

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 22.Б11-мм

Реализация и экспериментальное исследование графового метода обнаружения разладки

Ахметов Темерлан Витальевич

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:

Ассистент кафедры системного программирования, Гориховский В.

Консультант:

Инженер-исследователь, Лаборатория технологий программирования инфраструктурных
решений СПбГУ, Кутуев В. А.

Санкт-Петербург
2023

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Постановка задачи	6
2.2. Существующие методы обнаружения разладок	6
2.3. Графовые методы в обнаружении разладок	7
3. Реализация	10
3.1. Построение графа	11
3.2. Вычисление сетевых характеристик	12
3.3. Обнаружение разладок	12
4. Эксперимент	14
4.1. Дизайн эксперимента	14
4.2. Результаты эксперимента	14
4.3. Обобщение результатов	17
Заключение	18
Список литературы	19

Введение

Современное общество сталкивается с огромным объемом данных, собираемых в различных областях, от социальных сетей до биологических экспериментов и анализа изображений. Анализ этих данных требует разработки эффективных методов обнаружения структурных изменений или разрывов в последовательностях, что в свою очередь стимулирует развитие новых статистических методов.

Обнаружение точек изменений, или разладок, является важным этапом анализа данных. Это позволяет выявлять периоды, в которых характеристики данных существенно меняются. Применительно к сетевым структурам, такие изменения могут указывать на важные события, такие как:

1. *Финансовые рынки[5]*: В изменениях в структуре торговли на финансовых рынках могут быть отражены важные экономические события или изменения инвестиционной стратегии.
2. *Медицинская диагностика[3]*: В медицине, изменения в кардиограммах или других биомедицинских данных могут свидетельствовать о заболеваниях, изменениях в состоянии пациента или эффективности лечения.
3. *Производственные процессы[2]*: В мире производства, разладки могут указывать на нештатные ситуации или сбои в работе оборудования, что критически важно для обеспечения качества и эффективности производства.

Одним из перспективных подходов к решению задачи обнаружения разладок являются графовые методы. В этом контексте, данные представляются в виде графов, где узлы представляют отдельные наблюдения, а рёбра отображают взаимосвязь между ними. Такой подход позволяет визуально представить структуру данных и обнаруживать её изменения во времени.

Цель этой семестровой практики заключается в рассмотрении и реализации графового метода[1] для обнаружения разладок с использованием сканирующей статистики. Этот метод позволяет производить сканирование данных с целью выявления периодов статистически значимых изменений в сетевых характеристиках.

1 Постановка задачи

Цель данной учебной практики заключается в изучении и реализации графового метода для решения задачи о разладке в данных

Для достижения были поставлены следующие задачи:

- Проанализировать современные исследования по применению графовых методов в задачах обнаружения разладки.
- Реализовать графовый метод для решения задачи обнаружения разладки в разнообразных данных.
- Провести эксперименты на обширных данных с различными характеристиками, включая различные распределения и размеры выборок.

Дальнейшие задачи:

- Рассмотреть реализацию для онлайн версии алгоритма
- Анализ чувствительности метода к различным распределениям и размерам выборок.

2 Обзор

В этом разделе проведен обзор существующих методов обнаружения точек изменения[4]. А также были рассмотрены различные аспекты работы с графами, включая методы их подсчета и хранения.

2.1 Постановка задачи

Рассматривается последовательность наблюдений y_i где $i = 1, \dots, n$, упорядоченных по некоторому осмысленному признаку, например времени. Основная задача заключается в проверке нулевой гипотезы, которая заключается в том что у нас нет точек изменений:

$$H_0 : y_i \sim F_0, i = 1, \dots, n \quad (1)$$

А так же у нас есть альтернативная гипотеза которая говорит о том что у нас есть одна точка изменения:

$$H_1 : \exists 1 \leq \tau < n, y_i \sim \begin{cases} F_1, & i > \tau \\ F_0, & \text{иначе} \end{cases} \quad (2)$$

2.2 Существующие методы обнаружения разладок

В современных исследованиях активно разрабатываются методы обнаружения разладок в различных областях.

Параметрические методы:

Параметрические методы предполагают, что данные подчиняются определенному параметрическому распределению. Это означает, что форма распределения данных полностью определяется набором параметров, таких как среднее значение и стандартное отклонения. К ним относятся:

- Probability density ratio.
- Subspace models.
- Probabilistic method.

- Binary class classifier.

Непараметрические методы:

Непараметрические методы не предполагают заранее определённой формы распределения данных. Они являются менее зависимыми от конкретных предположений о структуре данных. Таковыми методами являются:

- Kernel-based methods.
- Multi-class classifier.
- Graph-based methods.

Преимущества непараметрических методов включают их способность обрабатывать данные с неизвестным или сложным распределением, а также более широкую применимость в условиях, когда предположения о структуре данных трудно сделать.

В данной работе был выбран графовый метод (Graph-based methods) в виду его отсутствия в предшествующих реализаций исследований. Это позволяет внести новый вклад в область обнаружения разладки и проверить эффективность графового метода. В то время как другие методы такие как Probability density ratio уже нашли своё применение в других реализациях.

2.3 Графовые методы в обнаружении разладок

Современные графовые методы для обнаружения разладок предоставляют обширные возможности в анализе данных. В данном разделе мы рассмотрим перспективы использования графовых методов.

2.3.1 Принцип работы

Графовые методы строят представление данных в виде графа, где узлы представляют наблюдения, а ребра отражают взаимосвязи между ними. Ключевой шаг метода заключается в определении сетевых характеристик.

Далее процесс включает в себя следующие шаги:

1. *Вычисление математического ожидания с дисперсией:*

$$E(R_G(t)) = p_1(t)|G|, \quad (3)$$

$$Var(R_G(t)) = p_1(t)|G| + \left(\frac{1}{2}p_1(t) - p_2(t)\right) \sum_i |G_i|^2 + (p_2(t) - p_1^2(t)) |G|^2, \quad (4)$$

где p_1 (5) и p_2 (6) это:

$$p_1(t) = \frac{2t(n-t)}{n(n-1)}, \quad (5)$$

$$p_2(t) = \frac{4t(t-1)(n-t)(n-t-1)}{n(n-1)(n-2)(n-3)}, \quad (6)$$

В процессе детального анализа формулы с научным руководителем и консультантом была выявлена опечатка в формуле для дисперсии(7) и внесены соответствующие коррекции.

$$Var(R_G(t)) = p_2(t)|G| + \left(\frac{1}{2}p_1(t) - p_2(t)\right) \sum_i |G_i|^2 + (p_2(t) - p_1^2(t)) |G|^2, \quad (7)$$

2. *Оценка сканирующей статистики:*

С использованием полученных параметров, рассчитывается сканирующая статистика по заданной формуле:

$$Z_G(t) = -\frac{Rg(t) - E[Rg(t)]}{\sqrt{Var[Rg(t)]}}, \quad (8)$$

3. *Сравнение со значением порога:*

Рассчитанное значение ($Zg(t)$) сравнивается с пороговым значением (threshold). Если $Zg(t)$ превосходит пороговое значение, это сигнализирует о выявлении точки разладки в данных.

2.3.2 Преимущества графовых методов

Одним из существенных преимуществ графовых методов является возможность визуализации изменений в структуре данных. Графовые методы обладают высокой гибкостью в моделировании различных типов данных. Возможность представления данных в виде графа позволяет адаптировать методы под конкретные особенности сетевых структур.

2.3.3 Ограничения графовых методов

Однако существуют и определённые ограничения при применении графовых методов. Прежде всего, сложность анализа графов и высокая вычислительная нагрузка могут стать барьером для их эффективного использования.

3 Реализация

В данном разделе представлены ключевые компоненты разработанного инструмента для обнаружения разладок с использованием графового подхода.

Разработанный инструмент является частью обширного проекта по статистическому анализу на языке программирования Python. Этот проект охватывает различные аспекты статистических методов и алгоритмов, предоставляя возможность исследования каждого компонента по отдельности.

Для реализации данного инструмента был выбран язык программирования Python и библиотека NumPy. Python предоставляет широкие возможности для эффективной разладки и отладки, а NumPy обеспечивает высокую производительность при операциях над массивами данных.

Процесс реализации включает в себя построение графа, анализ его структуры, а также расчет сканирующей статистики для точного обнаружения изменений. Рассмотрим каждый этап подробнее.

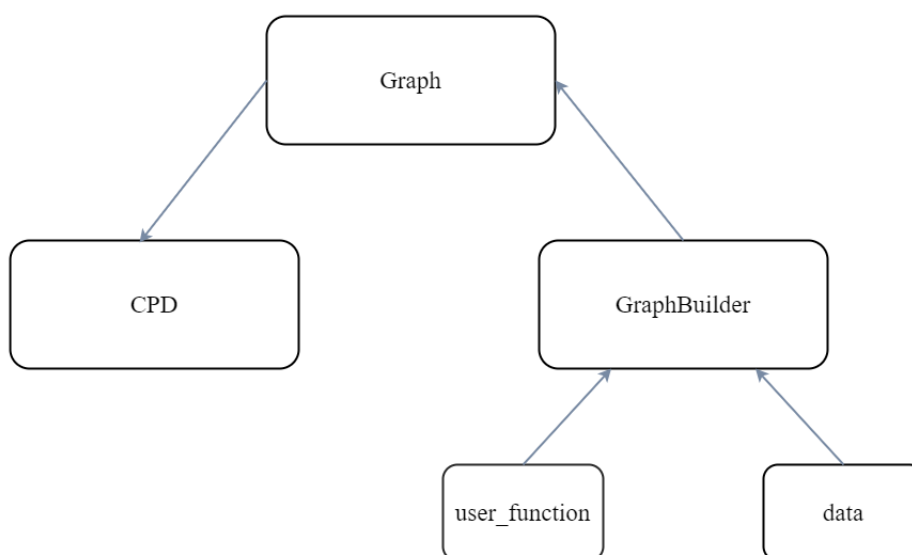


Рис. 1: Graph Based method

3.1 Построение графа

Для представления данных в виде графа был разработан класс “GraphBuilder” который обеспечивает построение графа на через матрицу смежности или списка смежности. Также граф строится на основе минимального остовного дерева (MST), который служит основой для дальнейших вычислений.

Матрица смежности

Плюсы:

- Простота добавления и удаления ребра между двумя узлами.
- Простота определения наличия ребра между двумя узлами
- Простота представления

Минусы:

- Затраты по памяти. Матрица смежности может потреблять значительное количество памяти, особенно при хранения большого числа узлов и связей

Список смежности

Плюсы:

- Список смежности эффективен с точки зрения хранения, потому что нам нужно хранить значения только для рёбер. Для разреженного графа с миллионами вершин и рёбер это может означать большую экономию места.
- Простота нахождения всех вершин, смежных с вершиной

Минусы:

- Поиск в списке смежности не происходит быстрее, чем в матрице смежности, так как сначала необходимо исследовать все соединенные узлы для их обнаружения

Оптимальное построение графа является важным звеном, поскольку от выбора структуры графа зависит точность и качество дальнейшей работы. Класс “GraphBuilder” позволяет выбирать оптимальный способ представления графа. Использование матрицы смежности обеспечивает быстрый доступ к связям между узлами, в то время как список смежности экономит память при работе с разреженными графами

3.2 Вычисление сетевых характеристик

Важным этапом в разработке метода обнаружения разладок с использованием графового подхода является вычисления сетевых характеристик, представляющих собой ключевые элементы анализа структуры данных. Класс “Graph” в нашей реализации содержит методы специально разработанные для вычисления различных сетевых характеристик:

1. *Степени вершин*
2. *Общее количество ребер*
3. *Количество ребер, соединяющих узлы из разных групп*

Сетевые характеристики предоставляют информацию о взаимосвязи. Уникальной особенностью нашего подхода является возможность представления данных в виде графа. Эти характеристики, вычисленные с использованием класса “Graph”, становятся основой для последующих этапов обнаружения разладок.

3.3 Обнаружение разладок

Основной этап - обнаружения разладок в данных. Класс CPD (Change point detection) тщательно реализует методы, позволяющие выявить периоды статистически значимых изменений в сетевых характеристиках. Это включает в себя вычисление scan-статистики и определения точек изменений.

Реализация алгоритма обнаружения разладок включает в себя тщательный анализ временных рядов с использованием сканирующей статистики, что позволяет выявлять изменения в структуре данных.

Процесс сканирования включает в себя вычисления метрик и последующее сравнение её с пороговым значением. Сканирующая статистика представляет собой меру отклонения сетевых характеристик, вычисляется согласно формуле Нео Чен $Z_G(t)$ (8)

Таким образом, если оценка превышает пороговое значение, это сигнализирует о статистически значимых изменениях в сетевых характеристиках, что определяет момент времени как точку разладки. Подробное подтверждение этого критерия будет представлено в ходе проведения экспериментов, где проверим чувствительность данного метода на различных распределениях и выборках

Для более детального взгляда на реализацию, вы можете ознакомиться в данной репозитории <https://github.com/Lesh79/Change-point-detection><https://www.overleaf.com/project/657c3d023d5ba4575e019b61>

4 Эксперимент

4.1 Дизайн эксперимента

Для оценки эффективности разработанного метода обнаружения разладок был проведен эксперимент на различных типах данных с различными характеристиками.

Первая группа данных: В этой группе значения были сгенерированы из одного распределения, где известно что точно нет точек изменений.

Вторая группа данных: Вторая группа представляла собой данные, в которых объединены два различных распределения и в которых точно присутствует разладка

Гипотеза: Разработанный метод обладает высокой чувствительностью к различным распределениям, а также размеру выборки, и специфичностью при обнаружении разладок в данных различных характеристик.

Характеристики используемого оборудования и программного обеспечения:

- Python 3.11
- NumPy 1.26.2
- Ноутбук (11th Gen Intel(R) Core(TM) i5-11400H, NVIDIA GeForce RTX 3050 Ti Laptop GPU, 16gb RAM)

4.2 Результаты эксперимента

В ходе каждого эксперимента, была проведена генерация каждой группы данных 1000 раз. В каждом наборе выбиралась максимальная scan-статистика. После этого строились гистограммы максимальных значений для каждой группы.

Пример генерации данных:

,

```

1 import numpy as np
2
3 without_change_point = normal(loc=5, scale=1, size=400)
4
5 def combination_data():
6     normal_set = np.random.normal(loc=5, scale=1, size = 150)
7     uniform_set = np.random.uniform(low=4, high=5, size=150)
8     dataset = np.concatenate([normal_set, uniform_set], axis = 0)
9     return dataset

```

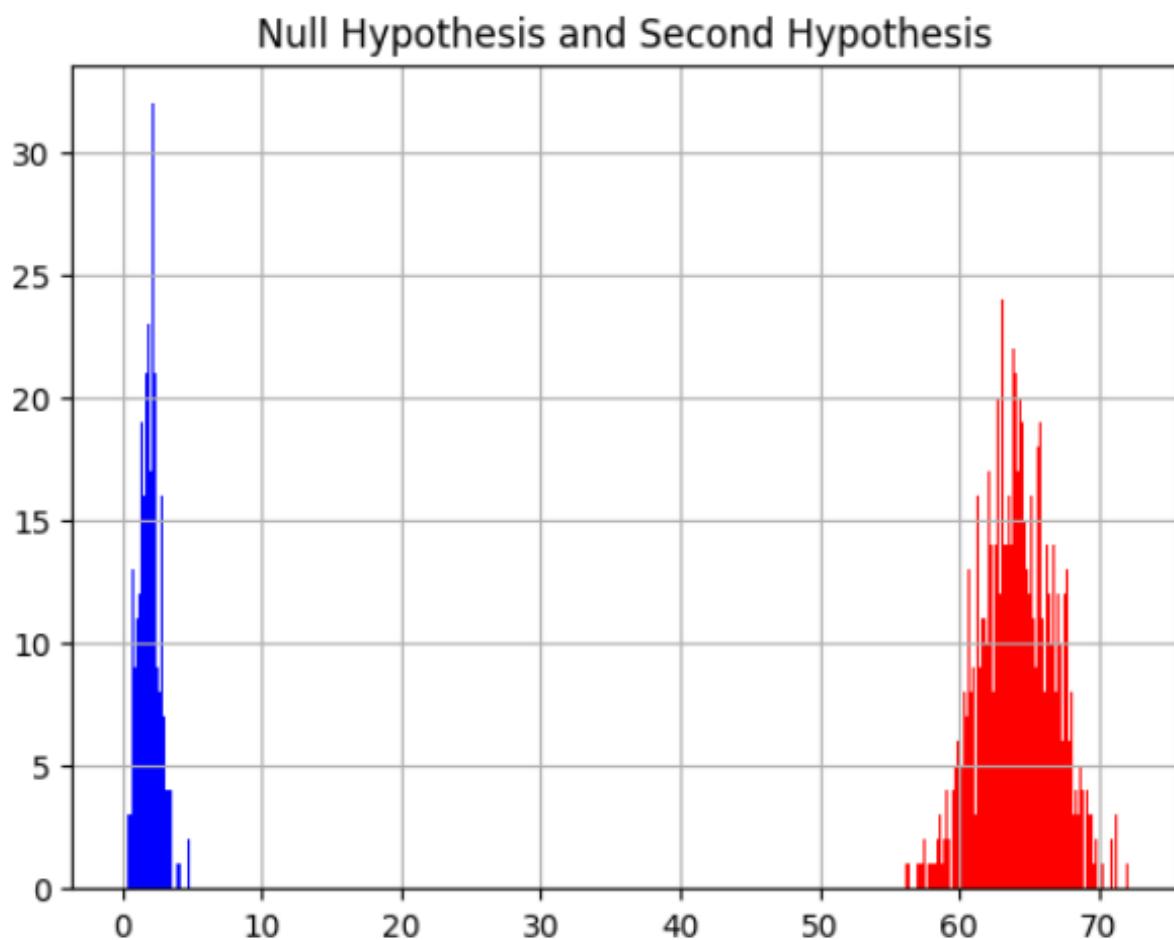


Рис. 2: Good results

Анализ результатов позволяет сделать вывод о том, что при данных распределениях и размерах выборок у нас имеется хорошая разделимость.

Рассмотрим другой пример данных:

,

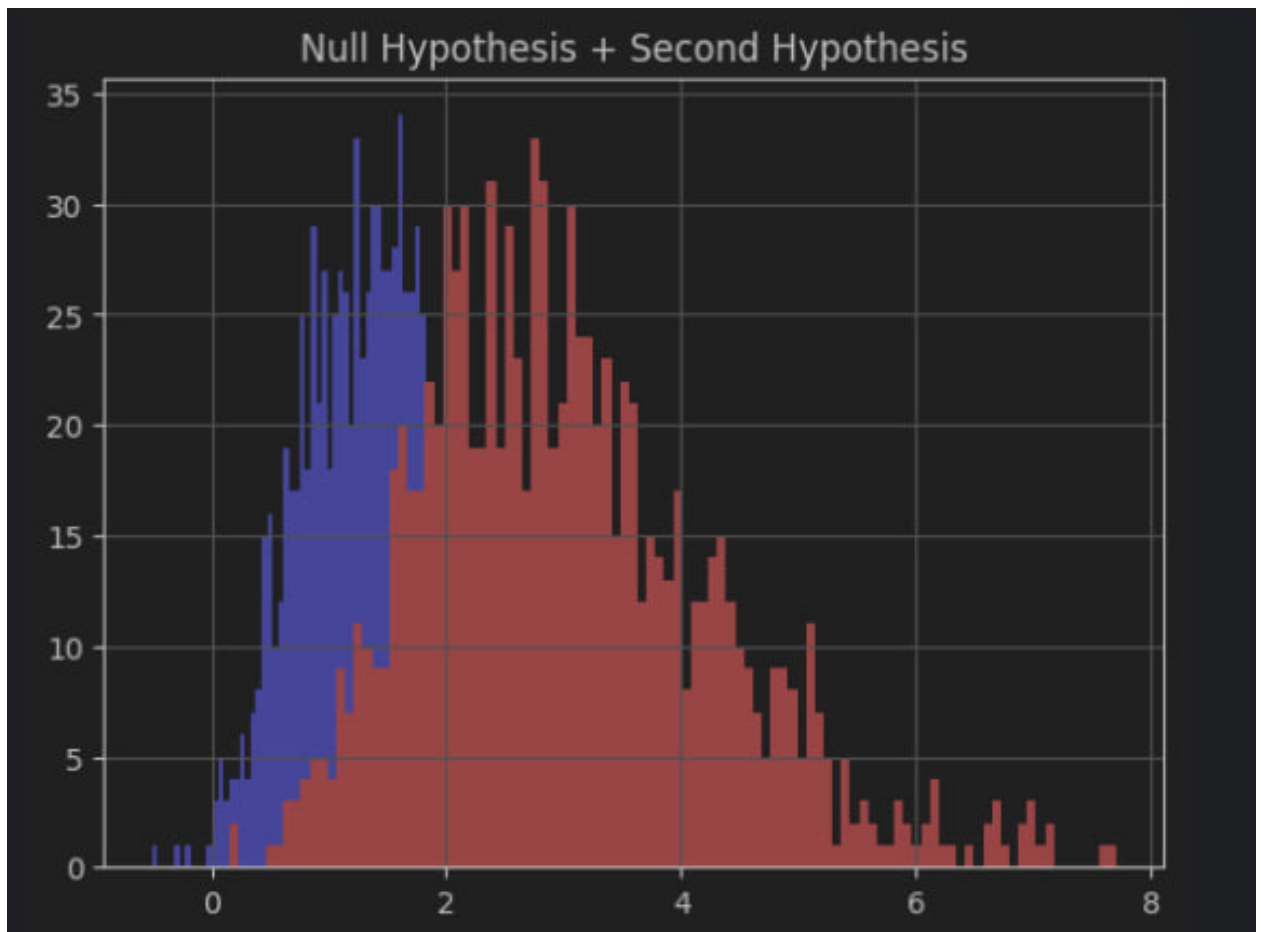


Рис. 3: Bad results

```

1 import numpy as np
2
3 without_change_point = gamma(shape=2, scale=2, size=50)
4
5 def combination_data():
6     exponential_set = np.random.exponential(scale=1, size=50)
7     normal_set = np.random.normal(low=1, high=10, size=150)
8     dataset = np.concatenate([exponential_set, normal_set], axis=0)
9     return dataset

```

В данном примере мы наблюдаем, что при маленьком размере выборки в объединении данных возникают существенные неточности в определении разладки.

4.3 Обобщение результатов

Обобщая результаты эксперимента, можно утверждать, что разработанный метод обнаружения точек изменений демонстрирует хорошую разделимость на разнообразных данных. Однако стоит отметить, что разделимость сильно зависит от характера распределений и выборки в данных и момента объединения данных. Это обстоятельство подчеркивает важность адаптации данного метода под конкретные условия и подразумевает возможные ограничения метода при работе с различными типами данных.

Заключение

В ходе данной работы были выполнены следующие задачи:

- Проанализированы современные исследования, посвященные использованию графовых методов в задачах обнаружения разладки, что позволило освоить актуальные тенденции в этой области.[1, 4]
- Реализован графовый метод для обнаружения разладки в данных. Результатом является инструмент, способный выявлять статистически значимые изменения в структуре данных.
- Проведены эксперименты на обширных данных с различными характеристиками, такими как различные распределения и размеры выборок.

Список литературы

- [1] Hao Chen Nancy Zhang. Graph-based change-point detection. — 2015. — URL: <https://www.projecteuclid.org/journals/annals-of-statistics/volume-43/issue-1/Graph-based-change-point-detection/10.1214/14-AOS1269.full> (дата обращения: 20 декабря 2023 г.).
- [2] Lai Tze Leung. Sequential Changepoint Detection in Quality Control and Dynamical Systems. — 1995. — URL: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1995.tb02052.x> (дата обращения: 20 декабря 2023 г.).
- [3] M. Staudacher S. Telser A. Amann H. Hinterhuber M. Ritsch-Marte. A new method for change-point detection developed for on-line analysis of the heart beat variability during sleep. — 2005. — URL: <https://www.sciencedirect.com/science/article/abs/pii/S0378437104013640> (дата обращения: 20 декабря 2023 г.).
- [4] Samaneh Aminikhanghahi Diane J. Cook. A survey of methods for time series change point detection. — 2017. — URL: <https://link.springer.com/article/10.1007/s10115-016-0987-z> (дата обращения: 20 декабря 2023 г.).
- [5] Sayantan Banerjee Kousik Guhathakurta. Change-point analysis in financial networks. — 2020. — URL: <https://www.sciencedirect.com/science/article/abs/pii/S0378437104013640> (дата обращения: 20 декабря 2023 г.).