ОГЛАВЛЕНИЕ

УДК 004

РАЗРАБОТКА МОБИЛЬНОЙ СИСТЕМЫ СБОРА ЦИФРОВОГО СЛЕДА ДЛЯ ИСПОЛЬЗОВАНИЯ ПРИ ГОРИЗОНТАЛЬНОМ ОБУЧЕНИИ

Р. Р. Алимбеков[1, [0000-0002-9306-8463]], А. Ф. Хасьянов[2, [0000-0002-1819-593X]]

[1, 2]Казанский (Приволжский) федеральный университет, ул. Кремлевская, д. 35, г. Казань, 420008

[1]arr1998@gmail.com, [2]ak@it.kfu.ru

Аннотация

Горизонтальное обучение — это современная модель, альтернативная традиционному вертикальному обучению и основанная на сотрудничестве, взаимодействии между студентами в рамках образовательного процесса. При этом для промежуточной аттестации по дисциплине преподавателю необходимо оценить вклад каждого студента в решение групповой задачи.

На сегодняшний день пользователями мобильных приложений в разных областях оставляется огромное количество цифровых следов. Основными типами оставляемого цифрового следа являются текст, фотографии, видеозаписи, аудиозаписи, а также текущее местоположение.

Для содействия преподавателю при горизонтальном обучении нами разработано мобильное приложение, собирающее все вышеперечисленные виды цифрового следа, а также веб-приложение, анализирующее его.

Ключевые слова: *сотовая связь, мобильное приложение, цифровой след, сбор цифрового следа, учет, анализ.*

ВВЕДЕНИЕ

В настоящее время существуют разные методы сбора цифрового следа, среди которых можно выделить три основных: платформенный метод, метод очной фиксации и метод анализа общего результата. Все эти методы напрямую связаны с платформой-агрегатором, которая хранит собранный цифровой след, и различаются лишь в том, какой след и каким образом туда попадет [8].

Одним из основных методов сбора цифрового следа является платформенный метод. В этом случае платформа-агрегатор, например, образовательная онлайн-платформа или тестирующая система, имеющая свою собственную систему оценки и фиксации цифрового следа, самостоятельно регистрирует деятельность пользователя системы. Единожды создав такую платформу, больше нет необходимости прибегать к ручному сбору цифрового следа, однако платформа не всегда находится в быстром доступе, и таким образом часть цифрового следа может пропасть [7].

Другой метод сбора цифрового следа – метод очной фиксации. Он предполагает, что если два человека взаимодействуют друг с другом, то в платформу информацию об этом может занести вручную третий участник - наблюдатель со стороны. При этом в контексте этого метода наблюдатель должен иметь объективный взгляд на результат взаимодействия студентов и указать только такой цифровой след, который отражает этот результат. Метод малоприменим в образовательных целях, так как наблюдатель выполняет функции преподавателя, который не может в постоянном режиме вести наблюдение за студентами [6].

Третий метод сбора – метод анализа общего результата. Он применим в ситуациях, когда команда выполняет какой-то общий проект, и информацию о деятельности одного из членов команды заносит в платформу другой член команды. В этом случае полученная информация необъективна, но единой системы оценки здесь не может быть. В этом случае оценивать полученный цифровой след следует по анализу итога, полученного командой в результате выполнения проекта [10].

Все три метода анализа цифрового следа применяются в различных ситуациях, в том числе при повышении качества образовательного процесса.

В случае, когда занятие проходит в аудитории, применим метод очной фиксации. Преподаватель, наблюдая за работой студентов или небольших студенческих групп, может оценивать вклад каждого студента в общий проект. В случае взаимодействия в команде среднего количества студентов (от 3 до 5 человек) применим метод анализа результата. Однако при существенном увеличении числа студентов и нагрузки на одного преподавателя, результат этого метода может быть нерелевантным, так как проверить ответ студента о своем

товарище по команде не представляется возможным. Помимо этого, оба эти методы неприменимы в случае, если работа над задачей идет за пределами аудитории в учебном заведении.

На сегодняшний день большинство молодых людей пользуется мобильными телефонами каждый день и в том числе постоянно – самыми разными мобильными приложениями.

Л. Севилианно Гарсиа и Эстебана Васкез-Кано на основании результатов исследования, проведенного в одном из испанских университетов, получили вывод о пользе использования мобильных телефонов в процессе обучения, так как они способствуют получению более легкого доступа к информации и стимулируют соревнование между студентами по зарабатыванию как можно большего количества баллов в конце семестра [11].

Одной из проблем интеграции цифрового следа в горизонтальное обучение является отсутствие мотивации у студента делиться информацией о горизонтальном обучении, так как помимо обычного образовательного процесса, который аналогичен процессу вертикального обучения, студенту будет необходимо делиться информацией о своем обучении через какой-нибудь сервис, например, сайт или мобильное приложение. Одним из решений этой проблемы является геймификация [16].

Основным преимуществом использования цифрового следа в образовательном процессе является возможность связать цифровой след с человеком, оставившим его. Для отслеживания прогресса у студентов можно использовать цифровой отпечаток, который студент, мотивированный геймификацией, добровольно оставляет в информационной системе и на основании которого у преподавателя появляется возможность в поощрении студента, тем самым повышая мотивацию к распространению информации о горизонтальном обучении.

Таким образом, разработанное приложение для мобильных телефонов с использованием платформенного метода дает пользователю удобную возможность оставить цифровой след, а преподавателям – наиболее эффективно собирать и анализировать полученные данные.

АНАЛИЗ МОБИЛЬНЫХ ОПЕРАЦИОННЫХ СИСТЕМ

С целью разработки мобильного приложения нами проведен анализ распределения мобильных операционных систем и мобильных устройств в настоящее время. В мире существует несколько разных мобильных операционных систем: iOS, Android, Tizen, Аврора, Harmony OS.

1) iOS — мобильная операционная система для смартфонов, электронных планшетов, носимых проигрывателей, разрабатываемая и выпускаемая американской компанией Apple [18].

2) Android — операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, фитнес-браслетов, игровых приставок, ноутбуков, нетбуков, смартбуков, очков Google Glass, телевизоров, проекторов и других устройств (в 2015 году появилась поддержка автомобильных развлекательных систем и бытовых роботов) [19].

3) Tizen — открытая операционная система на базе ядра Linux, поддерживает аппаратные платформы на процессорах архитектур ARM и x86 [20].

4) Аврора — российская мобильная операционная система, включающая проекты с открытым исходным кодом и компоненты с закрытым исходным кодом, создана для построения доверенной мобильной инфраструктуры, защиты чувствительной информации в государственных организациях/учреждениях и крупных и средних коммерческих компаниях. Востребована компаниями, которые ориентируются на импортозамещение и снижение операционных рисков. Единственная мобильная ОС, включенная в Единый реестр российских программ для ЭВМ и БД [21].

5) Harmony OS — операционная система на базе Android, разрабатываемая компанией Huawei с 2012 года. Она разработана для интеллектуальных устройств, таких как умные часы, смарт-ТВ смартфоны, и используется в качестве мобильной операционной системы [22].

Несмотря на существенное разнообразие, в настоящее время самым активным образом развиваются лишь две мобильные операционные системы – Android и iOS. Согласно статистике, менее 1% устройств, которыми владеют пользователи, являются платформами для операционных систем, отличных от Android и iOS [11]. При этом количество пользователей, владеющих мобильным телефоном под управлением операционной системы Android, превышает

количество пользователей, владеющих мобильным телефоном под управлением операционной системы iOS более чем в 2 раза, как показано на рисунке 1. Также любое приложение, разработанное для Android, будет работать на Harmony OS, Авроре и Tizen [9].

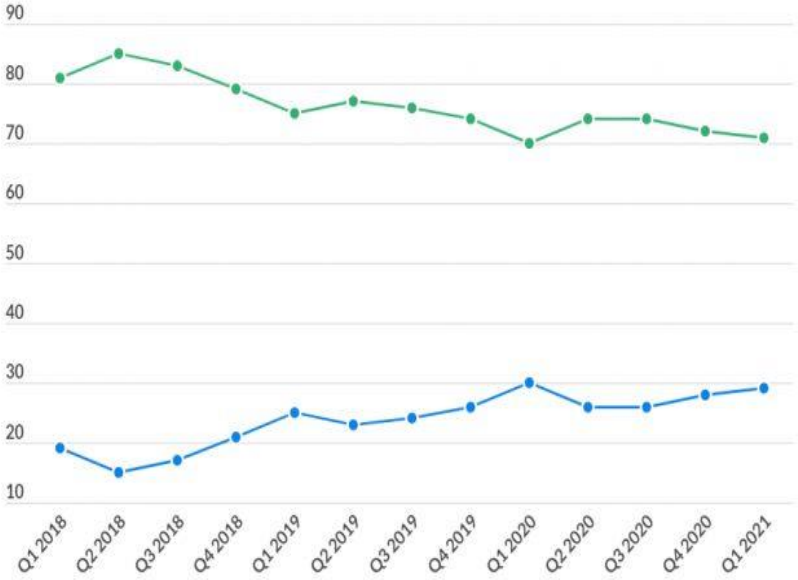

Рис. 1. Количество устройств Android и iOS

Таким образом, для разработки мобильного приложения по сбору цифрового следа в целях повышения качества горизонтального обучения была выбрана операционная система Android.

АНАЛИЗ МЕХАНИЗМОВ СБОРА ЦИФРОВОГО СЛЕДА

Следующим шагом разработки мобильного приложения стало рассмотрение механизмов сбора цифрового следа. Как уже было описано ранее, для реализации проекта был выбран единственно подходящий в данной ситуации платформенный метод. Учитывая специфику операционной системы Android, а также тот факт, что мобильный телефон чаще всего находится рядом со своим владельцем, в качестве активного цифрового следа будут рассмотрены заметки, фотографии, видеозаписи, и в качестве пассивного – геолокация. Также для имплементации элементов геймификации в приложении реализованы цифровые метки, выступающие в качестве наград за более частое количество предоставленного разнообразного цифрового следа. Для реализации возможности сбора активного цифрового следа использованы виджеты ввода

текста для ввода текста, а также встроенная в смартфон камера для съемки фотографий и видеозаписей. Сбор пассивного цифрового следа в свою очередь подразделяется на две подзадачи: определение местоположения с точностью до здания и определение местоположения внутри здания. Чтобы определить свое местоположение с точностью до здания, телефону необходимо использовать датчик GPS. В качестве сервиса, который будет предоставлять информацию о геолокации, был выбран сервис Google-карты, так как он позволяет отслеживать местоположение телефона, когда приложение находится в фоне, с помощью удобного API. Так как определение местоположения с помощью GPS внутри здания затруднено, вместо этого предлагается использовать QR-коды, RUKA-маркеры, NFC-метки или другие маячки разного рода, позволяющие определить местоположение с помощью телефона внутри здания. Примером такого определения может служить работа [17], где продемонстрировано эффективное отслеживание контактов внутри помещения с помощью QR-кодов.

Цифровые метки будут добавлены для повышения стимула студентов делиться цифровым следом, чтобы преодолеть проблему, показанную в исследовании о цифровых метках, связанную с низкой мотивацией студентов делиться цифровым следом [12].

Этическую сторону сбора цифрового следа характеризует исследование, проведенное группой ученых по получению географических данных пользователя [12]. В этой работе показано, что сбор цифрового следа о перемещении человека помогает лучше понять его цифровой профиль, который потом может быть использован для различного рода персонализаций деятельности, в которой он принимает участие. В то же время этот сбор является угрозой безопасности, так как дополнительная информация о передвижениях человека может стать целью кибератак или быть использована в криминальных целях.

Примером пользы цифрового следа в образовании может служить работа Боккони и Трентина, в которой они рассматривают смешанные способы сбора цифрового следа для высшего образования с помощью мобильных и сетевых технологий. Остается неизученным вопрос, может ли преподаватель корректировать и направлять группу студентов на основе собранного ими цифрового следа [1, 15].

Операционная система Android позволяет использовать встроенную в телефон камеру для фотографирования и видеосъемки, встроенный микрофон – для записи аудио, а также датчик GPS – для отслеживания текущей геолокации.

Несмотря на то, что Android является открытой операционной системой, она, тем не менее, дает возможность сохранять файлы в папку приложения. В случае такого сохранения файлы, созданные этим приложением, не будут видны в общем списке файлов и не доступны другим программам для чтения и редактирования.

Таким образом, для сохранения файлов использована вышеупомянутая возможность. Для сохранения геолокации использована база данных Room, интегрируемая в Android.

Благодаря этой технологии можно упростить разработку приложения, а также улучшить пользовательский опыт при использовании приложения.

С использованием вышеперечисленных инструментов разработки нами создано мобильное приложение для Android, позволяющее авторизованному пользователю оставлять цифровой след.

Для реализации системы, помогающей оценить успехи обучающегося на основании полученного цифрового следа, главной задачей является проведение анализа собранного цифрового следа [4, 6]. При этом, после анализа цифрового следа, должен быть сформирован отчет по каждому студенту, на основании которого можно делать вывод о его личных образовательных успехах, а также о том, кто наиболее из его одногруппников наиболее сильно повлиял на улучшение его успеваемости.

Важными видами собираемого цифрового следа являются аудиозаписи и видеозаписи, так как они позволяют фиксировать речь студентов, а также являются видами цифрового следа [5]

Для реализации такой системы был составлен следующий алгоритм анализа сбора цифрового следа:

- извлечение текста из аудио и видео записей;
- грамматический разбор и анализ ошибок, а также анализ тональности текстов;

- выявление инфлюенсеров - студентов, частота взаимодействия с которыми наиболее сильно сказывается на медиане тональности или числе грамматических ошибок.

На основании полученного результата формируется отчет, показывающий информацию об успехах обучающихся, Отчет показывает количество грамматических ошибок и распределение тональности текста для одного обучающегося, группы обучающихся и всех обучающихся. Также в нем указаны инфлюенсеры – студенты, частота взаимодействия с которыми наиболее сильно сказывается на медиане тональности или числе грамматических ошибок

Чтобы извлечь текст из аудиозаписи, была применена предобученная модель машинного обучения для распознавания текста с использованием набора Google Cloud Speech.

Google Cloud Speech – набор инструментов, позволяющий разработчикам преобразовывать звук в текст, применяя модели нейронных сетей.

Примененная модель распознавания текста из речи работает для всех аудиофайлов в формате «.wav», имеющих один канал и частоту дискретизации 16000 Гц. Так как микрофон мобильных телефонов позволяет записывать текст при таких параметрах, то для распознавания аудиозаписи не требуется никакой предобработки.

Чтобы распознать речь из видеозаписи, вместо поиска другой модели распознавания был реализован механизм конвертации видеозаписи в аудиозапись. Подавляющее большинство смартфонов записывает по умолчанию двухканальное видео в формате mp4 и не поддерживает запись в моно-режиме из коробки, так что для извлечения из видеозаписи аудиодорожки в «.wav» формате ее также необходимо предварительно обработать, сделав моноканальной, а также уменьшив частоту дискретизации, которая по умолчанию составляет 44100 Гц, чтобы модель могла извлечь текст из полученной аудиозаписи.

Следующим шагом проведения анализа являются грамматический разбор и анализ тональностей. Для выполнения этих процедур были использованы предобученные модели машинного обучения «transformers» и «language_tool».

Transformers – модель, позволяющая классифицировать текст по его тональности как нейтральный, позитивный или негативный [16].

Language Tool – модель, позволяющая провести грамматический разбор текста, вычленив допущенные ошибки и классифицировав виды этих ошибок [16].

С помощью моделей transformers и language_tool распознанный ранее текст был проанализирован на грамматические ошибки и тональность, таким образом выявив эти параметры для соответствующего пользователя.

Также строится график распределения тональности.

Последним шагом при проведении анализа является выявление инфлюенсеров. Для каждого пользователя с помощью ранее отсканированного QR-кода и выявленного с помощью GPS местоположения был определен список всех, с кем пользователь очно взаимодействовал. На основании этих списков в системе формируется список наиболее значимых других пользователей. Для формирования такого списка вычисляется среднее количество взаимодействий всех пользователей. После этого вычисляются среднее количество ошибок, допущенных пользователем за время обучения, а также медиана распределения тональности его речи. После этого пользователь становится инфлюенсером, если для большинства других пользователей выполняется следующее условие: если количество взаимодействий с пользователем больше(меньше) среднего количества взаимодействий среди всех обучающихся, а количество ошибок и медиана распределения тональности ниже(выше) чем средняя для пользователя, то такой пользователь является инфлюенсером. После выявления списка инфлюенсеров среди всех обучающихся они заносятся в отчет, который затем отображается по каждому пользователю в администраторской панели разрабатываемой системы.

РЕЗУЛЬТАТЫ

На основании изучения различных параметров технологий и методов, которые могут быть использованы для получения цифрового следа, был разработан и протестирован следующий прототип системы (Рис. 2):

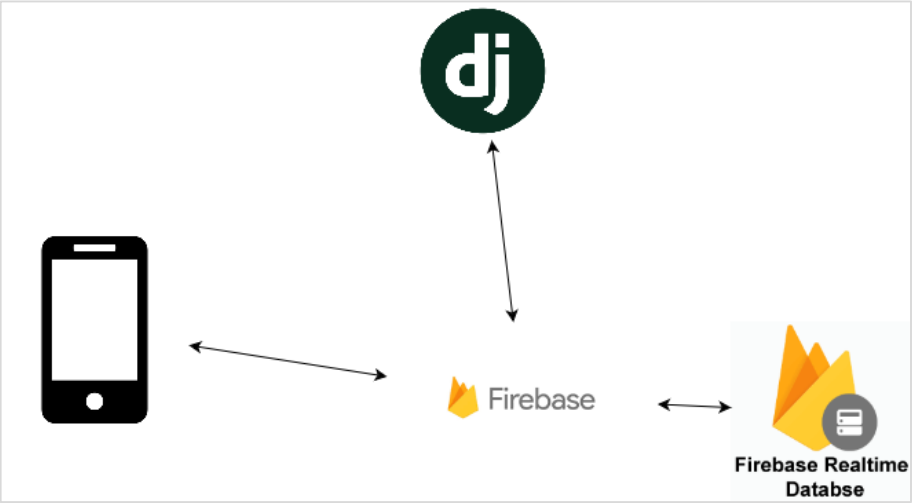

Рис. 2. Общая схема прототипа системы

Этапы работы системы:

1. С помощью сервиса Firebase пользователь авторизуется в мобильном приложении.

2. Мобильное приложение фиксирует цифровой след и отправляет в его Firebase, где он фиксируется и записывается в базу данных Realtime Database.

3. Администраторская панель, в качестве которой выступает Django, соединяется с firebase и получает данные о списке пользователей, которые авторизовались в приложении.

4. На основании полученного цифрового следа сервер вычисляет для каждого студента параметры: активность, основанная на количестве цифровых следов, посещаемость, основанная на определенном местоположении (GPS и QR-коды) в определенный промежуток времени и частые заметки, основанные на самых часто встречающихся словах в заметках, перечень грамматических ошибок в порядке убывания на основе аудио- и видеозаписей, медианную тональность и то, является ли пользователь инфлюенсером.

5. На основе полученных параметров сервер генерирует отчет (Рис. 3) по каждому студенту, понятный пользователю.

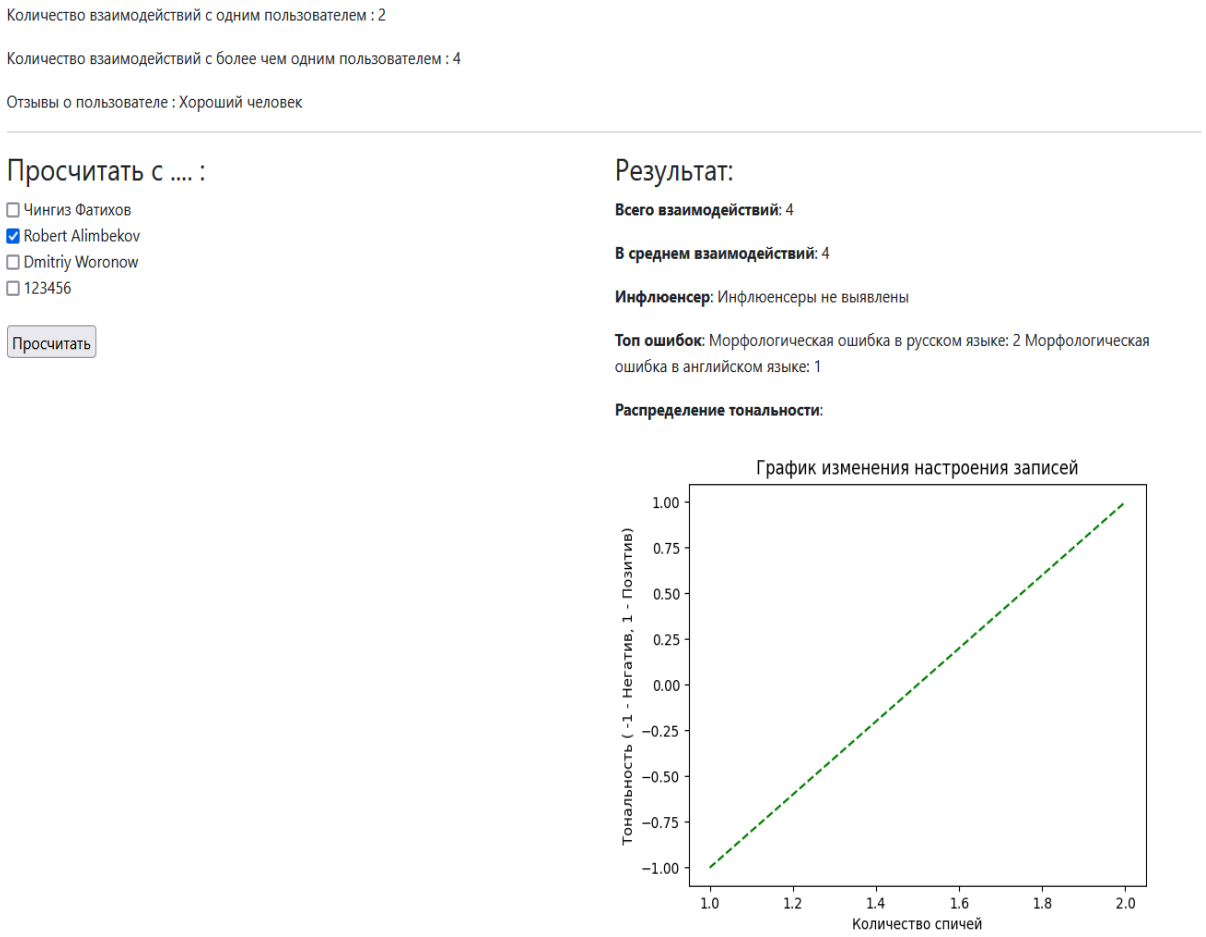

Рис. 3. Результат работы системы

ЗАКЛЮЧЕНИЕ

Разработанное мобильное приложение позволяет пользователю оставлять цифровой след при обучении русскому или английскому языкам. Разработанное веб-приложение позволяет анализировать собранный мобильным приложением цифровой след и формировать отчет, содержащий ключевые параметры для оценки преподавателем результатов обучения. Разработанные приложения могут быть использованы в образовательных учреждениях.

СПИСОК ЛИТЕРАТУРЫ

1. *Bruffee K.A.* Collaborative learning: Higher education, interdependence, and the authority of knowledge. Johns Hopkins University Press, 2715 North Charles Street, Baltimore, MD 21218-4363, 1999.
URL: https://eric.ed.gov/?id=ed430508.

2. *Daniel J., Kanwar A., Uvalić-Trumbić S.* A tectonic shift in global higher education //Change: The magazine of higher learning. 2006. Vol. 38. No. 4. P. 16–23. URL: https://www.tandfonline.com/doi/pdf/10.3200/CHNG.38.4.16-23.

3. *Altbach P.G., Reisberg L., Rumbley L.E.* Trends in global higher education: Tracking an academic revolution. Brill, 2019. 270 p. URL: https://books.google.ru/books?hl=ru&lr=&id=-t-mDwAAQBAJ&oi=fnd&pg=PP3&dq=global+higher+education&ots=rGE74gmJnN&sig=gd8d4-lllnRYlppv8T7V-6Om950&redir_esc=y#v=onepage&q=global%20higher%20education&f=false.

4. *Machekhina O.N.* Digitalization of education as a trend of its modernization and reforming // Revista Espacios. 2017. Vol. 38. No. 40.
URL: http://www.revistaespacios.com/a17v38n40/17384026.html.

5. *Galimova E.G. et al.* Digital Educational Footprint as a Way to Evaluate the Results of Students' Learning and Cognitive Activity in the Process of Teaching Mathematics // Eurasia Journal of Mathematics, Science and Technology Education. 2019. Vol. 15. No. 8. URL: https://www.ejmste.com/download/digital-educational-footprint-as-a-way-to-evaluate-the-results-of-students-learning-and-cognitive-7689.pdf.

6. *Buchanan R. et al.* Expert insights into education for positive digital footprint development // Scan: The Journal for Educators. 2018. Vol. 37. P. 49–64. URL: https://search.informit.org/doi/abs/10.3316/informit.243445474304969.

7. *Baranova E., Shvetsov G., Noskova T.* Educational Data Mining Methods for the Analysis of Student’s Digital Footprint // CEUR Workshop Proceedings. 2021. Vol. 2920. P. 44–58. URL: http://ceur-ws.org/Vol-2920/paper_4.pdf

8. EdCrunch Томск: материалы международной конференции по новым образовательным технологиям, Томск, 2019. URL: vital.lib.tsu.ru/vital/access/services/Download/vtls:000661888/SOURCE1.

9. *Мантуленко В.* Перспективы использования цифрового следа в высшем образовании, Преподаватель XXI век, 2020. URL: https://cyber-leninka.ru/article/n/perspektivy-ispolzovaniya-tsifrovogo-sleda-v-vysshem-obrazovanii.

10. *Camacho M., Minelli J., Grosseck G.* Self and identity: Raising undergraduate students' awareness on their digital footprints // Procedia-Social and Behavioral Sciences. 2012. Vol. 46. P. 3176–3181.
URL: https://www.sciencedirect.com/science/article/pii/S1877042812017685

11. *Sevillano-Garcia M.L., Vázquez-Cano E*. The Impact of Digital Mobile Devices in Higher Education // Educational Technology & Society. 2015. No. 1. P. 106–118. URL: https://www.jstor.org/stable/jeductechsoci.18.1.106.

12. *Gibson D. et al*. Digital badges in education // Education and Information Technologies. 2015. Vol. 20. No. 2. – P. 403–410.
URL: https://link.springer.com/article/10.1007/s10639-013-9291-7.

13. *Boase J.* Augmenting Survey and Experimental Designs with Digital Trace Data // Communication Methods and Measures. 2016. P. 165–166.
URL: https://www.tandfonline.com/doi/full/10.1080/19312458.2016.1150975

14. *Höhnle S., Michel B., Glasze G., Uphues R.* Digital geodata traces – new challenges for geographic education // International Research in Geographical and Environmental Education. 2013. P. 97–108.
URL: https://www.tandfonline.com/doi/full/10.1080/10382046.2013.778713?scroll=top&needAccess=true

15. *Bocconi S., Trentin G.* Modelling blended solutions for higher education: teaching, learning, and assessment in the network and mobile technology era // Educational Research and Evaluation. 2015. P. 516–535.
URL: https://www.tandfonline.com/doi/full/10.1080/13803611.2014.996367.

16. *Liu D.Y.T. et al.* Data-Driven Personalization of Student Learning Support in Higher Education // Learning Analytics: Fundaments, Applications, and Trends. 2017. P. 143–169. URL: https://link.springer.com/chapter/10.1007/978-3-319-52977-6_5 Data-Driven Personalization of Student Learning Support in Higher Education.

17. *Nakamoto I. et al.* A qr code–based contact tracing framework for sustainable containment of covid-19: Evaluation of an approach to assist the return to normal activity // JMIR mHealth and uHealth. 2020. Vol. 8. No. 9. P. e22321. URL: https://mhealth.jmir.org/2020/9/e22321.

18. IOS. URL: https://www.apple.com/ru/ios/ios-15/

19. Android. URL: https://www.android.com/intl/ru_ru/

20. Tizen. URL: https://www.tizen.org/

21. Aurora. URL: https://auroraos.ru/

22. Harmony OS. URL: https://www.harmonyos.com/en/

A MOBILE SYSTEM FOR COLLECTING A DIGITAL TRACE FOR THE TASK OF ACCOUNTING AND ANALYZING HORIZONTAL LEARNING IN THE LEARNING PROCESS WITHOUT USING A CELLULAR CONNECTION

R. R. Alimbekov[1][0000-0002-9306-8463], A. F. Khasianov[2][0000-0002-1819-593X]

[1, 2] *Kazan (Volga Region) Federal University, 35 Kremlevskaya str., Kazan, 420008*

[1]arr1998@gmail.com, [2]ak@it.kfu.ru

Abstract

Today, users of mobile applications in different areas leave a huge amount of digital footprint. The main types of digital footprints are text, photos, videos, audio, and current location. To assist the teacher in horizontal learning, a mobile application that collects all of the above types of digital footprint was developed as well as web application that analyzes it.

Key words: cellular communication, mobile application, digital footprint, digital footprint collection, accounting, analysis.

REFERENCES

1. *Bruffee K.A.* Collaborative learning: Higher education, interdependence, and the authority of knowledge. Johns Hopkins University Press, 2715 North Charles Street, Baltimore, MD 21218-4363, 1999.
URL: https://eric.ed.gov/?id=ed430508.

2. *Daniel J., Kanwar A., Uvalić-Trumbić S.* A tectonic shift in global higher education //Change: The magazine of higher learning. 2006. Vol. 38. No. 4. P. 16–23. URL: https://www.tandfonline.com/doi/pdf/10.3200/CHNG.38.4.16-23.

3. *Altbach P.G., Reisberg L., Rumbley L.E.* Trends in global higher education: Tracking an academic revolution. Brill, 2019. 270 p.
URL: https://books.google.ru/books?hl=ru&lr=&id=-t-mDwAAQBAJ&oi=fnd&pg=PP3&dq=global+higher+education&ots=rGE74gmJnN&sig=gd8d4-lllnRYlppv8T7V-6Om950&redir_esc=y#v=onepage&q=global%20higher%20education&f=false.

4. *Machekhina O.N.* Digitalization of education as a trend of its modernization and reforming // Revista Espacios. 2017. Vol. 38. No. 40.
URL: http://www.revistaespacios.com/a17v38n40/17384026.html.

5. *Galimova E.G. et al.* Digital Educational Footprint as a Way to Evaluate the Results of Students' Learning and Cognitive Activity in the Process of Teaching Mathematics // Eurasia Journal of Mathematics, Science and Technology Education. 2019. Vol. 15. No. 8. URL: https://www.ejmste.com/download/digital-educational-footprint-as-a-way-to-evaluate-the-results-of-students-learning-and-cognitive-7689.pdf.

6. *Buchanan R. et al.* Expert insights into education for positive digital footprint development // Scan: The Journal for Educators. 2018. Vol. 37. P. 49–64. URL: https://search.informit.org/doi/abs/10.3316/informit.243445474304969.

7. *Baranova E., Shvetsov G., Noskova T.* Educational Data Mining Methods for the Analysis of Student's Digital Footprint // CEUR Workshop Proceedings. 2021. Vol. 2920. P. 44–58. URL: http://ceur-ws.org/Vol-2920/paper_4.pdf

8. EdCrunch Tomsk: materialy mezhdunarodnoj konferencii po novym obrazovatel'nym tekhnologiyam, Tomsk, 2019. URL: vital.lib.tsu.ru/vital/access/services/Download/vtls:000661888/SOURCE1.

9. *Mantulenko V.* Perspektivy ispol'zovaniya cifrovogo sleda v vysshem obrazovanii, Prepo-davatel' HKHI vek, 2020. URL: https://cyberleninka.ru/article/n/perspektivy-ispolzovaniya-tsifrovogo-sleda-v-vysshem-obrazovanii.

10. *Camacho M., Minelli J., Grosseck G.* Self and identity: Raising undergraduate students' awareness on their digital footprints // Procedia-Social and Behavioral Sciences. 2012. Vol. 46. P. 3176–3181.
URL: https://www.sciencedirect.com/science/article/pii/S1877042812017685

11. *Sevillano-Garcia M.L., Vázquez-Cano E*. The Impact of Digital Mobile Devices in Higher Education // Educational Technology & Society. 2015. No. 1. P. 106–118. URL: https://www.jstor.org/stable/jeductechsoci.18.1.106.

12. *Gibson D. et al*. Digital badges in education // Education and Information Technologies. 2015. Vol. 20. No. 2. P. 403–410.
URL: https://link.springer.com/article/10.1007/s10639-013-9291-7.

13. *Boase J.* Augmenting Survey and Experimental Designs with Digital Trace Data // Communication Methods and Measures. 2016. P. 165–166.
URL: https://www.tandfonline.com/doi/full/10.1080/19312458.2016.1150975

14. *Höhnle S., Michel B., Glasze G., Uphues R.* Digital geodata traces – new challenges for geographic education // International Research in Geographical and Environmental Education. 2013. P. 97–108.
URL: https://www.tandfonline.com/doi/full/10.1080/10382046.2013.778713?scroll=top&needAccess=true

15. *Bocconi S., Trentin G.* Modelling blended solutions for higher education: teaching, learning, and assessment in the network and mobile technology era // Educational Research and Evaluation. 2015. P. 516–535.
URL: https://www.tandfonline.com/doi/full/10.1080/13803611.2014.996367.

16. *Liu D.Y.T. et al.* Data-Driven Personalization of Student Learning Support in Higher Education // Learning Analytics: Fundaments, Applications, and Trends. 2017. P. 143–169. URL: https://link.springer.com/chapter/10.1007/978-3-319-52977-6_5 Data-Driven Personalization of Student Learning Support in Higher Education.

17. *Nakamoto I. et al.* A qr code–based contact tracing framework for sustainable containment of covid-19: Evaluation of an approach to assist the return to normal activity // JMIR mHealth and uHealth. 2020. Vol. 8. No. 9. P. e22321.
URL: https://mhealth.jmir.org/2020/9/e22321.

18. IOS. URL: https://www.apple.com/ru/ios/ios-15/

19. Android. URL: https://www.android.com/intl/ru_ru/

20. Tizen. URL: https://www.tizen.org/

21. Aurora. URL: https://auroraos.ru/

22. Harmony OS. URL: https://www.harmonyos.com/en/

СВЕДЕНИЯ ОБ АВТОРАХ

АЛИМБЕКОВ Роберт Ринатович – магистрант, Казанский (Приволжский) федеральный университет, г. Казань.

Robert Rinatovich ALIMBEKOV – Master's student, Kazan (Volga region) Federal University, Kazan.

Email: arr1998@gmail.com
ORCID: 0000-0002-9306-8463

ХАСЬЯНОВ Айрат Фаридович – PhD (физико-математические науки), заведующий кафедрой программной инженерии, Казанский (Приволжский) федеральный университет, г. Казань.

Airat Faridovich Khasyanov – PhD (Physical and Mathematical Sciences), Head of the Department of Software Engineering, Kazan (Volga region) Federal University, Kazan.

Email: ak@it.kfu.ru
ORCID: 0000-0002-1819-593X

Материал поступил в редакцию 2 июня 2022 года

УДК 004

РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ ПО ПОСТРОЕНИЮ АРХИТЕКТУРЫ ПРОГРАММНЫХ ПРОДУКТОВ

А. Е. Гришин[1] [0000-0002-7355-4878], К. А. Григорян[2] [0000-0001-6470-1832]

[1, 2]Казанский (Приволжский) федеральный университет, ул. Кремлевская, 35, г. Казань, 420008

[1]andrey.grishin.work@gmail.com, [2]karigri@yandex.ru

Аннотация

Статья посвящена автоматизации этапа проектирования программного обеспечения. Проанализированы причины высокого значения данного этапа и актуальность его автоматизации. Рассмотрены основные стадии названного этапа и существующие системы, позволяющие автоматизировать каждую из них. Предложено собственное решение в рамках задачи рефакторинга структуры классов на основе метода комбинаторной оптимизации. Разработан и протестирован на реальной модели метод решения, позволяющий улучшить качество иерархии классов.

***Ключевые слова**: автоматизация, проектирование, рефакторинг, архитектура ПО, ООП, оптимизация*

ВВЕДЕНИЕ

В последние несколько лет наблюдается стремительный рост объема рынка разработки программного обеспечения (ПО) [1], увеличивается количество разрабатываемых программных продуктов (ПП) и повышается спрос на квалифицированных специалистов в этой области. В условиях большого объема работы и ограниченности трудовых ресурсов оптимизация экономической эффективности процесса разработки становится необходимостью.

Одним из наиболее важных в процессе разработки ПО является этап проектирования, который представляет собой формализованное описание внутренних и внешних свойств разрабатываемой системы, а также того, как система взаимодействует с внешними компонентами и конечными пользователями [2]. Вместе с

языком программирования, фреймворком, базой данных и другими программными компонентами все вышеперечисленное составляет архитектуру ПО.

Значимость проектирования заключается в том, что на основе решений, принятых на этом этапе, осуществляется вся последующая разработка [3, 4]. Соответственно, при качественно проведенном проектировании можно получить следующие организационные преимущества:

- правильную оценку времени и стоимости разработки;
- увеличение вероятности избежать дополнительных доработок и согласований требований;
- возможность избежать неудовлетворенности конечным результатом и разногласий между заказчиком и исполнителем.

Кроме того, правильно спроектированная архитектура позволяет добиться и технических преимуществ:

- снижения скорости накопления технического долга;
- ускорения процесса разработки;
- уменьшения вероятности возникновения дефектов.

В каждом из перечисленных пунктов можно достичь и обратного эффекта, если совершить большое количество ошибок на данном этапе или же вовсе пренебречь им [3, 4]. Суммарно все полученные негативные эффекты будут выражаться в увеличении сроков и стоимости разработки, что всегда является крайне нежелательным.

Сам по себе этап проектирования является затратным в плане экономических и временных ресурсов, так как, в зависимости от размера разрабатываемой системы, его продолжительность может варьироваться от коротких до очень длительных сроков [5]. Также, ввиду высокой цены ошибки на данном этапе, предъявляются строгие требования к квалификации специалистов, вовлеченных в него [5]. Перспектива полной или частичной автоматизации этого процесса позволит сократить издержки и снизить влияние человеческого фактора на конечные результаты. Таким образом, на сегодняшний день задача автоматизации проектирования является крайне актуальной.

ОБЗОР ЛИТЕРАТУРЫ

Перед чем начать рассматривать аналоги, описанные в литературе, определим, из чего состоит процесс проектирования. Упрощенно его можно разделить на три основных этапа: анализ требований, непосредственно разработка архитектуры и преобразование полученных теоретических программных компонентов в реальные.

Анализ требований традиционно выделяют в качестве отдельного этапа жизненного цикла ПО. Однако он неразрывно связан и с проектированием, ведь именно на основе бизнес-требований формируются программные компоненты и их связи друг с другом, а также модели архитектуры системы, технологии разработки и реализации. Рассмотрим системы, позволяющие автоматизировать этап анализа требований.

TOVE (TOronto Virtual Enterprise) [6]. Этот проект позволяет создавать универсальные и повторно используемые корпоративные модели данных. Авторы разработали несколько адаптивных онтологий, которые можно использовать для собственных проектов, в зависимости от предметной области. Каждая онтология оснащена поддержкой общей терминологии предметной области с определениями каждого термина, поддержкой семантики в наборе аксиом, позволяющей автоматически генерировать связи между компонентами в проекте, а также многими другими характеристиками. Реализован данный программный комплекс в большей степени на языке Prolog.

Marrying Ontology and Software Technologies (MOST) [7]. Проект направлен на улучшение качества разработки программного обеспечения с помощью использования онтологий и технологии рассуждения. Для достижения этой цели авторы планируют реализовать бесшовную интеграцию технологии онтологий в разработку ПО на основе моделей MDSD. В результате разработка будет вестись на основе онтологий, что в свою очередь ускорит процесс проектирования и снизит вероятность несогласованностей в требованиях. В данный момент проект находится на стадии реализации.

KOntoR [8]. Этот проект реализует основанный на онтологии подход к повторному использованию программного обеспечения. Авторы задались целью

при помощи базовых знаний, представленных в виде онтологий, повысить ценность повторно используемых библиотек. Это было достигнуто путем семантической интеграции явных и неявных метаданных, что обеспечило средства для получения новых фактов. Иными словами, при возникновении требований, которые ранее уже были проанализированы и для которых уже был подобран соответствующий программный комплекс, система будет автоматически связывать их с уже разработанными решениями.

SEON: The Software Engineering Ontology Network [9]. Данный проект фактически является хорошо обоснованной сетью эталонных онтологий и механизмов для получения и включения в сеть новых интегрированных онтологий предметной области. Благодаря этому достигается возможность многократного использования этих онтологий в качестве шаблонов для других.

Заключительный этап проектирования — преобразование полученных формальных структур в реальные программные компоненты, то есть кодогенерация. В отличие от предыдущих этапов данная стадия, как правило, не требует высокого уровня квалификации, абстрактного мышления или практических навыков в области проектирования архитектуры ПО. Она представляет собой преимущественно реализацию уже разработанного проекта на том или ином технологическом стеке. В результате формализованности данного этапа он наиболее часто подвергается автоматизации. Рассмотрим системы, позволяющие автоматизировать преобразования формально описанных структур в реальные программные компоненты.

UML/Code Generation Tool [10]. Является инструментом для генерации кода из унифицированного языка моделирования UML, который работает на Windows, Linux и MacOS X. Этот инструмент дает средство моделирования, которое включает диаграммы UML, такие как диаграммы вариантов использования, классов, последовательностей, связей. Помимо этого, он позволяет генерировать код на C++, Java, Idl, PHP, Python и MySQL или импортировать код в диаграммы. Последняя его версия была выпущена в июле 2018 года.

BPwin [11]. Это программный продукт, разработанный компанией ltd. Logic Works. Он предназначен для поддержки процесса создания информационных систем. Относится к категории CASE средств верхнего уровня. Данный инструмент

является достаточно развитым средством моделирования, позволяющим проводить анализ, документирование и улучшение бизнес-процессов. С его помощью можно моделировать действия в процессах, определять их порядок и необходимые для них ресурсы. Модели BPwin создают структуру, необходимую для понимания бизнес-процессов, выявления управляющих событий и порядка взаимодействия элементов процесса между собой.

Erwin Data Modeler [11]. Это программное обеспечение для проектирования и документирования баз данных. Модели данных помогают визуализировать структуру данных, обеспечивая эффективный процесс организации, управления и администрирования таких аспектов деятельности предприятия, как уровень сложности данных, технологии баз данных и среды развертывания. По своей сути Erwin является CASE-средством. Пользователи могут использовать Erwin Data Modeler как способ создания концептуальной модели данных или создания логической модели, не зависящей от конкретной технологии базы данных. В дальнейшем эта схематическая модель может быть использована для создания физической модели данных.

Design/IDEF [12]. Это CASE-пакет, который по заверениям авторов автоматизирует многие этапы проектирования сложных систем различного назначения: формулировку требований и целей проектирования, разработку спецификаций, определение компонентов и взаимодействий между ними, документирование проекта, проверку его полноты и непротиворечивости. Наиболее успешно пакет применяется для описания и анализа деятельности предприятия. Он позволяет оценить такую структуру, как единую сущность, сочетающую в себе управленческие, производственные и информационные процессы. В основе пакета лежит методология структурного проектирования и анализа сложных систем IDEF0/SADT.

Silverrun [13]. Данное CASE-средство американской фирмы Computer Systems Advisers, Inc. используется для анализа и проектирования больших информационных систем и ориентировано в большей степени на спиральную модель жизненного цикла. Оно применимо для поддержки любой методологии, основанной на раздельном построении функциональной и информационной моделей (диаграмм потоков данных и диаграмм «сущность–связь»).

Rational Rose [14]. Данное ПО является еще одним CASE-средством проектирования и разработки информационных систем и программного обеспечения для управления предприятиями. Как и другие вышеперечисленные CASE-средства, его можно применять для анализа и моделирования бизнес-процессов. Принципиальное отличие Rational Rose от других средств заключается в объектно-ориентированном подходе. Графические модели, создаваемые с его помощью, основаны на объектно-ориентированных принципах и языке UML. Данный инструмент моделирования позволяет разработчикам создавать целостную архитектуру процессов предприятия, сохраняя все взаимосвязи и управляющие методы между различными уровнями иерархии.

Vantage Team Builder [15]. Этот интегрированный программный продукт ориентирован на реализацию каскадной модели жизненного цикла ПО. В качестве отличительных особенностей этой системы можно выделить возможность программирования на языке C со встроенным SQL и многопользовательский доступ к репозиторию проекта, который осуществляется за счет возможности работы приложения в конфигурации «клиент–сервер».

Нетрудно заметить, что среди аналогов не представлены решения, позволяющие автоматизировать основной этап проектирования, а именно, разработку архитектуры. На данный момент не существует систем, позволяющих автоматизировать ее непосредственную разработку в той же мере, в которой ее выполняет соответствующий специалист.

МЕТОДОЛОГИЯ

В чем же заключаются причины проблем автоматизации проектирования? Деятельность в этом направлении ведется уже более сорока лет [16], и были достигнуты определенные успехи в этой области, описанные в предыдущей главе. Однако проблема так и не была решена в полной мере: на текущий момент, основываясь только на бизнес-требованиях, нельзя получить архитектурную модель, готовую к дальнейшей разработке. Происходит это по разным причинам.

Во-первых, разработка качественной архитектурной модели невозможна без нетривиальных знаний и опыта программного архитектора, а именно, понимания того, какие из требований нуждаются в соответствующих программных решениях. Проблема заключается в том, что разная совокупность таких требований может

быть связана с разными программными компонентами, и попытка перебрать все возможные комбинации будет приводить к комбинаторному взрыву. Кроме того, сами выбранные программные компоненты должны корректно взаимодействовать между собой. И плюс ко всему – программные компоненты имеют разный уровень представления, например, имеется ряд баз данных, для каждой из которых существует множество драйверов для множества языков программирования, в каждом из которых существуют свои паттерны представления данных и обеспечения доступа к ним. Выбор каждого из этих пунктов должен быть аргументирован наличием соответствующих бизнес-требований.

Во-вторых, уже на текущий момент развития отрасли разработки программного обеспечения количество программных компонентов и подходов к ним достигло огромных масштабов. Понятие разработки постоянно делится на специализации, каждая из которых имеет свою тенденцию развития с точки зрения как программной, так и методологической частей. Даже просто учесть их все и структурировать между собой представляется сложной задачей.

Третья проблема заключается в скорости развития технологий. Теоретически, решив две проблемы, описанные выше, и начав разработку подобной системы на основе текущей ситуации в отрасли, по ее окончании можно с высокой вероятностью получить неактуальный результат. Причина этого – высокая скорость появления на рынке новых программных решений, некоторые из которых могут изменять саму концепцию разработки в той или иной области. Таким образом, встает задача постоянного встраивания новых решений в уже разработанную систему.

На основе проведенного исследования мы приняли решение провести разработку в такой области этапа проектирования, как рефакторинг структуры классов. Подавляющее большинство достаточно больших систем программного обеспечения реализуется с помощью подхода ООП [17]. Если говорить о его преимуществах, то следует выделить гибкость, экономию времени разработчиков по мере развития системы и сокращение общей кодовой базы, что не позволяет проекту засоряться. Из недостатков имеет смысл сказать о достаточно сложном старте разработки в сравнении с процедурным подходом и некоторое снижение производительности работы программы. Однако следует сделать оговорку, что в

последнее время набирает популярность микросервисная архитектура [18], в основе которой часто используют языки с меньшим уровнем абстракций. Тем не менее вопрос проектирования качественной иерархии классов все еще актуален.

Однако даже такая задача является достаточно сложной, ведь задачи объектно-ориентированного проектирования часто могут быть противоречивыми и зависеть от контекста. Решаются они на основе опыта программного архитектора и его понимания бизнес-задачи. Однако существует определенный класс свойств объектно-ориентированной иерархии, характерных для подавляющего большинства систем. Набор таких свойств возьмем из работы [19]:

- количество абстрактных суперклассов;
- количество повторяющихся методов;
- количество неиспользуемых методов;
- количество безликих классов.

Именно эти свойства были выбраны потому, что они являются максимально непротиворечивыми, а также измеримыми. Однако при желании ничего не мешает добавить дополнительные свойства или же изменить существующие.

Далее необходимо определить, с помощью каких действий будет происходить рефакторинг архитектуры. Сохраняя баланс между эффективностью и реализуемостью, мы выбрали следующие действия:

- Перемещение методов по иерархии;
  - переместить метод в суперкласс;
  - переместить метод в дочерний класс;
- Управление подклассами;
  - извлечь подкласс из суперкласса;
  - объединить подкласс с суперклассом;
- Управление абстракциями;
  - сделать класс абстрактным;
  - сделать класс конкретным;
- Удаление
  - удалить класс;
  - удалить метод.

Обозначив методы рефакторинга и свойства архитектуры, необходимо также определить, как оценивать полученный результат. Составив функцию, зависимую от числового выражения вышеперечисленных свойств, можно получить

$$q_d = \sum_{m=1}^{n} w_m metric_m(d). \quad (1)$$

В формуле (1): q_d – значение качества полученной архитектуры; w_m – вес метрики m; $metric_m(d)$ – значение свойства m в архитектуре d. Отсюда следует вывод, что свойства или же метрики должны быть взвешенными. Веса были расставлены в таблице 1 в соответствии с описанием приоритетов того или иного свойства в работе [19].

Таблица 1. Описание весов метрик

Метрика	Количество абстрактных суперклассов	Количество повторяющихся методов	Количество неиспользуемых методов	Количество безликих классов
Вес	3	2	1	-1

Последним шагом являлся выбор того, каким образом будет осуществляться переход между вариантами архитектуры. Поскольку задача фактически сводится к задаче глобальной оптимизации, необходимо было выбрать алгоритмический метод решения задач соответствующего класса [20]. Выбор был сделан в пользу метода имитации отжига [21]. В целом для решения данной задачи можно было использовать и любой другой похожий алгоритм, однако обычные стохастические методы поиска в этом случае требовали бы крайне много времени для получения результата [22]. В добавление к этому отметим, что данный метод обладает более успешными возможностями выхода из локальных минимумов. В качестве недостатка можно отметить, что не будет возможности доказать оптимальность полученного решения. Однако, поскольку в данном случае достаточно будет

найти качественное решение за ограниченное время, а не самое лучшее, этого алгоритма будет вполне достаточно.

РЕЗУЛЬТАТЫ

Для проверки качества работы полученного решения была создана небольшая иерархия классов, представленная на рисунке 1. Заглавные буквы в названии класса означают, что данный класс использует указанные методы. Как можно заметить, исходная структура классов имеет следующие недостатки: классы ConcreteAB, ConcreteABC, ConcreteABDE, ConcreteABD наследуют метод f, но не используют его, а метод b дублируется в классах ConcreteABDE, ConcreteAB, ConcreteABC.

Реализовав данный алгоритм на языке Python и применив его на протяжении 1000 эпох к данной структуре, мы получили новую иерархию классов, представленную на рисунке 2. Как можно увидеть, вышеперечисленные проблемы были решены путем добавления новых абстрактных классов AbstractBaseSubClass и AbstractABDE и выносом соответствующих методов в них. Это позволило избежать дублирования методов и добиться того, что классы наследуют только те методы, которые используют.

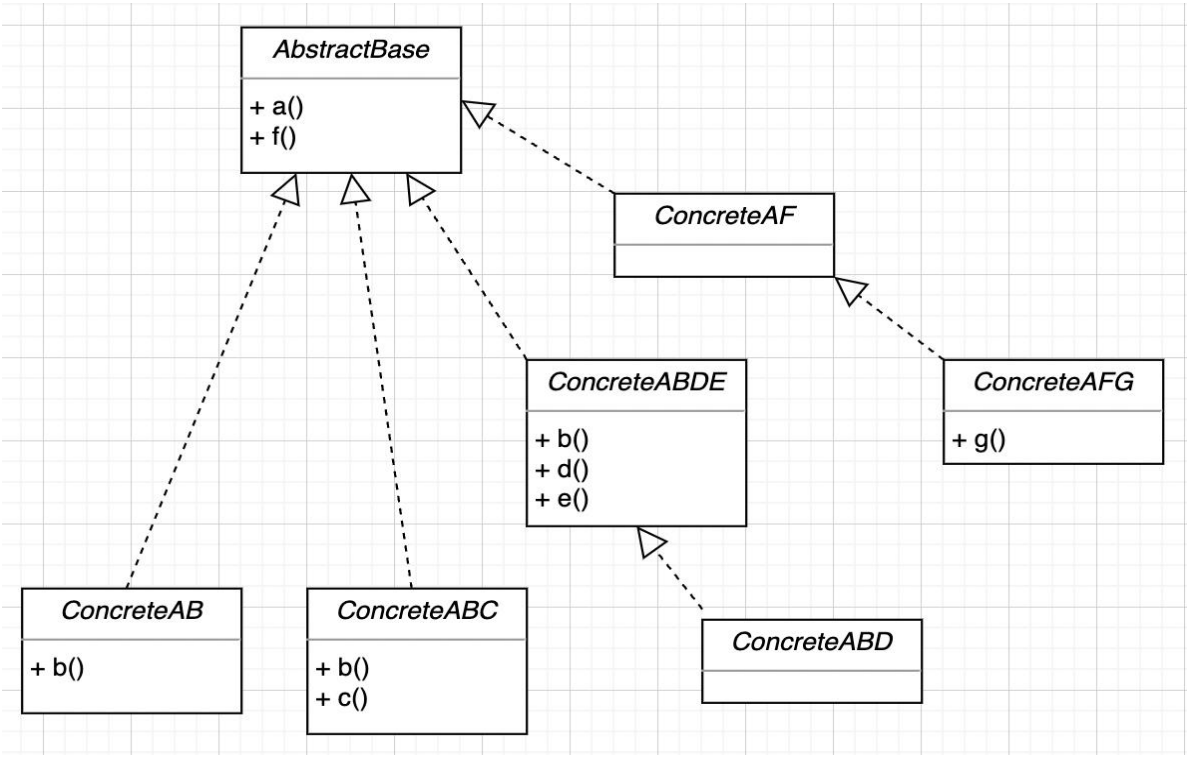

Рис. 1. Изначальная иерархия классов

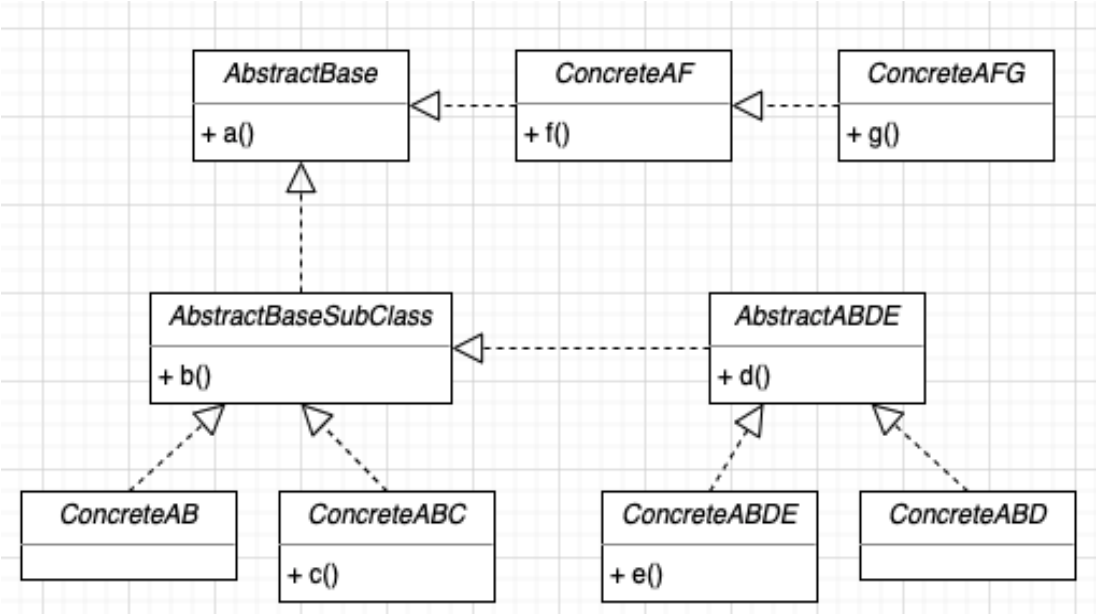

Рис. 2. Конечная иерархия классов

ОБСУЖДЕНИЕ

В результате исследования этапа проектирования ПО были проанализированы текущие способы его автоматизации, а также предложено собственное решение для автоматизации проектирования архитектуры ПО. Предложенное решение реализует автоматизацию рефакторинга структуры классов на основе метода комбинаторной оптимизации, что частично способствует автоматизации проектирования ПО и, как следствие, снижению стоимости этапа проектирования ПО. На основании выбранных метрик качества архитектуры и составленной целевой функции задача была сведена к задаче глобальной оптимизации. После этого на основе примера реальной структуры классов была протестирована работа алгоритма.

Набор метрик, выбранный в данной работе, можно заменить на любой другой, для этого достаточно будет определиться с их составом, расставить приоритеты, выбрав веса, и определить способ их подсчета в структуре классов.

Перспективой развития данной работы можно считать тестирование алгоритма на реальной достаточно крупной структуре классов и оценку его с точки зрения реальной практической пользы, а также подбор дополнительных метрик оценки качества архитектуры и методов ее изменения.

СПИСОК ЛИТЕРАТУРЫ

1. *Водзинская Э.В.* Оценка стоимости компаний российского рынка разработки программного обеспечения методами DCF и EVA // Экономические исследования и разработки. 2016. № 4. С. 163–168.

2. *Щенников А.Н.* Проектирование программного обеспечения для информационных систем. Saarbruken: LAP LAMBERT, 2018. 126 с.

3. *Макконнелл С.* Совершенный код. СПб.: Питер, 2005. 59 с.

4. *Фаулер М.* Архитектура корпоративных программных приложений: Пер. с англ. М.: Издательский дом Вильямс, 2006. 544 с.

5. *Влацкая И.В., Заельская Н.А., Надточий Н.С.* Проектирование и реализация прикладного программного обеспечения: учебное пособие. Оренбург: Оренбургский гос. ун-т, 2015. 118 с.

6. *Fox M.S., Gruninger M.* Enterprise modeling // AI magazine. 1998. Vol. 19. No. 3. 109 p. https://doi.org/10.1609/aimag.v19i3.1399

7. *Miksa K. et al.* Case Studies for Marrying Ontology and Software Technologies // Ontology-Driven Software Development. Springer, Berlin, Heidelberg, 2013. P. 69–94.

8. *Happel H.J. et al.* KOntoR: an ontology-enabled approach to software reuse // In: Proc. of The 18Th Int. Conf. On Software Engineering and Knowledge Engineering. 2006. P. 91.

9. *Borges Ruy F. et al.* SEON: A software engineering ontology network // European Knowledge Acquisition Workshop. Springer, Cham, 2016. P. 527–542.

10. *Chauvel F., Jézéquel J.M*. Code generation from UML models with semantic variation points // International Conference on Model Driven Engineering Languages and Systems. Springer, Berlin, Heidelberg, 2005. P. 54–68.

11. *Маклаков С.В.* BPwin и ERwin. CASE-средства разработки информационных систем. М.: Диалог-мифи, 2001. 121 с.

12. *Lakin R., Capon N., Botten N.* BPR enabling software for the financial services industry // Management services. 1996. Vol. 40. No. 3. P. 18–20.

13. *Gryphon R.* Design better apps with SilverRun // Data Based Advisor. 1994. Vol. 12. No. 1. P. 103–107.

14. *Quatrani T.* Visual modeling with Rational Rose 2000 and UML. Addison-Wesley Professional, Second Edition. Addison Wesley, 2000. 288 p.

15. *Kopyltsov A.V. et al.* Algorithm of estimation and correction of wireless telecommunications quality // 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA). IEEE, 2018. P. 1–4.

16. *Vathsavayi S. et al.* Tool support for software architecture design with genetic algorithms // 2010 Fifth International Conference on Software Engineering Advances. IEEE, 2010. P. 359–366.

17. *Мейер Б.* Объектно-ориентированное программирование и программная инженерия: учебное пособие. 2-е изд., испр. М.: Национальный Открытый Университет «ИНТУИТ», 2016. 286 с.
URL: https://biblioclub.ru/index.php?page=book&id=429034

18. *Джамшиди П. и др.* Микросервисы: пройденный путь и дальнейшие цели // Открытые системы. СУБД. 2018. № 3. С. 19–23.

19. *Riel A.J.* Object-Oriented Design Heuristics. Addison-Wesley Professional; Illustrated edition, 1996. 400 p.

20. *Орлянская И.В.* Современные подходы к построению методов глобальной оптимизации // Исследовано в России. 2002. Т. 5. С. 2097–2108.

21. *Глушань В.М.* Метод имитации отжига // Известия Южного федерального университета. Технические науки. 2003. Т. 31. № 2. С. 148–150.

22. *Матренин П.В., Гриф М.Г., Секаев В.Г.* Методы стохастической оптимизации: учеб. пособие. Новосибирск: Изд-во НГТУ, 2016. 66 с.

DEVELOPMENT OF THE EXPERT SYSTEM FOR BUILDING THE ARCHITECTURE OF SOFTWARE PRODUCTS

Andrey Grishin[1] [0000-0002-7355-4878], Karen Grigoryan[2] [0000-0001-6470-1832]

[1, 2]Kazan (Volga Region) Federal University, 35 Kremlevskaya str., Kazan, 420008

[1]andrey.grishin.work@gmail.com, [2]karigri@yandex.ru

Abstract

The article is devoted to automation of the software design stage. In the course of the study, the reasons for the high importance of this stage and the relevance of its automation were analyzed. The main stages of this stage were also considered and the existing systems that allow automating each of them were considered. In addition, an own solution was proposed within the framework of the problem of class structure refactoring based on the combinatorial optimization method. A solution method has been developed to improve the quality of the class hierarchy and tested on a real model.

Keywords*: automation, design, refactoring, software architecture, OOP, optimization.*

REFERENCES

1. *Vodzinskaya E.V.* Ocenka stoimosti kompanij rossijskogo rynka razrabotki programmnogo obespecheniya metodami DCF i EVA // Ekonomicheskie issledovaniya i razrabotki. 2016. № 4. S. 163–168.
2. *Shchennikov A.N.* Proektirovanie programmnogo obespecheniya dlya informacionnyh sistem. Saarbruken: LAP LAMBERT, 2018. 126 s.
3. *Makkonnell S.* Sovershennyj kod. SPb.: Piter, 2005. 59 s.
4. *Fauler M.* Arhitektura korporativnyh programmnyh prilozhenij: Per. s angl. M.: Izdatel'skij dom Vil'yams, 2006. 544 s.
5. *Vlackaya I.V., Zael'skaya N.A., Nadtochij N.S.* Proektirovanie i realizaciya prikladnogo programmnogo obespecheniya: uchebnoe posobie. Orenburg: Orenburgskij gos. un-t., 2015. 118 s.

6. *Fox M.S., Gruninger M.* Enterprise modeling // AI magazine. 1998. Vol. 19. No. 3. 109 p. https://doi.org/10.1609/aimag.v19i3.1399.

7. *Miksa K. et al.* Case Studies for Marrying Ontology and Software Technologies // Ontology-Driven Software Development. Springer, Berlin, Heidelberg, 2013. P. 69–94.

8. *Happel H.J.* et al. KOntoR: an ontology-enabled approach to software reuse // In: Proc. of The 18Th Int. Conf. on Software Engineering and Knowledge Engineering. 2006. P. 91.

9. *Borges Ruy F. et al.* SEON: A software engineering ontology network // European Knowledge Acquisition Workshop. Springer, Cham, 2016. P. 527–542.

10. *Chauvel F., Jézéquel J.M.* Code generation from UML models with semantic variation points // International Conference on Model Driven Engineering Languages and Systems. Springer, Berlin, Heidelberg, 2005. P. 54–68.

11. *Maklakov S.V.* BPwin i ERwin. CASE-sredstva razrabotki informacionnyh sistem. M.: Dialog-mifi, 2001. 121 s.

12. *Lakin R., Capon N., Botten N.* BPR enabling software for the financial services industry // Management services. 1996. Vol. 40. No. 3. P. 18–20.

13. *Gryphon R.* Design better apps with SilverRun // Data Based Advisor. 1994. Vol. 12. No. 1. P. 103–107.

14. *Quatrani T.* Visual modeling with Rational Rose 2000 and UML. Addison-Wesley Professional, Second Edition. Addison Wesley, 2000. 288 p.

15. *Kopyltsov A.V. et al.* Algorithm of estimation and correction of wireless telecommunications quality // 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA). IEEE, 2018. P. 1–4.

16. *Vathsavayi S. et al.* Tool support for software architecture design with genetic algorithms // 2010 Fifth International Conference on Software Engineering Advances. IEEE, 2010. P. 359–366.

17. *Mejer B.* Ob"ektno-orientirovannoe programmirovanie i programmnaya inzheneriya: uchebnoe posobie. 2-e izd., ispr. M.: Nacional'nyj Otkrytyj Universitet «INTUIT», 2016. 286 s. URL: https://biblioclub.ru/index.php?page=book&id=429034

18. *Dzhamshidi P. i dr.* Mikroservisy: projdennyj put' i dal'nejshie celi // Otkrytye sistemy. SUBD. 2018. № 3. S. 19–23.

19. *Riel A.J.* Object-Oriented Design Heuristics. Addison-Wesley Professional; Illustrated edition, 1996. 400 p.

20. *Orlyanskaya I.V.* Sovremennye podhody k postroeniyu metodov global'noj optimizacii // Issledovano v Rossii. 2002. T. 5. S. 2097–2108.

21. *Glushan' V.M.* Metod imitacii otzhiga // Izvestiya YUzhnogo federal'nogo universiteta. Tekhnicheskie nauki. 2003. T. 31. № 2. S. 148–150.

22. *Matrenin P.V., Grif M.G., Sekaev V.G.* Metody stohasticheskoj optimizacii: ucheb. posobie. Novosibirsk: Izd-vo NGTU, 2016. 66 s.

СВЕДЕНИЯ ОБ АВТОРАХ

ГРИШИН Андрей Евгеньевич – магистрант, Казанский (Приволжский) федеральный университет, г. Казань.

Andrey Evgenyevich GRISHIN – graduate, Kazan (Volga region) Federal University, Kazan.

Email: andrey.grishin.work@gmail.com
ORCID: 0000-0001-6470-1832

ГРИГОРЯН Карен Альбертович – кандидат экономических наук, доцент, Казанский (Приволжский) федеральный университет, г. Казань.

Karen Albertovich GRIGORIAN – Candidate of Economics, Associate Professor, Kazan (Volga region) Federal University, Kazan.

Email: karigri@yandex.ru
ORCID: 0000-0001-6470-1832

Материал поступил в редакцию 20 мая 2022 года

УДК 004

РАЗРАБОТКА МЕТОДИКИ СЕГМЕНТАЦИИ ПОЛЬЗОВАТЕЛЕЙ С ПОМОЩЬЮ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ И РАСШИРЕННОЙ АНАЛИТИКИ

Д. А. Клинов[1] [0000-0002-3623-9596], К. А. Григорян[2] [0000-0001-6470-1832]

[1, 2]Казанский (Приволжский) федеральный университет, ул. Кремлевская, 35, г. Казань, 420008

[1]daniil.klinov@bk.ru, [2]karigri@yandex.ru

Аннотация

Статья посвящена созданию эффективного решения по сегментации пользователей. Представлены анализ существующих сервисов сегментации пользователей и подходов к их сегментации (ABCDx сегментация, демографическая сегментация, сегментация на основании карты пути пользователя), а также анализ алгоритмов кластеризации (K-means, Mini-Batch K-means, DBSCAN, Agglomerative Clustering, Spectral Clustering). Исследование названных подходов нацелено на создание решения по сегментации, «гибкого» и адаптирующегося под каждую пользовательскую выборку. Также применены дисперсионный анализ (тест ANOVA) и разбор метрик кластеризации для оценки качества сегментации пользователей. С помощью указанных методов разработано эффективное решение по сегментации пользователей с использованием технологии расширенной аналитики и машинного обучения.

Ключевые слова: *Сегментация, кластеризация, дисперсионный анализ, машинное обучение, расширенная аналитика, тест ANOVA, продуктовая аналитика.*

ВВЕДЕНИЕ

В современном конкурентном мире крайне важно понимать поведение клиентов и классифицировать клиентов на основе их демографии и покупательского поведения. Это критический аспект сегментации клиентов, который позволяет

маркетологам лучше адаптировать свои маркетинговые усилия к различным подгруппам аудитории с точки зрения стратегий продвижения, маркетинга и разработки продуктов.

Сегментация пользователей — это процесс сегментирования группы лиц в соответствии с определенными характеристиками, чтобы максимально точно определить их ожидания и потребности. Исследования показывают, что сегментации клиентов помогают привести к тому, что компании тратят менее 20% рабочего времени своих сотрудников на развитие продукта для удовлетворения потребностей клиентов, приносящих более 80% общей выручки продукта [1].

Ведущие IT-компании разрабатывают свои внутренние алгоритмы сегментации клиентов. У небольших IT-компаний, работающих в B2C (коммерческие взаимоотношения между организацией и частными лицами), нет выделенных средств для создания эффективной сегментации. Исходя из этого, малому и среднему бизнесу приходится использовать базовые алгоритмы сегментации, которые не учитывают индивидуальную пользовательскую аналитику определенного продукта. Помимо этого, используемые открытые алгоритмы сегментации пользователей обладают рядом недочетов, которые можно избежать с помощью исследований в области расширенной аналитики [2].

Стремление разработчиков продуктов расширить критерии сегментации, чтобы включить интересы и предпочтения большого круга пользователей, приводит к проблеме устаревания используемого алгоритма сегментации. Данная статья направлена на изучение способов адаптации разрабатываемого алгоритма к изменению поведения пользователей продукта в результате взаимодействия с ним.

ИССЛЕДОВАНИЕ СУЩЕСТВУЮЩИХ СЕРВИСОВ ДЛЯ СЕГМЕНТАЦИИ ПОЛЬЗОВАТЕЛЕЙ

Рассмотрим особенности существующих решений.

- **BlueVenn –** https://www.bluevenn.com. Сервис предоставляет возможности аналитики данных, прогнозирования событий и сегментации пользователей. BlueVenn предоставляет API-интерфейс и удобную загрузку данных: настраиваемый механизм идентификационных данных позволяет обрабатывать, сопоставлять, объединять, избавляться от дубликатов в данных клиентов, дает

возможность работы с транзакциями клиентов, маркетинговыми каналами и источниками данных в режиме реального времени.

- **Commence Cloud CRM –** https://www.commence.com. Сервис предоставляет возможности по автоматизации демографической сегментации клиентов. Позволяет клиентам получать доступ к ряду заранее созданных сегментов, а пользователям – создавать новые демографические сегменты, объединяя данные о клиентах. Встроенная аналитика сегментации дает возможность проводить дальнейшие исследования и оценивать эффективность клиентских сегментов.
- **Qualtrics –** https://www.qualtrics.com**.** Сервис предоставляет возможности настройки исследований, создания целевых групп, анализа результатов исследования. Реализует сегментацию клиентов на единой платформе, что обеспечивает оперативный доступ к необходимым данным и сведениям о различных событиях.
- **Experian –** https://www.experian.com. Сервис предоставляет возможности настройки событий для сегментации клиентов. У пользователя есть возможность составлять и контролировать портфель наиболее прибыльных клиентов. Решения, предлагаемые этим ПО, в первую очередь направлены на идентификацию «портрета» клиента.
- **HubSpot –** https://www.hubspot.com. Продукт помогает работать с контактами клиентов, которые есть в базе данных пользователя. С помощью этого продукта пользователь может продумывать стратегии маркетинга, продаж и работы с клиентами.

Нами были изучены и проанализированы 5 сервисов сегментации пользователей, которые суммарно имеют на своих сайтах более 4 миллионов уникальных посещений в месяц. Все изученные сервисы отличаются высокой ценой и не используют в своих решениях технологии машинного обучения. Все сервисы используют фильтрацию данных и не располагают расширенной аналитикой для прогнозирования сегмента новых пользователей [3].

После детального анализа текущих решений в сегментации пользователей не удалось найти решения, которое бы использовало «гибкий» подход к выбору алгоритма сегментации пользователей в зависимости от набора атрибутов для

сегментации или индивидуальных особенностей продукта, пользователи которого сегментируются.

ОПИСАНИЕ АЛГОРИТМА СЕГМЕНТАЦИИ ПОЛЬЗОВАТЕЛЕЙ

Целью конечного алгоритма является выявление подгрупп пользователей (сегментов), отличающихся между собой покупательским потенциалом, активностью посещений и обращений в техническую поддержку, демографическими признаками или иными количественными и качественными метриками.

Проведем теперь анализ алгоритмов кластеризации.

K-Means – алгоритм на основе центроидов, в котором каждая точка данных размещается ровно в одном из K непересекающихся кластеров, выбранных до запуска алгоритма [4].

Mini-Batch K-Means – алгоритм использует небольшие случайные группы, так называемые «пакеты», размер которых установлен изначально, чтобы их можно было хранить в памяти, а затем при каждой итерации случайная выборка из набора данных собирается и используется для обновления кластеров [4].

DBSCAN – этому алгоритму требуются два параметра:

- *Eps*: если расстояние между двумя точками меньше или равно *eps*, то они считаются соседями. Если значение *eps* выбрано слишком маленьким, большая часть данных будет рассматриваться как выбросы. Если этот параметр выбран очень большим, то кластеры объединятся, и большинство точек данных будет в одних и тех же кластерах.

- *MinPts*: чем больше набор данных, тем должно быть выбрано большее значение *MinPts*. Как правило, минимальные *MinPts* могут быть получены из числа измерений D в наборе данных как MinPts>=D+1. Минимальное значение MinPts должно быть выбрано не меньшим 3 [7].

HAC (Hierarchical Agglomerative Clustering) – алгоритм, результатом реализации которого является древовидное представление объектов, называемое «дендрограммой», которая показывает прогрессивную группировку данных. Этот алгоритм кластеризации не требует предварительного указания количества кластеров. Алгоритмы «снизу вверх» сначала обрабатывают данные как отдельный кластер, а затем последовательно объединяют пары кластеров, пока все кластеры не будут объединены в один кластер, содержащий все данные [7].

Метод спектральной кластеризации – этот алгоритм использует информацию из собственных значений (спектра) специальных матриц, построенных из графика или набора данных. Он рассматривает каждую точку данных как узел графа и, таким образом, преобразует задачу кластеризации в задачу разделения графа. Метод спектральной кластеризации не делает сильных предположений о статистике кластеров – в отличие от алгоритма K-средних, который предполагает, что точки, назначенные кластеру, имеют сферическую форму относительно центра кластера. В таких случаях спектральная кластеризация помогает создавать более точные кластеры [4–7].

Этапы сегментации пользователей сводятся к последовательному выполнению следующих шагов:

1. Загрузка пользовательских данных;
2. Выбор подхода к сегментации пользователей: ABCDx, Demographics, Journey map;
3. Реализация метода главных компонент для определения атрибутов кластеризации;
4. Применение алгоритмов кластеризации к пользовательским данным: K-means, Mini-Batch K-means, DBSCAN, Agglomerative Clustering, Spectral Clustering;
5. Анализ эффективности метрик алгоритмов кластеризации;
6. Применение теста ANOVA на тех же пользовательских данных для оценки количественных показателей эффективности алгоритма кластеризации;
7. Определение наиболее эффективного алгоритма кластеризации на данных, загруженных пользователем;
8. Финальная сегментация пользователей по определенному алгоритму кластеризации.

АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМА КЛАСТЕРИЗАЦИИ

Метрики качества кластеризации

Прежде чем анализировать качество кластеризации, нами определен термин *«эталонный кластер». Эталонные* кластеры существуют на исследуемом

множестве независимо от алгоритма кластеризации. Кластеризация на эталонные кластеры – это лучший результат работы алгоритма кластеризации среди всех возможных результатов, принимая во внимание, что может не существовать алгоритма, обеспечивающего кластеризацию на эталонные кластеры.

Анализ метрик кластеризации

1. Однородность кластеров. Качество кластеризации ухудшается, если происходит объединение двух эталонных кластеров в один (Рис. 1).

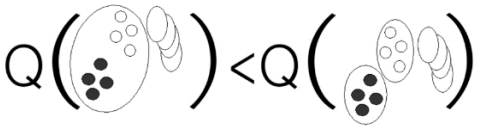

Рис. 1. Качество кластеризации в зависимости от однородности кластеров.

2. Полнота кластеров. Качество кластеризации ухудшается, если происходит разделение эталонного кластера на два других кластера (Рис. 2).

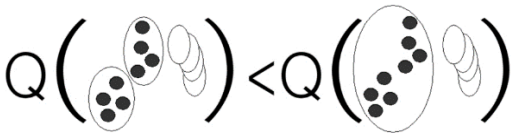

Рис. 2. Качество кластеризации в зависимости от полноты кластеров.

3. Чистота кластеров. Пусть на множестве есть эталонный кластер и несколько нерелевантных элементов, каждый из которых представляет эталонный кластер. Качество кластеризации увеличивается, если эталонный кластер выделяется в отдельный кластер без добавления других элементов (Рис. 3).

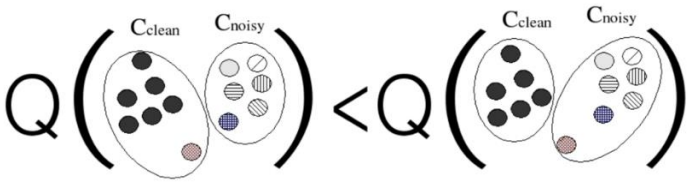

Рис. 3. Качество кластеризации в зависимости от чистоты кластеров.

4. Количество и размер кластеров. Качество кластеризации ухудшается, если отсутствует большое число небольших эталонных кластеров, но при этом присутствует один крупный эталонный кластер [8] (Рис. 4).

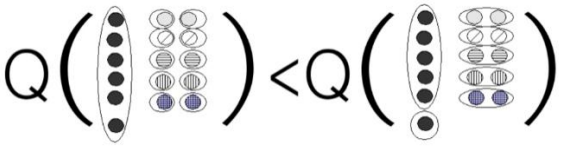

Рис. 4. Качество кластеризации в зависимости размера и количества кластеров.

Тест ANOVA

Правила, по которым применяется тест ANOVA для анализа качества кластеризации пользовательских данных, состоят в следующем:

- Исследуемой выборкой является база данных пользователей;
- Подгруппами являются сегменты пользователей, определенные по результатам кластеризации;
- Качественной характеристикой является сегмент пользователя;
- Количественными характеристиками являются генерируемая пользователем прибыль, сессии пользователя, обращения пользователя в техническую поддержку.

Метрикой качества кластеризации является статистическая значимость отличий между сформированными подгруппами (кластерами). Чем большую статистическую значимость имеют зависимости количественных характеристик от подгрупп, тем качественней считается алгоритм кластеризации.

На основании метрик кластеризации и теста ANOVA определяется эффективность алгоритма кластеризации [9].

ЗАКЛЮЧЕНИЕ

В результате проведенного исследования были проанализированы существующие сервисы сегментации пользователей. Было разработано решение с использованием различных алгоритмов кластеризации для эффективной сегментации пользовательских данных.

Существующие подходы к сегментации пользователей не являются «гибкими», они не адаптируются под определенные пользовательские данные. Решение, разобранное нами, предполагает анализ качества нескольких алгоритмов кластеризации с помощью метрик кластеризации и теста ANOVA и последующее использование эффективного алгоритма кластеризации.

Продолжить развитие данного исследования можно в сторону развития базы алгоритмов кластеризации, подходов к сегментации, улучшения оценки качества кластеризации и эффективности сегментации.

СПИСОК ЛИТЕРАТУРЫ

1. *Чурин В.В.* Роль маркетинговых исследований в проектной деятельности: Учебно-методическое пособие // Московский автомобильно-дорожный государственный технический университет (МАДИ). 2019. С. 1–111.

2. *An J., Kwak H., Jung S., Salminen J., Jansen B.* Customer segmentation using online platforms: isolating behavioral and demographic segments for persona creation via aggregated user data // Social Network Analysis and Mining. 2018. P. 1–19.

3. *Старкова Н В.* Кластеризация стран Европы по демографическим признакам // Молодой ученый. 2016. № 9 (113). С. 418–426.
URL: https://moluch.ru/archive/113/28811/ (дата обращения: 06.06.2022)

4. *Черезов Д.С.* Обзор основных методов классификации и кластеризации данных // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2009. №2. С. 23–27.
URL: https://rucont.ru/efd/519732 (дата обращения: 06.06.2022)

5. *Jagabathula S., Rusmevichientong P., Venkataraman A., Zhao X.* Estimating Large-Scale Tree Logit Models // NYU Stern School of Business, 2022.

6. *Amigó E., Gonzalo J., Artiles J.* A comparison of extrinsic clustering evaluation metrics based on formal constraints // Information Retrieval volume. 2009. No. 12. P. 461–486.

7. *Топалович Н*. Алгоритмы кластеризации в машинном обучении // Молодой ученый. 2020. № 52 (342). С. 47–49.
URL: https://moluch.ru/archive/342/77003/ (дата обращения: 06.06.2022)

8. *Rodriguez M.Z., Comin C.H., Casanova D., Bruno O.M., Amancio D.R., Costa L.F., Rodrigues F.A.* Clustering algorithms: A comparative approach // PLoS One. 2019. No. 14. P. 15–30.

9. *Байков И.И.* Метод ансамблирования алгоритмов кластеризации для решения задачи совместной кластеризации // Сенсорные системы. 2021. Т. 35. № 1. С. 43–49.

DEVELOPMENT OF A METHOD FOR USER SEGMENTATION USING CLUSTERING ALGORITHMS AND ADVANCED ANALYTICS

D. A. Klinov[1] [0000-0002-3623-9596], K. A. Grigorian[2] [0000-0001-6470-1832]

[1, 2]Kazan (Volga Region) Federal University, 35 Kremlevskaya str., Kazan, 420008

[1]daniil.klinov@delion.ru, [2]karigri@yandex.ru

Abstract

The article is devoted to the creation of an effective solution for user segmentation. The article presents an analysis of existing user segmentation services, an analysis of approaches to user segmentation (ABCDx segmentation, demographic segmentation, segmentation based on a user journey map), an analysis of clustering algorithms (K-means, Mini-Batch K-means, DBSCAN, Agglomerative Clustering, Spectral Clustering). The study of these areas is aimed at creating a “flexible” segmentation solution that adapts to each user sample. Dispersion analysis (ANOVA test), analysis of clustering metrics is also used to assess the quality of user segmentation. With the help of these areas, an effective solution for user segmentation has been developed using advanced analytics and machine learning technology.

Keywords: *Segmentation, clustering, analysis of variance, machine learning, advanced analytics, ANOVA test, product analytics.*

REFERENCES

1. *Churin V.V.* Rol' marketingovyh issledovanij v proektnoj dejatel'nosti: Uchebno-metodicheskoe posobie // Moskovskij avtomobil'no-dorozhnyj gosudarstvennyj tehnicheskij universitet (MADI). 2019. S. 1–111.

2. *An J., Kwak H., Jung S., Salminen J., Jansen B.* Customer segmentation using online platforms: isolating behavioral and demographic segments for persona creation via aggregated user data // Social Network Analysis and Mining. 2018. P. 1–19.

3. *Starkova N V.* Klasterizacija stran Evropy po demograficheskim priznakam // Molodoj uchenyj. 2016. № 9 (113). S. 418–426.
URL: https://moluch.ru/archive/113/28811/ (date of the application: 06.06.2022)

4. *Cherezov D.S.* Obzor osnovnyh metodov klassifikacii i klasteri-zacii dannyh // Vestnik Voronezhskogo gosudarstvennogo universiteta. Serija: Sistemnyj analiz i informacionnye tehnologii. 2009. №2. S. 23–27. URL: https://rucont.ru/efd/519732 (date of the application: 06.06.2022)

5. *Jagabathula S., Rusmevichientong P., Venkataraman A., Zhao X.* Estimating Large-Scale Tree Logit Models // NYU Stern School of Business, 2022.

6. *Amigó E., Gonzalo J., Artiles J.* A comparison of extrinsic clustering evaluation metrics based on formal constraints // Information Retrieval volume. 2009. No. 12. P. 461–486.

7. *Topalovich N.* Algoritmy klasterizacii v mashinnom obuchenii // Molodoj uchenyj. 2020. № 52 (342). S. 47–49.
URL: https://moluch.ru/archive/342/77003/ (date of the application: 06.06.2022)

8. *Rodriguez M.Z., Comin C.H., Casanova D., Bruno O.M., Amancio D.R., Costa L.F., Rodrigues F.A.* Clustering algorithms: A comparative approach // PLoS One. 2019. No. 14. P. 15–30.

9. *Bajkov I.I.* Metod ansamblirovanija algoritmov klasterizacii dlja reshenija zadachi sovmestnoj klasterizacii // Sensornye sistemy. 2021. T. 35. № 1. S. 43–49.

СВЕДЕНИЯ ОБ АВТОРАХ

КЛИНОВ Даниил Андреевич[1] – магистр Института информационных технологий и интеллектуальных систем по направлению «Программная инженерия», изучает продуктовую аналитику.

KLINOV Daniil Andreevic[1]– the magister of Institute of Information Technologies and Intelligent Systems in the direction “Software engineering”, middle of product analytics.

Email: daniil.klinov@bk.ru
ORCID: 0000-0002-3623-9596

ГРИГОРЯН Карен Альбертович[2] – кандидат экономических наук, доцент, Казанский (Приволжский) федеральный университет, г. Казань.

GRIGORIAN Karen Albertovich[2] – candidate of Economics, Associate Professor, Kazan (Volga region) Federal University, Kazan.

Email: karigri@yandex.ru
ORCID: 0000-0001-6470-1832

Материал поступил в редакцию 31 мая 2022 года

УДК 004.550

ПОСТРОЕНИЕ ЦИФРОВОЙ СИСТЕМЫ УПРАВЛЕНИЯ ГЕОЛОГИЧЕСКИМИ ЗНАНИЯМИ ДЛЯ ПОДДЕРЖКИ НАУЧНЫХ ИССЛЕДОВАНИЙ

М. И. Патук[1, [0000-0003-3036-2275]], В. В. Наумова[2, [0000-0002-3001-1638]]

[1, 2] *ФГБУН Государственный геологический музей им. В.И. Вернадского РАН Москва, Россия*

[1]patuk@mail.ru, [2]naumova_new@mail.ru

Аннотация

Описаны новые подходы к сбору данных о научных публикациях из систем открытого доступа с тематикой «Науки о земле». На основе разработанных и адаптированных подходов созданы архив научных публикаций (репозиторий) и комплекс программ доступа к научным публикациям для сбора, поиска, фильтрации, каталогизации и управления публикациями и их метаданными. Для улучшения доступности публикаций и других связанных с ними данных, находящихся на сайтах Государственного геологического музея им. В.И. Вернадского РАН, разработана система Wiki – Геология России. Эта система является тематическим рубрикатором по направлению «Месторождения полезных ископаемых России», с дополнительной тематикой «Минералогия». Все статьи имеют ссылку на источник информации из архива научных публикаций и, опционально, дополнительные ссылки по сходной тематике. Wiki – Геология России являются первым шагом в создании базы знаний по месторождениям полезных ископаемых.

***Ключевые слова**:* Wiki – Геология России, системы управления знаниями, репозиторий

В Государственном геологическом музее (ГГМ) им. В.И. Вернадского РАН накоплено большое количество открытых данных по наукам о земле. Эти данные, в основном, находятся на двух сайтах – Геология России (http://geologyscience.ru/) [1] и Портал открытых данных ГГМ РАН (http://data.sgm.ru/)[2]. Большой объем данных имеет свою негативную сторону – с увеличением объема растут и затраты на поиск нужной информации. Один из путей для повышения доступности информации – использовать тематическую группировку данных [3]. Такие тематически

сгруппированные данные, представленные в сжатой форме, могут служить объединяющим началом, синтезом разнородной информации, позволяющим формировать новое знание. С другой стороны, предлагаемый подход можно рассматривать как систему «легкого доступа» в электронную библиотеку и связанные с ней, возможно в неявной форме, данные в виде карт, фото и другой текстовой информации. Такой подход частично перекликается с развивающимися в последнее время системами управления знаниями, использующими в качестве базы знаний накопленный опыт организации и, в частности, научно-технические библиотеки [4].

В качестве платформы такой группировки нами была выбрана технология wiki – технология построения веб-систем, предназначенных для коллективной разработки, хранения, структуризации текста, гипертекста, файлов, мультимедиа. В качестве программной платформы была выбрана MediaWiki как самая популярная, свободно распространяемая платформа. В настоящее время версия этой платформы – 1.36.1 [5]. Созданный сайт (wiki – Геология России) расположен по адресу http://wiki.geologyscience.ru.

В интернете имеется огромное число сайтов, созданных с помощью названной или подобных ей платформ. Самой известной, безусловно, является Википедия. Принятый в ней принцип энциклопедичности принес ей заслуженную популярность. Однако не все статьи в ней одинаково равнозначны; имеет место неавторитетность и ненадежность информации, зачастую отсутствуют ссылки на первоисточник.

Более близкий нам по своему подходу, отношению к надежности информации и указанию её источников является сайт Геовикипедия (https://wiki.web.ru), созданный на Геологическом факультете Московского государственного университета им. М.В. Ломоносова. Тематический каталог (научный рубрикатор) сайта показывает широкий охват терминологии предметной области. Авторы проекта акцентируют внимание прежде всего на всестороннем представлении научной терминологии и корректном определении терминов. Можно сказать, что статьи в Геовикипедии являются конечным пунктом в определении геологических терминов. Мы же в своем подходе хотим сделать такие статьи отправной точкой в построении базы знаний по выбранным тематикам.

В качестве исходных тематик для группировки данных были выбраны «Месторождения полезных ископаемых» и «Минералогия». Естественно, что эти тематики не покрывают всего объема данных, имеющихся в распоряжении на наших сайтах. Их можно рассматривать в качестве теста для отработки технологии группировки информации. Основной тематикой в нашем подходе является «Месторождения полезных ископаемых», а «Минералогия» служит уточняющим и изобразительным дополнением к основной тематике, по крайней мере в текущей реализации системы.

Информация в MediaWiki хранится в виде статей с заголовками. Заголовками в нашем случае служат наименования месторождений и минералов. Как отмечалось [6], главным недостатком Wiki-технологии является отсутствие явно выраженной и удобно представленной структуры. При увеличении объема информации в системе это будет негативно влиять на ее доступность. Одним из возможных подходов к решению этой проблемы является добавление семантической информации в статьи, публикуемые в Wiki. Такой информацией является система категорий MediaWiki, которая позволила организовать алфавитный доступ к данным (вкладка «Месторождения по алфавиту») (рис. 1).

Кроме алфавитного доступа по наименованию месторождений, аналогичный подход был использован для организации доступа по полезным ископаемым (вкладка «Месторождения по полезным ископаемым»). Для обеспечения этого функционала также были использованы тэги категорий (например – «Полезные ископаемые–Железо»). Использование возможностей категорий в MediaWiki позволило осуществить доступ к информации по алфавиту (по наименованиям страниц) и извлекаемым полезным ископаемым (по наименованию полезных ископаемых).

Вся информация, вносимая в «Wiki – Геология России», берется из научных статей либо из монографий с обязательным указанием авторов и ссылки на статью или монографию. Сами первоисточники располагаются в архиве научных публикаций ГГМ им. В.И. Вернадского (https://repository.geologyscience.ru/) [7]. По ссылкам на первоисточник можно перейти на сайт архива научных публикаций и ознакомиться с полным текстом источника информации.

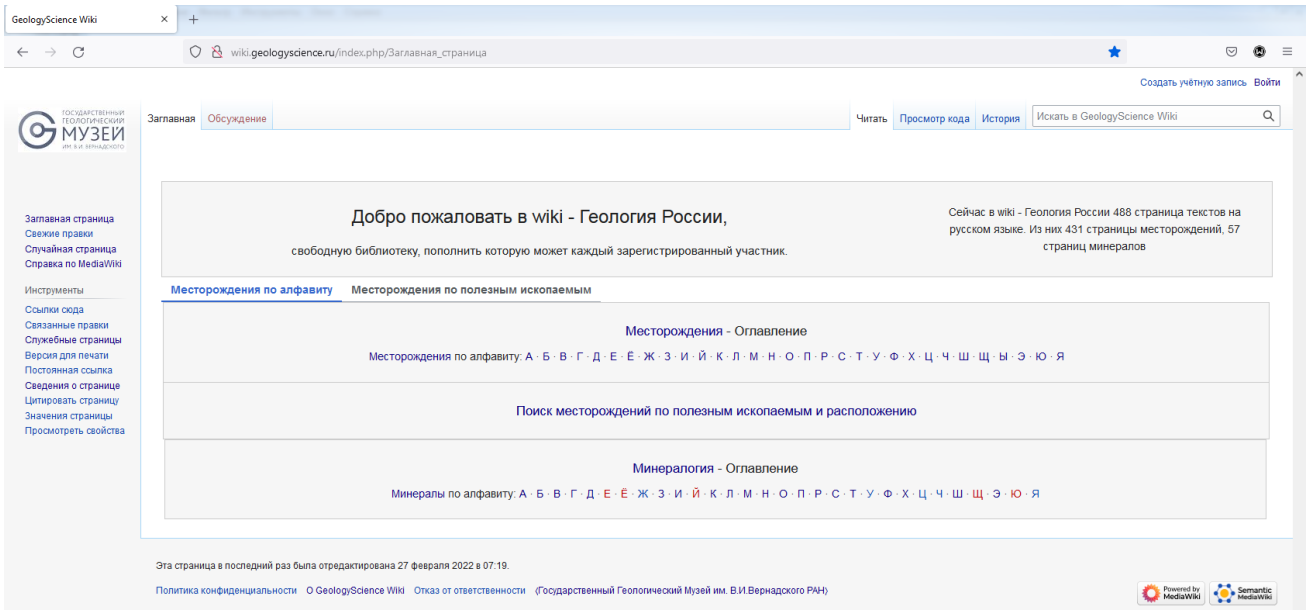

Рис. 1. Главная страница «Wiki-Геология России»

Кроме ссылки на первоисточник на странице месторождения находятся ссылки на описание месторождения в Государственном кадастре месторождений (https://www.rfgf.ru) и паспорт месторождения (данные находятся на сайте http://geologyscience.ru/) (рис. 2).

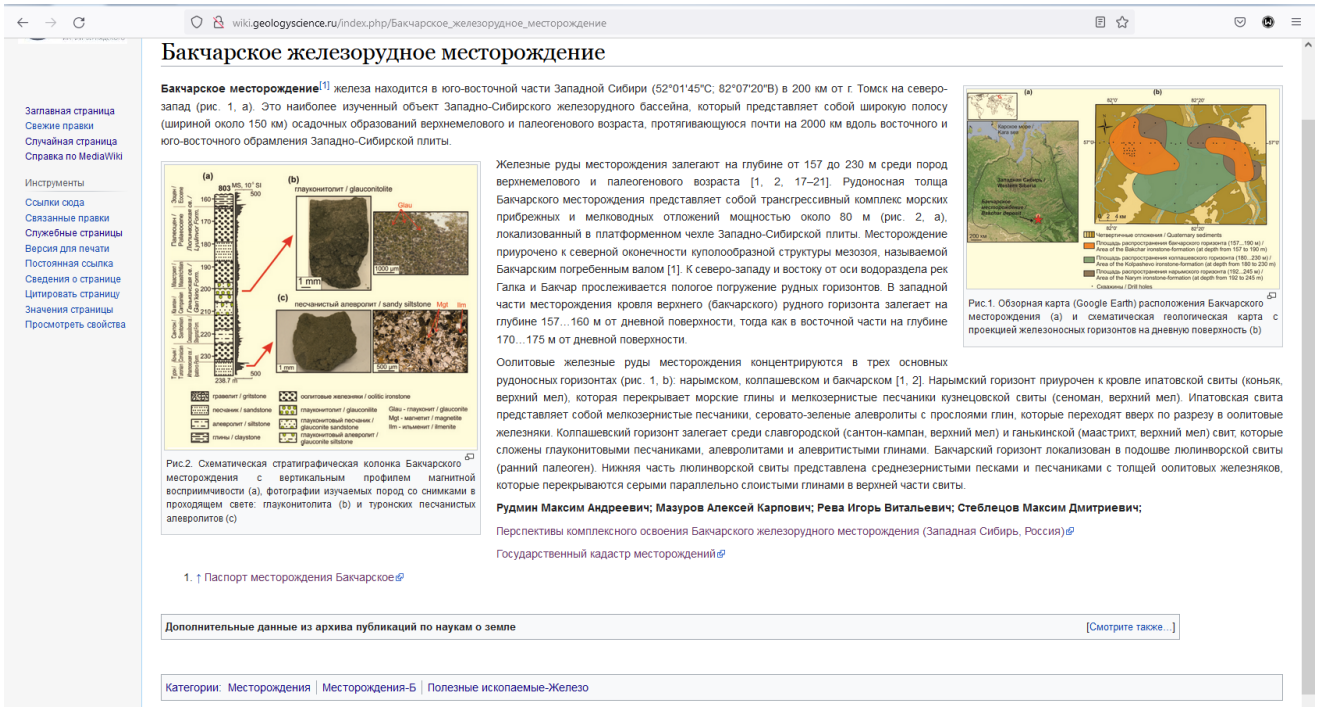

Рис. 2. Страница с описанием месторождения полезного ископаемого

Как правило, данные по конкретному месторождению не исчерпываются одной публикацией. Получить доступ к дополнительной информации можно по ссылке «Дополнительные данные из архива публикаций по наукам о земле» (рис. 3).

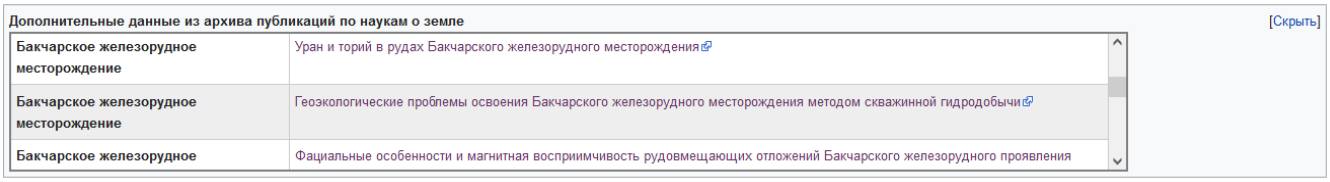

Рис. 3. Окно «Дополнительные данные из архива публикаций по наукам о земле»

Для извлечения дополнительных ссылок на публикации используется расширение MediaWiki – ExternalData – для доступа к базе данных репозитория, работающего на СУБД PostgreSQL. Хранение ссылок на публикации в базе MediaWiki (MySQL) достигается с использованием расширения Cargo.

Увеличение числа страниц месторождений полезных ископаемых выявило необходимость создания отдельного модуля поиска месторождений, с использованием информации об их территориальном расположении. Для реализации такой возможности было установлено дополнительное расширение MediaWiki – Semantic MediaWiki [8]. Это расширение позволяет наряду с категориями добавлять на страницы различные свойства и выполнять семантические запросы на основе внесенной информации. С помощью этого инструментария был реализован блок «Поиск месторождений по полезным ископаемым и их расположению».

Выбор месторождений происходит на основе одного или двух значений полезных ископаемых и территориального размещения (края, области, республики). Можно заполнить как одно поле (т. е. осуществлять поиск только по полезному ископаемому или территории), так и все три поля (рис. 4).

В разделе «Минералогия» представлены краткие описания конкретных минералов с фотографией указанного минерала из Портала открытых данных ГГМ РАН (http://data.sgm.ru/) или Портала Минералогического музея им. А.Е. Ферсмана РАН (https://fmm.ru). По ссылке под фотографией можно перейти на сайт источника и ознакомиться с более полной информацией по конкретному образцу

минерала. На странице минерала, как и на странице месторождения, можно получить доступ к дополнительной информации о выбранном минерале по ссылке «Дополнительные данные из архива публикаций по наукам о земле». Дополнительно указаны ссылки на данный минерал на 3-х популярных ресурсах – Wikipedia, энциклопедии GeoWiki, базе catalogmineralov.ru (рис. 5).

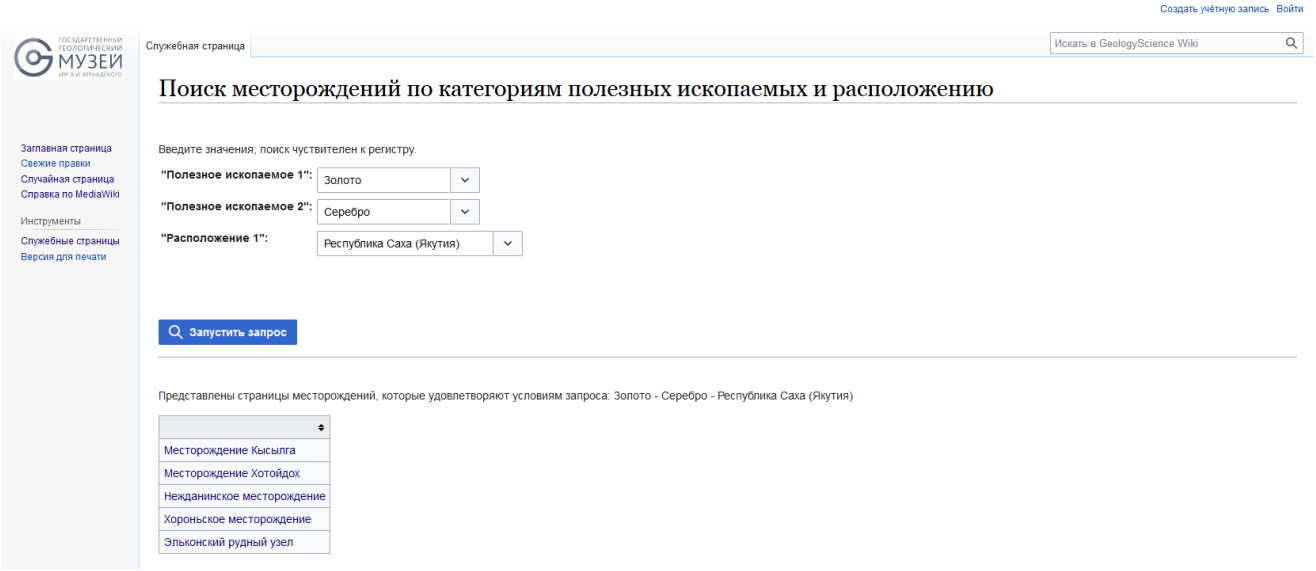

Рисунок 4. Страница поиска месторождений полезных ископаемых.

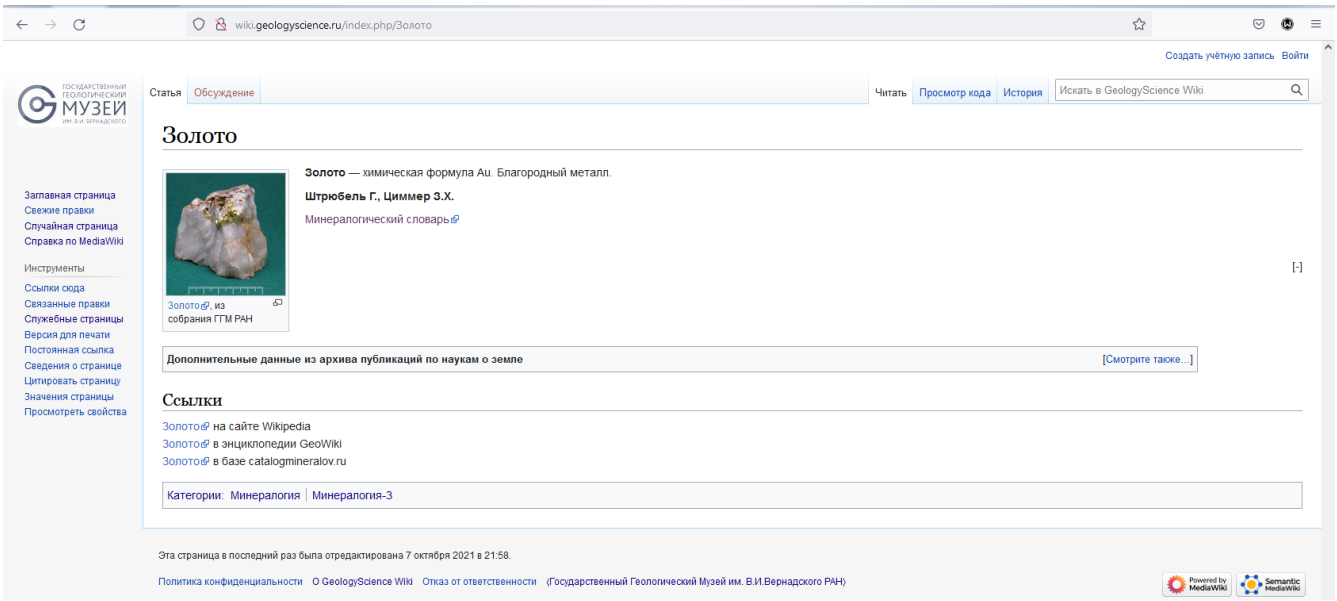

Рисунок 5. Страница с описанием минерала

На страницах описаний месторождений используются внутренние ссылки на другие страницы месторождений и страницы минералов. Они отображаются синим цветом.

Таким образом, создана первая версия сайта «Wiki – Геология России», содержащая 483 страницы описаний месторождений и 57 страниц описаний минералов. Каждая страница содержит ссылку на источник информации. Кроме обязательной ссылки страница может содержать перечень дополнительной литературы, относящейся к выбранному объекту. Ссылки ведут на репозиторий научных публикаций ГГМ им. В.И. Вернадского РАН. (https://repository.geologyscience.ru/). Кроме этого, присутствуют ссылки на паспорт месторождения и государственный кадастр месторождения, расположенные на сайте (http://geologyscience.ru/). Дополнительно указаны категории полезных ископаемых и территориальное размещение месторождений. Достигнутое объединение разнородной информации, связанной одной тематикой (месторождения полезных ископаемых), делает этот сайт первым шагом к созданию базы знаний в выбранном направлении.

Работы выполняются в рамках Государственного задания ГГМ РАН по теме № 0140-2019-0005 «Разработка информационной среды интеграции данных естественнонаучных музеев и сервисов их обработки для наук о Земле», а также темы государственного задания № 1021061009468-8-1.5.1 «Цифровая платформа интеграции и анализа геологических и музейных данных».

СПИСОК ЛИТЕРАТУРЫ

1. *Naumova V.V., Eremenko V.S., Platonov K.A., Dyakov S.E., Patuk M.I., Eremenko A.S.* Development of geographically distributed information-analytical geological environment // Russian Journal of Earth Sciences. 2019. Vol. 19, ES6012. https://doi.org/10.2205/2019ES000696, 2019

2. *Naumova V.V., Platonov K.V., Dyakov S.E., Eremenko V.S., Starodubtseva I.A.* Basic principles of development of open access to data of the Vernadsky State Geological Museum of RAS // Information Technologies in Earth Sciences and Applications for Geology, Mining and Economy. ITES&MP-2019: Proceedings of the V International Conference, Moscow (Russia), 14–18 October 2019. Moscow: VNIIgeosystem, 2019. P. 31.

3. *Ахмадеева И.Р., Загорулько Ю.А., Саломатина Н.В., Серый А.С., Сидорова Е.А., Шестаков В.К.* Подход к формированию тематических коллекций текстов на основе интернет-ресурсов // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2013. Т. 11, Вып. 4. С. 5–15.

4. *Жмайло С.В., Ульянин О.В.* Научно-техническая библиотека как составная часть системы управления знаниями организации: взгляд информационного работника // Научные и технические библиотеки. 2020. № 2. С. 9–23. https://doi.org/10.33186/1027-3689-2020-2-9-23

5. MediaWiki. https://www.mediawiki.org/wiki/MediaWiki

6. *Шестаков В.К.* Разработка и сопровождение информационных систем, базирующихся на онтологии и Wiki-технологии // Труды 13-й Всерос. науч. конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL’2011, Воронеж, Россия, 2011. С. 299–306.

7. *Патук М.И., Наумова В.В., Еременко В.С.* Цифровой репозиторий "GeologyScience.ru": открытый доступ к научным публикациям по геологии России // Электронные библиотеки. 2020. Т. 23, №6. С. 1324–1338. https://doi.org/10.26907/1562-5419-2020-23-6-1324-1338

8. Semantic MediaWiki. https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki

BUILDING A DIGITAL GEOLOGICAL KNOWLEDGE MANAGEMENT SYSTEM TO SUPPORT SCIENTIFIC RESEARCH

M. I. Patuk[1], V. V. Naumova [2]

[1, 25] *State Geological Museum named after Vladimir Vernadsky, Moscow*

[1]patuk@mail.ru, [2]naumova_new@mail.ru

Abstract

The paper describes new approaches to collecting data on scientific publications from open access systems with the subject of Earth Science. Based on the developed and adapted approaches, an archive of scientific publications (repository) and a set of programs for accessing scientific publications for collecting, searching, filtering, cataloging and managing publications and their metadata have been created. In order to improve the availability of publications and other related data on the websites of the SGM RAS, the Wiki – Geology of Russia system has been developed. This system is a thematic rubric in the direction of "Mineral deposits of Russia", with an additional topic "Mineralogy". All articles must have a link to the source of information from the archive of scientific publications and, optionally, additional links on similar topics. Wiki – Geology of Russia is the first step in creating a knowledge base on mineral deposits.

***Keywords**:* Wiki – Geology of Russia, knowledge management systems, repository.

REFERENCES

1. *Naumova V.V., Eremenko V.S., Platonov K.A., Dyakov S.E., Patuk M.I., Eremenko A.S.* Development of geographically distributed information-analytical geological environment // Russian Journal of Earth Sciences. 2019. Vol. 19, ES6012. https://doi.org/10.2205/2019ES000696, 2019

2. *Naumova V.V., Platonov K.V., Dyakov S. E., Eremenko V.S., Starodubtseva I.A.* Basic principles of development of open access to data of the Vernadsky State Geological Museum of RAS // Information Technologies in Earth Sciences and Applications for Geology, Mining and Economy. ITES&MP-2019: Proceedings of the V International Conference, Moscow (Russia), 14–18 October 2019. Moscow: VNIIgeosystem, 2019. P. 31.

3. *Ahmadeeva I.R., Zagorul'ko Yu.A., Salomatina N.V., Seryj A.S., Sidorova E.A., Shestakov V.K.* Podhod k formirovaniyu tematicheskih kollekcij tekstov na osnove internet-resursov // Vestn. Novosib. gos. un-ta. Seriya: Informacionnye tekhnologii. 2013. T. 11, vyp. 4. S. 5–15.

4. *Zhmajlo S.V., Ul'yanin O.V.* Nauchno-tekhnicheskaya biblioteka kak sostavnaya chast' sistemy upravleniya znaniyami organizacii: vzglyad informacionnogo rabotnika, Nauchnye i tekhnicheskie biblioteki. 2020, № 2. S. 9–23. https://doi.org/10.33186/1027-3689-2020-2-9-23

5. MediaWiki. https://www.mediawiki.org/wiki/MediaWiki

6. *Shestakov V.K.* Razrabotka i soprovozhdenie informacionnyh sistem, baziruyushchihsya na ontologii i Wiki-tekhnologii, Trudy 13j Vserossijskoj nauchnoj konferencii “Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kollekcii” – RCDL’2011, Voronezh, Rossiya, 2011. S. 299–306.

7. Patuk M.I., Naumova V.V., Eryomenko V.S. (2020). Cifrovoj repozitorij "geologyscience.ru": otkrytyj dostup k nauchnym publikaciyam po geologii Rossii // Elektronnye biblioteki. 2020. T. 23, No. 6. S. 1324–1338. https://doi.org/10.26907/1562-5419-2020-23-6-1324-1338

8. Semantic MediaWiki. https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki

СВЕДЕНИЯ ОБ АВТОРАХ

ПАТУК Михаил Иванович – к. г.-м. н., и. о. н. с., научный отдел Государственного геологического музея им. В.И. Вернадского РАН, Москва, Россия.

Michail I. PATUK – PhD, scientific department, Vernadsky State Geological Museum RAS, Moscow (Russia).

Email: patuk@mail.ru

ORCID: 0000-0003-3036-2275

НАУМОВА Вера Викторовна – д. г.-м. н., г. н. с., зав. Научным отделом Государственного геологического музея им. В.И. Вернадского РАН, Москва, Россия.

Vera V. NAUMOVA – Prof., head of SGM scientific department, Vernadsky State Geological Museum RAS, Moscow (Russia).

Email: naumova_new@mail.ru,

ORCID: 0000-0002-3001-1638

Материал поступил в редакцию 6 апреля 2022 года

УДК 004

РАЗРАБОТКА МОДУЛЯ ПРОВЕРКИ ДАННЫХ ДЛЯ УДОВЛЕТВОРЕНИЯ МЕТРИКИ УСТАРЕВАНИЯ

А. И. Сибгатуллина[1][0000-0003-4014-9558], **А. Ш. Якупов**[2][0000-0002-2333-8819]

[1, 2]Казанский (Приволжский) федеральный университет, ул. Кремлевская, 35, г. Казань, 420008

[1]aigul.sibgatulli@gmail.com, [2]azat.yakupov@it.kfu.ru

Аннотация

Из года в год возрастает объем мирового рынка больших данных. Их анализ является неотъемлемой частью для принятия немедленных и надежных решений. Технологии больших данных ведут к значительному снижению стоимости за счет использования облачных сервисов, распределенных файловых систем, когда возникает потребность в хранении больших объемов информации. Их аналитика неразрывно связана с понятием качества данных, что особенно важно, если они имеют определенный срок хранения – метрику устаревания – и мигрируют из одного источника в другой, увеличивая риск потери данных. Предупреждение негативных последствий достигается за счет процесса сверки данных – комплексной проверки больших объемов информации с целью подтверждения их согласованности.

В статье рассмотрены вероятностные структуры данных, которые могут быть использованы для решения задачи, а также предложена реализация – модуль проверки целостности данных с использованием фильтра Блума с подсчетом. Данный модуль интегрирован в Apache Airflow для автоматизации процесса.

***Ключевые слова**: большие данные, метрика устаревания, партиция, parquet файл, фильтр Блума*

ВВЕДЕНИЕ

В эпоху больших данных каждая компания ежедневно обрабатывает огромные объемы информации. Согласно исследованию, опубликованному компанией MarketsandMarkets [1], мировой объем рынка Big Data оценивается в 162,6 миллиарда долларов по итогам 2021 года, однако прогнозируется, что к 2026 году он

вырастет до 273,4 миллиарда долларов при средней динамике в 11,9% в год. Одними из преимуществ аналитики больших данных являются скорость и эффективность. Если всего несколько лет назад предприятия собирали и извлекали информацию, проводили аналитику для принятия только будущих решений, то сегодня этот процесс ориентирован на немедленные и более обоснованные решения, что предоставляет организациям конкурентное преимущество: с одной стороны, возможность работать быстрее, с другой – сохранять гибкость.

Аналитика больших данных неразрывно связана с такими понятиями, как управление данными и качество данных. Перед проведением анализа необходимо удостовериться в их целостности, что особенно важно во время проведения миграции, так как существует высокий риск потери части данных вследствие таких факторов, как: низкая пропускная способность сети или ее отключение, нестабильное соединение, прерывание транзакций, сбои во время выполнения заданий, выход из строя узлов кластера. Все перечисленные ошибки могут привести к тому, что данные будут сохранены в недопустимом состоянии, например, содержать неверные или некорректно сформатированные значения, дублирующиеся строки, или, наоборот, часть информации будет потеряна.

Кроме того, многие компании [2] руководствуются политикой хранения данных (или записей), которая является ключевой составляющей жизненного цикла информации. Она представляет собой установленный в организации протокол и описывает, как долго необходимо хранить данные, в каком месте и каким образом их уничтожить по истечении срока использования. Эта политика важна вследствие ряда причин. Во-первых, она снижает затраты на хранение ненужных данных, начиная с вопросов места и памяти и заканчивая экономическими аспектами. Во-вторых, повышает релевантность существующих данных, так как ранняя информация становится менее актуальной по мере устаревания. Однако, если удаление данных внутри организации не представляется возможным, две вышеупомянутые причины становятся проблемами, которые необходимо решить.

Наиболее распространенным способом является использование «холодного хранилища» (например, Hadoop HDFS, Amazon S3) для длительного содержания объектов с редкими запросами на чтение. По истечении срока использования часть нерелевантных данных из СУБД перемещается в холодное хранилище.

При необходимости хранить данные на протяжении длительного периода возрастает важность обеспечения их качества. Если после истечения срока, установленного политикой хранения, обнаружится, что мигрированные данные повреждены, то они будут безвозвратно утеряны. Предупреждение негативных последствий достигается за счет процесса сверки данных – комплексной проверки больших объемов информации с целью подтверждения их согласованности. Во время него происходит сравнение исходных данных с целевыми для определения стабильности архитектуры миграции.

Так как это область больших данных, использование стандартных методов, например, построчного сравнения, неприменимо, поэтому становится актуальной проблема отсутствия автоматизированного процесса проверки качества больших данных.

ОБЗОР ЛИТЕРАТУРЫ

Вероятностные структуры данных

Неограниченный рост данных привел к смене парадигмы в методах хранения и поиска от традиционных структур данных к вероятностным. Детерминированные структуры всегда дают точные ответы и, как и вероятностные, могут выполнять то же множество операций, но только с малыми наборами данных. Если размер датасета велик и не помещается в память, то детерминированные структуры дают сбой, и их использование не представляется возможным. Вероятностные, в свою очередь, подходят для работы с большими данными и потоковыми приложениями, так как позволяют избежать высокой задержки аналитических процессов. Эти структуры используют хеш-функции для компактного представления набора элементов. Они не могут дать определенного ответа, только приближенный, но по сравнению с детерминистическими структурами они требуют гораздо меньше памяти и имеют постоянное время обработки сложных запросов.

Существует несколько типов вероятностных структур данных, изображенных на рисунке 1, которые решают следующие задачи:

- проверка на членство в множестве;
- подсчет частоты;

- оценка кардинальности – подсчет количества раз, когда элемент встречался в массивных наборах данных;
- поиск сходства – поиск ближайших соседей в датасете.

Рисунок 1. Вероятностные структуры данных

В статье [3] обсуждаются различные сферы применения подобных структур. Например, Bloom Filter изначально создавался с целью представления слов в словаре. Постепенно он стал широко использоваться в сетевых алгоритмах и алгоритмах безопасности, таких как аутентификация, отслеживание IP-адресов, поиск подстроки. Кроме того, сотовые сети обеспечивают связь между устройствами с применением фильтра Блума для идентификации мобильных приложений [4]. Другим примером является метод MinHash, позволяющий найти сходства между двумя элементами, вычисляя коэффициент Жаккара. В настоящий момент она используется в различных областях: при кластеризации изображений для поиска дубликатов [5], для обнаружения вредоносных программ [6].

В [3] также экспериментально подтверждено, что асимптотическая сложность вероятностных структур данных гораздо меньше, чем детерминистических при выполнении операций вставки, деления, обхода, поиска наряду с другими статическими запросами. Следовательно, учитывая экспоненциальный рост данных и доменных областей, такие структуры позволяют ускорить процессы по обработке информации.

Для решения задачи принадлежности элемента множеству две структуры из обозначенных на рисунке 1 могут быть использованы в качестве его основы: Bloom Filter и Quotient Filter.

Несмотря на то что оба подхода поддерживают один набор операций и одну и ту же сложность, Quotient Filter имеет некоторую вероятность ложноотрицательного срабатывания [7], то есть при наличии элемента в множестве возвращать результат об его отсутствии, что несвойственно для Bloom Filter. Помимо этого, он использует больше памяти, но по скорости сравним с фильтром Блума. Другим недостатком данной структуры является резкое снижение производительности более, чем в два раза, по мере ее заполнения. Временные расходы происходят из-за комплексной процедуры хеширования [8]: нахождение подходящей позиции для элемента при вставке со сдвигом является трудоемкой задачей ввиду возможности коллизий. Также в [8] проиллюстрирован сравнительный анализ частоты коллизий в зависимости от числа хеш-функций. Изначально Bloom Filter с меньшим набором функций имеет большее число коллизий по сравнению с Quotient Filter, однако оно значительно уменьшается с увеличением количества функций.

Имплементации фильтра Блума

На текущий момент существует более 60 различных вариаций фильтра Блума [9], десятки из которых посвящены сокращению ложноположительных срабатываний и упрощению реализации с целью улучшения производительности алгоритма. Их сравнение производилось на основе нескольких параметров:

- подсчет – подсчет количества использованных битов, который может быть осуществлен как напрямую, так и с помощью набора хеш-функций или других альтернативных решений;
- группировка – нахождение подмножества, к которому принадлежит определенный элемент;
- удаление и масштабируемость;
- распад – исключение устаревших элементов для фокусировки на более новых;
- параллелизм – параллельные вычисления, обеспечивающие ускорение запроса и увеличение его пропускной способности;
- ложноотрицательное срабатывание.

Стоит отметить, что ни один из рассмотренных вариантов не удовлетворяет всем вышеописанным характеристикам, при этом треть из них имеет сложность

вставки и запроса выше, чем стандартная реализация. Например, для Bloomier Filter [10] вследствие того, что он рекурсивно кодирует все элементы, применяя несколько фильтров Блума, сложность вставки составляет $O(n\,logn)$, а запроса – $O(\lambda k)$, где λ – количество фильтров Блума. Кроме того, многие из подтипов применимы только к определенным доменным областям, а именно: сетевое взаимодействие (Adaptive BF [11], Energy efficient BF [12]), обработка биометрических (k-mer BF [13]) и пространственных (Spatial BF [14]) данных, системы хранения (BloomStore [15]), дублирование (Stable BF [16]) и обнаружение копий (Matrix BF [17]). А также 15 типов из представленных 60 вводят ложноотрицательное срабатывание, которое отсутствует в оригинальном фильтре Блума.

В статье [18] предложена Persistent Bloom Filter (PBF) – вероятностная структура, которая поддерживает тестирование временного членства в множестве. Под временным диапазоном понимается разница между двумя моментами времени для каждого элемента. В качестве примера для тестирования используются IP-адреса: например, пользователь мог войти сеть несколько раз в течение одного заданного интервала, поэтому дубликаты допустимы. Таким образом, цель данного тестирования – обнаружить строки или столбцы с определенными значениями, которые были добавлены или изменены в течение указанного периода. Решение данной проблемы – PBF – представляет собой цепочку фильтров Блума, где каждый из них ответственен за отобранное подмножество элементов. Он декомпозирует запрос и отправляет отдельные части разным фильтрам. К его особенностям относятся: отсутствие ложноотрицательных срабатываний, эффективное использование памяти, производительное обновление данных и высокая точность за счет регулирования общего количества битов и для каждого отдельного фильтра.

Одним из вариаций фильтра Блума является также фильтр Блума с подсчетом (Counting Bloom Filter). Он поддерживает операцию удаления благодаря использованию счетчика, при этом не вводит ложноотрицательное срабатывание [19]. Эта функциональность критически важна, когда предоставляемый набор данных динамичен, то есть его размер может изменяться со временем. Кроме того, описываемый фильтр использует меньший объем памяти по сравнению со

стандартным подходом, но потребляет больший при хранении отпечатка каждого элемента.

Для решения поставленных задач нами выбран фильтр Блума с подсчетом в качестве основы для реализации алгоритма вследствие следующих причин. Во-первых, он имеет низкий уровень расходов по времени, памяти и вычислениям [3]. Во-вторых, он является структурой общего назначения [9], то есть не относится к конкретной предметной области и может быть использован в совокупности с любыми форматами входных данных. В-третьих, позволяет хранить повторную информацию и исключать элементы из множества. В-четвертых, в фильтре Блума с подсчетом точное значение счетчика не учитывается помимо того, положительное оно или отрицательное [20], что предоставляет достаточный результат для определения принадлежности к множеству.

ИСПОЛЬЗУЕМЫЕ ФОРМАТЫ И СТРУКТУРЫ ДАННЫХ

Партиционная таблица

Партиционирование (или секционирование) – это разделение хранимых объектов баз данных (таблиц, индексов, материализованных представлений) на более мелкие логические части по заданным критериям. Партиционная таблица представляет собой специального вида таблицу, которая поделена на сегменты, называемые партициями, для того, чтобы обеспечить более удобное и быстрое управление данными. Физически партиции могут находиться в одном табличном пространстве, в разных или комбинируя оба подхода. За счет разбиения большой таблицы на более мелкие секции повышается производительность запросов, так как обращение идет только к части данных, и уменьшаются расходы на память, что является ключевым фактором при работе в области больших данных. Таким образом, основная цель партиционирования – это помощь в обслуживании объемных таблиц и сокращение общего времени отклика на чтение и загрузку данных для определенных SQL операций.

Apache Parquet

Apache Parquet представляет собой бинарный, колоночно-ориентированный формат хранения больших данных. Он предоставляет возможности задавать

схемы сжатия на уровне столбцов и добавлять новые кодировки. Изначально созданный для экосистемы Hadoop, он является одним из наиболее распространенных форматов Big Data наряду с Apache Avro, RCFile и ORC.

Parquet впервые появился в 2013 году, и с тех пор получил широкое распространение в качестве бесплатного формата хранения данных с открытым исходным кодом, ориентированного для быстрого выполнения аналитических запросов. Когда компания AWS анонсировала [21] функциональность экспорта озер данных, она охарактеризовала его как формат, позволяющий выгружать в два раза быстрее и использующий в шесть раз меньше места в хранилище Amazon S3 по сравнению с текстовыми форматами.

Структура Parquet файла проиллюстрирована на рисунке 2, в которой можно выделить три составляющие:

1. Группа строк (row group);
2. Фрагмент столбца (column chunk);
3. Страница (page).

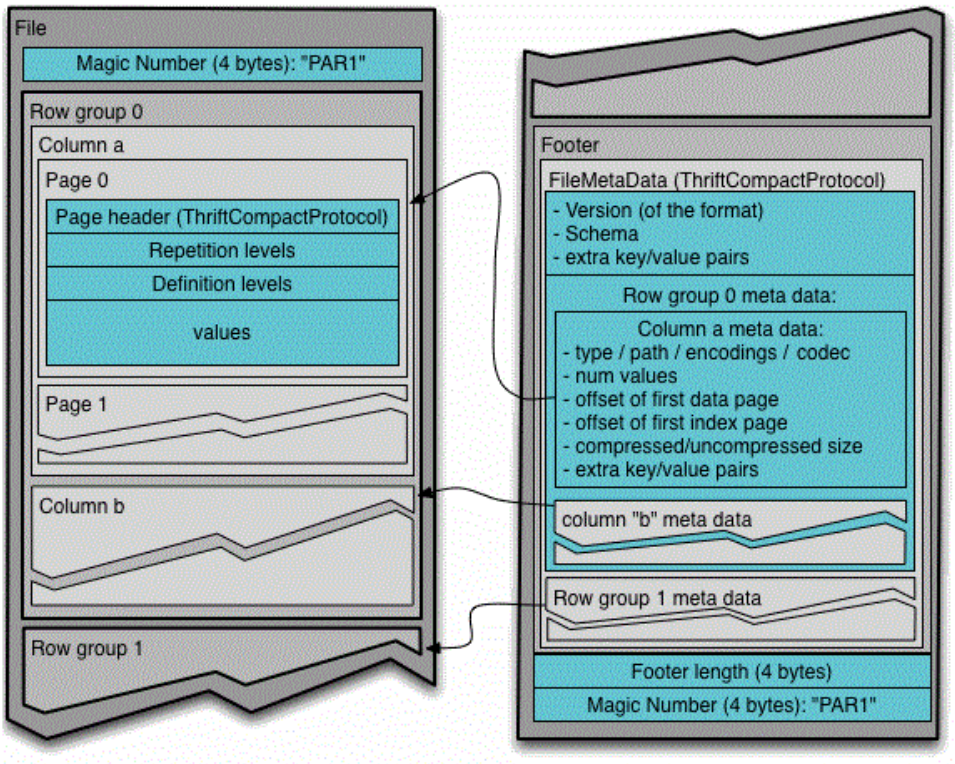

Рисунок 2. Структура файла формата Apache Parquet

Группа строк представляет собой набор строк в колоночно-ориентированном формате размером от 50 мегабайт до 1 гигабайта. Она состоит из фрагмента каждого столбца в наборе данных. Фрагмент столбца – это блок данных конкретного столбца в определенной группе строк. Страница – это концептуально неделимая единица, которая содержит фрагменты столбцов. Страницы записаны друг за другом, содержат метаинформацию и закодированные данные, поэтому при необходимости можно считать только определенные страницы.

В отличие от линейно-ориентированных форматов, parquet оптимизирован для повышения производительности. При выполнении запросов фокусировка происходит только на необходимых данных. Кроме того, объем сканируемой информации сводится к минимуму, что ведет к рациональному использованию операций ввода-вывода. В одном из практических экспериментов [22], проведенном американской компанией Databricks, было подтверждено, что использование формата parquet сократило расходы на память как минимум на одну треть для больших объемов данных, а также уменьшило временные затраты на сканирование и десериализацию более, чем в 34 раза. Таким образом, при работе с большими данными Apache Parquet является одним из наиболее подходящих форматов, позволяющим оптимизировать время запросов и увеличить производительность.

Фильтр Блума

Фильтр Блума – это эффективная вероятностная структура данных, созданная Бертоном Блумом в 1970 году, которая определяет, является ли элемент частью множества. Он предотвращает выполнение лишних вычислений, проверяя тот факт, что элемент совершенно точно не входит в множество. В отличие от линейного и бинарного поисков, сложности которых составляют $O(n)$ и $O(\log n)$ соответственно, сложность вставки и проверки принадлежности к множеству с помощью фильтра Блума – $O(1)$. Он никогда не выдаст ложноотрицательные результаты, но, вследствие того, что структура является вероятностной, в этом подходе существует возможность получения ложноположительных (false positive) результатов. Под ложноположительным (ложноотрицательным) срабатыванием будет понимать положительный (отрицательный) ответ структуры данных при отсутствии (наличии) в ней элемента. Другими словами, все элементы множества

распознаются корректно, однако есть вероятность получить положительный итог при отсутствии значения.

Для описания работы фильтра Блума необходимо ввести несколько переменных. На начальном этапе пустой фильтр представляет собой битовый массив из m битов, равных 0:

	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
m	0	1	2	3	4	5	6	7	8	9

Под n будем считать количество элементов множества S. Фильтр Блума основан на хешировании, поэтому количество хеш-функций обозначим k. Каждая функция сопоставляет элементы с корзинами (бакетами) соответственно битовому массиву. При добавлении элемента e в фильтр рассчитываются значения всех хеш-функций, затем индексы при помощи оператора деления по модулю, которые заменяются на 1 в битовом массиве:

$$\forall e: h_1(e), \dots, h_k(e) => S = \{S_1, S_2, \dots, S_m\} \cup \{e\}$$

Например, для элемента e с применением двух хеш-функций верно:

$$h_1(e)\ \%\ m = 2,$$
$$h_2(e)\ \%\ m = 6,$$

	0	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
m	0	1	2	3	4	5	6	7	8	9

Для осуществления проверки необходимо пройти через весь процесс в обратном порядке: рассчитать результаты хеш-функций для входного значения и посмотреть, все ли индексы битового массива равны 1. В случае, если все биты содержат 1, элемент либо точно существует в множестве, либо отсутствует ввиду ложноположительного срабатывания. Если хотя бы один из битов равен 0, то можно утверждать, что элемент отсутствует:

$$\forall e: h_1(e), \dots, h_k(e);\ \forall i = \overline{1,n};\ \exists BF_i = 1 => \{e\} \in S$$
$$\forall e: h_1(e), \dots, h_k(e);\ \forall i = \overline{1,n};\ \exists BF_i = 1 => false\ positive$$
$$\forall e: h_1(e), \dots, h_k(e);\ \forall i = \overline{1,n};\ \exists BF_i = 0 => \{e\} \notin S$$

Ложноположительный результат может возникнуть тогда, когда все биты входного значения установлены 1 при добавлении других элементов. Его вероятность можно контролировать путем изменения размеров фильтра Блума: чем больше битовый массив, тем ниже вероятность ложных срабатываний. Кроме

того, увеличение количества хеш-функций также ведет к уменьшению вероятности, однако добавляет временную задержку при дополнении и поиске.

Фильтр Блума поддерживает две операции – вставку и проверку – но не позволяет произвести удаление элемента. Если изменять биты в массиве, то это может привести к ложноотрицательным результатам. Наиболее популярным расширением классического фильтра Блума является фильтр Блума с подсчетом. Он вводит массив из m счетчиков, соответствующий каждому биту в массиве.

Фильтр Блума с подсчетом позволяет приблизительно определить, сколько раз каждый элемент был отмечен в фильтре, увеличивая счетчик при добавлении нового элемента. При этой операции сначала вычисляются соответствующие ему битовые позиции, а затем для каждой из них инкрементируется счетчик. Благодаря этому при удалении элемента возможны модификации массива, так как его значения всегда будут неотрицательными. Нами использовано данное расширение ввиду того, что оно дает возможность хранить повторную информацию и не теряет своей функциональности при удалении.

РЕЗУЛЬТАТЫ

После изучения различных вероятностных структур, которые могут быть использованы для определения принадлежности элемента множеству, был разработан и протестирован модуль проверки партиции и parquet файла на основе фильтра Блума с подсчетом, затем он был интегрирован в процесс в Apache Airflow.

Процесс запускается ежедневно по расписанию и проверяет, появилась ли новая партиция, у которой истек срок хранения. Он задается в днях в отдельной базе данных на уровне партиционной таблицы. Каждая партиция содержит информацию за отдельный месяц. Работа процесса осуществляется в несколько этапов:

1. Формируется список всех партиций, которые в текущий момент хранятся в базе данных;
2. Запрашивается метрика устаревания таблицы. Каждая партиция из списка, созданного на предыдущем шаге, проверяется на соответствие метрике. Если разница между максимальным значением даты и текущей датой превышает допустимый порог, то такая партиция считается устаревшей;

3. Для найденной партиции запрашиваются маппинг столбцов с их типами и сами данные;
4. Считывается файл формата parquet из HDFS кластера;
5. Инициализируется фильтр Блума. В качестве n подается на вход количество строк в партиции. Вероятность P задается вручную в конфигурационном файле. Ориентиром в этом случае служит партиция – она всегда содержит актуальную неискаженную информацию, в то время как паркетный файл может иметь различного рода аномалии: замещение слепков данных, неверное копирование – в связи с нестабильностью сетевого трафика;
6. При успешной проверке партиция отсоединяется и удаляется из базы данных, так как соответствие с parquet файлом подтверждено;
7. При неуспешной проверке либо отсутствии или повреждении файла по заданному пути отправляется информационное сообщение в телеграм-чат.

Полный граф процесса изображен на рисунке 3:

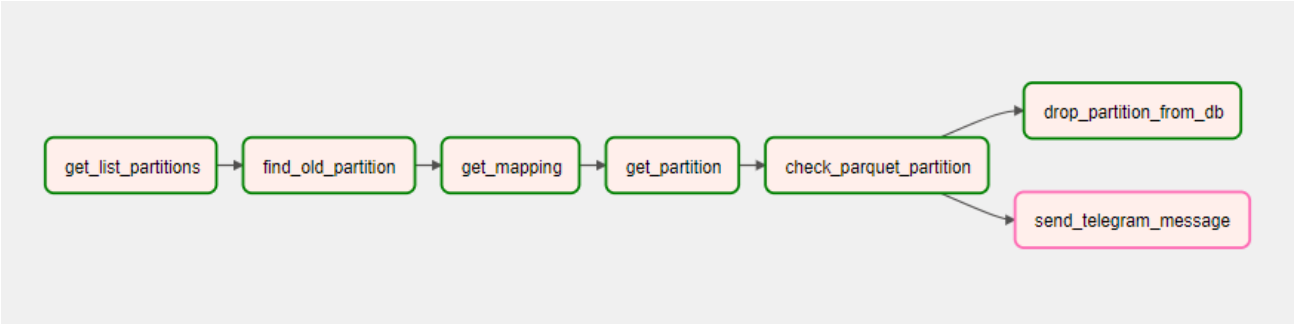

Рисунок 3. Граф процесса проверки parquet файла с данными партиции

ЗАКЛЮЧЕНИЕ

Таким образом, проведенное исследование показало, что на практике для решения задачи проверки целостности файлов с данными могут использоваться вероятностные структуры, определяющие принадлежность элемента множеству. Кроме того, были рассмотрены различные вариации фильтра Блума и реализован наиболее подходящий под критерии задачи.

Результатом исследования является разработка модуля проверки данных партиции и parquet файла на основе фильтра Блума с подсчетом. При разработке процесса использованы: инструмент для создания, планирования и мониторинга рабочих процессов Apache Airflow, распределенная файловая система Apache Hadoop и база данных PostgreSQL.

В дальнейшем планируются подключение брокеров сообщений для получения информации из HDFS кластера, а также применение процесса на основе реальных данных.

СПИСОК ЛИТЕРАТУРЫ

1. Big Data Market worth $273.4 billion by 2026. URL: https://www.marketsandmarkets.com/Market-Reports/big-data-market-1068.html.

2. Data Retention Policy: What Is It and How to Build One. URL: https://www.techtarget.com/searchdatabackup/definition/data-retention-policy.

3. *Batra S., Garg S., Kaur R., Kumar N., Singh A., Zomaya A.Y.* Probabilistic data structures for big data analytics: A comprehensive review // Knowledge-Based Systems. 2019. Vol. 188. No. 104987. P. 54–75.

4. *Choi K.W., Hossain E., Wiriaatmadja D.T.* Discovering mobile applications in cellular device-to-device communications: Hash function and bloom filter-based approach // IEEE Transactions on Mobile Computing. 2016. Vol. 15. No. 2. P. 336–349.

5. *Sasikala J., Thaiyalnayaki S.* Indexing near-duplicate images in web search using minhash algorithm // International Conference on Processing of Materials, Minerals and Energy. 2018. Vol. 5. No. 1. P. 1943–1949.

6. *Drew J., Hahsler M., Moore T.* Polymorphic Malware Detection Using Sequence Classification Methods // IEEE Security and Privacy Workshops (SPW). 2016. P. 81–87.

7. *Borgohain S.K., Nayak S., Patgiri R.* rDBF: A r-Dimensional Bloom Filter for massive scale membership query // Journal of Network and Computer Applications. 2019. Vol. 136. P. 100–113.

8. *Batra S., Garg S., Kumar N., Singh A.* Probabilistic data structure-based community detection and storage scheme in online social networks // Future Generation Computer Systems. 2019. Vol. 94. P. 173–184.

9. *Guo D., Luo L., Luo X., Ma R. T. B., Rottenstreich O.* Optimizing Bloom Filter: Challenges, Solutions, and Comparisons // IEEE Communications Surveys & Tutorials. 2019. Vol. 21. No. 2. P. 1912–1949.

10. *Boy O., Chazelle B., Kilian J., Rubinfeld R., Tal A.* The Bloomier filter: An efficient data structure for static support lookup tables // SODA. 2004. P. 30–39.

11. *Hazeyama H., Kadobayashi Y., Matsumoto Y.* Adaptive Bloom filter: A space-efficient counting algorithm for unpredictable network traffic // IEICE Transactions on Information and Systems. 2008. Vol. 91. No. 5. P. 1292–1299.

12. *Song T., Wang X., Zhou Y.* EABF: Energy efficient self-adaptive Bloom filter for network packet processing // IEEE International Conference on Communications (ICC). 2012. P. 2729–2734.

13. *Filippova D., Kingsford C., Pellow D.* Improving Bloom filter performance on sequence data using k-mer Bloom filters // J. Comput. Biol. 2017. Vol. 26. No. 6. P. 547–557.

14. *Calderoni L., Maio D., Palmieri P.* Location privacy without mutual trust: The spatial Bloom filter // Computer Communications. 2015. Vol. 68. P. 4–12.

15. *Du D.H.C., Lu G., Nam Y.J.* BloomStore: Bloom filter based memory-efficient key-value store for indexing of data de-duplication on flash // IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST). 2012. P. 1–11.

16. *Deng F., Rafiei D.* Approximately detecting duplicates for streaming data using stable Bloom filters // ACM SIGMOD international conference on Management of data. 2006. P. 25–36.

17. *Ahmadi M., Geravand S.* A novel adjustable matrix Bloom filterbased copy detection system for digital libraries // IEEE 11th International Conference on Computer and Information Technology. 2011. P. 518–525.

18. *Guo J., Li F., Peng Y., Qian W., Zhou A.* Persistent Bloom Filter: Membership Testing for the Entire History // International Conference on Management of Data. 2018. P. 1037–1052.

19. *Nayak S., Patgiri R.* A Review on Role of Bloom Filter on DNA Assembly // IEEE Access. 2019. Vol. 7. P. 66939–66954.

20. *Reviriego P., Rottenstreich O.* The Tandem Counting Bloom Filter – It Takes Two Counters to Tango // IEEE/ACM Transactions on Networking. 2019. Vol. 27. No. 6. P. 2252–2265.

21. Announcing Amazon Redshift data lake export: share data in Apache Parquet format. URL: https://aws.amazon.com/about-aws/whats-new/2019/12/announcing-amazon-redshift-data-lake-export/#:~:text=The%20Parquet%20format%20is%20up,lake%20in%20an%20open%20format.

22. Parquet. URL: https://databricks.com/glossary/what-is-parquet.

DEVELOPMENT A DATA VALIDATION MODULE TO SATISFY THE RETENTION POLICY METRIC

Aigul Sibgatullina[1][0000-0003-4014-9558], **Azat Yakupov**[2][0000-0002-2333-8819]

[1, 2]*Kazan (Volga Region) Federal University, 35 Kremlevskaya str., Kazan, 420008*

[1]aigul.sibgatulli@gmail.com, [2]azat.yakupov@it.kfu.ru

Abstract

Every year the size of the global big data market is growing. Analysing these data is essential for good decision-making. Big data technologies lead to a significant cost reduction with use of cloud services, distributed file systems, when there is a need to store large amounts of information. The quality of data analytics is dependent on the quality of the data themselves. This is especially important if the data has a retention policy and migrates from one source to another, increasing the risk of a data loss. Prevention of negative consequences from data migration is achieved through the process of data reconciliation – a comprehensive verification of large amounts of information in order to confirm their consistency.

This article discusses probabilistic data structures that can be used to solve the problem, and suggests an implementation – data integrity verification module using a Counting Bloom filter. This module is integrated into Apache Airflow to automate its invocation.

Keywords: *big data, retention policy, partition, parquet file, Bloom filter*

REFERENCES

1. Big Data Market worth $273.4 billion by 2026. URL: https://www.marketsandmarkets.com/Market-Reports/big-data-market-1068.html.

2. Data Retention Policy: What Is It and How to Build One. URL: https://www.techtarget.com/searchdatabackup/definition/data-retention-policy.

3. *Batra S., Garg S., Kaur R., Kumar N., Singh A., Zomaya A.Y.* Probabilistic data structures for big data analytics: A comprehensive review // Knowledge-Based Systems. 2019. Vol. 188. No. 104987. P. 54–75.

4. *Choi K.W., Hossain E., Wiriaatmadja D.T.* Discovering mobile applications in cellular device-to-device communications: Hash function and bloom filter-based approach // IEEE Transactions on Mobile Computing. 2016. Vol. 15. No. 2. P. 336–349.

5. *Sasikala J., Thaiyalnayaki S.* Indexing near-duplicate images in web search using minhash algorithm // International Conference on Processing of Materials, Minerals and Energy. 2018. Vol. 5. No. 1. P. 1943–1949.

6. *Drew J., Hahsler M., Moore T.* Polymorphic Malware Detection Using Sequence Classification Methods // IEEE Security and Privacy Workshops (SPW). 2016. P. 81–87.

7. *Borgohain S.K., Nayak S., Patgiri R.* rDBF: A r-Dimensional Bloom Filter for massive scale membership query // Journal of Network and Computer Applications. 2019. Vol. 136. P. 100–113.

8. *Batra S., Garg S., Kumar N., Singh A.* Probabilistic data structure-based community detection and storage scheme in online social networks // Future Generation Computer Systems. 2019. Vol. 94. P. 173–184.

9. *Guo D., Luo L., Luo X., Ma R. T. B., Rottenstreich O.* Optimizing Bloom Filter: Challenges, Solutions, and Comparisons // IEEE Communications Surveys & Tutorials. 2019. Vol. 21. No. 2. P. 1912–1949.

10. *Boy O., Chazelle B., Kilian J., Rubinfeld R., Tal A.* The Bloomier filter: An efficient data structure for static support lookup tables // SODA. 2004. P. 30–39.

11. *Hazeyama H., Kadobayashi Y., Matsumoto Y.* Adaptive Bloom filter: A space-efficient counting algorithm for unpredictable network traffic // IEICE Transactions on Information and Systems. 2008. Vol. 91. No. 5. P. 1292–1299.

12. *Song T., Wang X., Zhou Y.* EABF: Energy efficient self-adaptive Bloom filter for network packet processing // IEEE International Conference on Communications (ICC). 2012. P. 2729–2734.

13. *Filippova D., Kingsford C., Pellow D.* Improving Bloom filter performance on sequence data using k-mer Bloom filters // J. Comput. Biol. 2017. Vol. 26. No. 6. P. 547–557.

14. *Calderoni L., Maio D., Palmieri P.* Location privacy without mutual trust: The spatial Bloom filter // Computer Communications. 2015. Vol. 68. P. 4–12.

15. *Du D.H.C., Lu G., Nam Y.J.* BloomStore: Bloom filter based memory-efficient key-value store for indexing of data de-duplication on flash // IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST). 2012. P. 1–11.

16. *Deng F., Rafiei D.* Approximately detecting duplicates for streaming data using stable Bloom filters // ACM SIGMOD international conference on Management of data. 2006. P. 25–36.

17. *Ahmadi M., Geravand S.* A novel adjustable matrix Bloom filterbased copy detection system for digital libraries // IEEE 11th International Conference on Computer and Information Technology. 2011. P. 518–525.

18. *Guo J., Li F., Peng Y., Qian W., Zhou A.* Persistent Bloom Filter: Membership Testing for the Entire History // International Conference on Management of Data. 2018. P. 1037–1052.

19. *Nayak S., Patgiri R.* A Review on Role of Bloom Filter on DNA Assembly // IEEE Access. 2019. Vol. 7. P. 66939–66954.

20. *Reviriego P., Rottenstreich O.* The Tandem Counting Bloom Filter – It Takes Two Counters to Tango // IEEE/ACM Transactions on Networking. 2019. Vol. 27. No. 6. P. 2252–2265.

21. Announcing Amazon Redshift data lake export: share data in Apache Parquet format. URL: https://aws.amazon.com/about-aws/whats-new/2019/12/announcing-amazon-redshift-data-lake-export/#:~:text=The%20Parquet%20format%20is%20up,lake%20in%20an%20open%20format.

22. Parquet. URL: https://databricks.com/glossary/what-is-parquet.

СВЕДЕНИЯ ОБ АВТОРАХ

СИБГАТУЛЛИНА Айгуль Ильдаровна – магистрант, Казанский (Приволжский) федеральный университет, г. Казань.

Aigul Ildarovna SIBGATULLINA – Master's student, Kazan (Volga region) Federal University, Kazan.

Email: aigul.sibgatulli@gmail.com
ORCID: 0000-0003-4014-9558

ЯКУПОВ Азат Шавкатович – старший преподаватель, Казанский (Приволжский) федеральный университет, г. Казань.

Azat Shavkatovich YAKUPOV – Senior Lecturer, Kazan (Volga region) Federal University, Kazan.

E-mail: azat.yakupov@it.kfu.ru
ORCID: 0000-0002-2333-8819

Материал поступил в редакцию 6 июня 2022 года

УДК 004

АНАЛИЗ И РАЗРАБОТКА КОНВЕЙЕРА MLOPS ДЛЯ РАЗВЕРТЫВАНИЯ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Р. Р. Ямиков[1, [0000-0001-9240-5168]], К. А. Григорян[2, [0000-0001-6470-1832]]

[1, 2]Казанский (Приволжский) федеральный университет, ул. Кремлевская, 35, г. Казань, 420008

[1]jamrustem@yandex.ru, [2]karigri@yandex.ru

Аннотация

Рост числа IT-продуктов с внедренными элементами машинного обучения (Machine Learning – ML) обуславливает повышение актуальности автоматизации процессов машинного обучения. Использование методов MLOps направлено на обеспечение обучения и эффективного развертывания приложений с производственной среде, автоматизируя решение побочных инфраструктурных вопросов слабо связанных с непосредственно разработкой модели.

Мы рассматриваем компоненты, принципы и подходы MLOps и анализируем существующие платформы и решения для построения конвейеров машинного обучения. Кроме того, предлагаем подход к построению конвейера машинного обучения на основе основных инструментов DevOps и библиотек с открытым исходным кодом.

***Ключевые слова**: MLOps, DevOps, CI/CD, CT, ML, конвейер машинного обучения.*

ВВЕДЕНИЕ

На текущий момент растет сложность полноценной разработки приложения с машинным обучением. Полностековому специалисту необходимо изучать достаточно широкую область знаний, требуется обладать не только навыками в области науки о данных, но и иных областях, таких как инфраструктура машинного обучения, развертывание приложения. Поэтому появляется тенденция к росту спроса на услуги MLOps-инженеров. Таким образом, сегодня сфера MLOps имеет высокую актуальность и востребованность среди организаций, связанных с управлением и обработкой данных.

Согласно проведенному опросу «The State of ML 2020» 331 специалиста по машинному обучению из 63 различных стран, до 40% опрошенных занимаются как непосредственно работами с проработкой моделей, так и инфраструктурными вопросами. Одной из частых проблем, с которой сталкивались респонденты в ходе работы, являлись процессы, связанные с развертыванием моделей в производственной среде [1]. В виду этого множество проектов с использованием машинного обучения терпит неудачу на стадиях проверки концепции (proof of concept) и экспериментирования еще до внедрения в производство [2]. Провалы обуславливается тем, что специалисты машинного обучения уделяют основное внимание проектированию и построению ML-модели, а не созданию готового продукта с машинным обучением. Кроме того, системы машинного обучения довольно сложны, и их бывает трудно соединить с инфраструктурой для использования в производственной среде [3].

MLOps помогает решить проблему внедрения моделей машинного обучения в производственную среду путем автоматизации как процессов машинного обучения, так и процессов развертывания ML-моделей в производстве.

ОБЗОР ЛИТЕРАТУРЫ

MLOps – это сборник различных методов, практик и инструментов для развертывания моделей машинного обучения в производстве [4]. MLOps можно рассматривать как пересечение практик машинного обучения и DevOps. DevOps – это методология, которая включает в себя практики автоматизации процессов сборки, настройки и развёртывания программного обеспечения (ПО) и объединяет рабочие процессы разработки ПО с процессами тестирования и эксплуатации для минимизации времени выпуска ПО [5].

В основе методов MLOps лежат принципы методологии DevOps. Основными из них являются непрерывная интеграция (CI) и непрерывное развертывание (CD). Непрерывная интеграция – это практика разработки ПО, которая заключается в постоянной интеграции программного кода и выполнении автоматизированной сборки продукта через частые промежутки времени для постоянного тестирования текущего состояния кода и скорейшего исправления ошибок [6]. Непрерывная доставка – подход к разработке ПО, при котором разработка проходит короткими итерациями, когда постоянно и автоматизировано выпускается новая

стабильная версия продукта для тестирования [7]. MLOps кроме непрерывной доставки и непрерывного развертывания включает в себя еще и непрерывное обучение (CT) – переобучение ML-модели при необходимости.

Уровни зрелости MLOps

В зависимости от степени автоматизации процессов продукта с машинным обучением принято относить его к одному из уровней зрелости процесса MLOps [8]. Компании Google и Microsoft выделяют две основные классификации по уровню зрелости [6].

В Google выделяют три уровня зрелости процесса MLOps-проектов по степени автоматизации шагов процесса доставки модели машинного обучения в производство [9]:

1. Ручной процесс;
2. Автоматизация конвейера машинного обучения;
3. Автоматизация конвейера CI/CD.

Microsoft в свою очередь выделяет в классификации пять уровней [10]:

1. Отсутствие процессов MLOps;
2. Наличие DevOps, но без MLOps;
3. Автоматизация обучения ML-модели;
4. Автоматизация развертывания ML-модели;
5. Полная автоматизация процессов MLOps.

Принципы MLOps

В MLOps выделяют девять принципов или наилучших практик для разработки продуктов с машинным обучением [11]:

1. Автоматизация CI/CD;
2. Оркестрация рабочих процессов – координация порядка выполнения задач конвейера машинного обучения;
3. Воспроизводимость;
4. Версионирование – отслеживание данных, модели, кода в системах контроля версий для обеспечения воспроизводимости и аудита;
5. Коллаборация – возможность совместной работы над проектом машинного обучения;

6. Непрерывное обучение и оценка ML-модели;
7. Отслеживание метаданных – каждой итерации обучения модели, ее параметров, метрик;
8. Мониторинг;
9. Обратная связь – по результатам оценок качества модели, мониторинга развернутой модели.

Компоненты MLOps

Для реализации принципов MLOps используются следующие компоненты системы MLOps [11], [12]:

1. Компонент CI/CD (реализует принципы 1, 6, 9);
2. Репозиторий программного кода (принципы 4, 5);
3. Компонент оркестрации рабочих процессов (принципы 2, 3, 6);
4. Система хранения функций и данных (feature store, принципы 3, 4);
5. Инфраструктура для обучения моделей (принцип 6);
6. Реестр моделей (принципы 3, 4);
7. Хранилище метаданных машинного обучения (принципы 4, 7);
8. Компонент обслуживания моделей (принцип 1);
9. Компонент мониторинга (принципы 8, 9).

Итеративно-инкрементный процесс MLOps

Согласно сайту MLOps полный итеративно-инкрементный процесс MLOps включает в себя три основных этапа [12]:

1. Проектирование приложения на базе машинного обучения – происходят сбор требований, определение проблематики бизнеса и дальнейшее проектирование модели машинного обучения для решения проблемы пользователя и повышения его производительности. Дополнительно идут оценка данных для обучения алгоритма и разработка архитектуры модели машинного обучения.
2. Эксперименты и разработка машинного обучения – проверка применимости алгоритма машинного обучения, происходит проверка концепции. Целью этапа будет получение стабильной модели требуемого качества для производственной среды.

3. Операции машинного обучения – непосредственно само развертывание обученной модели в производство с использованием методов DevOps.

ОБЗОР СУЩЕСТВУЮЩИХ ПЛАТФОРМ

Yandex DataSphere – это облачный сервис для разработки и дальнейшей эксплуатации моделей машинного обучения, который предоставляет все необходимые инструменты и ресурсы для полного цикла разработки машинного обучения [13].

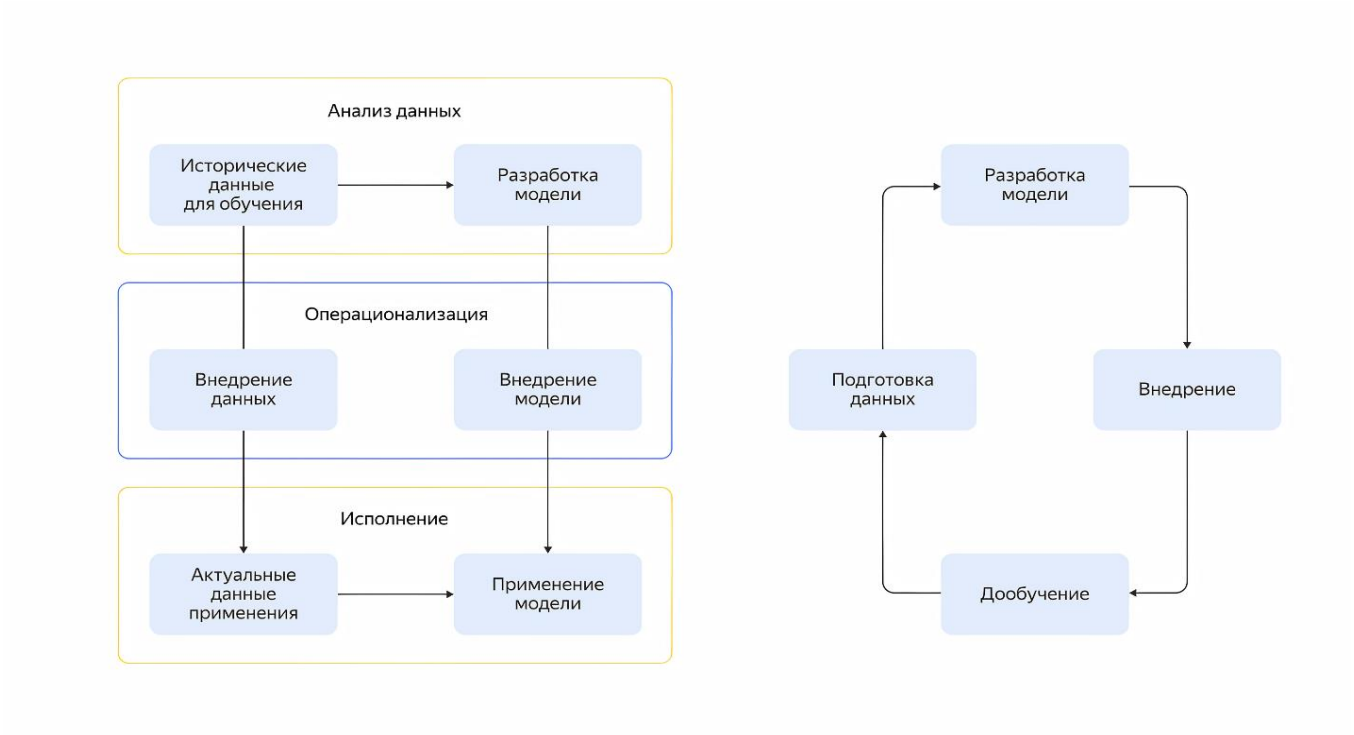

Рисунок 1. Полный цикл машинного обучения [13]

Разработка ML-моделей происходит в среде для интерактивных вычислений Jupyter Notebook. Блокнот содержит набор из множества ячеек, в каждой из которых независимо исполняется код [14].

Проекты в Yandex DataSphere представляют собой блокнот Jupiter. В них происходит сохранение полного состояния блокнота, включая переменные, установленные пакеты и др. [14].

В проект Yandex DataSphere данные можно загружать как через интерфейс (при небольшом объеме), так и через сетевые хранилища и базы данных [14].

Yandex DataSphere сохраняет состояния ноутбука в проекте при помощи системы контрольных точек, в которых хранятся как данные блокнота (код ячеек, вывод и значения переменных в определенный момент времени), так и данные хранилища проекта [14].

После разработки моделей их можно легко развернуть в виде микросервисов. Предобученные модели разворачиваются на инстансах — виртуальных машинах, где зафиксированы состояние интерпретатора и код модели. Затем эти инстансы объединяются в ноды (группы виртуальных машин). Для доступа к ним используется API [15].

Главный недостаток сервиса заключается в том, что это коммерческое ПО. Кроме того, сервис скорее рассчитан на экспериментирование с машинным обучением, чем на полноценный конвейер с оркестрацией и развертыванием модели.

MLFlow — это платформа с открытым исходным кодом, для управления целиком всем жизненным циклом машинного обучения [16]. Фреймворк включает в себя решение задач с экспериментами, воспроизводимостью и развертыванием, так же содержит центральный реестр моделей.

Платформа MLFlow поддерживает интеграцию с Docker и Kubernetes и интегрирована с основными фреймворками машинного обучения. Содержит четыре основных модуля, которые можно использовать независимо друг от друга:

- MLflow Tracking – необходим для записи экспериментов, параметров моделей, версионирования кода, логирования метрик моделей для дальнейшей визуализации в пользовательском интерфейсе [17];
- MLflow Projects – упаковка кода проекта для дальнейшего использования и воспроизведения конвейеров машинного обучения; проект описан в файле MLProject в формате yaml [18];
- MLflow Models – упаковка, хранение и развертывание моделей машинного обучения в различных средах; поддерживает развертывание в виде REST API эндпоинтов и упаковку в Docker-образы [19];
- Model Registry – хранение, аннотирование и управление жизненным циклом модели в центральном репозитории моделей [20].

Среди крупных недостатков MLFlow выделяется проблема отсутствия ролей

пользователей и функционала безопасности, что приводит к затруднениям совместной работы над моделями [21]. Кроме того, еще одни из проблем MLFlow – сложности с развёртыванием моделей на различных платформах и отсутствие мониторинга производительности моделей [21], [22].

KubeFlow — это бесплатная платформа для машинного обучения с открытым исходным кодом, которая делает развертывание конвейеров рабочих процессов машинного обучения в контейнерах Kubernetes простым и легко масштабируемым [23]. Платформа предоставляет полный набор инструментов для оркестрации стека машинного обучения для использования при развертывании, масштабировании и управлении системами машинного обучения.

Особенности платформы Kubeflow [24]:

- сервисы для создания интерактивных блокнотов Jupyter и управления ими при процессе обработке данных;
- операторы для обучения моделей машинного обучения TensorFlow, настройки гиперпараметров и управления нагрузками;
- обслуживание моделей ML – экспорт обученных моделей в Kubernetes через TensorFlow Serving; также есть интеграция с Seldon Core для развертывания моделей;
- конвейер Kubeflow Pipelines — платформа для создания и развертывания масштабируемых рабочих процессов машинного обучения на основе контейнеров Docker;
- поддержка различных фреймворков машинного обучения: TensorFlow, PyTorch, Apache MXNet, MPI, XGBoost, Chainer и др.

Kubeflow содержит множество инструментов и может быть представлен как платформа для размещения компонентов системы машинного обучения поверх Kubernetes [25].

К недостаткам Kubeflow относится отсутствие версионирования данных и конвейера [22]. Кроме этого, Kubeflow сложно настроить, у него высокий порог входа, а для его использования необходимы глубокие знания в Kubernetes [26].

DVC и CML

DVC (data version control) – инструмент с открытым исходным кодом для

управления версиями данных проектов машинного обучения, основанный на уже существующем инструментарии Git, CI/CD и т. д. Система DVC позволяет создавать воспроизводимые конвейеры машинного обучения; отслеживает и обрабатывает файлы, наборы данных, модели машинного обучения и их метрики в виде кода [27].

DVC – это инструмент, работающий совместно с Git. DVC использует файл dvc для контроля артефактов машинного обучения, а Git отвечает за контроль версий кода и файла dvc [28].

Этапы конвейера машинного обучения описаны в файле dvc.yaml с указанием используемых скриптов Python для каждого шага, его параметров, используемых и генерируемых файлов [29].

CML (continuous machine learning) – библиотека с открытым исходным кодом для реализации непрерывной интеграции и непрерывной доставки (CI/CD) в проектах машинного обучения. Библиотека используется для автоматизации процесса разработки, обучения и оценки моделей, сравнения экспериментов с моделями машинного обучения и отслеживания изменений наборов данных [30].

CML может применяться совместно с DVC, при этом DVC будет управлять данными и моделями, а CML – оркестровкой инфраструктуры, тестированием и мониторингом [31].

CML интегрируется в GitHub Actions или GitLab CI/CD и работает совместно с ними, создавая совместный конвейер.

CML использует GitFlow на основе рабочего процесса Git для управления экспериментами через версионирование DVC данных и моделей. CML может создавать отчеты в пулл-реквестах с метриками и графиками оценок моделей, позволяя команде принимать информированные решения [32].

Среди недостатков связки DVC и CML выделяется то, что это неполноценная платформа, а лишь часть конвейера MLOps, для работы которым дополнительно требуются инструменты CI/CD, развертывания и оркестрации модели, мониторинга модели. Однако это в свою очередь обеспечивает гибкость подхода.

МЕТОДОЛОГИЯ

Для разработки конвейера MLOps нами использовался подход, базирующийся на основных инструментах DevOps и библиотек с открытым исходным кодом. Были использованы следующие технологии для реализации компонентов MLOps:

1. Gitlab в качестве компонента репозитория программного кода;
2. Gitlab CI/CD и CML для компонент CI/CD;
3. Gitlab CI/CD с несколькими gitlab runner для выполнения заданий конвейера обучения модели на высокопроизводительной системе и развертывания на сервере; компоненты оркестрации рабочих процессов и инфраструктура для обучения моделей;
4. DVC и удаленное хранилище в Google Drive для компонентов системы хранения функций и данных, реестра моделей, хранилище метаданных машинного обучения;
5. Микросервис FastApi как компонент обслуживания моделей;
6. Сбор и хранение метрик в Prometheus и их визуализация в виде графиков в Grafana в качестве компонента мониторинга.

Был построен конвейер машинного обучения с указанной ниже блок-схемой для проведения экспериментов (рис. 2). Эксперименты проводились с использованием средств Git и Gitlab на удаленном сервере. Разработчик вносит локально правки к модели, создает коммит и отправляет его в репозиторий Gitlab, где в дальнейшем открывает pull request. Следующие процессы конвейера MLOps по обучению модели происходят удаленно. После завершения работ конвейера результаты публикуются в виде комментария к коммиту в PR. Затем разработчик может закрыть PR и слить ветку в главную, тем самым запустив дальнейшие шаги конвейера по интеграции и развертыванию модели.

Схема конвейера MLOps представлена на рис. 3.

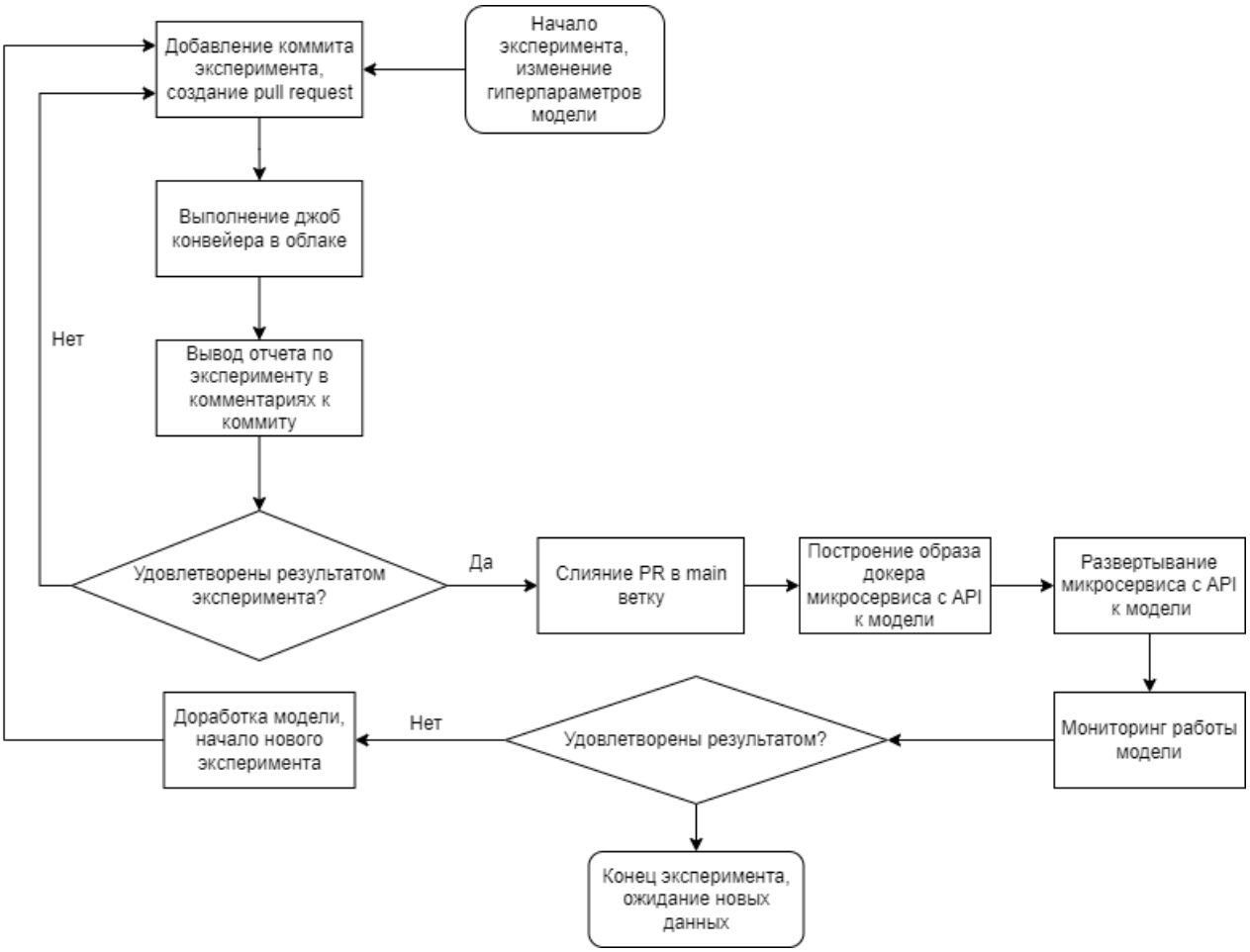

Рисунок 2. Блок схема процесса экспериментирования

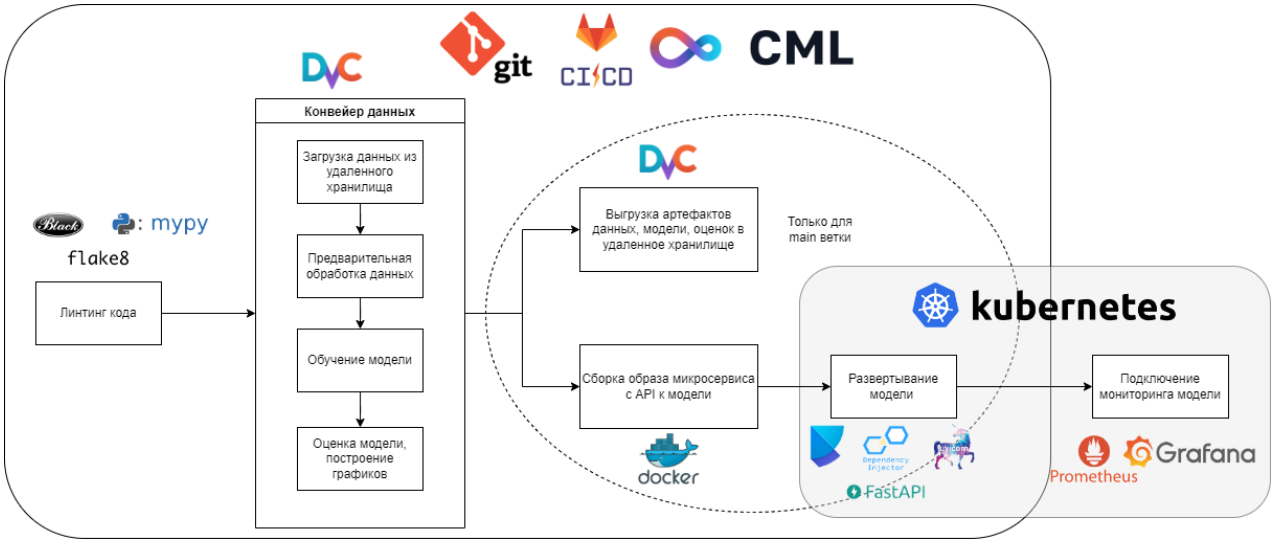

Рисунок 3. Схема конвейера

Конвейер процессов MLOps был разработан с использованием инструментов для интеграции Git, Gitlab CI/CD и CML и включает в себя 4 основных блока:

1. Линтинг программного кода файлов Python процессов обработки данных и построения модели машинного обучения; кода приложения микросервиса к API данной модели. Для синтаксического анализа и форматирования кода были использованы библиотеки: форматер кода black под стандарты PEP8, статический анализатор типов mypy, статический анализатор flake8;
2. Конвейер данных, состоящий из 4 шагов:

- загрузка данных;
- предварительная обработка данных;
- обучения модели на основе этих данных;
- оценка модели с построением графиков.

Для построения конвейера данных использовалась библиотека DVC.

3. Процессы интеграции:

- выгрузка артефактов в удаленное облачное хранилище, использовался DVC;
- сборка образа микросервиса с API к модели с использованием Docker;

4. Процессы развертывания на сервере в кластере оркестратора Docker контейнеров Kubernetes [33]:

- развертывание микросервиса на асинхронном фреймворке FastApi, с веб-сервером Uvicorn, с использованием библиотеки для внедрения зависимостей Dependency Injector и инструмента для управления библиотеками Poetry;
- подключение к ранее развернутым сервисам для мониторинга; использовался инструмент Prometheus для сбора и хранения событий, метрик и Grafana – для аналитики и визуализации собранных метрик [34] [35].

Стадии конвейера с процессами интеграции и процессами развертывания выполняются только в главной ветке.

РЕЗУЛЬТАТЫ

Был успешно реализован прототип конвейера MLOps. API микросервиса с моделью содержит три эндпоинта (рис. 4):

1. эндпоинт /predict с загрузкой файла для классификации моделью –

возвращает спрогнозированный класс;

2. эндпоинт /get_experiment_metas, возвращающий json с метаданными экспериментов: тег коммита эксперимента, метрики и параметры модели машинного обучения;

3. эндпоинт /replace_ml_model для «горячей» замены используемой модели машиного обучения без перезапуска сервиса; замена производится по тегу эксперимента, который можно просмотреть через эндпоинт /get_experiment_metas.

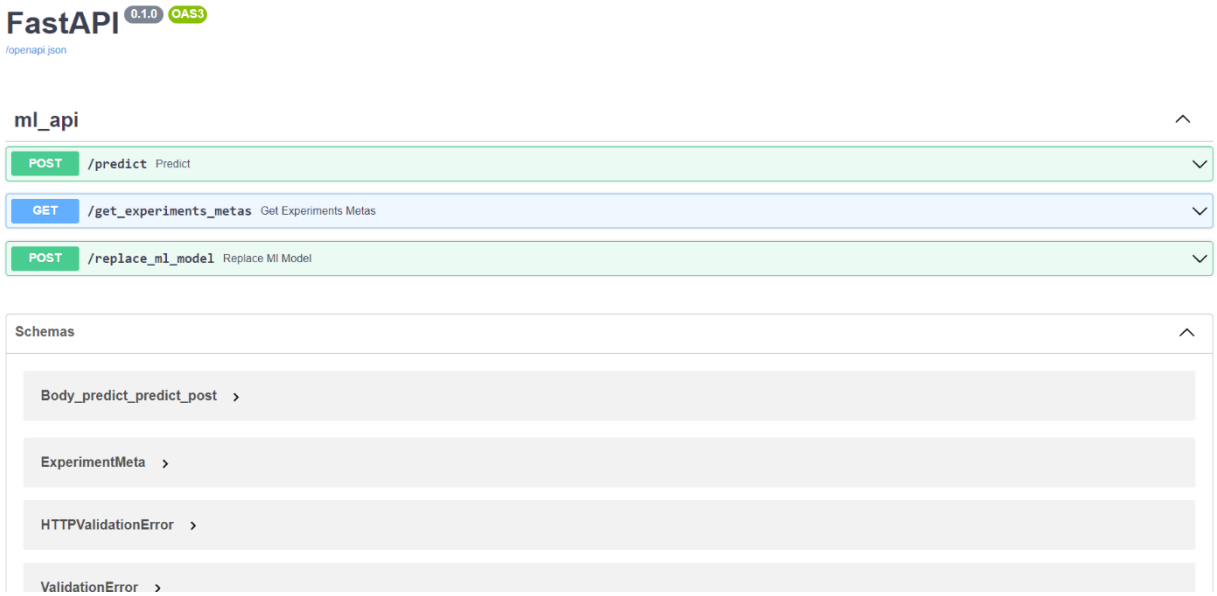

Рисунок 4. Страница интерфейса к API микросервиса

Показатели производительности модели выводятся на графики в Grafana (рис. 5 и 6).

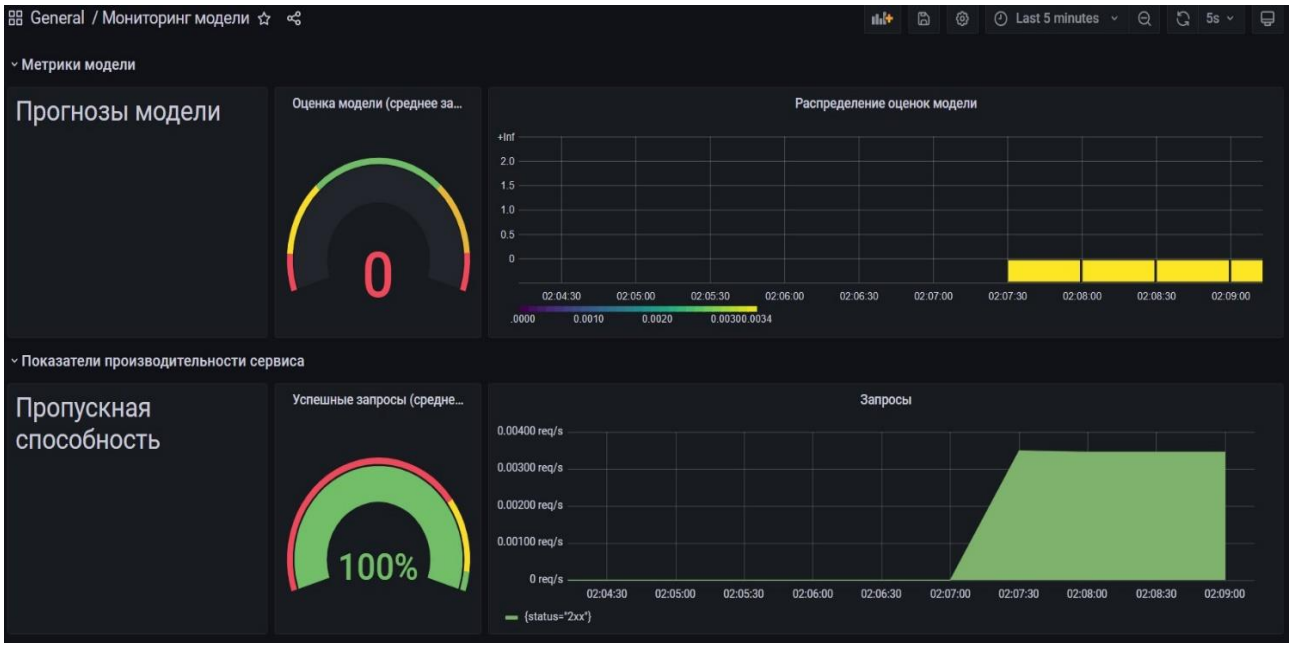

Рисунок 5. Графики метрик микросервиса с моделью

Рисунок 6. Показатели загруженности системы

ЗАКЛЮЧЕНИЕ

В ходе исследования был проведен анализ области MLOps, изучены уровни развития MLOps, его принципы и компоненты. Рассмотрены различные подходы и готовые платформы MLOps.

В результате исследования построен прототип конвейера системы MLOps для автоматизации задач разработки, развертывания и поддержки моделей ма-

шинного обучения на основе основных инструментов DevOps и библиотек с открытым исходным кодом.

В дальнейшем планируется более глубокое внедрение мониторинга и триггеров для дообучения модели.

СПИСОК ЛИТЕРАТУРЫ

1. *Makinen S., Skogstrom H., Laaksonen E., Mikkonen T.* Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? // Software Engineering for AI (WAIN) of 43rd International Conference on Software Engineering (ICSE). 2021.

2. *Van der Meulen R., McCall T.* Gartner Says Nearly Half of CIOs Are Planning to Deploy Artificial Intelligence // Gartner. 2018. URL: https://www.gartner.com/en/newsroom/press-releases/2018-02-13-gartner-says-nearly-half-of-cios-are-planning-to-deploy-artificial-intelligence (дата обращения: 01.06.2022).

3. *Posoldova A.* Machine Learning Pipelines: From Research to Production // IEEE Potentials. 2020. Vol. 39, No. 6. P. 38–42. https://doi.org/10.1109/MPOT.2020.3016280

4. *Alla S., Adari S.K.* What is mlops? // In: Beginning MLOps with MLFlow. Berkeley: Apress, 2021. P. 79–124.

5. *Gift N., Deza A.* Practical MLOps. O'Reilly Media, Inc., 2021.

6. *Symeonidis G., Nerantzis E., Kazakis A., Papakostas G.A.* MLOps – Definitions, Tools and Challenges // IEEE Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas. 2022. Vol. 12. P. 453–460. https://doi.org/10.48550/arXiv.2201.00162

7. *Chen L.* Continuous Delivery: Huge Benefits, but Challenges Too // IEEE Software. 2015. Vol. 32. P. 50–54. https://doi.org/10.1109/MS.2015.27

8. *John M., Olsson H., Bosch J.* Towards MLOps: A Framework and Maturity Model // Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Palermo. 2021. Vol. 47. P. 1–8. https://doi.org/10.1109/SEAA53835.2021.00050

9. MLOps: Continuous delivery and automation pipelines in machine learning // Google Cloud. 2021. URL: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning (дата обращения:

03.07.2021).

10. Machine Learning operations maturity model // Microsoft. URL: https://docs.microsoft.com/en-us/azure/architecture/example-scenario/mlops/mlops-maturity-model, last accessed 2022/05/30.

11. *Kreuzberger D., Kühl N., Hirschl S.* Machine Learning Operations (MLOps): Overview, Definition, and Architecture // arXiv preprint arXiv:2205.02302. 2022. https://doi.org/10.48550/arXiv.2205.02302

12. MLOps Principles // MLOps. URL: https://ml-ops.org/content/mlops-principles (дата обращения: 03.07.2021).

13. Yandex DataSphere // Yandex Cloud. URL: https://cloud.yandex.ru/services/datasphere, (дата обращения: 05.03.2022).

14. Проект // Yandex datasphere документация. URL: https://cloud.yandex.ru/docs/datasphere/concepts/project, (дата обращения: 05.03.2022).

15. Развертывание эксплуатации моделей // Yandex datasphere документация. URL: https://cloud.yandex.ru/docs/datasphere/concepts/deploy, (дата обращения: 05.03.2022).

16. MLFlow. URL: https://mlflow.org (дата обращения: 28.12.2021).

17. MLflow Tracking // MLflow. URL: https://mlflow.org/docs/latest/tracking.html (дата обращения: 28.12.2021).

18. MLflow Projects // MLflow. URL: https://mlflow.org/docs/latest/projects.html, (дата обращения: 28.12.2021).

19. MLflow Models // MLflow. URL: https://mlflow.org/docs/latest/models.html (дата обращения: 28.12.2021).

20. MLflow Model Registry // MLflow. URL: https://mlflow.org/docs/latest/model-registry.html (дата обращения: 28.12.2021).

21. *Khandelwal N.* MLflow Alternatives for Data Version Control: DVC vs. MLflow // Censious. URL: https://censius.ai/blogs/dvc-vs-mlflow (дата обращения: 30.05.2022).

22. *Hewage N., Meedeniya D.* Machine Learning Operations: A Survey on MLOps Tool Support // arXiv preprint arXiv:2202.10169. 2022. https://doi.org/10.48550/arXiv.2202.10169

23. Introduction // Kubeflow documentation. URL: https://www.kubeflow.org/docs/started/introduction (дата обращения: 11.03.2022).

24. What is Kubeflow? // Kubeflow. URL: https://www.kubeflow.org (дата обращения: 11.03.2022).

25. Architecture // Kubeflow documentation. URL: https://www.kubeflow.org/docs/started/architecture (дата обращения: 11.03.2022).

26. *Kaewsanmua K.* Best 8 Machine Learning Model Deployment Tools That You Need to Know // Neptune. 2021. URL: https://neptune.ai/blog/best-8-machine-learning-model-deployment-tools (дата обращения: 01.06.2022).

27. DVC. URL: https://dvc.org (дата обращения: 27.12.2021).

28. *Zhao Y.* MLOps: Data versioning with DVC — Part Ⅰ // Medium. 2020. URL: https://yizhenzhao.medium.com/mlops-data-versioning-with-dvc-part-ⅰ-8b3221df8592 (дата обращения: 27.12.2021).

29. *Mesquita D.* The ultimate guide to building maintainable Machine Learning pipelines using DVC // Towards data science. 2020. URL: https://towardsdatascience.com/the-ultimate-guide-to-building-maintainable-machine-learning-pipelines-using-dvc-a976907b2a1b (дата обращения: 27.12.2021).

30. CML Documentation // CML. URL: https://cml.dev/doc (дата обращения: 27.12.2021).

31. Continuous Integration and Deployment for Machine Learning // DVC. URL: https://dvc.org/doc/use-cases/ci-cd-for-machine-learning (дата обращения: 27.12.2021).

32. Continuous Integration with CML and Github Actions // MLOps Guide. URL: https://mlops-guide.github.io/CICD/cml_testing (дата обращения: 27.12.2021).

33. Kubernetes Documentation // Kubernetes. URL: https://kubernetes.io/docs/hom (дата обращения: 22.05.2022).

34. What is Prometheus? // Prometheus.
URL: https://prometheus.io/docs/introduction/overview

35. Grafana // Grafana Labs. URL: https://grafana.com/grafana (дата обращения: 22.05.2022).

ANALYSIS AND DEVELOPMENT OF THE MLOPS PIPELINE FOR ML MODEL DEPLOYMENT

Rustem Yamikov[1][0000-0001-9240-5168], Karen Grigoryan[2][0000-0001-6470-1832]

[1, 2] Kazan (Volga Region) Federal University, 35 Kremlevskaya str., Kazan, 420008

[1]jamrustem@yandex.ru, [2]karigri@yandex.ru

Abstract

The growth in the number of IT products with machine-learning features is increasing the relevance of automating machine-learning processes. The use of MLOps techniques is aimed at providing training and efficient deployment of applications in a production environment by automating side infrastructure issues that are not directly related to model development.

In this paper, we review the components, principles, and approaches of MLOps and analyze existing platforms and solutions for building machine learning pipelines. In addition, we propose an approach to build a machine learning pipeline based on basic DevOps tools and open-source libraries.

Keywords: *MLOps, DevOps, CI/CD, CT, ML, machine learning pipeline.*

REFERENCES

1. *Makinen S., Skogstrom H., Laaksonen E., Mikkonen T.* Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? // Software Engineering for AI (WAIN) of 43rd International Conference on Software Engineering (ICSE). 2021.

2. *Van der Meulen R., McCall T.* Gartner Says Nearly Half of CIOs Are Planning to Deploy Artificial Intelligence // Gartner. 2018. URL: https://www.gartner.com/en/newsroom/press-releases/2018-02-13-gartner-says-nearly-half-of-cios-are-planning-to-deploy-artificial-intelligence, last accessed 2022/06/01.

3. *Posoldova A.* Machine Learning Pipelines: From Research to Production // IEEE Potentials. 2020. Vol. 39, No. 6. P. 38–42. https://doi.org/10.1109/MPOT.2020.3016280

4. *Alla S., Adari S.K.* What is mlops? // In: Beginning MLOps with MLFlow.

Berkeley: Apress, 2021. P. 79–124.

5. *Gift N., Deza A.* Practical MLOps. O'Reilly Media, Inc., 2021.

6. *Symeonidis G., Nerantzis E., Kazakis A., Papakostas G.A.* MLOps - Definitions, Tools and Challenges // IEEE Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas. 2022. Vol. 12. P. 453–460. https://doi.org/10.48550/arXiv.2201.00162

7. *Chen L.* Continuous Delivery: Huge Benefits, but Challenges Too // IEEE Software. 2015. Vol. 32. P. 50–54. https://doi.org/10.1109/MS.2015.27

8. *John M., Olsson H., Bosch J.* Towards MLOps: A Framework and Maturity Model // Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Palermo. 2021. Vol. 47. P. 1–8. https://doi.org/10.1109/SEAA53835.2021.00050

9. MLOps: Continuous delivery and automation pipelines in machine learning // Google Cloud. 2021. URL: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning, last accessed 2021/07/03.

10. Machine Learning operations maturity model // Microsoft. URL: https://docs.microsoft.com/en-us/azure/architecture/example-scenario/mlops/mlops-maturity-model, last accessed 2022/05/30.

11. *Kreuzberger D., Kühl N., Hirschl S.* Machine Learning Operations (MLOps): Overview, Definition, and Architecture // arXiv preprint arXiv:2205.02302. 2022. https://doi.org/10.48550/arXiv.2205.02302

12. MLOps Principles // MLOps. URL: https://ml-ops.org/content/mlops-principles, last accessed 2021/07/03.

13. Yandex DataSphere // Yandex Cloud. URL: https://cloud.yandex.ru/services/datasphere, last accessed 2022/03/05.

14. Proekt // Yandex datasphere dokumentaciya. URL: https://cloud.yandex.ru/docs/datasphere/concepts/project, last accessed 2022/03/05.

15. Razvertyvanie ekspluatacii modelej // Yandex datasphere dokumentaciya. URL: https://cloud.yandex.ru/docs/datasphere/concepts/deploy, last accessed 2022/03/05.

16. MLFlow. URL: https://mlflow.org, last accessed 2021/12/28.

17. MLflow Tracking // MLflow. URL: https://mlflow.org/docs/latest/tracking.html, last accessed 2021/12/28.

18. MLflow Projects // MLflow. URL: https://mlflow.org/docs/latest/projects.html, last accessed 2021/12/28.

19. MLflow Models // MLflow. URL: https://mlflow.org/docs/latest/models.html, last accessed 2021/12/28.

20. MLflow Model Registry // MLflow. URL: https://mlflow.org/docs/latest/model-registry.html, last accessed 2021/12/28.

21. *Khandelwal N.* MLflow Alternatives for Data Version Control: DVC vs. MLflow // Censious. URL: https://censius.ai/blogs/dvc-vs-mlflow, last accessed 2022/05/30.

22. *Hewage N., Meedeniya D.* Machine Learning Operations: A Survey on MLOps Tool Support // arXiv preprint arXiv:2202.10169. 2022. https://doi.org/10.48550/arXiv.2202.10169

23. Introduction // Kubeflow documentation. URL: https://www.kubeflow.org/docs/started/introduction, last accessed 2022/03/11.

24. What is Kubeflow? // Kubeflow. URL: https://www.kubeflow.org, last accessed 2022/03/11.

25. Architecture // Kubeflow documentation. URL: https://www.kubeflow.org/docs/started/architecture, last accessed 2022/03/11.

26. *Kaewsanmua K.* Best 8 Machine Learning Model Deployment Tools That You Need to Know // Neptune. 2021. URL: https://neptune.ai/blog/best-8-machine-learning-model-deployment-tools, last accessed 2022/06/01.

27. DVC. URL: https://dvc.org, last accessed 2021/12/27.

28. *Zhao Y.* MLOps: Data versioning with DVC — Part Ⅰ // Medium. 2020. URL: https://yizhenzhao.medium.com/mlops-data-versioning-with-dvc-part-ⅰ-8b3221df8592, last accessed 2021/12/27.

29. *Mesquita D.* The ultimate guide to building maintainable Machine Learning pipelines using DVC // Towards data science. 2020. URL: https://towardsdatascience.com/the-ultimate-guide-to-building-maintainable-machine-learning-pipelines-using-dvc-a976907b2a1b, last accessed 2021/12/27.

30. CML Documentation // CML. URL: https://cml.dev/doc, last accessed

2021/12/27.

31. Continuous Integration and Deployment for Machine Learning // DVC. URL: https://dvc.org/doc/use-cases/ci-cd-for-machine-learning, last accessed 2021/12/27.

32. Continuous Integration with CML and Github Actions // MLOps Guide. URL: https://mlops-guide.github.io/CICD/cml_testing, last accessed 2021/12/27.

33. Kubernetes Documentation // Kubernetes. URL: https://kubernetes.io/docs/home, last accessed 2022/05/22.

34. What is Prometheus? // Prometheus. URL: https://prometheus.io/docs/introduction/overview, last accessed 2022/05/22.

35. Grafana // Grafana Labs. URL: https://grafana.com/grafana, last accessed 2022/05/22.

СВЕДЕНИЯ ОБ АВТОРАХ

ЯМИКОВ Рустем Рафикович – магистрант, Казанский (Приволжский) федеральный университет, г. Казань.

Rustem Raficovich YAMIKOV – Master's student, Kazan (Volga region) Federal University, Kazan.

Email: jamrustem@yandex.ru
ORCID: 0000-0001-9240-5168

ГРИГОРЯН Карен Альбертович – кандидат экономических наук, доцент, Казанский (Приволжский) федеральный университет, г. Казань.

Karen Albertovich GRIGORIAN – Candidate of Economics, Associate Professor, Kazan (Volga region) Federal University, Kazan.

Email: karigri@yandex.ru
ORCID: 0000-0001-6470-1832

Материал поступил в редакцию 25 мая 2022 года