МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

Тема работы

КУРСОВАЯ РАБОТА
по дисциплине «Программная инженерия»
ЮУрГУ – 09.03.04.2023.308-066.КР

Нормоконтролер, профессор
кафедры СП

_____ М.Л. Цымблер

“____”_____ 2024 г.

Научный руководитель
доктор физ.-мат. наук

_____ М.Л. Цымблер

Автор работы,
студент группы КЭ-303

_____ А.В. Толмачева

Работа защищена
с оценкой: _____

“____”_____ 2024 г.

Челябинск-2024

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
“____”_____ 2024 г.

ЗАДАНИЕ
на выполнение выпускной курсовой работы
по дисциплине «Программная инженерия»
студенту группы КЭ-303
Толмачевой Анастасии Вячеславовне,
обучающемуся по направлению
09.03.04 «Программная инженерия»

1. **Тема работы**
   Разработка системы для выявления фиктивных аккаунтов Open Journal System.
2. **Срок сдачи студентом законченной работы:** 31.05.2024.
3. **Исходные данные к работе**
   3.1. Open Journal Systems. [Электронный ресурс]. URL: https://openjournalsystems.com/ (дата обращения 18.09.2023)
   3.2. Кластеризация . [Электронный ресурс]. URL: https://scikit-learn.ru/clustering/ (дата обращения 18.09.2023)
   3.3. Rokach L., Maimon O. Clustering Methods // The Data Mining and Knowledge Discovery Handbook, 2005. - 321-352 pp. DOI: 10.1007/0-387-25465-X_15
   3.4. Breuer A., Khosravani N., Tingley M., Cottel B. Preemptive Detection of Fake Accounts on Social Networks via Multi-Class Preferential Attachment Classifiers // KDD ’23: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023. – 105–116 pp. DOI: 10.1145/3580305.3599471
4. **Перечень подлежащих разработке вопросов**
   4.1. Анализ предметной области и литературы по теме работы.
   4.2. Разработка алгоритма выявления фиктивных аккаунтов.
   4.3. Проектирование интерфейса программной системы и модульной структуры приложения.
   4.4. Реализация программной системы, выявляющей фиктивные аккаунты, на основе разработанного алгоритма.
   4.5. Подготовка набора тестов и тестирование программной системы.

5. **Дата выдачи задания:** ____”_____ 2024 г.

Научный руководитель М.Л. Цымблер

Задание принял к исполнению А.В. Толмачева

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы

Интернет в наши дни развиваются быстро. Это очень важно для маркетинговых компаний и знаменитостей, которые желают увеличить количество покупателей и фанатов; для обычных пользователей, которые быстрее узнают новости и находят новых знакомых; для исследователей, которые делятся последними открытиями и актуальной информацией. Социальные сети делают нашу жизнь лучше, но есть некоторые аспекты, которые требуют внимания.

Вместе с расширением онлайн-сообществ возникает проблема поддельных аккаунтов. «Фейки» могут быть созданы с разными целями: получить коммерческую выгоду, дискредитировать настоящего пользователя, заполучить личную информацию, осветить вредный контент и так далее [2]. Иногда их тяжело отличить от аккаунта живого человека, а массовость проблемы может набирать социально-опасный характер: например, «боты», которые сыграли роль в выборах президента [10].

К сожалению, поддельные аккаунты могут встречаться и в среде научных публикаций. Open Journal System не защищена от появления в ней ненастоящих пользователей, что может вызывать различные трудности при поиске научных статей и обмене исследованиями. Это может играть роль во время написания новых публикаций, обучения, преподавания – «фейки», распространяющие ложную информацию, подрывают доверие пользователей к ресурсам в Интернете.

Технический прогресс, особенно в области машинного обучения и анализа данных, предоставляет инструменты для обнаружения поддельных аккаунтов. Алгоритмы машинного обучения могут быть применены для выявления аномального поведения аккаунтов, их классификации как фиктивных [9].

Таким образом, разработка системы для выявления фиктивных аккаунтов в Open Journal System становится актуальной и востребованной, учитывая масштаб проблемы появления «ботов» в Интернете и технический прогресс в области искусственного интеллекта и машинного обучения.

Цель и задачи

Целью курсовой работы является реализация программной системы, выявляющей фиктичные аккаунты в системе Open Journal Systems. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области и литературы по теме работы.
2. Разработать алгоритм выявления фиктивных аккаунтов.
3. Спроектировать интерфейс программной системы и модульной структуры приложения.
4. Реализовать программную систему, выявляющую фиктивные аккаунты, на основе разработанного алгоритма.
5. Подготовить набор тестов, выполнить тестирование программной системы.

Структура и содержание работы

Написать разделы, из которых состоит курсовая работа.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В разделе 1.1 представлен обзор предметной области проекта – что такое Open Journal System и что представляет аккаунт в ней. Далее в рассмотрены аналогичные решения, их описание расположено в разделе 1.2. Методы, которые могут использоваться для поиска фиктивных аккаунтов описаны в разделе 1.3. Обзор набора данных для обучения, принципы разметки и примеры аккаунтов из датасета можно увидеть в разделе 1.4.

1.1. Предметная область

Open Journal System (OJS) – это программное обеспечение, которое позволяет публиковать статьи и организовать рабочий процесс издательства. На ее основе разработаны многие порталы, работают институты, научные центры и журналы в разных странах мира (интерфейс OPS переведен более чем на 30 языков). Платформа обладает модульной структурой и имеет возможность подключения плагинов.

OJS может быть рассмотрена как электронная библиотека: программа обеспечивает доступ к контенту, поиск по нему (автора, ключевые слова, названия статей, год выпуска и так далее).

Аккаунт в Open Journal Systems представляет собой учетную запись пользователя, которая позволяет взаимодействовать с системой в качестве автора, рецензента, читателя или редактора в научных журналах, использующих OJS для управления процессом публикации.

Авторы могут создавать и отправлять свои научные статьи на публикацию в журнал. Их аккаунт позволяет им отслеживать статус статей, вносить изменения в данные и взаимодействовать с рецензентами.

Рецензенты имеют доступ к предоставленным им статьям. Их аккаунт дает им возможность отправлять свои оценки, взаимодействовать с редакцией и следить за обновлениями в процессе рецензирования.

Читатели могут создавать свои учетные записи для отслеживания интересующих статей, подписываться на уведомления о новых публикациях, а также участвовать в дискуссиях в комментариях к статьям.

Редакторы имеют полный доступ к управлению процессом публикации в журнале. Это включает в себя принятие и отклонение статей, управ-

ление рецензиями, управление составом редакционной коллегии и другие аспекты редакционной работы.

Администраторы имеют права доступа ко всем функциям системы и могут управлять пользователями, настройками, архивами и другими аспектами системы.

Поля, доступные в учетной записи пользователя OJS, могут немного различаться в зависимости от версии системы и настроек конкретного журнала, но учетные записи в обычно включают следующие базовые характеристики: имя пользователя, пароль, электронная почта, имя, фамилия, страна, аффилиация (принадлежность пользователя к организации, университету или другому учреждению), язык, ссылка, телефон, почтовый адрес.

1.2. Обзор аналогичных проектов

Нахождение фиктивных аккаунтов в Open Journal System является темой данной работы. Для выбора наилучшего подхода к решению задач и достижения цели были рассмотрены некоторые статьи на тему выявления «фейков» и их влияния на аккаунты реальных людей. Далее представлены исследования, которые были использованы в ходе работы.

Авторы статьи [9] раскрывают проблему поддельных страниц. Их количество растет вместе с увеличением числа активных пользователей. Поддельные профили на сайтах социальных сетей создают ненастоящие новости и распространяют нежелательные материалы, содержащие спам-ссылки. В этой статье приводится контролируемый алгоритм машинного обучения, называемый машиной опорных векторов, который используется вместе с методом случайного леса. Эту концепцию можно применить для идентификации большого количества учетных записей, которые невозможно проверить вручную. Данная модель сравнивается с другими методами идентификации, и результаты показывают, что предложенный авторами алгоритм работает с большей точностью.

Статья [5] посвящена выявлению поддельных профилей в социальных сетях. Для их обнаружения было предложено множество алгоритмов и методов, и авторы этой работы оценивают точность использования де-

рева решений и наивного алгоритма Байеса для классификации профилей пользователей на поддельные и подлинные.

В работе [21] исследователи используют модель, которая определяет подлинность новостных статей, публикуемых в Twitter, на основе подлинности и предвзятости пользователей, которые взаимодействуют с этими статьями. Предлагаемая модель включает в себя идею оценки соотношения подписчиков, возраст аккаунта и т.д. Для анализа изображений предлагается использовать нейронную сеть (CredNN). Подход исследователей помогает обнаруживать поддельные изображения с точностью около 76%.

В исследовании [8] проведен анализ 62 миллионов общедоступных профилей пользователей социальной сети Twitter, и разработана стратегия идентификации поддельных профилей. Используя алгоритм сопоставления шаблонов имен, анализ времени обновления твитов и даты создания профилей, были выявлены фиктивные учетные записи пользователей.

В статье [3] описывается новый алгоритм для обнаружения поддельных учетных записей в социальной сети. Авторы работы собирают некоторые из первых аналитических данных о том, как новые (поддельные и реальные) аккаунты пытаются завести друзей, ориентируясь на их первые запросы о дружбе после регистрации в социальной сети (Facebook).

В работе [13] авторы представляют метод обнаружения, основанный на сходстве пользователей, с учетом их сетевых коммуникаций. На первом этапе измеряются такие параметры, как общие соседи, ребра графа общих соседей, косинус и коэффициент подобия Жаккарда [16], которые вычисляются на основе матрицы смежности соответствующего графа социальной сети. На следующем шаге, чтобы уменьшить сложность данных, к каждой вычисленной матрице подобия применяется компонентный анализ для получения набора информативных признаков. Затем с помощью метода локтя выбирается набор высокоинформативных собственных векторов. Извлеченные функции используются для обучения алгоритма классификации.

В статье [19] представлено исследование поддельных аккаунтов в социальных сетях с использованием искусственной нейронной сети для их идентификации. Специально разработанное и внедренное приложение

было использовано для выявления специфических особенностей поддельных аккаунтов и изучения принципов и причин их генерации. На основе изучения 500 реальных и 500 поддельных аккаунтов социальносй сети ВКонтакте был сделан ряд выводов об особенностях поддельных аккаунтов. Проведенное исследование позволило расширить список критериев идентификации поддельных аккаунтов набором шаблонов.

В статье [4] авторы нацеливаются на модель фиктивного аккаунта, который может не только автоматически публиковать сообщения или комментарии, но и рассылать рекламный спам или распространять ложную информацию. В этом исследовании был предложен метод обнаружения «фейков». Он основан на шаблоне активности пользователя в Facebook с использованием машинного обучения, чтобы предсказать, контролируется ли учетная запись поддельным пользователем.

Работа [10] представляет результаты экспериментальной визуализации более 200 000 твитов из подтвержденных поддельных аккаунтов Twitter. Авторы анализируют учетные записи пользователей, изучая их имена пользователей, описания или биографии, твиты, их частоту и содержание. Исследователи обнаружили, что поддельные учетные записи узнаваемы благодаря политическим и религиозным убеждениям. Они присоединялись к популярным хэштегам в Twitter и публиковали твиты в острые моменты (например, во время дебатов).

Авторы статьи [7] рассматривают влияние фиктивных аккаунтов на аккаунты людей. Цель работы – представить новую модель распространения информации. Исследователи вводят два типа пользователей с разным уровнем «зараженности»: пользователи, «инфицированные» человеком, и пользователи, «зараженные» учетными записями ботов. Было измерено влияние поддельных аккаунтов на скорость распространения лжи среди записей людей. Результаты эксперимента показывали, что точность предложенной модели превосходит классическую при моделировании процесса распространения слухов. Был сделан вывод, что «фейки» ускоряют процесс распространения невподтвержденной информации, поскольку они воздействуют на многих людей за короткое время.

1.3. Обзор основных методов, используемых для поиска аномалий

Методы, которые могут использоваться для поиска аномалий:

Isolation Forest (лес изоляции) [11] представляет собой алгоритм обнаружения аномалий, который строит ансамбль изолирующих деревьев решений.

Дерево изоляции. Пусть T – узел дерева изоляции. T может быть либо внешним узлом без дочерних узлов, либо внутренним узлом с одним тестом и ровно двумя дочерними узлами (T_l, T_r). Тест состоит из атрибута q и значения разделения p, такого, что тест $q < p$ разделяет точки данных на T_l и T_r.

Дана выборка данных $X = \{x_1, \ldots, x_n\}$ из n экземпляров из d-мерного распределения. Для построения дерева изоляции мы рекурсивно делим X, случайно выбирая атрибут q и значение разделения p, пока дерево не достигнет предельной высоты, либо $|X| = 1$, либо все данные в X будут иметь одинаковые значения. Дерево изоляции – это правильное бинарное дерево, где каждый узел в дереве имеет ровно ноль или два дочерних узла.

Одним из способов обнаружения аномалий с помощью дерева изоляции является сортировка точек данных по их длинам пути или оценкам аномалий; и аномалии – это точки, которые занимают верхние позиции в списке. Длина пути $h(x)$ точки x измеряется количеством рёбер, которые x проходит в дереве изоляции от корневого узла до завершения обхода во внешнем узле.

Работа алгоритма леса изоляции:

1. Случайное разбиение. Дерево случайным образом выбирает атрибут и значение разделения данных. Это создает разделение, которое помогает выделить аномалии.
2. Рекурсивная изоляция. Дерево повторяет процесс, разделяя данные на более мелкие группы. Цель состоит в том, чтобы изолировать аномалии от обычных данных.
3. Идентификация аномалий. Аномалии идентифицируются как точки данных, для выделения которых требуется меньшее количество раз-

биений.

4. Определение изолирующего пути. Определяется количество разбиений, необходимых для изоляции точки данных, служит мерой её аномальности.
5. Ансамбль деревьев. Алгоритм создает несколько деревьев изоляции, независимых друг от друга, образующих ансамбль, который коллективно оценивает аномалии.
6. Расчет оценки различий. Вычисляется среднее расстояние разделения между всеми деревьями для каждой точки данных, что дает оценку аномалии.
7. Классификация. Используются предопределенные пороги для отличения нормальных и аномальных точек данных. Объекты с высокими оценками помечаются как аномалии.

Hierarchical clustering (иерархическая кластеризация) – это алгоритм группировки данных, который строит иерархию кластеров. Существует два основных подхода: агломеративный и дивизивный.

Агломеративная кластеризация. Каждая точка рассматривается как отдельный кластер. На каждом шаге ближайшие кластеры объединяются в новый кластер (на основании расстояния между кластерами или другими мерами сходства). Повторяется до тех пор, пока все точки не объединятся в один кластер.

Дивизивная кластеризация. Вся выборка рассматривается как один кластер. На каждом шаге один из кластеров разбивается на более мелкие кластеры (на основании мер несходства между подгруппами). Процесс повторяется до тех пор, пока каждая точка не станет отдельным кластером.

Результат иерархической кластеризации представляет собой дерево, называемое дендрограммой. Дендрограмма отображает последовательное объединение или разделение кластеров в зависимости от расстояний или мер сходства.

В данной работе рассматривался алгоритм агломеративной кластеризации [1]. Можно выделить несколько методов вычисления связи.

Метод одиночной связи (single linkage): расстояние между кластера-

ми равно минимальному расстоянию между точками из разных кластеров.

$$d_{\min}(C_i, C_j) = \min_{x \in C_i, y \in C_j} \rho(x, y)$$

Метод полной связи (complete linkage): расстояние между кластерами равно максимальному расстоянию между точками из разных кластеров.

$$d_{\max}(C_i, C_j) = \max_{x \in C_i, y \in C_j} \rho(x, y)$$

Метод средней связи (average linkage): расстояние между кластерами равно среднему расстоянию между всеми парами точек из разных кластеров.

$$d_{\text{avg}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x \in C_i} \sum_{y \in C_j} \rho(x, y)$$

Метод Уорда (Ward's linkage): расстояние между кластерами равно приросту суммы квадратов расстояний от точек до центроидов кластеров при объединении этих кластеров. Этот метод стремится минимизировать внутрикластерную дисперсию.

$$d_{\text{ward}}(C_i, C_j) = \frac{n_i n_j}{n_i + n_j} \rho^2(\bar{x}_i, \bar{x}_j)$$

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [6] – это алгоритм кластеризации, который определяет кластеры на основе плотности данных в пространстве. Он может обнаруживать кластеры произвольной формы и выделять точки, не принадлежащие ни одному кластеру (шум).

Для нахождения кластера DBSCAN начинает с произвольной точки p и извлекает все точки, достижимые по плотности из p относительно Eps и $MinPts$. Если p является ядром кластера, этот процесс дает кластер относительно Eps и $MinPts$. Если p является граничной точкой, и нет точек, достижимых по плотности, и DBSCAN переходит к следующему объекту в базе данных.

Для алгоритма DBSCAN необходимо указывать параметры Eps и $MinPts$.

Параметр Eps определяет окрестности вокруг точек данных: если расстояние между двумя точками меньше или равно Eps, то они считаются соседними. Если значение Eps выбрано слишком маленьким, то большая

часть данных будет считаться шумом. Если же выбрано очень большое значение Eps, то кластеры объединятся, и большинство объектов окажется в одних и тех же кластерах.

Параметр $MinPts$ представляет собой минимальное количество соседей (точек данных) в радиусе Eps. Чем больше набор данных, тем большее значение $minPts$ следует выбрать.

Метрики

Для оценки качества работы алгоритма будут использоваться метрики accuracy (аккуратность), precision (точность) и recall (полнота), F-measure (F-мера). Первая показывает долю верно классифицированных объектов, вторая – долю объектов, которые модель классифицировала как положительные, и которые действительно являются положительными, а третья – долю объектов положительного класса, которые модель определила правильно, четвертая – это комбинированная метрика, объединяющая точность и полноту в единственное число.

Данные метрики можно вычислить по следующим формулам:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP + TN}{TP + FN}$$

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Термины TP, TN, FP, FN используются в контексте матрицы ошибок (таблица 1), которая является инструментом для оценки производительности моделей.

Таблица 1 – Матрица ошибок

	Positive	Negative
True	True Positives	True Negatives
False	False Positives	False Negatives

TP (True Positives) – количество верно предсказанных положительных примеров. Это случаи, когда модель правильно предсказала, что объект принадлежит к положительному классу. В данной работе это будут те аккаунты, которые являются фиктивными, и модель классифицировала их как аномальные.

TN (True Negatives) – количество верно предсказанных отрицательных примеров. Это случаи, когда модель правильно предсказала, что объект принадлежит к отрицательному классу. В данной работе это будут те аккаунты, которые являются реальными, и модель не классифицировала их как аномальные.

FP (False Positives) – количество ложно положительных примеров. Это случаи, когда модель ошибочно предсказала, что объект принадлежит к положительному классу, хотя на самом деле он принадлежит отрицательному классу. В данной работе это будут те аккаунты, которые являются реальными, но модель классифицировала их как аномальные.

FN (False Negatives) – количество ложно отрицательных примеров. Это случаи, когда модель ошибочно предсказала, что объект принадлежит отрицательному классу, хотя на самом деле он принадлежит положительному классу. В данной работе это будут те аккаунты, которые являются фиктивными, но модель не классифицировала их как аномальные.

1.4. Набор данных для обучения

Для обучения модели был применен датасет с реальными данными аккаунов, который имеет следующие столбцы: user_id, username, password, email, url, phone, mailing_address, billing_address, country, locales, date_last_email, date_registered, date_validated, date_last_login, must_change_password, auth_id, auth_str, disabled, disabled_reason, inline_help, gossip.

Была произведена его разметка с добавлением колонки fake, в которой 1 показывает, что аккаунт фиктивный, а 0 – настоящий. Разметка происходила по следующим признакам:

1. Проверка email на валидность (просмотр синтаксиса и возможности доставки на домен). Если почта не валидна, то акка-

унт считается фиктивным (например, аккаунт с id 4237 и email admin@cr6ptobrowser.site);

2. Проверка mailing_address на валидность (просмотр синтаксиса и возможности доставки на домен). Если почта не валидна, то аккаунт считается фиктивным (например, аккаунт с id 10666 и mailing_address trinapv1@sora81.sorataki.in.net);
3. Проверка работы ссылки, указанной в аккаунте. Если ссылка не работала, то аккаунт считается фиктивным (например, аккаунт с ссылкой http://viaforsl.com);
4. Проверка содержимого сайта, указанного в ссылке аккаунта. Если там находится что-либо, не связанное с научными публикациями, аккаунт считается фиктивным (например, аккаунт с id 3895 ссылкой на казино: https://vsem-zabor.ru);
5. Просмотр ближайших строк сверху и снизу по списку – если имена пользователей, email или billing_addres различаются на 1 – 5 символов, то такие аккаунты считаются фиктивными. Также если в характеристиках 2 и более аккаунтов встречается одинаковая часть (например, dtyekzneeplomo и tkbpfdtnfneeplomo, которые идут друг за другом, считаются фиктивными);
6. Проверка символов в имени пользователя, email и mailing_address: представляют они осмысленный текст или произвольный набор. В случаях, когда почта была валидна, но состояла из рандомного набора символов, проверялись дата последнего захода в аккаунт и дата регистрации. В случаях, когда они были в диапазоне суток друг от друга, аккаунт считается фиктивным (например, аккаунт с username ramonjap, email bvbxzroi@dimail.xyz и mailing_addressvljufcpn@dimail.xyz).

После разметки было выяснено, что в данном датасете из 540 записей 204 (38%) являются фиктивными, и 336 (62%) – настоящими. Примеры «фейков» и реальных аккаунтов рассмотрены в Приложении А.

1.5. Заключение

В данном разделе изучена предметная область курсовой работы (что такое Open Journal System, что является аккаунтом в данной системе), а также рассмотрены различные подходы к нахождению фиктивных аккаунтов, и какое влияние «фейки» оказывают. Можно сделать вывод, что из существующих решений, ни одно не затрагивало проблему нахождения фиктивных аккаунтов в Open Journal Systems.

Исходя из рассмотренных статей были выбраны три алгоритма для поиска аномалий и описаны метрики, с помощью которых будет оцениваться эффективность работы алгоритмов.

Был описан представленный датасет, с помощью которого производились эксперименты, представлены примеры размеченных записей.

2. ПРОЕКТИРОВАНИЕ

В разделе 2.1. описаны варианты использования системы. В разделе 2.2. представлены функциональные и нефункциональные требования к системе. Раздел 2.3 содержит модули системы. В разделе 2.4. показаны зарисовки графического интерфейса.

2.1. Варианты использования системы

Описание способов взаимодействия с системой, кто с ней может работать и каким образом отображено на рисунке 1.

Рисунок 1 – Диаграмма вариантов использования

Единственный актер, который может взаимодействовать с системой распознавания фиктивных аккаунтов в Open Journal System – это исследователь. Он может выполнять следующие действия:

1. Загрузить данные. Исследователь совершает выгрузку данных аккаунтов.

2. Редактировать данные. Исследователь может отредактировать данные, которые загрузил в систему.
3. Выполнить аугментацию. Исследователь может искусственно расширить имеющийся набор данных, сгенерировав новые, и привести выборку к равному количеству фиктивных и подлинных аккаунтов.
4. Выполнить настройку. Исследователь выполняет настройку того, какой будет выполняться алгоритм, с какими параметрами, в каком формате будут выводиться результаты.
5. Обучить модель. Исследователь может обучить модель.
6. Найти фиктивные аккаунты. Исследователь может найти «фейки» с помощью выбранного алгоритма.

Спецификация вариантов использования системы представлена в Приложении Б.

2.2. Требования к системе

Функциональные требования

Функциональные требования – это спецификация того, каким образом должно вести себя разрабатываемое программное обеспечение. Они определяют, какие конкретные функции и возможности должны быть включены в приложение, чтобы оно соответствовало ожиданиям пользователей. Эти требования описывают конкретные действия, которые система должна выполнять, и каким образом пользователь взаимодействует с приложением для достижения определенных целей.

Функциональные требования к системе:

1. Система должна предоставлять пользователю возможность загружать данные для исследования.
2. Система должна предоставлять пользователю возможность выполнить настройку ее работы: выбрать алгоритм, изменить его характеристики, добавить вывод результатов в графической форме.
3. Система должна предоставлять возможность обучать модель.
4. Система должна находить фиктивные аккаунты.

Нефункциональные требования

Ненфункциональные требования определяют свойства и характеристики, которыми должна обладать система. Эти требования касаются аспектов, не связанных напрямую с конкретными функциями, а определяют общие качественные аспекты, которые необходимы для функционирования системы.

Нефункциональные требования к системе:

1. Система должна быть написана на языке программирования Python.
2. Система должна работать с файлами формата csv.

2.3. Архитектура приложения

В данном разделе представлена архитектура системы. Она представляет из себя диаграмму компонентов, которая представлена на рисунке 2.

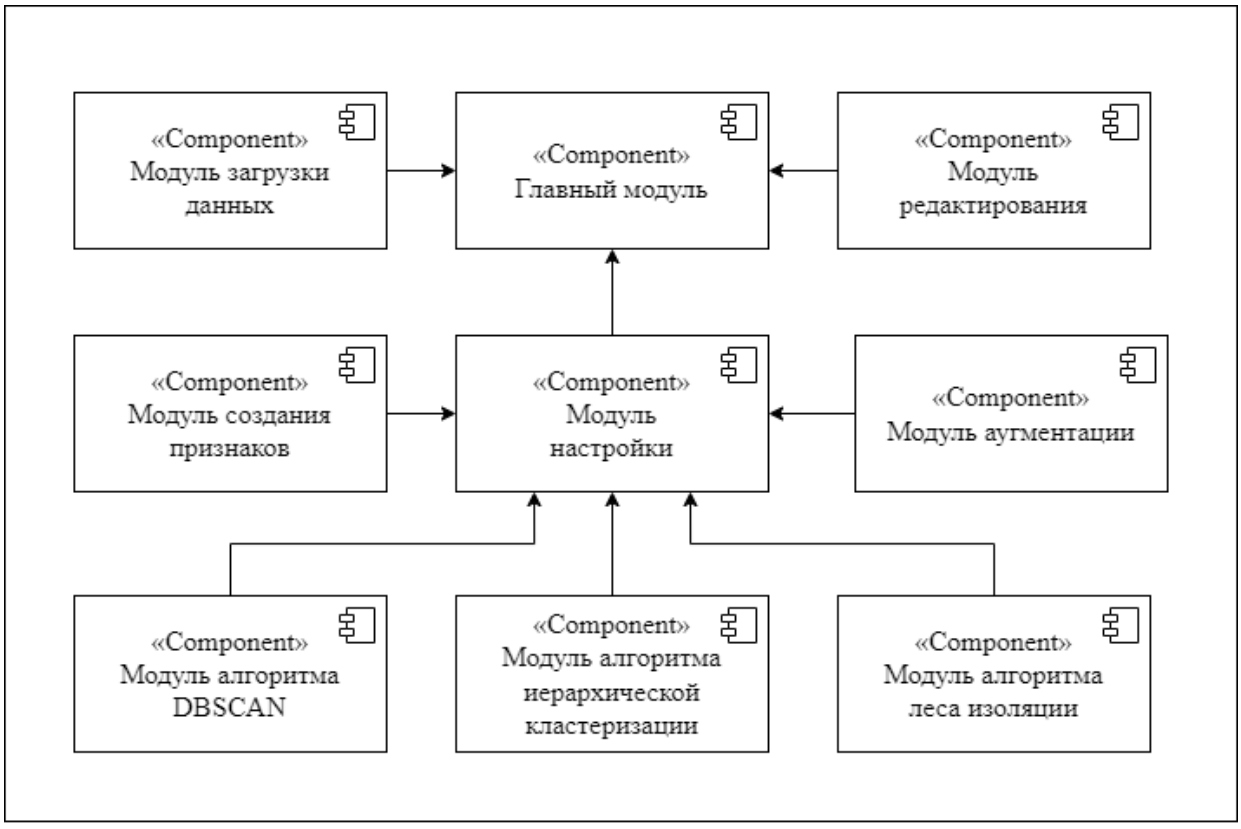

Рисунок 2 – Архитектура системы

Модули, представленные на схеме:

Модуль загрузки данных отвечает за работу с файлами, которые исследователь загружает в систему: открытие окна для выбора файла форма-

та CSV, добавление его в базу данных приложения.

Модуль редактирования предоставляет функционал, который позволяет исследователю редактировать загруженный датасет: изменять значение ячеек, добавлять и удалять столбцы и строки.

Модуль аугментации помогает исследователю сбалансировать набор данных, добавляя искусственно созданные записи на основе уже существующих. Результатом работы модуля является аугментированный датасет, содержащий равное количество фиктивных и настоящих аккаунтов. Компонент добавляет новые строки в конец таблицы и загружает ее в базу данных приложения.

Модуль настройки позволяет исследователю выполнить настройку алгоритмов, отображения результатов, сохранения файлов, а также найти «фейки». Данный компонент передает параметры в алгоритмы, вызывает модули создания признаков и выполняения аугментации.

Модуль создания признаков создает новую таблицу, на основе выбранной пользователем, которая форматирует текстовые данные датасета в числовые. Данная таблица загружается в новую базу данных.

Модуль алгоритма DBSCAN содержит алгоритм DBSCAN для нахождения аномалий в виде фиктивных аккаунтов, его визуализацию и вычисление метрик.

Модуль алгоритма иерархической кластеризации содержит алгоритм агломеративной кластеризации, подбор количества оптимальных кластеров для нее, визуализацию работы алгоритма (создание дендрограммы), а также вычисление метрик.

Главный модуль показывает исследователю, как работать с приложением, а также отвечает за запуск остальных окон приложения.

2.4. Графический интерфейс

В данном разделе представлены макеты модулей системы. Они являются примерными и содержат в себе основной функционал.

На рисунке 3 представлен главный экран системы. На нем располагаются кнопки, по которым происходит работа с приложением. При нажатии на кнопку «Загрузить данные» открывается форма для выбора файла

для загрузки, кнопка «Отредактировать данные» открывает новое окно, в котором можно выбрать таблицу для редактирования. По кнопке «Найти фейки» открывается раздел, в котором можно выбрать необходимые параметры алгоритмов и запустить их работу.

Рисунок 3 – Главный экран

Экран с редактированием данных представлен на рисунке 4. На нем располагается таблица с данными, а также кнопки, которые позволяют взаимодействовать с ней. Вверху экрана можно переключиться между загруженными данными, если пользователь добавлял несколько файлов. Также на макете можно увидеть кнопку «Сохранить», которая отвечает за сохранение изменений в данных.

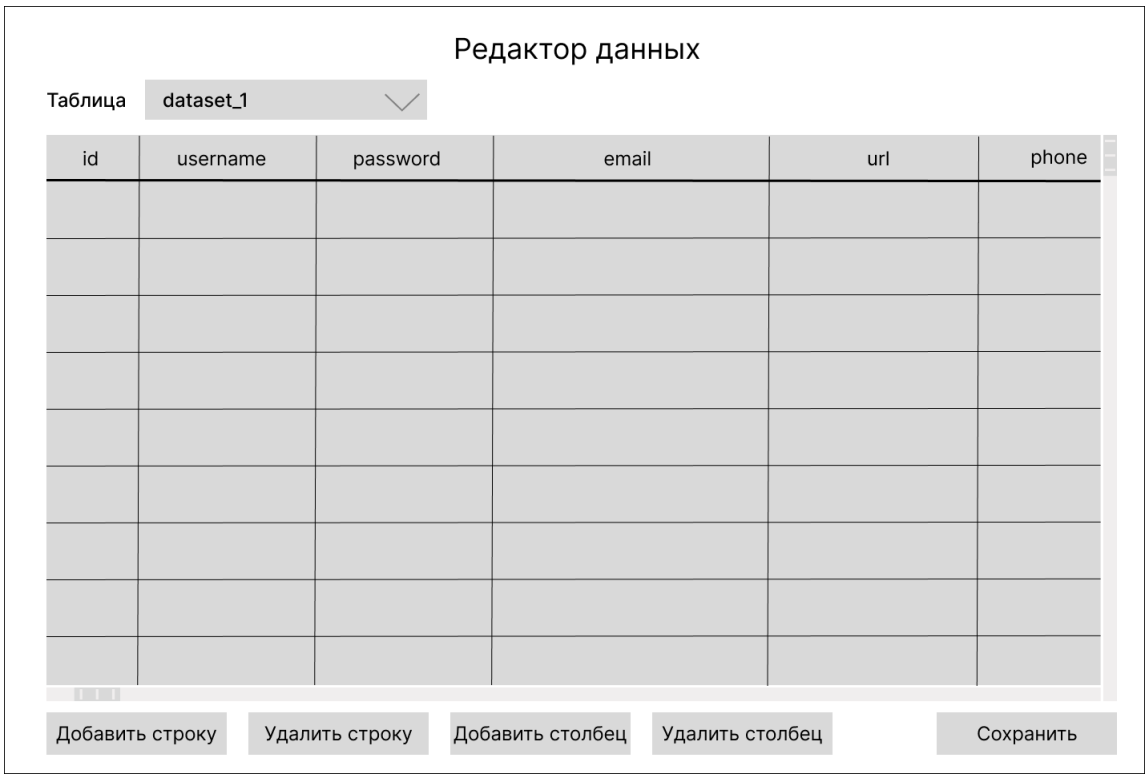

Рисунок 4 – Экран разметки

На рисунке 5 показан экран с настройками. На нем можно выбрать алгоритм, таблицу с данными, отображать результаты или нет, а также попку для вывода результатов. Помимо этого после выбора конкретного алгоритма будут появляться новые поля для настройки конкретного алгоритма (пример представлен на рисунке 6).

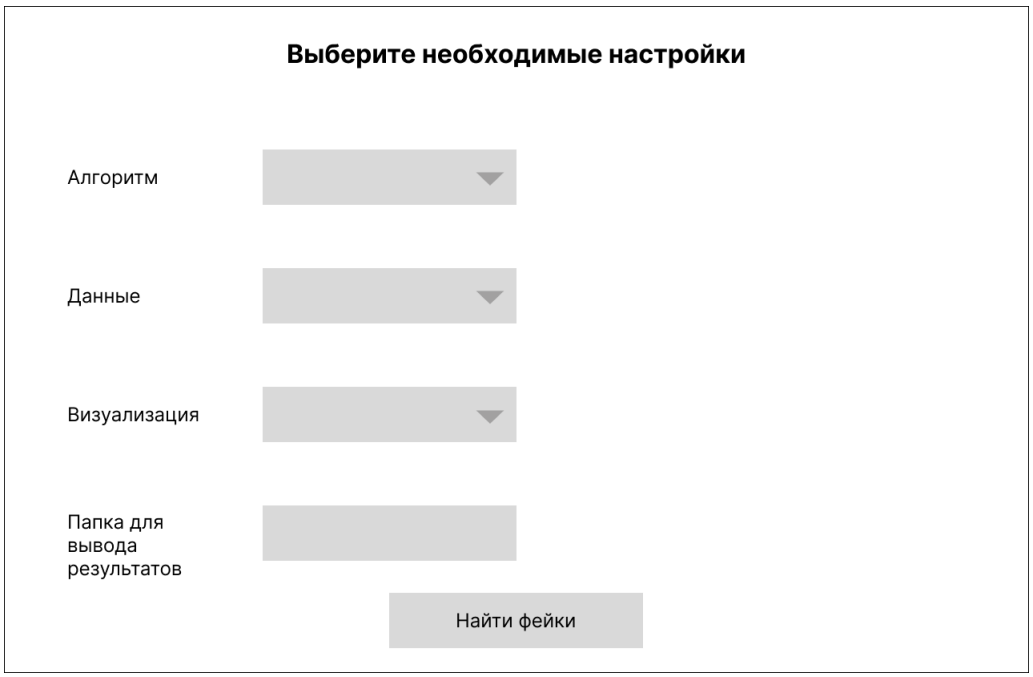

Рисунок 5 – Экран настроек

Рисунок 6 – Пример появления новых полей при выборе алгоритма

2.5. Базы данных

Для того, чтобы пользователь мог загружать данные, при использовании приложения создаются базы данных. Заранее нельзя определить, какие таблицы с какими колонками будут загружены, но для того, чтобы иссле-

дования проходили корректно, будут выводиться предупредления об отсутствии основных столбцов в таблице (user_id, username, email, country, date_registered, date_last_login), так как в основном именно эти данные преобразовываются в числовые значения.

Кроме того, нельзя заранее сказать, какие могут содержаться таблицы в базах данных, поэтому база данных app_database.db создается только при загрузке каких-либо датасетов (пример таблицы: в разделе 1.4«Набор данных для обучения» описана таблица, с которой происходила работа при создании системы). База данных app_database_features.db создается в определенный момент при настройке алгоритмов, она не видна пользователю, он не может взаимодействовать с ней. Она содержит в себе преобразованные в числовые значения параметры выбранного датасета. Столбцы и описание данных, созданной на основе набора данных для обучения таблицы, представлены в таблице **??**

3. РЕАЛИЗАЦИЯ

В разделе 3.1 описаны средства реализации: выбранный язык программирования, редактор кода и подключаемые библиотеки. В разделе 3.2 описана реализация компонентов приложения (какие данные получают модули, как их преобразуют). В разделе 3.2 показана реализация пользовательского интерфейса с изображениями окон.

3.1. Программные средства реализации

Система нахождения фиктивных аккаунтов была написана на языке программирования Python версии 3.9.13. Разработка велась в текстовом редакторе Visual Studio Code 1.89.0. Основные библиотеки, которые использовались при реализации:

1. Tkinter – библиотека для создания графических пользовательских интерфейсов в Python [20].
2. Sqlite3 – библиотека для работы с базой данных SQLite. Позволяет создавать, читать, изменять и удалять записи в реляционных базах данных, используя SQL-запросы [18].
3. Pandas – библиотека для анализа и обработки данных в Python [15].
4. Scikit-learn – библиотека для машинного обучения и анализа данных [17].
5. Matplotlib.pyplot – подбиблиотека Matplotlib, позволяющая создавать графики и визуализировать данные [12].
6. Numpy – библиотека для работы с массивами и матрицами [14].

3.2. Реализация компонентов приложения

Расписать работу модулей приложения, какие данные получают на входе, какие на выходе.

3.3. Реализация пользовательского интерфейса

Реализация пользовательского интерфейса проводилась по разработанным макетам. Было создано три окна: главное меню, оекно редактиро-

вания данных и окно настроек алгоритмов. Для реализации использовалась библиотека tkinter [20].

На рисунке 7

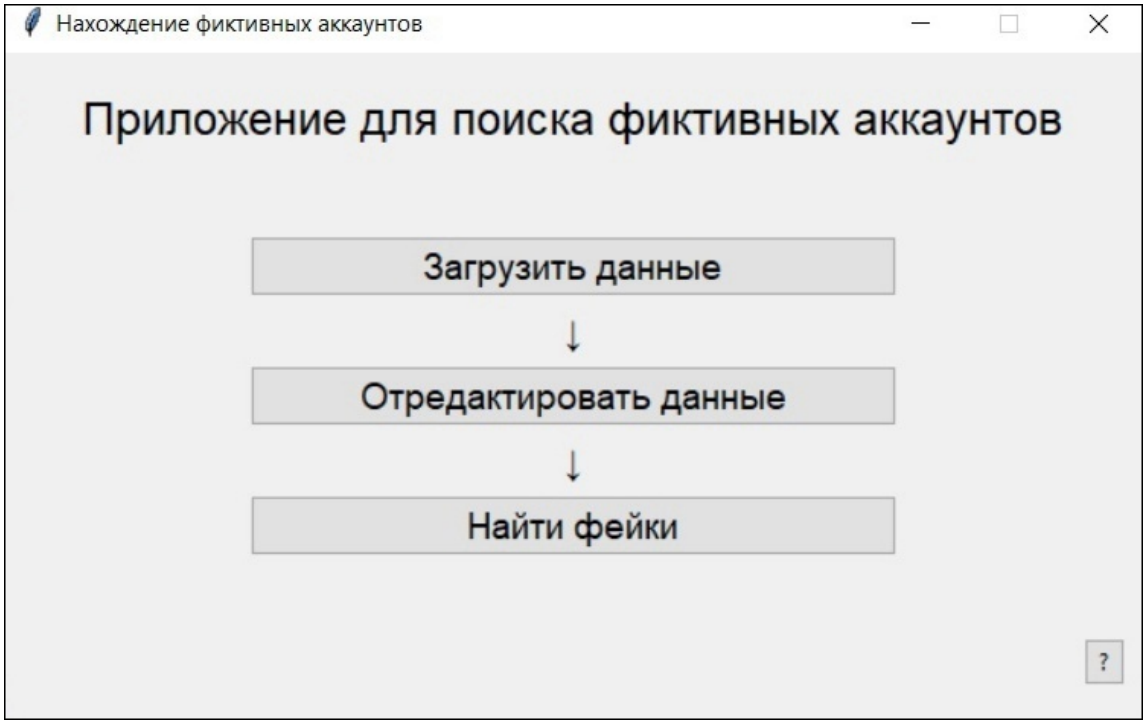

Рисунок 7 – Кластеризация методом изолированного леса

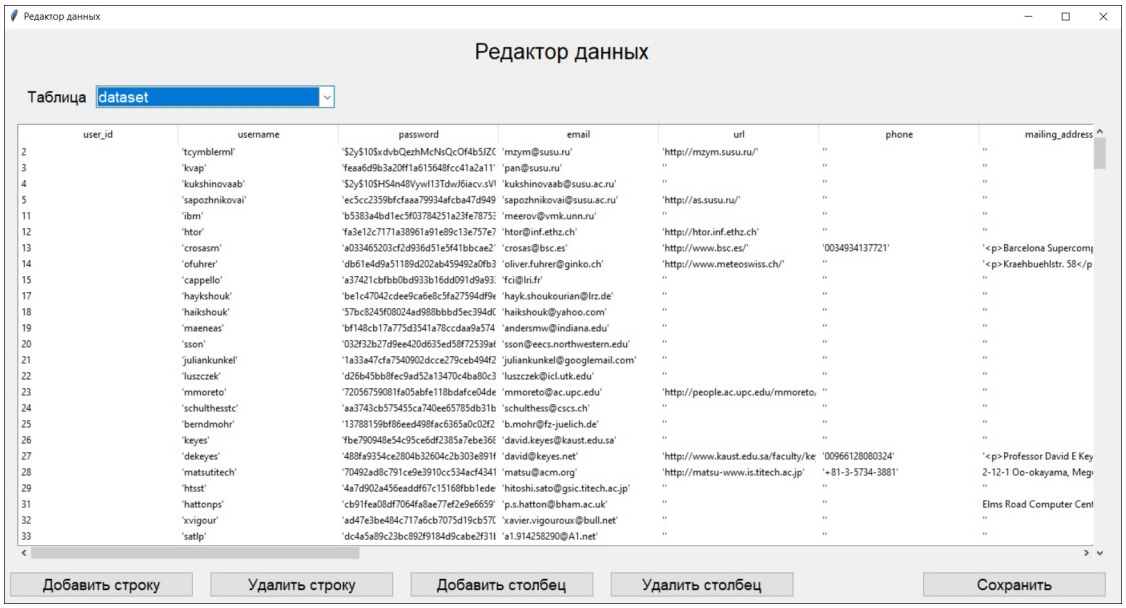

Рисунок 8 – Кластеризация методом изолированного леса

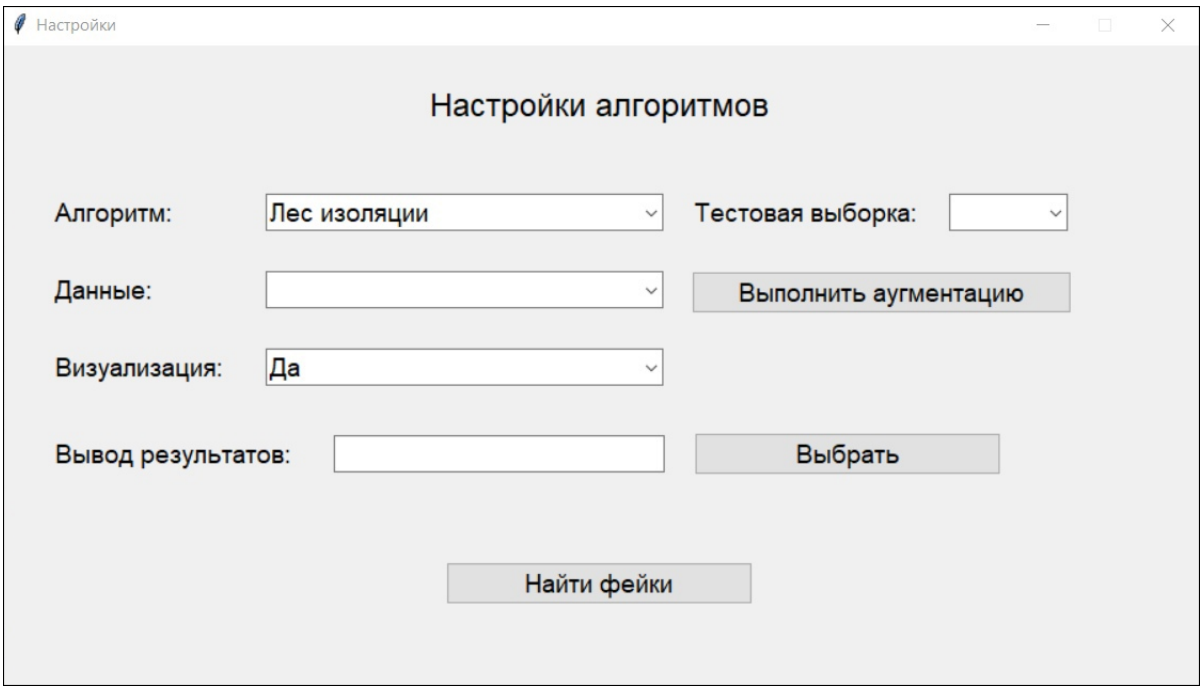

Рисунок 9 – Кластеризация методом изолированного леса

4. ТЕСТИРОВАНИЕ

4.1. Сравнение алгоритмов

Для нахождения более подходящего алгоритма выявления фиктивных аккаунтов были разработаны пробные модели, а после найдены их критерии качества.

Во время разметки были обнаружены скопления фиктивных аккаунтов с близкими по значениям характеристиками. В связи с этим были предложены новые признаки «соседства», влияние которых также было рассмотрено в работе.

Первый шаг - извлечение признаков, по которым проводилась кластеризация, они описаны в разделе 3.4 Инженерия признаков.

Второй шаг - написание моделей и сравнение их метрик. Далее подробнее рассмотрим алгоритмы кластеризации, которые были использованы для нахождения аномалий в виде фиктивных аккаунтов.

Алгоритм изолированного леса

Была произведена подготовка данных в виде заполнения пропущенных значений средним значением по столбцу. Была реализована нормализация признаков: они преобразованы в данные в диапазоне от 0 до 1.

Установлена ожидаемая доля аномалий - 5%. Аномалии (данные с индексом -1) были занесены в файл, а также создано графическое изображение для результатов кластеризации (рисунок 10).

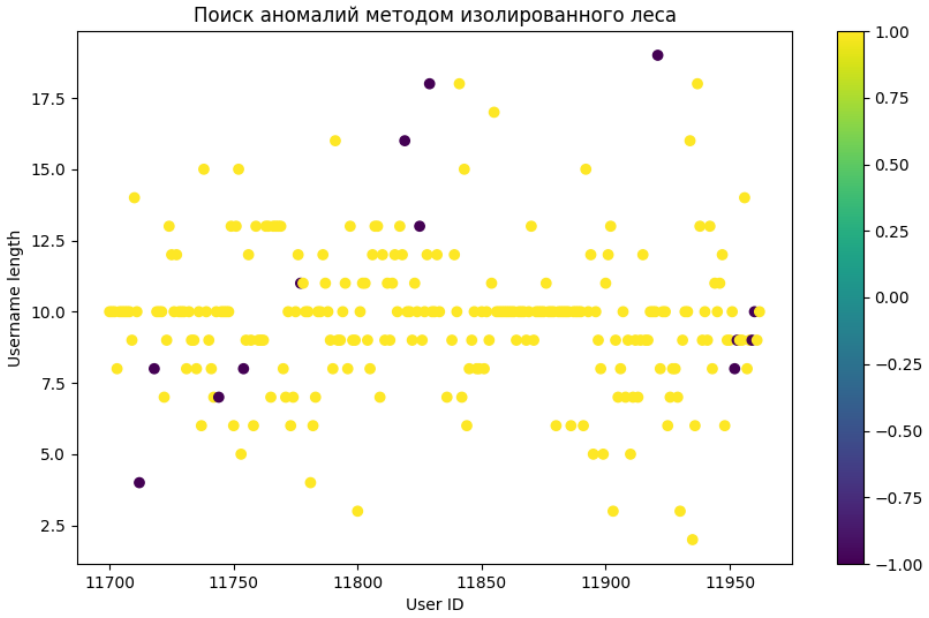

Рисунок 10 – Кластеризация методом изолированного леса

После этого были произведены расчеты метрик алгоритма, с использованием матрицы ошибок (табл. 2).

Таблица 2 – Матрица ошибок

	Positive	Negative
True	8	152
False	5	89

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0,63$$

$$precision = \frac{TP}{TP + FP} = 0,62$$

$$recall = \frac{TP + TN}{TP + FN} = 0,08$$

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 0,14$$

Расчеты метрик алгоритма, который работал со всеми признаками, включая признаки «соседства». Матрица ошибок представлена в табл. 3

Таблица 3 – Матрица ошибок

	Positive	Negative
True	7	151
False	6	90

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0,62$$

$$precision = \frac{TP}{TP + FP} = 0,54$$

$$recall = \frac{TP + TN}{TP + FN} = 0,07$$

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 0,12$$

Алгоритм иерархической кластеризации

Была произведена подготовка данных в виде заполнения пропущенных значений средним значением по столбцу. Была реализована нормализация признаков: они преобразованы в данные в диапазоне от 0 до 1. Была произведена сортировка по username_length. Создана матрица расстояний с методом Single Linkage, так как он сильнее реаширует на выбросы. Создано графическое изображение для результатов кластеризации (рисунок 11).

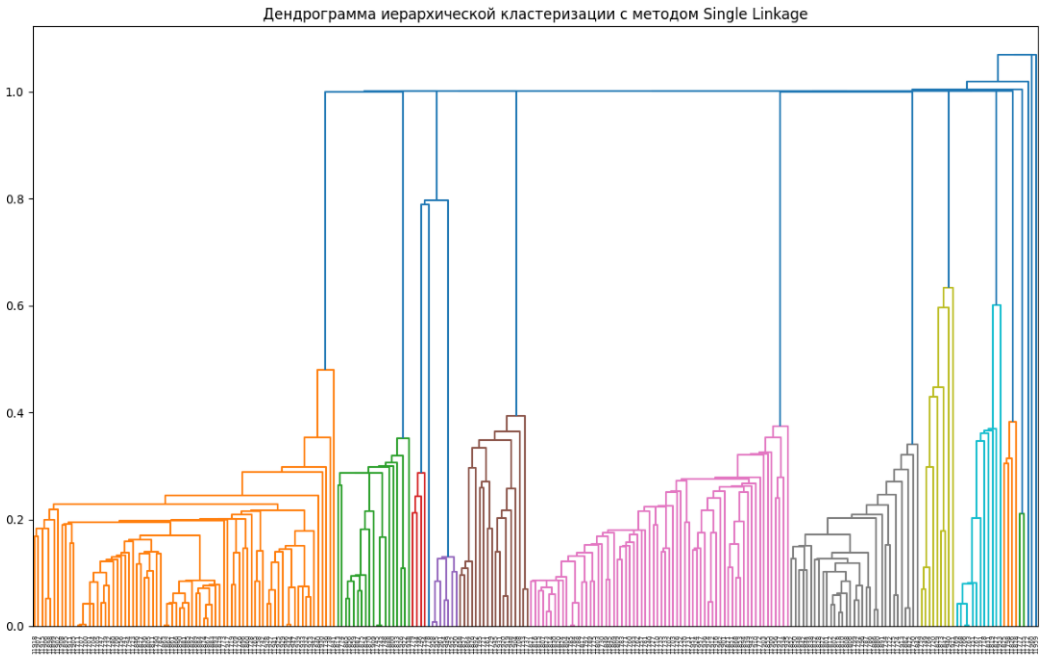

Рисунок 11 – Иерархическая кластеризация

После этого были произведены расчеты метрик алгоритма, с использованием ячеек матрицы ошибок (табл. 4).

Таблица 4 – Матрица ошибок

	Positive	Negative
True	3	150
False	7	94

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0,60$$

$$precision = \frac{TP}{TP + FP} = 0,3$$

$$recall = \frac{TP + TN}{TP + FN} = 0,03$$

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 0,05$$

Расчеты метрик алгоритма, который работал со всеми признаками, включая признаки «соседства». Матрица ошибок представлена в табл. 5

Таблица 5 – Матрица ошибок

	Positive	Negative
True	4	149
False	8	93

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0,60$$

$$precision = \frac{TP}{TP + FP} = 0,33$$

$$recall = \frac{TP + TN}{TP + FN} = 0,04$$

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 0,07$$

Алгоритм DBSCAN

Была произведена подготовка данных в виде заполнения пропущенных значений средним значением по столбцу. Была реализована нормализация признаков: они преобразованы в данные в диапазоне от 0 до 1. Кластеризация была произведена с параметрами eps = 1, min_samples = 6. Выбраны аномалии с индексом -1 и создано графическое изображение для результатов кластеризации (рисунок 12).

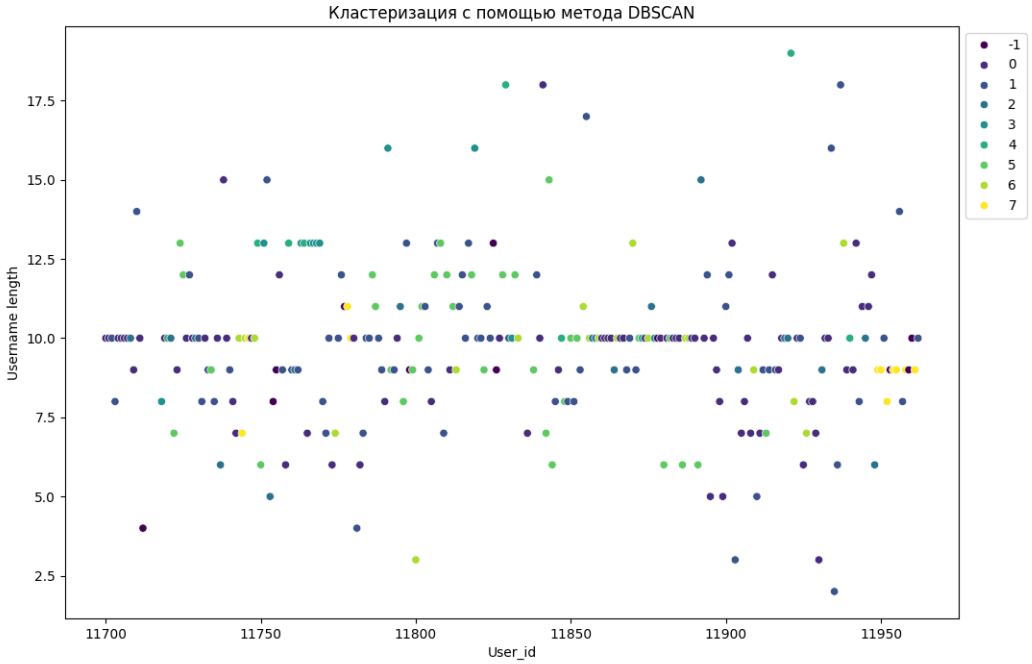

Рисунок 12 – Кластеризация с помощью метода DBSCAN

После этого были произведены расчеты метрик алгоритма, с использованием ячеек матрицы ошибок (табл. 6).

Таблица 6 – Матрица ошибок

	Positive	Negative
True	7	155
False	2	90

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0,60$$

$$precision = \frac{TP}{TP + FP} = 0,3$$

$$recall = \frac{TP + TN}{TP + FN} = 0,03$$

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 0,05$$

Расчеты метрик алгоритма, который работал со всеми признаками,

включая признаки «соседства». Матрица ошибок представлена в табл. 7

Таблица 7 – Матрица ошибок

	Positive	Negative
True	8	154
False	3	89

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0,64$$

$$precision = \frac{TP}{TP + FP} = 0,73$$

$$recall = \frac{TP + TN}{TP + FN} = 0,08$$

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 0,14$$

Сравнение результатов алгоритмов

Результаты рассчитанных метрик для всех реализованных алгоритмов представлены в таблице 8.

Таблица 8 – Метрики алгоритмов

	accuracy	precision	recall	F-measure
Без признаков «соседства»				
Isolation forest	0,63	0,62	0,08	0,14
Hierarchical clustering	0,60	0,30	0,03	0,05
DBSCAN	0,64	0,78	0,07	0,13
С признаками «соседства»				
Isolation forest	0,62	0,54	0,07	0,12
Hierarchical clustering	0,60	0,33	0,04	0,07
DBSCAN	0,64	0,73	0,08	0,14

По данным результатам можно сделать вывод, что без дополнительных признаков больше подходят для обрнаружения фиктивных аккаунтов Isolation forest и DBSCAN, с признаками же выигрывает DBSCAN.

4.2. Функциональное тестирование

Провести тестирование на соответствие приложения предъявленным требованиям.

4.3. Вычислительные эксперименты

В данном разделе представлены вычислительные эксперименты для набора данных.

ЗАКЛЮЧЕНИЕ

ЛИТЕРАТУРА

1. Banfield J.D., Raftery A.E. Model-based Gaussian and non-Gaussian clustering. // Biometrics. – 1993. – Vol. 49. – P. 803–821. – URL: https://api.semanticscholar.org/CorpusID:17507406.

2. Boshmaf Y. The socialbot network: when bots socialize for fame and money. / Y. Boshmaf, I. Muslukhov, K. Beznosov, M. Ripeanu. // Twenty-Seventh Annual Computer Security Applications Conference, ACSAC 2011, Orlando, FL, USA, 5-9 December 2011 / Ed. by R.H. Zakon, J.P. McDermott, M.E. Locasto. – ACM, 2011. – P. 93–102. – URL: https://doi.org/10.1145/2076732.2076746.

3. Breuer A. Preemptive Detection of Fake Accounts on Social Networks via Multi-Class Preferential Attachment Classifiers. / A. Breuer, N.K. Tehrani, M. Tingley, B. Cottel. // CoRR. – 2023. – Vol. abs/2308.05353. – arXiv : 2308.05353.

4. Chen Y., Wu S.F. FakeBuster: A Robust Fake Account Detection by Activity Analysis. // 9th International Symposium on Parallel Architectures, Algorithms and Programming, PAAP 2018, Taipei, Taiwan, December 26-28, 2018. – IEEE, 2018. – P. 108–110. – URL: https://doi.org/10.1109/PAAP.2018.00026.

5. Elyusufi Y., Elyusufi Z., Kbir M.A. Social networks fake profiles detection based on account setting and activity. // Proceedings of the 4th International Conference on Smart City Applications, SCA 2019, Casablanca, Morocco, October 02-04, 2019. – ACM, 2019. – P. 37:1–37:5. – URL: https://doi.org/10.1145/3368756.3369015.

6. Ester M. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. / M. Ester, H. Kriegel, J. Sander, X. Xu. // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA / Ed. by E. Simoudis, J. Han, U.M. Fayyad. – AAAI Press, 1996. – P. 226–231. – URL: http://www.aaai.org/Library/KDD/1996/kdd96-037.php.

7. Fahmy S.G. Modeling the Influence of Fake Accounts on User Behavior and Information Diffusion in Online Social Networks. / S.G. Fahmy, S. AbdelGaber, O.H. Karam, D.S. Elzanfaly. // Informatics. – 2023. – Vol. 10.

– No. 1. – P. 27. – URL: https://doi.org/10.3390/informatics10010027.

8. Gurajala S. Fake Twitter accounts: profile characteristics obtained using an activity-based pattern detection approach. / S. Gurajala, J.S. White, B. Hudson, J.N. Matthews. // Proceedings of the 2015 International Conference on Social Media & Society, Toronto, ON, Canada, July 27-29, 2015 / Ed. by A.A. Gruzd, J. Jacobson, P. Mai, B. Wellman. – ACM, 2015. – P. 9:1–9:7. – URL: https://doi.org/10.1145/2789187.2789206.

9. Hassan A., Alhalangy A.G.I., Al-Zahrani F. Fake Accounts Identification in Mobile Communication Networks Based on Machine Learning. // Int. J. Interact. Mob. Technol. – 2023. – Vol. 17. – No. 4. – P. 64–74. – URL: https://doi.org/10.3991/ijim.v17i04.37645.

10. Hsu S., Kes D., Joshi A. Visualizing Tweets from Confirmed Fake Russian Accounts. // Visualization and Data Analysis 2019, Burlingame, CA, USA, 16-17 January 2019 / Ed. by T. Wischgoll, S. Zhang, D.L. Kao, Y. Chiang. – Society for Imaging Science and Technology, 2019. – URL: https://doi.org/10.2352/ISSN.2470-1173.2019.1.VDA-678.

11. Liu F.T., Ting K.M., Zhou Z. Isolation Forest. // Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy. – IEEE Computer Society, 2008. – P. 413–422. – URL: https://doi.org/10.1109/ICDM.2008.17.

12. Maptolib. – [Электронный ресурс] URL: https://matplotlib.org/stable/api/pyplot_summary.html (дата обращения: 2023-10-17).

13. Mohammadrezaei M., Shiri M.E., Rahmani A.M. Detection of Fake Accounts in Social Networks Based on One Class Classification. // ISC Int. J. Inf. Secur. – 2019. – Vol. 11. – No. 2. – P. 173–183. – URL: https://doi.org/10.22042/isecure.2019.165312.450.

14. Numpy. – [Электронный ресурс] URL: https://numpy.org/doc/stable/ (дата обращения: 2023-10-17).

15. Pandas. – [Электронный ресурс] URL: https://pandas.pydata.org/ (дата обращения: 2023-10-15).

16. Santisteban J., Tejada-Cárcamo J. Unilateral Jaccard Similarity Coefficient. // Proceedings of the First International Workshop on Graph

Search and Beyond, GSB 2015, co-located with The 38th Annual SIGIR Conference (SIGIR'15), Santiago, Chile, August 13th, 2015 / Ed. by O. Alonso, M.A. Hearst, J. Kamps. – Vol. 1393 of CEUR Workshop Proceedings. – CEUR-WS.org, 2015. – P. 23–27. – URL: https://ceur-ws.org/Vol-1393/paper-10.pdf.

17. Scikit-learn. – [Электронный ресурс] URL: https://scikit-learn.org/stable/ (дата обращения: 2023-10-16).

18. Sqlite3. – [Электронный ресурс] URL: https://docs.python.org/3/library/sqlite3.html (дата обращения: 2023-10-15).

19. Stolbova A., Ganeev R., Ivaschenko A. Intelligent Identification of Fake Accounts on Social Media. // 30th Conference of Open Innovations Association, FRUCT 2021, Oulu, Finland, October 27-29, 2021. – IEEE, 2021. – P. 279–284. – URL: https://doi.org/10.23919/FRUCT53335.2021.9599974.

20. Tkinter. – [Электронный ресурс] URL: https://docs.python.org/3/library/tkinter.html (дата обращения: 2024-02-24).

21. Uppada S.K. Novel approaches to fake news and fake account detection in OSNs: user social engagement and visual content centric model. / S.K. Uppada, K. Manasa, B. Vidhathri, R. Harini, B. Sivaselvan. // Soc. Netw. Anal. Min. – 2022. – Vol. 12. – No. 1. – P. 52. – URL: https://doi.org/10.1007/s13278-022-00878-9.

ПРИЛОЖЕНИЯ

Приложение А. Примеры размеченных аккаунтов

Примеры аккаунтов, которые были размечены, представлены в таблицах 9– 10.

Таблица 9 – Фиктивные аккаунты

Характеристика	Значения
user_id	6758
username	fishzupic438
password	6cd28fca7bdf8937ee45eb65d0a126a5
email	fishgvnpo334@gmail.com
url	https://aktivator-kleva.com
phone	88298893596
mailing_address	fishfmmzj822@gmail.com
billing_address	NULL
country	MT
locales	
date_last_email	NULL
date_registered	2021-03-04 22:22:07
date_validated	NULL
date_last_login	2021-03-04 22:22:07
must_change_password	0
auth_id	NULL
auth_str	NULL
disabled	0
disabled_reason	NULL
inline_help	0
gossip	NULL
fake	1
user_id	8872
username	charlestab
password	c129710afdd27c0eaffbb3487763c46e
email	shishakova69@mail.ru

Продолжение таблицы 9

Характеристика	**Значения**
url	https://fortnite-proswapper.com
phone	89486237981
mailing_address	shishakova69@mail.ru
billing_address	NULL
country	LB
locales	
date_last_email	NULL
date_registered	2021-04-29 09:44:52
date_validated	NULL
date_last_login	2021-04-29 09:44:52
must_change_password	0
auth_id	NULL
auth_str	NULL
disabled	0
disabled_reason	NULL
inline_help	0
gossip	NULL
fake	1
user_id	11662
username	ionka
password	244d4b7728e312ba73ddb0c68f536b10
email	soetedom@hotmail.com
url	
phone	89741875247
mailing_address	lrdrjrmr@aol.com
billing_address	NULL
country	CG
locales	
date_last_email	NULL
date_registered	2021-09-15 09:40:32
date_validated	NULL

Продолжение таблицы 9

Характеристика	**Значения**
date_last_login	2021-09-15 09:40:32
must_change_password	0
auth_id	NULL
auth_str	NULL
disabled	0
disabled_reason	NULL
inline_help	0
gossip	NULL
fake	1

Таблица 10 – Настоящие аккаунты

Характеристика	**Значения**
user_id	2
username	tcymblerml
password	$2y$10$xdvbQezhMcNsQcOf4b5JZOZ g2butksNxug7/TiEV.sClODdS4djJK
email	mzym@susu.ru
url	http://mzym.susu.ru/
phone	
mailing_address	
billing_address	NULL
country	Ru
locales	
date_last_email	2020-01-03 18:42:34
date_registered	2014-01-24 13:57:45
date_validated	NULL
date_last_login	2022-12-04 08:56:49
must_change_password	0
auth_id	1
auth_str	NULL

Продолжение таблицы 10

Характеристика	**Значения**
disabled	0
disabled_reason	NULL
inline_help	0
gossip	NULL
fake	0
user_id	3238
username	masa
password	8624ecbdf495eacb1c01bd27e10be7fc
email	masa@tohoku.ac.jp
url	
phone	
mailing_address	
billing_address	NULL
country	JP
locales	
date_last_email	NULL
date_registered	2019-07-01 01:44:03
date_validated	NULL
date_last_login	2021-09-19 10:05:20
must_change_password	0
auth_id	NULL
auth_str	NULL
disabled	0
disabled_reason	NULL
inline_help	0
gossip	NULL
fake	0
user_id	4564
username	and_debol
password	$2y$10$UIT6GMrsZ4tqOaD02/LFj. nXpia0T2IE1M/AZe/SK1SgNpkb2f9Bm

Продолжение таблицы 10

Характеристика	**Значения**
email	and.debol@srcc.msu.ru
url	
phone	+79060367065
mailing_address	Leninskiye Gory street 1 building 4, 119234, Moscow
billing_address	NULL
country	RU
locales	
date_last_email	NULL
date_registered	2020-03-24 04:57:07
date_validated	NULL
date_last_login	2022-09-10 17:14:14
must_change_password	0
auth_id	NULL
auth_str	NULL
disabled	0
disabled_reason	NULL
inline_help	0
gossip	NULL
fake	0

Приложение Б. Спецификация вариантов использования

Спецификация вариантов использования (ВИ) разработанной системы приведена в таблицах 11– 16.

Таблица 11 – Спецификация варианта «Загрузить данные»

Прецедент: Загрузить данные
ID: 1
Краткое описание: Исследователь загружает данные в систему.
Главные актеры: Исследователь.
Второстепенные актеры: Нет.
Предусловия: отсутствуют
Основной поток: 1. Прецедент начинается, когда Исследователь нажимает на кнопку «Загрузить данные». 2. Исследователь выбирает необходимый файл. 3. Система сохраняет загруженные данные.
Постусловия: Исследователь загрузил данные
Альтернативные потоки: Нет

Таблица 12 – Спецификация варианта «Редактировать данные»

Прецедент: Редактировать данные
ID: 2
Краткое описание: Исследователь редактирует загруженные данные.
Главные актеры: Исследователь.
Второстепенные актеры: Нет.
Предусловия: данные для редактирования были загружены.
Основной поток: 1. Прецедент начинается, когда Исследователь нажимает на кнопку «Редактировать данные». 2. Исследователь выбирает данные, которые будет редактировать. 3. Исследователь редактирует данные. 4. Система сохраняет изменения.
Постусловия: Данные были отредактированы.
Альтернативные потоки: Нет

Таблица 13 – Спецификация ВИ «Выполнить аугментацию»

Прецедент: Выполнить аугментацию
ID: 3
Краткое описание: Исследователь генерирует новые искусственные записи, чтобы уравновесить выборку.
Главные актеры: Исследователь.
Второстепенные актеры: Нет.
Предусловия: Наличие размеченных данных.
Основной поток: 1. Прецедент начинается, когда Исследователь нажимает кнопку «Выполнить аугментацию». 2. Приложение запускает алгоритм для создания новых записей фиктивных или настоящих аккаунтов. 3. Данные сохраняются.
Постусловия: Датасет пользователя уравновешен.
Альтернативные потоки: Нет

Таблица 14 – Спецификация ВИ «Выполнить настройку»

Прецедент: Выполнить настройку
ID: 4
Краткое описание: Исследователь выбирает алгоритм и настраивает его параметры.
Главные актеры: Исследователь.
Второстепенные актеры: Нет.
Предусловия: Нет.
Основной поток 1. Прецедент начинается, когда пользователь нажимает на кнопку «Настройки». 2. Пользователь выбирает необходимый алгоритм и его параметры.
Постусловия: Выполнена настройка алгоритма.
Альтернативные потоки: Нет

Таблица 15 – Спецификация ВИ «Обучить модель»

Прецедент: Обучить модель
ID: 5
Краткое описание: Исследователь обучает модель на данных.
Главные актеры: Исследователь.
Второстепенные актеры: Нет.
Предусловия: Загружен файл с размеченными данными, а также выполнена настройка алгоритма.
Основной поток 1. Прецедент начинается, когда пользователь нажимает на кнопку «Обучить модель». 2. Система обучает модель на данных исследователя.
Постусловия: Выполнено обучение модели.
Альтернативные потоки: Нет

Таблица 16 – Спецификация ВИ «Найти фиктивные аккаунты»

Прецедент: Найти фиктивные аккаунты
ID: 6
Краткое описание: Исследователь находит фиктивные аккаунты.
Главные актеры: Исследователь.
Второстепенные актеры: Нет.
Предусловия: Загружен файл с данными, выполнена настройка алгоритма.
Основной поток 1. Прецедент начинается, когда пользователь нажимает на кнопку «Найти фиктивные аккаунты». 2. Система находит фиктивные аккаунты и выводит результат в соответствии с настройкой.
Постусловия: Выполнен поиск фиктивных аккаунтов.
Альтернативные потоки: Нет