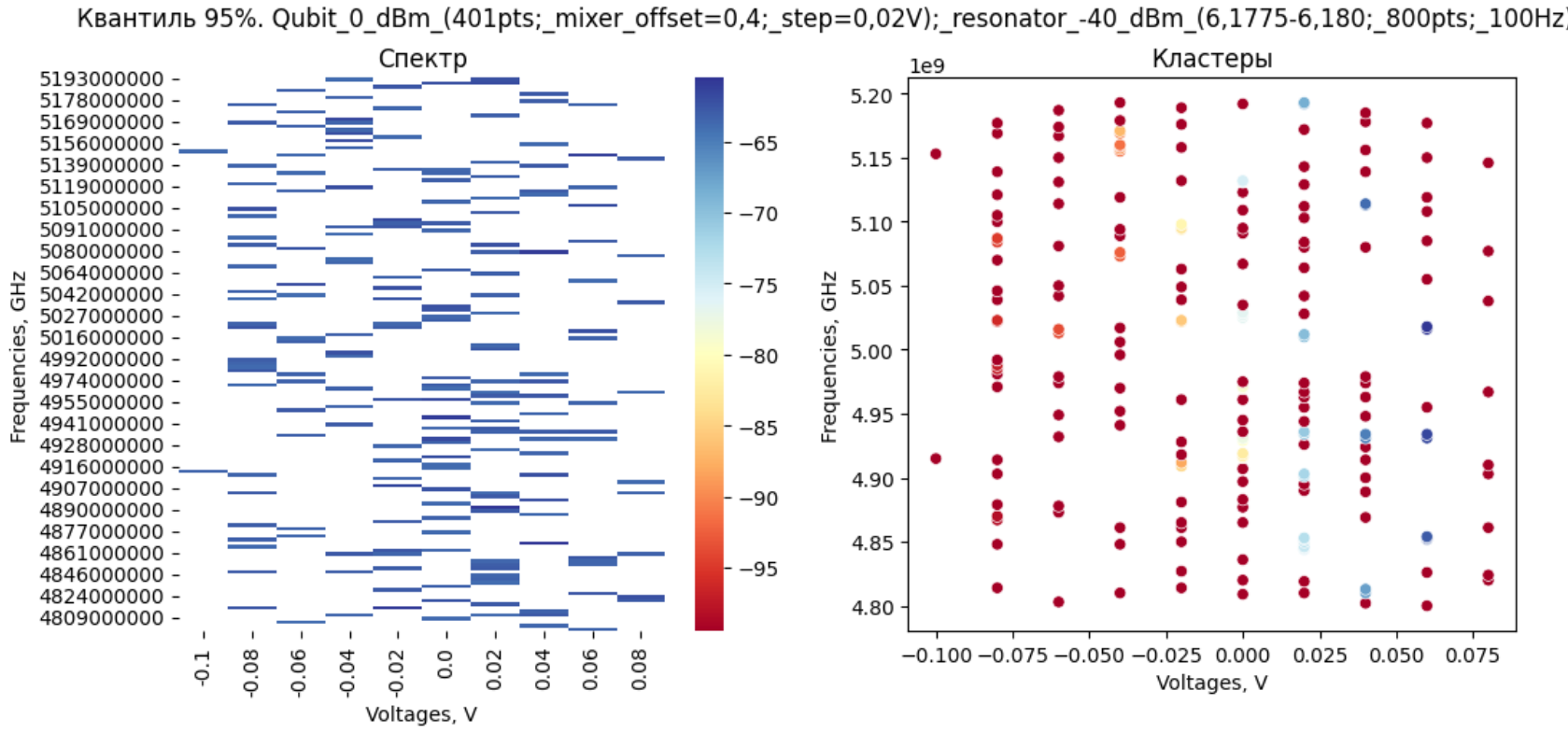


Рисунок 21 – Обработка и кластеризация по 95 квантили плохих данных

Тем самым в конечном итоге мы получаем не красивую параболу, а большое количество точек, попавших в один кластер, и отчет о том, что парабола не найдена (рисунок 22). Экранные формы показаны на рисунках А.2, А.3, А4.

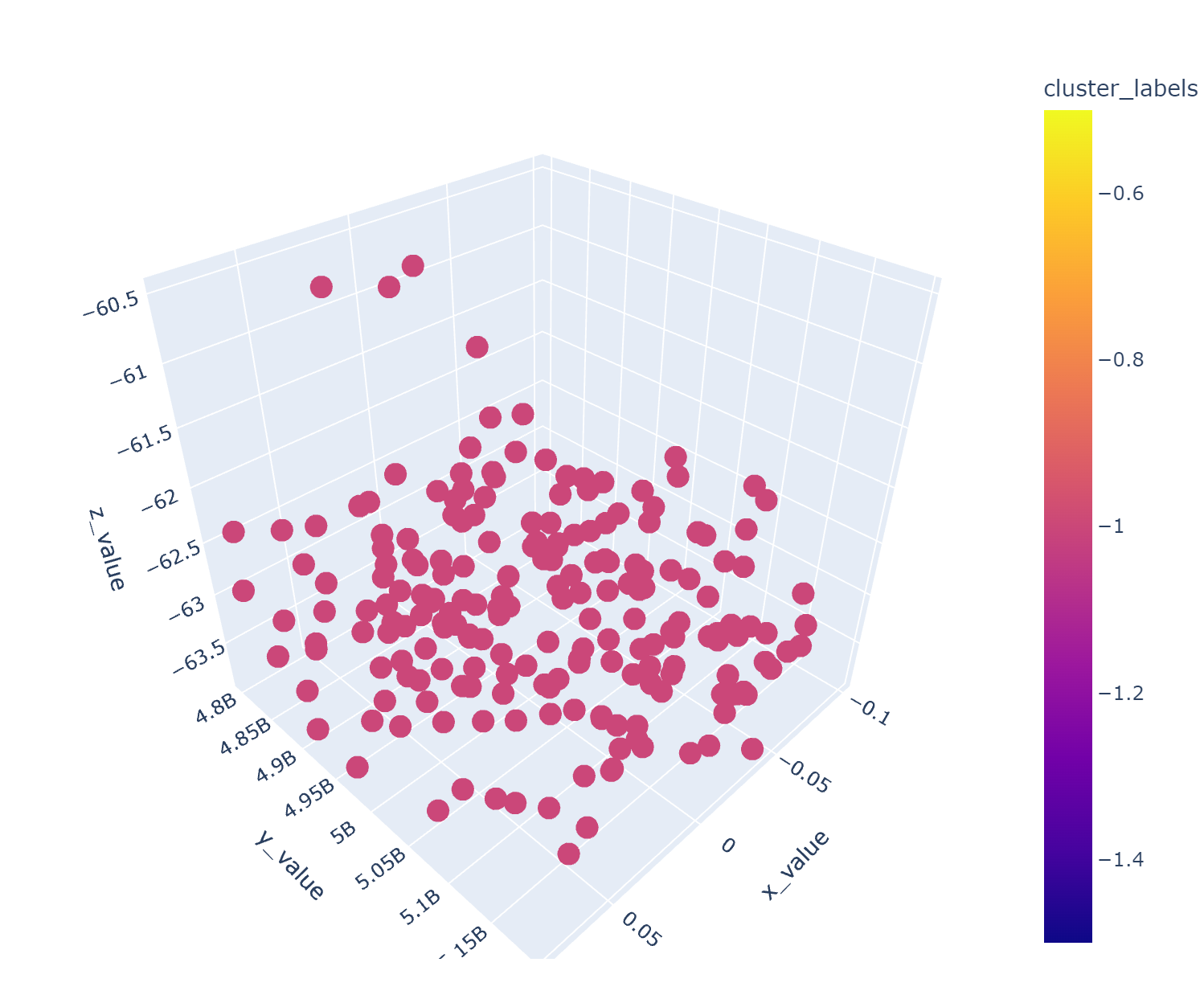


Рисунок 22 – Отрисовка данных в плохом наборе

# **ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ**

## **Проблемы, решенные в процессе разработки**

### **Проблема фильтрации данных**

Для вычленения среди данных искомых выбросов в форме параболы, целесообразно проводить фильтрацию данных, то есть выделять из полного набора точек такое их множество, которое, с одной стороны, позволяет полностью охватить выброс, но, с другой стороны, обеспечивает минимальное количество лишних данных для упрощения последующего анализа. Подбор алгоритма фильтрации был одной из задач, решенных в ходе исследования.

Были на практике проверены и проанализированы различные способы фильтрации:

* Нахождение нужных данных на основе мод в определенных промежутках. Данный метод предполагает нахождение наиболее часто встречающихся значений по оси Z внутри исследуемого набора точек, на основе которых и предлагается осуществление фильтрации. Недостатки этого метода в нашем случае проявлялись в ситуациях, когда в наборе данных встречались иные выбросы, не относящиеся к искомой параболе. Действительно, зачастую наборы данных содержат ложные вертикальные выбросы с левой стороны, которые сложно автоматически отделять от исследуемых (рисунок 23);

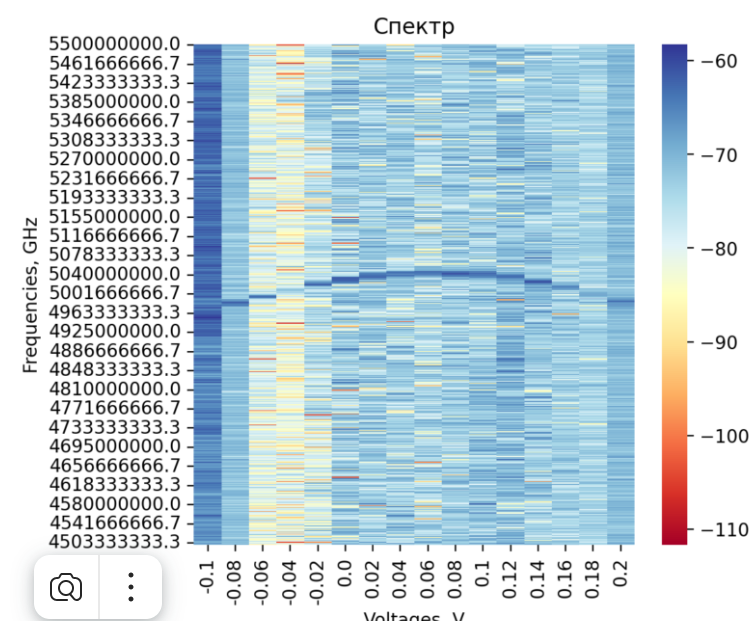


Рисунок 23 – Ложные вертикальные выбросы с левой стороны в виде полоски

* Фильтрация на основе заданного значения. Предполагалась возможность использования в качестве метода фильтрации данных отделение набора, содержащего искомый параболический выброс, по заранее заданному фиксированному порогу. В этом методе для упрощения нахождения порога и увеличения «разброса» значений также допускалась возможность предварительного высчитывания для каждого значения по оси Z значения какой-либо нелинейной монотонной функции (к примеру, экспоненты или степенной функции). Этот метод мог бы быть успешно применен, если бы интервал значений по оси Z был заранее определен и не различался бы между разными наборами данных;
* Фильтрация на основе производных на графике распределения. Данный метод предполагал построение графика распределения значений Z. По этому графику предлагалось вычислять его вторые производные, чтобы с их использованием находить точки «перелома» этого графика, и, соответственно, выбросы. К сожалению, этот метод хорошо бы работал, только если бы была заранее известна функция, определяющая распределение значений. На практике же из-за недостаточного количества точек для точного вычисления производных, после чего могли образовываться ложные маленькие выбросы на графике, а также из-за серьезных требований к вычислительным мощностям от применения этого метода пришлось отказаться;
* Фильтрация на основе процентилей. Тогда как фильтрация на основе заданного значения не подошла для решения задачи, потому что интервал, в котором пролегают значения по оси Z, не определен, экспериментально все же было выявлено, что искомая парабола, как правило, находится в верхних или нижних 5% значений координат точек по оси Z. Хотя использование фиксированных порогов такого рода все же оставляет место для ошибок и неточностей, лучшего метода на практике найдено не было. В качестве компромисса итоговый модуль позволяет изменить значения процентилей, используемые для фильтрации по умолчанию (5% и 95%) (рисунок 24).

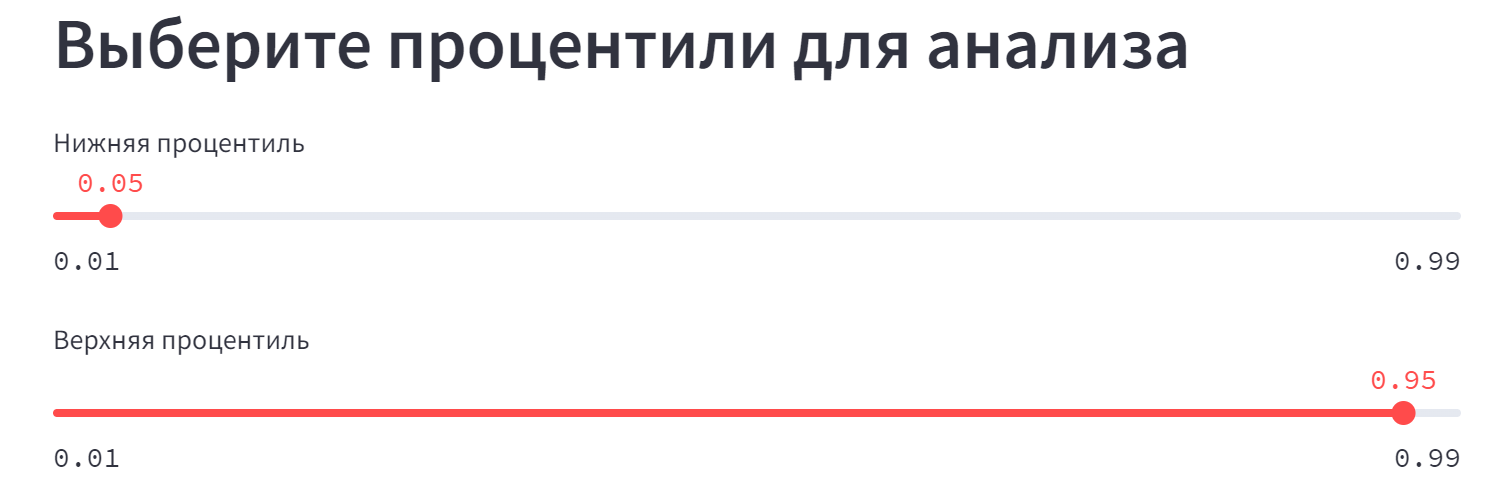


Рисунок 24 – Форма ввода процентилей, используемых для фильтрации

### **Проблема подбора алгоритма кластеризации**

Основная задача исследования заключается в подборе алгоритма кластеризации, способного находить параболические выбросы в наборе точек в трехмерном пространстве. Несколько алгоритмов кластеризации были проверены на имеющихся данных для нахождения среди них такого, который мог бы решать поставленные задачи:

* Алгоритм DBSCAN.

Прежде всего стоит разобраться, как работает DBSCAN.

1. Для начало необходимо найти корневые точки, от которых будет строиться новый кластер. DBSCAN берет точку и строит от нее буфер, предварительно указывается радиус. Если в буфере больше точек, чем минимально заданное количество, то точка становится корневой.
2. Затем находятся пограничные точки. Это точки, у которых в буфере меньше соседей, чем задано, но при этом есть корневая точка.
3. Затем ищутся выбросы. Это точки в буфере, у которых меньше соседей, чем указывалось, и также нет корневых элементов.
4. Если две корневых точки находятся рядом, то они объединяются в один кластер.
5. Пограничные точки будут относиться к группе корневого элемента из своей окрестности.
6. Процесс завершается, если ни к одному кластеру не может быть добавлено ни одного нового объекта.

Алгоритм DBSCAN был использован для нахождения кластеров точек с целью определения среди них параболических выбросов. На практике применение DBSCAN без модификаций оказалось недостаточно эффективным из-за ограничений, связанных с принципом его работы: зачастую не удавалось подобрать такие параметры или метрики расстояния, которые бы позволили избежать «срастания» парабол с точками, к ним не относящимся. Однако все же применение данного алгоритма показалось относительно эффективным для нахождения кластеров, и рассматривалось в дальнейшем [16];

* Алгоритм OPTICS.

Идея OPTICS очень напоминает идею DBSCAN, но OPTICS нужен для обнаружения кластеров в данных, имеющих различные плотности. OPTICS используется граф достижимости, который находит достижимое расстояние для каждой точки, которая в дальнейшем будет относиться к ближайшему кластеру. Если кластеры расположены близко друг к другу, это метод лучше определяет кластеры разной плотности. Однако за счет этого алгоритм требует больше времени.

Алгоритм OPTICS, в некоторой степени расширяющий DBSCAN, также рассматривался к применению. К сожалению, применение данного алгоритма не приводило к результатам, значительно лучшим, нежели результаты DBSCAN, которые ранее были признаны неудовлетворительными. К тому же, данный алгоритм обладает меньшей гибкостью относительно DBSCAN, поэтому от его использования пришлось отказаться [17];

* Собственный алгоритм кластеризации.

Ввиду того, что конвенциональные алгоритмы оказались недостаточными для решения достаточно специфичной поставленной задачи, было решено использовать алгоритм собственной разработки. Суть алгоритма заключается в том, чтобы провести предварительную кластеризацию точек с использованием DBSCAN и особой метрики для вычисления расстояний между точками (формула 5):

Совместно с применением параметров , , такая метрика расстояния позволяет находить наборы точек, имеющие одинаковое значение вдоль оси X. Здесь мы пользуемся ограниченной дискретизацией исследуемых наборов данных. Это делается в том числе для того, чтобы по большому скоплению кластеров для одного значения X находить большие выбросы-полосы, параллельные оси Y, которые часто встречаются в исследуемых наборах данных. Далее для найденных кластеров находятся характеристические точки, конкретно выбираются пики выбросов вдоль оси Z. Последний шаг – итеративное нахождение парабол по характеристическим точкам: обход начинается вдоль оси X с того конца, на котором находится наименьшее количество характеристических точек, после чего для каждой характеристической точки с заданным X последовательно выбираются ближайшие точки со следующим значением X с дополнительным ограничением на угол, под которым следующие точки лежат к до этого построенной части кривой (рисунок 25).

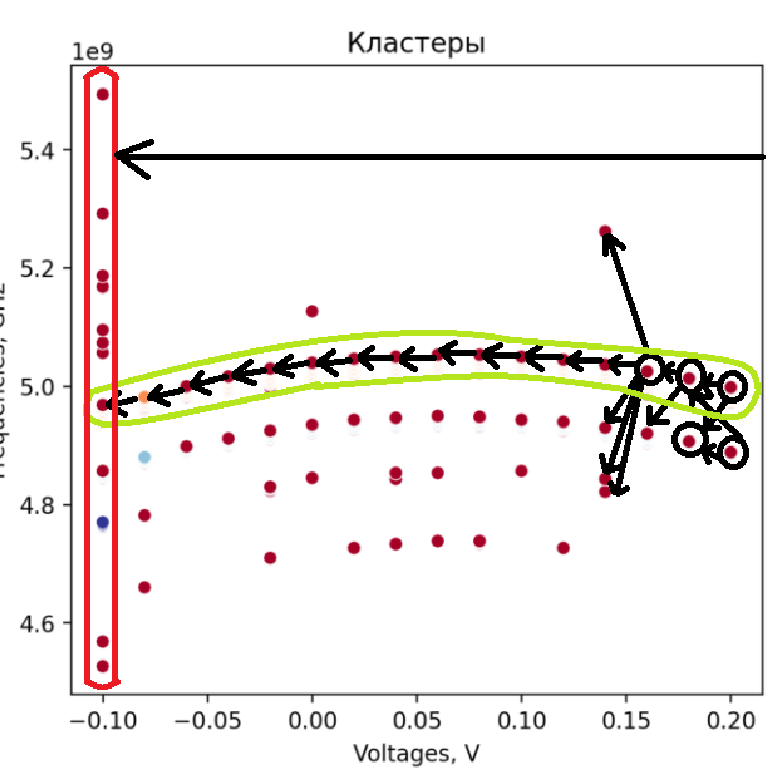


Рисунок 25 – Пример работы алгоритма нахождения парабол по кластерам DBSCAN

# **ЗАКЛЮЧЕНИЕ**

В рамках выпускной квалификационной работы бакалавра было разработано приложение «Модуль обработки спектроскопических данных квантового вычислителя». В ходе ВКРБ были выполнены следующие задачи:

* 1. Исследована предметная область;
  2. Выбрана программная платформа методом взвешенной суммы;
  3. Были описаны технологии;
  4. Описана функциональная модель;
  5. Был описан собственный алгоритм;
  6. Были описаны функции приложения;
  7. Были подробно описаны проблемы, решенные в процессе работы.

Получившееся приложение позволяет пользователю быстро находить наиболее толстую верхнюю параболу, в загруженном датасете, и анализировать получившийся результат. Тем самым автоматически отслеживать выбросы в квантовом компьютере.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Липаев В. В. Системное проектирование сложных программных средств для информационных систем. – М.: СИНТЕГ, 2002. - 224 с.
2. Методические рекомендации по подготовке и защите выпускной квалификационной работы бакалавра. / Кротов Ю.Н. [Электронный ресурс] – URL: https://drive.google.com/file/d/1pEcfTr3xDdJ81Hxz2F6GcbtNV1n3dan6/view. (дата обращения: 10.06.2024).
3. Документация R. [Электронный ресурс] – URL: https://www.r-project.org/ (дата обращения: 10.06.2024).
4. Современный учебник JavaScript. [Электронный ресурс] – URL: <https://learn.javascript.ru/> (дата обращения: 10.06.2024).
5. Документация Python. [Электронный ресурс] – URL: https://www.python.org/doc/ (дата обращения: 10.06.2024).
6. Пocтникoв B. M. Ocнoвы экcплyaтaции ACOИиУ Toм 1. Texничecкoe oбcлyживaниe ACOИиУ. yчeб. пocoбиe / B. M. Пocтникoв. — M.: Изд-вo MГTУ им. H. Э. Бayмaнa, 2015. — 214 [2] c.
7. Пocтникoв B. M., Ocнoвы экcплyaтaции ACOИиУ Toм 2. Администрирование и развитие ACOИиУ. yчeб. пocoбиe / B. M. Пocтникoв. — M.: Изд-вo MГTУ им. H. Э. Бayмaнa, 2015. — 205 [2] c.
8. Методология функционального моделирования IDEF0 [Электронный ресурс] // URL: https://advanced-quality-tools.ru/assets/idef0-rus.pdf/ (дата обращения: 10.06.2024).
9. Документация Matlplotlib. [Электронный ресурс] – URL: https://matplotlib.org/stable/index.html/(дата обращения: 10.06.2024).
10. Документация Sklearn [Электронный ресурс] – URL: https://scikit-learn.org/stable/user\_guide.html/(дата обращения: 10.06.2024).
11. Документация Numpy. [Электронный ресурс] – URL: https://numpy.org/doc/stable/(дата обращения: 10.06.2024).
12. Документация Pandas. [Электронный ресурс] – URL: https://pandas.pydata.org/docs/(дата обращения: 10.06.2024).
13. Документация Seaborn. [Электронный ресурс] – https://seaborn.pydata.org/(дата обращения: 10.06.2024).
14. Документация Plotly. [Электронный ресурс] – URL: https://plotly.com/python-api-reference/generated/plotly.html/(дата обращения: 10.06.2024).
15. Документация Streamlit. [Электронный ресурс] – URL: https://docs.streamlit.io/(дата обращения: 10.06.2024).
16. Ester M., Kriegel H.P., Jörg S., Xiaowei X. A density-based algorithm for discovering clusters in large spatial databases with noise // [Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)](https://archive.org/details/kdd96proceedings0000inte) / Evangelos S., Jiawei H., Usama M. F. — [AAAI Press](https://ru.wikipedia.org/w/index.php?title=AAAI_Press&action=edit&redlink=1), 1996. — С. [226](https://archive.org/details/kdd96proceedings0000inte/page/226)–231. — [ISBN 1-57735-004-9](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/1577350049).
17. Mihael A., Markus M. B., Hans-Peter K., Jörg S. OPTICS: Ordering Points To Identify the Clustering Structure // [=ACM SIGMOD international conference on Management of data](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.129.6542). — [ACM Press](https://ru.wikipedia.org/w/index.php?title=ACM_Press&action=edit&redlink=1), 1999. — С. 49–60.

# **ПРИЛОЖЕНИЕ А ГРАФИЧЕСКАЯ ЧАСТЬ**

В графическую часть выпускной квалификационной работы входят:

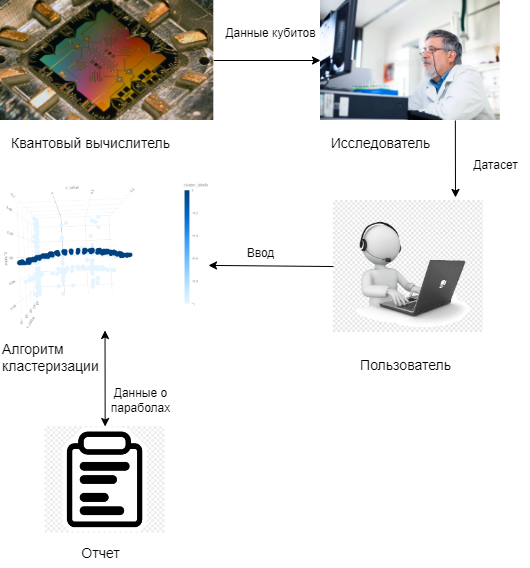
А.1 Предметная область,

А.2 Экранные формы часть 1,

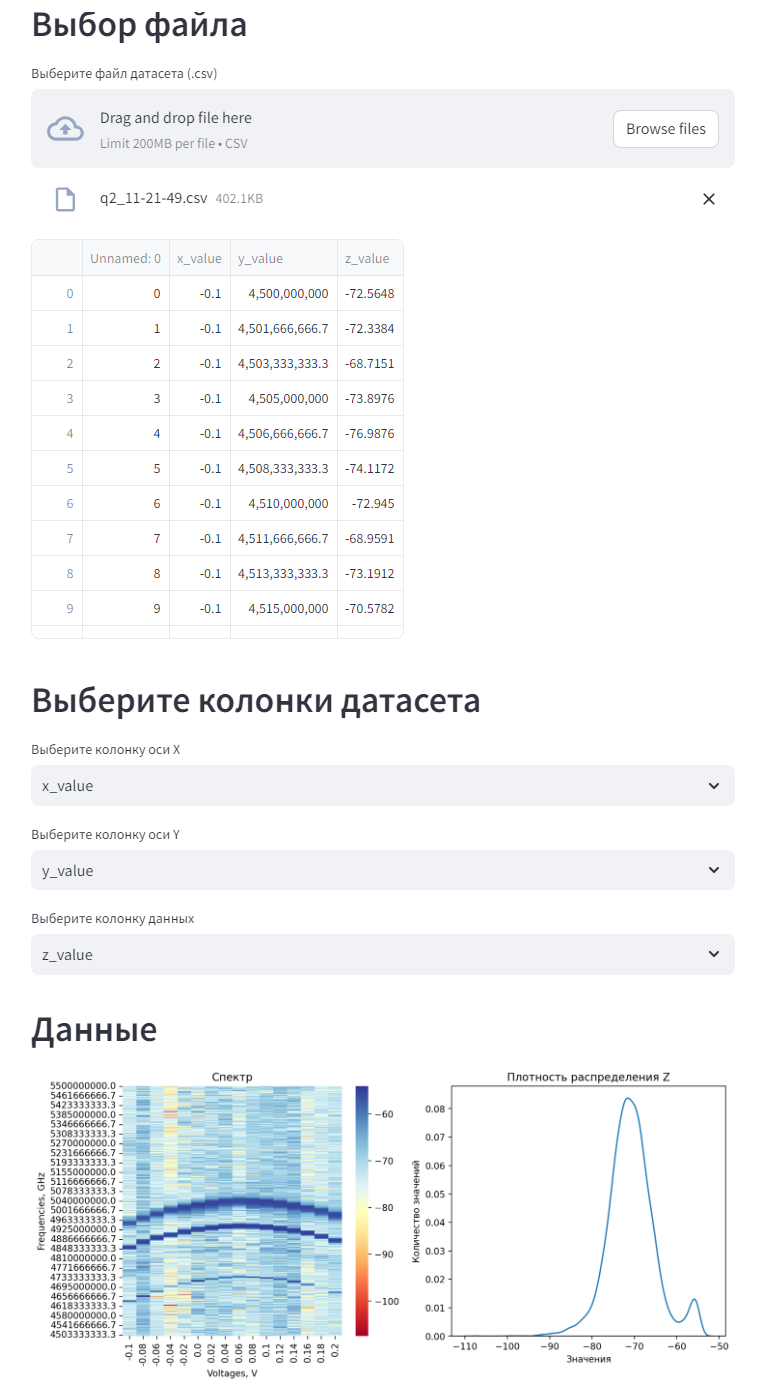
А.3 Экранные формы часть 2,

А.4 Экранные формы часть 3.

А.1 Предметная область



А.2 Экранные формы часть 1



А.3 Экранные формы часть 2



А.4 Экранные формы часть 3



# **ПРИЛОЖЕНИЕ Б ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
|  | \_\_\_\_\_\_\_\_\_\_\_ В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_ Ю.Е. Гапанюк  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Модуль обработки спектроскопических данных квантового вычислителя**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

5

(количество листов)

ИСПОЛНИТЕЛЬ: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Акулова Анастасия Алексеевна

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г.

Москва - 2024

**1. Введение**

Наименование – модуль обработки спектроскопических данных квантового вычислителя. Данная информационная система может быть использована в дальнейших исследованиях квантовых компьютеров в МГТУ им. Баумана.

**2. Основания для разработки**

Основанием для разработки является задание на выпускную квалификационную работу, подписанное руководителем выпускной работы и утверждённое заведующим кафедрой ИУ5 МГТУ им. Н.Э. Баумана 15 декабря 2023 года.

**3. Назначение разработки**

Назначением выпускной квалификационной работы является создание автоматизированной информационной системы «Модуля обработки спектроскопических данных квантового вычислителя», позволяющей пользователю отыскивать оптимальные параметры запуска, то есть выделять параболы на загруженных датасетатах, а также визуализировать данные.

**4. Требования к программе или программному изделию**

**4.1 Требования к функциональным характеристикам**

Программа должна выполнять следующие функции:

4.1.1 Загрузка данных.

4.1.2 Выбор колонки датасета.

4.1.3 Выбор процентили для анализа.

4.1.4 Предварительная обработка данных.

4.1.5 Визуализация данных.

4.1.6 Устранение пропусков в данных.

4.1.7 Нахождение оптимальных параметров.

4.1.8 Вывод оптимальных параметров или сообщения о том, что таковые не найдены.

4.1.9 Вывод пояснений к результатам.

**4.2 Требования к надежности**

Система должна надежно и устойчиво функционировать, при вводе некорректных данных выдавать сообщение об ошибке на русском языке. При сбоях восстанавливаться после перезагрузки.

**4.3 Требования к составу и параметрам технических средств**

Минимальные системные требования к устройству, выполняющему программу:

* Четырёхъядерный процессор с частотой не менее 2 ГГц;
* Объем оперативной памяти не менее 4 ГБ;
* Объем свободной памяти на диске не менее 2 ГБ;
* VGA-совместимый видеоадаптер и монитор;
* Компьютерная мышь;
* Клавиатура.

**4.4 Требования к информационной и программной совместимости**

Для корректной работы системы необходимо наличие на устройстве следующих программных средств:

* ОС Windows 10 и выше;
* Python 3.10;

**5. Требования к программной документации**

Для представления заказчику разрабатываются следующие документы:

5.1 Программа и методика испытаний;

5.2 Руководство пользователя – c описанием всех действий, которые пользователь может произвести, и реакцию системы на эти действия; порядок действий пользователя при зависании или сбое программы;

5.3 Руководство администратора;

5.3 Расчетно-пояснительная записка.

**6. Технико-экономические показатели**

Требования к данному разделу не предъявляются.

**7. Стадии и этапы разработки**

**7.1 Этапы разработки программы**

Разработка программы будет производиться в соответствии со следующими этапами:

* Исследование предметной области
* Выбор языка программирования
* Выбор среды разработки
* Разработка программного кода
* Тестирование программного кода
* Создание клиентского приложения
* Внедрение кода в клиентское приложение

**7.2 Перечень программных документов**

Перечень программных документов и сроки их оформления приведены в таблице 2.

Таблица 2 – перечень программных документов

| № п/п | Наименование документа | Сроки выполнения |
| --- | --- | --- |
| 1 | Техническое задание | 11-20 марта 2024 г. |
| 2 | Программа и методика испытания | апрель 2024 г. |

**8. Порядок контроля и приёмки**

Приём программного изделия в виде испытаний функциональных возможностей осуществляется в ходе «Защиты макетов программ – предварительной защиты ВКРБ» в период с 15 по 24 мая 2024 года в соответствии с разработанной программой и методикой испытаний.

На испытание представляются: программный продукт, техническое задание, программа и методика испытаний.

# **ПРИЛОЖЕНИЕ В ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ**

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
|  | \_\_\_\_\_\_\_\_\_\_\_ В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_ Ю.Е. Гапанюк  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Модуль обработки спектроскопических данных квантового вычислителя**

Программа и методика испытаний

(вид документа)

писчая бумага

(вид носителя)

7

(количество листов)

ИСПОЛНИТЕЛЬ: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Акулова Анастасия Алексеевна

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г.

Москва - 2024

**Аннотация**

В данном документе описываются последовательность и методы проведения испытаний при тестировании программного изделия, состав и структура технических и программных средств, необходимых для проведения испытаний, а также приводятся требования к предъявляемой документации, характеристикам программы применительно к условиям эксплуатации и требования к информационной и программной совместимости. Описывается тестовый пример и реакция системы на него.

* 1. **Объект испытаний**

Модуль обработки спектроскопических данных квантового вычислителя.

* 1. **Цель испытаний**

Цель испытания – проверка функционирования всех указанных в техническом задании функций программы.

* 1. **Состав предъявляемой документации**

На испытания программного продукта предъявляются следующие документы:

1) Техническое задание.

2) Программа и методика испытаний.

* 1. **Технические требования**

* 1. **Требования к программной документации**

Комплектность программной документации должна удовлетворять разделу данного документа "Состав предъявляемой документации".

* 1. **Требования к техническим характеристикам**
     1. **Требования к составу аппаратного обеспечения**

Модуль обработки спектроскопических данных квантового вычислителя должен выполняться на устройстве со следующими характеристиками:

* Четырёхъядерный процессор с частотой не менее 2 ГГц;
* Объем оперативной памяти не менее 4 ГБ;
* Объем свободной памяти на диске не менее 2 ГБ;
* VGA-совместимый видеоадаптер и монитор;
* Компьютерная мышь;
* Клавиатура.
  + 1. **Требование к составу программного обеспечения**

Для работы модуля обработки спектроскопических данных квантового вычислителя требуются:

* ОС Windows 10 и выше;
* Python 3.10.

* 1. **Порядок проведения испытаний**

Испытания данного программного продукта будут проводиться в следующем порядке:

1) Запуск модуля обработки спектроскопических данных квантового вычислителя.

2) Тестирование функционала модуля обработки спектроскопических данных квантового вычислителя.

**5.1 Требования к составу аппаратного обеспечения**

Требования к составу аппаратного обеспечения учитываются согласно пункту 4.2.1.

**5.2 Требования к составу программного обеспечения**

Требования к составу программного обеспечения учитываются согласно пункту 4.2.2.

**6. Методика проведения испытаний**

Методы испытаний и их последовательность для информационной системы «Модуль обработки спектроскопических данных квантового вычислителя» приведены в следующей таблице 1.

Таблица 1 - Методы испытаний и их последовательность

| **N** | **N пункта ТЗ** | **Выполняемые действия** | **Результат** |
| --- | --- | --- | --- |
| 1 |  | Пользователь: запускает приложение «Модуль обработки спектроскопических данных квантового вычислителя». | Отображается главная форма приложения. |
| 2 | 4.1.1  Загрузка данных. | Пользователь:  Нажимает на кнопку выбора файла датасета,  Выбирает файл датасета. | Отображаются названия датасетов, появляется выбор колонки датасета. |
| 3 | 4.1.2  Выбор колонки датасета. | Пользователь: Выбирает из выпадающего списка  Колонку оси X.  Колонку оси Y.  Колонку данных | Отображается список, сгенерированный на основе загруженного датасета с возможностью выбора полей оси X, оси Y, колонки данных. Появляется слайдер для выбора нижней и верхней квантили данных. |
| 4 | 4.1.3  Выбор квантили для анализа. | Пользователь: Выбирает для анализа  Нижнюю квантиль  Верхнюю квантиль | Отображается слайдер, где пользователь сам может выставить значения нижней и верхней квантили данных. Появляется кнопка «Обработать датасет». |
| 5 | 4.1.4  Предварительная обработка данных.  4.1.5  Визуализация данных.  4.1.6  Устранение пропусков в данных.  4.1.7  Нахождение оптимальных параметров.  4.1.8  Вывод оптимальных параметров или сообщения о том, что таковые не найдены.  4.1.9  Вывод пояснений к результатам. | Пользователь: нажимает на кнопку «Обработать результат» после добавления датасета с заведомо известной непрерывной параболой. | Отображаются наборы графиков по выбранной пользователем нижней и верхней квантили, по одному из которых явно видно наличие параболы, вывод пояснений к результатам и точек найденной параболы в текстовом виде. |
| 6 | 4.1.4 Предварительная обработка данных.  4.1.5  Визуализация данных.  4.1.6  Устранение пропусков в данных.  4.1.7  Нахождение оптимальных параметров.  4.1.8  Вывод оптимальных параметров или сообщения о том, что таковые не найдены.  4.1.9  Вывод пояснений к результатам. | Пользователь: нажимает на кнопку «Получить результат» после добавления датасета, заведомо не содержащего парабол. | Отображаются наборы графиков по выбранной пользователем нижней и верхней квантили, по которым явно видно отсутствие параболы, вывод сообщения о том, что парабола не найдена, в текстовом виде |

**7. Результат испытаний**

Основой испытаний является демонстрация работы основных функций приложения: загрузка данных, предварительная обработка данных, визуализация данных, устранение пропусков в данных, нахождение оптимальных параметров, вывод оптимальных параметров или сообщения о том, что таковые не найдены, вывод пояснений к результатам.

Испытание считается пройденным успешно, если в процессе демонстрации пользователь смог успешно ввести и обработать данные, а также получить вывод о наличии или отсутствии параболы в своем датасете.