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 5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7
1.1. Обзор основных методов, используемых для поиска аномалий 10
2. ПРОЕКТИРОВАНИЕ 13
2.1. Варианты использования системы 13
2.2. Требования к системе 14
2.3. Графический интерфейс 14
3. РЕАЛИЗАЦИЯ 15
3.1. Архитектура приложения 15
3.2. Программные средства реализации 15
3.3. Подготовка данных 15
3.4. Инженерия признаков 15
3.5. Реализация компонентов приложения 16
3.6. Реализация пользовательского интерфейса 16
4. ТЕСТИРОВАНИЕ 17
4.1. Сравнение алгоритмов 17
4.2. Функциональное тестирование 22
4.3. Вычислительные эксперименты 23
ЗАКЛЮЧЕНИЕ 24
ЛИТЕРАТУРА 25

ВВЕДЕНИЕ

Актуальность темы

Интернет может нести не только пользу, но также и вред. Одна из его опасностей — фиктивные аккаунты. Они существуют как в различных социальных сетях, так и на других платформах; некоторые используются в безобидных целях, а другие — для распространения ложной информации.

«Фейки» могут быть созданы с разными целями: получить коммерческую выгоду, дискредитировать настоящего пользователя, заполучить личную информацию и так далее. Это очень серьезная опасность, которую не всегда можно распознать с первого взгляда. Некоторые пользователи мировой сети не догадываются, кто скрывается в диалоге с интернет-знакомым — люди могут умело подделывать аккаунты в социальных сетях. Кроме того, часто бывает, что за страницей обычных пользователей могут скрываться автоматизированные программы. Они могут практически не отличаться от обычных аккаунтов.

Особенно сильно это наносит вред такой среде, как наука. Подделка информации в этой сфере несет огромный вред, который может распространяться на все общество. Научные данные используются везде: от строительства домов, до лечения животных, и неточности или ошибки в них могут стоить дорого.

Основная проблема заключается в том, что фиктивные аккаунты, которые создаются в системах, связанных с наукой, тяжело отличимы от обычных аккаунтов. Для поддержания стабильной работы публикации статей необходимо быстрое реагирование на «фейки» и их оперативное удаление.

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, выполнение тестирования программной системы.

Структура и содержание работы

Написать разделы, из которых состоит курсовая работа.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Open Journal System – 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).

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

Авторы статьи [6] 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 (SVM), 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 [3] посвящена выявлению поддельных профилей в социальных сетях. Для их обнаружения было предложено множество алгоритмов и методов, и авторы этой работы оценивают точность использования дерева решений (DT) и наивного алгоритма Байеса (NB) для классификации профилей пользователей на поддельные и подлинные.

В работе [11] исследователи используют модель SENAD, которая определяет подлинность новостных статей, публикуемых в Twitter, на ос-

нове 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). Предложенная гибридная идея объединения ELA и Sent и анализа настроений помогает обнаруживать поддельные изображения с точностью около 76%.

В исследовании [5] проведен анализ 62 миллионов общедоступных профилей пользователей Twitter, и разработана стратегия идентификации автоматически сгенерированных поддельных профилей. Используя алгоритм сопоставления шаблонов имен, анализ времени обновления твитов и даты создания профилей, были выявлены фиктивные учетные записи пользователей.

В статье [1] описывается новый алгоритм под названием Preferred attachment k-class Classifier (PreAttacK) 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). Исследователи используют эту модель для создания нового алгоритма PreAttacK.

В работе [9] 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, 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 [10] 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 поддель-

ных 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 [2] 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 [7] 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 критические моменты (например, во время дебатов).

Авторы статьи [4] 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.

1.1. Обзор основных методов, используемых для поиска аномалий

Методы, которые могут использоваться для поиска аномалий: **Isolation Forest** (изолированный лес) [8] представляет собой алгоритм обнаружения аномалий, который строит ансамбль изолирующих деревьев решений. В процессе построения этих деревьев, выбираются случайные признаки и случайные пороги для разбиения данных. Аномалии обычно требуют меньше разбиений, чтобы быть «изолированными» от остальных данных.

В основе метода лежит предположение, что аномальные точки будут находиться ближе к корню дерева, так как им потребуется меньше разбиений для выделения. Наоборот, нормальные точки должны быть более распределены по дереву. Путем измерения среднего пути до изоляции каждой точки, алгоритм может определить степень их аномальности.

Hierarchical clustering (иерархическая кластеризация) [13] – 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. Может использоваться расстояние между кластерами (например, евклидово расстояние) или другие меры 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. Обычно используются меры несходства между подгруппами, такие как расстояние или сходство. 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.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [12] – 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).

DBSCAN обеспечивает гибкость в выявлении кластеров различной формы и эффективно обрабатывает точки, которые можно посчитать шумом. Однако на результат влияет выбор параметров: радиуса и минимального числа точек, которые должны образовывать плотную область.

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:

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

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

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

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

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

Табл. 1. Матрица ошибок

	Positive	Negative
True	TP	TN
False	FP	FN

TP (True Positives) – количество верно предсказанных положитель-

ных примеров. Это случаи, когда модель правильно предсказала, что объект принадлежит к положительному классу.

TN (True Negatives) – количество верно предсказанных отрицательных примеров. Это случаи, когда модель правильно предсказала, что объект принадлежит к отрицательному классу.

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.

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.

2. ПРОЕКТИРОВАНИЕ

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

2.1. Варианты использования системы

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

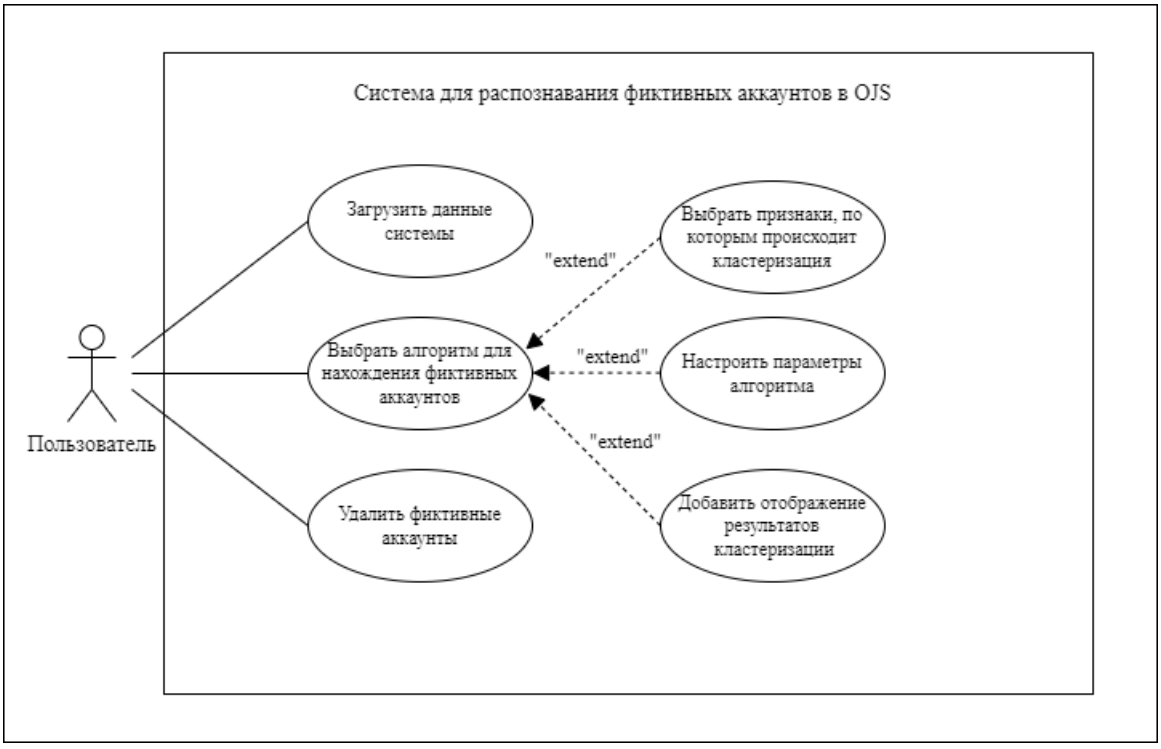

Рис. 1. Диаграмма вариантов использования

Единственный актер, который может взаимодействовать с системой распознавания фиктивных аккаунтов в OJS – это пользователь. Он может выполнять следующие действия:

1. Загрузить данные системы. Пользователь совершает выгрузку данных об аккаунтов.

2. Выбрать алгоритм для нахождения фиктивных аккаутов. Пользователь выбирает подходящий метод, который будет искать аномалии в виде поддельных учетных записей.

3. Выбрать признаки, по которым происходит кластеризация. Пользо-

ватель отмечает признаки, по которым необходимо отыскивать фиктивные аккаунты.

4. Настроить параметры алгоритма. Пользователь настраивает параметры выбранного метода.

5. Добавить отображение результатов кластеризации. Пользователь отображает результаты кластеризации после работы алгоритма.

6. Удалить фиктивные аккаунты. Пользователь совершает удаление всех аккаунтов, отмеченных как фиктивные, или выборочно некоторые из них.

2.2. Требования к системе

Функциональные требования

Функциональные требования определяют функциональность программного обеспечения, то есть описывают, какое поведение должна предоставлять разрабатываемое приложение

Нефункциональные требования

К нефункциональным требованиям системы относятся свойства, которыми она должна обладать. Например, удобство использования, безопасность, расширяемость и т.д.

2.3. Графический интерфейс

Представление графического интерфейса программной системы.

3. РЕАЛИЗАЦИЯ

3.1. Архитектура приложения

Показать модули приложения, расписать то, что делает каждый из них.

3.2. Программные средства реализации

Прописать используемые языки программирования и библиотеки.

3.3. Подготовка данных

Сбор данных о пользователях системы электронного журнал, определение, какие аккаунты являются фиктивными.

3.4. Инженерия признаков

У фиктивных аккаунтов есть некоторые признаки, которые могут отличать их от обычных аккаунтов. Далее рассмотрим их подробнее.

- user_id: уникальный идентификатор пользователя. Используется для идентификации каждого аккаунта;
- username_length: длина имени пользователя;
- numbers_in_name: переменная, которая означает наличие цифр в имени пользователя. Если нет – ставится 0, иначе – 1;
- email_length: длина email;
- matching_names: переменная, которая означает совпадение username с email по определенному порогу сходства (в случае несовпадения 0, иначе – 1);
- pattern_email: продходит ли email по шаблону user@domain.com (в случае прохождения по парамтру 0, иначе – 1);
- country: проверка, указана ли страна (если не указана - 0, иначе - 1);
- date_last_email: проверка даты последнего отправленного email. Если она есть – 1, иначе – 0;

- date_registered: дата регистрации аккаунта;
- date_last_login: дата последнего входа в аккаунт;
- matching_dates: характеристика, совпадают ли даты послегнего входа в аккаунт и даты регистрации.
- username_neighbour_above: расстояние до «соседа» по базе данных сверху рассматриваемого аккаунта, который имеет сходные символы в username.
- username_neighbour_below: расстояние до «соседа» по базе данных снизу рассматриваемого аккаунта, который имеет сходные символы в username.
- email_neighbour_above: расстояние до «соседа» по базе данных сверху рассматриваемого аккаунта, который имеет сходные символы в логине email.
- email_neighbour_below: расстояние до «соседа» по базе данных снизу рассматриваемого аккаунта, который имеет сходные символы в логине email.

3.5. Реализация компонентов приложения

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

3.6. Реализация пользовательского интерфейса

Каким образом и где реализован интерфейс приложения. Описать основные механизмы и привести изображения.

4. ТЕСТИРОВАНИЕ

4.1. Сравнение алгоритмов

Для нахождения более подходящего алгоритма выявления фиктивных аккаунтов были разработаны пробные модели, а после найдены их критерии качества. Так как были разработаны новые признаки «соседства», оценка методов производилась сначала без них, а потом с ними.

Первое, что было сделано – это разметка датасета (ссылка на датасет как сделать). Фиктивные аккаунты были обозначены 1, а реальные – 0. Далее было произведено извлечение признаков, по которым проводилась кластеризация. Признаки описаны в разделе 3.4 Инженерия признаков.

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 (рис. 2).

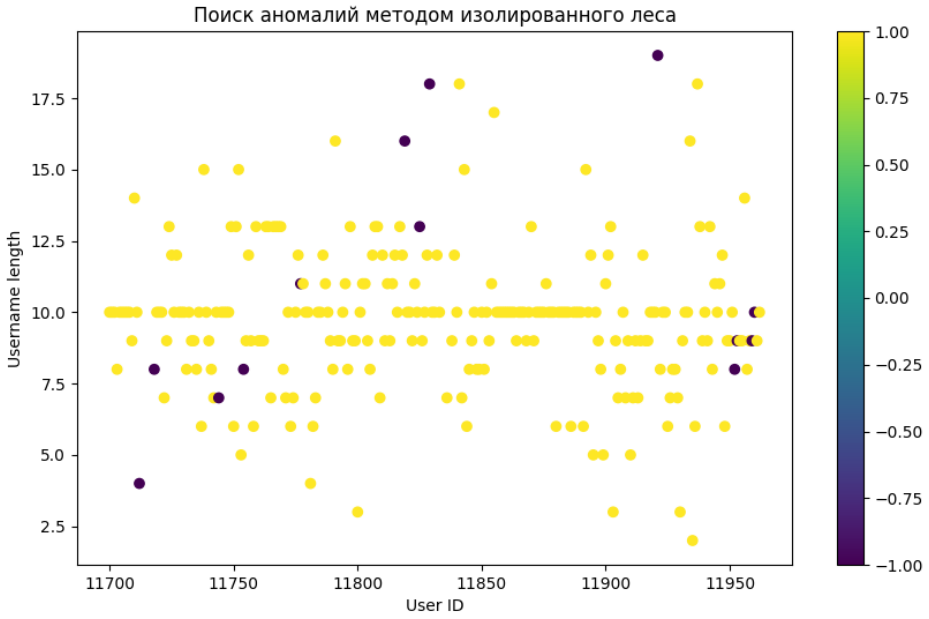

Рис. 2. Кластеризация методом изолированного леса

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

Табл. 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

Табл. 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 (рис. 3).

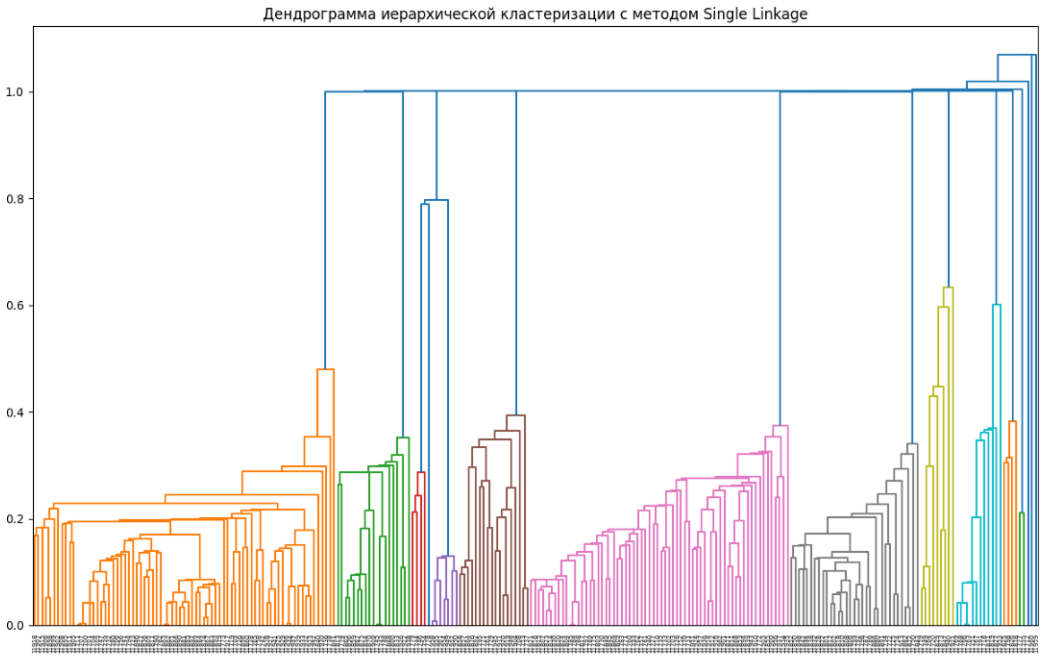

Рис. 3. Иерархическая кластеризация

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

Табл. 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

Табл. 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 (рис. 4).

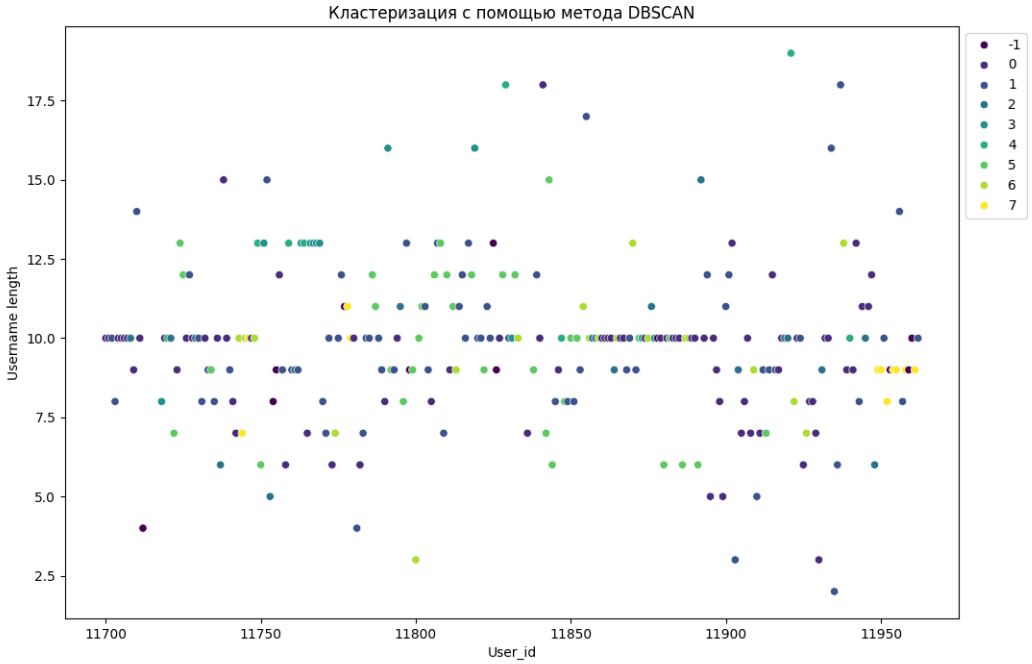

Рис. 4. Иерархическая кластеризация

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

Табл. 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

Табл. 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.

Табл. 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. 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.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.

11. 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.

12. DBSCAN. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html. Accessed on: 18 декабря 2023 г..

13. Кластеризация. – URL: https://scikit-learn.ru/clustering/hierarchical-clustering. Accessed on: 18 декабря 2023 г..