ТРУДЫ

ИНСТИТУТА СИСТЕМНОГО ПРОГРАММИРОВАНИЯ РАН

PROCEEDINGS OF THE INSTITUTE FOR SYSTEM PROGRAMMING OF THE RAS

ISSN Print 2079-8156
Том 36 Выпуск 5

ISSN Online 2220-6426
Volume 36 Issue 5

Институт системного
программирования
им. В.П. Иванникова РАН

Москва, 2024

ИСП РАН

ISSN 2079-8156 (Print), ISSN 2220-6426 (Online)

Труды Института системного программирования РАН
Proceedings of the Institute for System Programming of the RAS

Труды ИСП РАН – это издание с двойной анонимной системой рецензирования, публикующее научные статьи, относящиеся ко всем областям системного программирования, технологий программирования и вычислительной техники. Целью издания является формирование научно-информационной среды в этих областях путем публикации высококачественных статей в открытом доступе.
Издание предназначено для исследователей, студентов и аспирантов, а также практиков. Оно охватывает широкий спектр тем, включая, в частности, следующие:

- операционные системы;
- компиляторные технологии;
- базы данных и информационные системы;
- параллельные и распределенные системы;
- автоматизированная разработка программ;
- верификация, валидация и тестирование;
- статический и динамический анализ;
- защита и обеспечение безопасности ПО;
- компьютерные алгоритмы;
- искусственный интеллект.

Журнал издается по одному тому в год, шесть выпусков в каждом томе.
Поддерживается открытый доступ к содержанию издания, обеспечивая доступность результатов исследований для общественности и поддерживая глобальный обмен знаниями.
Труды ИСП РАН реферируются и/или индексируются в:

Proceedings of ISP RAS are a double-blind peer-reviewed journal publishing scientific articles in the areas of system programming, software engineering, and computer science. The journal's goal is to develop a respected network of knowledge in the mentioned above areas by publishing high quality articles on open access.
The journal is intended for researchers, students, and practitioners. It covers a wide variety of topics including (but not limited to):

- Operating Systems.
- Compiler Technology.
- Databases and Information Systems.
- Parallel and Distributed Systems.
- Software Engineering.
- Software Modeling and Design Tools.
- Verification, Validation, and Testing.
- Static and Dynamic Analysis.
- Software Safety and Security.
- Computer Algorithms.
- Artificial Intelligence.

The journal is published one volume per year, six issues in each volume.
Open access to the journal content allows to provide public access to the research results and to support global exchange of knowledge. **Proceedings of ISP RAS** is abstracted and/or indexed in:

УДК004.45

Редколлегия

Главный редактор - Аветисян Арутюн Ишханович, академик РАН, доктор физико-математических наук, профессор, ИСП РАН (Москва, Российская Федерация)

Заместитель главного редактора – Карпов Леонид Евгеньевич, д.т.н., ИСП РАН (Москва, Российская Федерация)

Члены редколлегии

Воронков Андрей Анатольевич, доктор физико-математических наук, профессор, Университет Манчестера (Манчестер, Великобритания)
Вирбицкайте Ирина Бонавентуровна, профессор, доктор физико-математических наук, Институт систем информатики им. академика А.П. Ершова СО РАН (Новосибирск, Россия)
Коннов Игорь Владимирович, кандидат физико-математических наук, Технический университет Вены (Вена, Австрия)
Ластовецкий Алексей Леонидович, доктор физико-математических наук, профессор, Университет Дублина (Дублин, Ирландия)
Ломазова Ирина Александровна, доктор физико-математических наук, профессор, Национальный исследовательский университет «Высшая школа экономики» (Москва, Российская Федерация)
Новиков Борис Асенович, доктор физико-математических наук, профессор, Санкт-Петербургский государственный университет (Санкт-Петербург, Россия)
Петренко Александр Федорович, доктор наук, Исследовательский институт Монреаля (Монреаль, Канада)
Черных Андрей, доктор физико-математических наук, профессор, Научно-исследовательский центр CICESE (Энсенада, Баха Калифорния, Мексика)
Шустер Ассаф, доктор физико-математических наук, профессор, Технион — Израильский технологический институтTechnion (Хайфа, Израиль)

Editorial Board

Editor-in-Chief - Arutyun I. Avetisyan, Academician of RAS, Dr. Sci. (Phys.–Math.), Professor, Ivannikov Institute for System Programming of the RAS (Moscow, Russian Federation)

Deputy Editor-in-Chief – Leonid E. Karpov, Dr. Sci. (Eng.), Ivannikov Institute for System Programming of the RAS (Moscow, Russian Federation)

Editorial Members

Igor Konnov, PhD (Phys.–Math.), Vienna University of Technology (Vienna, Austria)
Alexey Lastovetsky, Dr. Sci. (Phys.–Math.), Professor, UCD School of Computer Science and Informatics (Dublin, Ireland)
Irina A. Lomazova, Dr. Sci. (Phys.–Math.), Professor, National Research University Higher School of Economics (Moscow, Russian Federation)
Boris A. Novikov, Dr. Sci. (Phys.–Math.), Professor, St. Petersburg University (St. Petersburg, Russian Federation)
Alexandre F. Petrenko, PhD, Computer Research Institute of Montreal (Montreal, Canada)
Assaf Schuster, Ph.D., Professor, Technion - Israel Institute of Technology (Haifa, Israel)
Andrei Tchernykh, Dr. Sci., Professor, CICESE Research Centre (Ensenada, Baja California, Mexico).
Irina B. Virbitskaite, Dr. Sci. (Phys.–Math.), The A.P. Ershov Institute of Informatics Systems, Siberian Branch of the RAS (Novosibirsk, Russian Federation)
Andrey Voronkov, Dr. Sci. (Phys.–Math.), Professor, University of Manchester (Manchester, United Kingdom)

Адрес: 109004, г. Москва, ул. А. Солженицына, дом 25.
Телефон: +7(495) 912-44-25
E-mail: info-isp@ispras.ru
Сайт: http://www.ispras.ru/proceedings/

Address: 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
Tel: +7(495) 912-44-25
E-mail: info-isp@ispras.ru
Web: http://www.ispras.ru/en/proceedings

© Институт Системного Программирования им. В.П. Иванникова РАН, 2024

Содержание

Table of Contents

DOI: 10.15514/ISPRAS-2024-36(5)-1

CC BY

Конструирование программных систем, нацеленное на обеспечение безопасности

[1,2,3] *В.В. Кулямин, ORCID: 0000-0003-3439-9534 <kuliamin@ispras.ru>*
[1,2,3] *А.К. Петренко, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>*
[4] *Е.А. Рудина, ORCID: 0000-0003-2944-162X <Ekaterina.Rudina@kaspersky.com>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*
[2] *Московский государственный университет имени М.В. Ломоносова, Россия, 119991, г. Москва, Ленинские горы, д. 1.*
[3] *Национальный исследовательский университет «Высшая школа экономики», Россия, 101000, г. Москва, ул. Мясницкая, д. 20.*
[4] *АО «Лаборатория Касперского», Россия, 125212, г. Москва, Ленинградское шоссе, д. 39А, стр. 2.*

Аннотация. Конструктивная информационная безопасность — один из подходов, нацеленных на обеспечение надежности и безопасности программных систем, занимающий среди них достаточно важное место. Он развивается уже более 50 лет, однако широкие массы разработчиков не знакомы с его принципами и методами. Важной задачей в рамках популяризации этого подхода является создание типологии задач и техник обеспечения конструктивной информационной безопасности и определение приоритетных направлений их развития.

Ключевые слова: информационная безопасность; методы предотвращения атак; конструирование безопасных систем; архитектурные образцы и стили; усиление защищенности кода; моделирование архитектуры; анализ архитектуры; безопасные языки программирования.

Для цитирования: Кулямин В.В., Петренко А.К., Рудина Е.А. Конструирование программных систем, нацеленное на обеспечение безопасности. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 7–16. DOI: 10.15514/ISPRAS–2024–36(5)–1.

Благодарности: В статье использовались материалы, которые были собраны сотрудниками Лаборатории Касперского и ИСП РАН. В первую очередь мы выражаем благодарность А. Бухвалову как инициатору данной работы, а также Е.Баскову, Д. Буздалову, С. Ерошкину, Н. Горелиц, А. Максимову, М. Кричанову, Е. Корныхину, В. Падаряну, А. Угненко, А. Хорошилову и В.Чепцову.

Software Security by Design

[1,2,3] *V.V. Kuliamin, ORCID: 0000-0003-3439-9534 <kuliamin@ispras.ru>*
[1,2,3] *A.K. Petrenko, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>*
[4] *E.A. Rudina, ORCID: 0000-0003-2944-162X <Ekaterina.Rudina@kaspersky.com>*

[1] *V.P. Ivannikov Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
[2] *Lomonosov Moscow State University,*
GSP-1, Leninskie Gory, Moscow, 119991, Russia.
[3] *National Research University «Higher School of Economy»,*
20, Myasnitskaya st., Moscow, 101000, Russia.
[4] *AO Kaspersky Lab,*
39A, str. 2, Leningradskoye sh., Moscow, 125212, Russia.

Abstract. Security-by-Design is an important approach to ensure software security and reliability. It has been developing already for more than 50 years, but its principles and techniques are still not well known among wide society of software developers. To make the approach more familiar and popular we need to reestablish its goals and problems, to classify and explain its techniques, and formulate trends of its future development. This paper reformulates the main principles of Security-by-Design, provides some examples of security design patterns and anti-patterns, and also explores relations between the approach and software architecture analysis methods, hardening techniques, and safe programming languages.

Keywords: software security; attack prevention methods; secure software design; software architecture styles and patterns; hardening; software architecture modeling; software architecture analysis; safe programming languages.

For citation: Kuliamin V.V., Petrenko A.K., Rudina E.A. Software Security by Design. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024, pp. 7-16 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-1.

Acknowledgements. The paper used materials provided by many employees of Kaspersky Lab and Institute for System Programming. We thank A. Buhvalov for the initiative to prepare this paper, we also thank A.Buhvalov, D. Buzdalov, S. Eroshkin, H. Gorelits, A. Maksimov, M. Kritchanov, E. Kornykhin, V. Padaryan, A. Ugnenko, A. Khoroshilov and V. Cheptsov.

1. Введение

В последние десятилетия значимость и сложность проблем обеспечения безопасности программно-аппаратных систем значительно повысилась в связи с действием следующих факторов.

- Рост количества и сложности решаемых такими системами задач из-за увеличивающихся требований со стороны бизнеса и органов государственного управления, а также из-за высокой скорости изменений самих требований.
- Создание с использованием разнородных технологий все более сложных систем из гетерогенных компонентов с целью охватить как можно больше пользователей, относящихся к различным социальным слоям и распределенных географически.
- Обострение конкурентной борьбы между бизнес-группами и между государствами и совершенствование технологий атак, препятствующих функционированию программных систем и нарушающих их информационную безопасность.

Проблема обеспечения информационной безопасности является комплексной и многоаспектной, у нее нет простых и линейных решений. Систематические и тщательно продуманные подходы к выработке возможных решений предполагают использование разнообразных методов предотвращения, обнаружения и устранения проблем безопасности на всех стадиях создания, сопровождения и эксплуатации программно-аппаратных систем, причем ни один элемент этой триады не может быть проигнорирован без значительного

повышения рисков. В частности, без систематической работы над предотвращением проблем безопасности трудоемкость поддержки систем становится чрезмерно дорогой из-за постоянной потребности в закрытии все новых и новых обнаруживаемых уязвимостей.

Одним из наиболее значимых подходов к предотвращению проблем информационной безопасности является конструирование систем, с самого начала нацеленное на ее обеспечение [1,2]. В России этот подход получил название Конструктивная информационная безопасность (КИБ). Конечно же, предотвратить все возможные проблемы безопасности никогда не удается, но можно существенно уменьшить разнообразие потенциальных атак и риски их успешности, тем самым значительно снижая стоимость поддержания системы в высоко защищенном состоянии и повышая уровень доверия к ней. Для этого необходимо с самого начала проектирования не оставлять проблемы защиты системы на более поздние этапы, а одновременно решать задачи реализации основных функций системы и защиты этой системы как от внешних, так и от внутренних факторов, ведущих к нештатному функционированию или отказам.

Подход к конструированию, нацеленному на обеспечение безопасности, пока остается достаточно аморфным, разные авторы и группы выделяют в нем немного отличающиеся списки элементов. Но, в целом, все согласны, что в него входит систематическое использование нескольких принципов и правил, а также ряда учитывающих эти правила образцов (паттернов) проектирования и низкоуровневых паттернов организации данных и взаимодействия.

Конструктивная информационная безопасность не противопоставляется другим подходам и технологиям, нацеленным на повышение надежности и защищенности программ. Более того, данный подход предполагает, что жизненный цикл программ обеспечен, например, такими технологиями как SSDL (Secure Software Development Lifecycle) или РБПО (Разработка Безопасного Программного Обеспечения). То есть перечисленные подходы и технологии дополняют друг друга, и их следует применять совместно.

Принципы не всегда легко трансформируются в понятные разработчикам технологии, инструкции, техники разработки безопасного ПО, а в рамках конструктивного подхода они явно нужны. Далее мы сначала остановимся на принципах, а потом на тех конструктивных составляющих, которые должны развить и обогатить арсенал средств конструктивной информационной безопасности.

Важным вопросом в рамках популяризации конструктивной информационной безопасности является создание типологии задач и техник, присущих этому подходу, и определение приоритетных направлений их развития. Исследованию данного вопроса посвящен совместный проект Лаборатории Касперского и Института системного программирования. Данная статья является публикацией результатов исследований первой фазы этого проекта.

2. Принципы построения безопасных программных систем

Отдельные идеи, связанные с конструированием систем, нацеленным на обеспечение безопасности, (Secure by Design, Secure by Construction) [1] возникают достаточно давно, первые из них развиваются в рамках методов создания систем, корректных по построению (Correctness by Construction) [3-5]. Такие методы развиваются уже более 50-ти лет, но на практике применяются лишь к системам небольшого размера или в сочетании с другими более масштабируемыми технологиями. Тем не менее уже первые работы в этом направлении указали на важность успешного предотвращения многих проблем на этапе проектирования системы.

Можно утверждать, что основе методов и технологий конструирования программных систем лежат некоторые фундаментальные принципы, в том числе принципы безопасности. Поскольку само понятие безопасности зависит от назначения и условий применения системы, то принципы могут иметь разные приоритеты и применяться с различной степенью

строгости. Важно, что принципы применяются практически на всех этапах жизненного цикла систем, при этом они реализуются разными методами и средствами. Приведем несколько примеров принципов конструктивной информационной безопасности [2,6]:

- Принцип изоляции — нужно стараться свести данные, доступные одновременно нескольким компонентам или передаваемые между ними, к минимуму, необходимому для обеспечения решения ими своих задач.
- Принцип минимизации поверхности атаки — нужно стараться свести к минимуму количество компонентов, напрямую взаимодействующих с внешними системами или пользователями и получающих извне (возможно, через посредников) необработанные или непроверенные данные.
- Принцип минимизации привилегий — нужно стараться выдавать каждому компоненту или процессу минимальные привилегии, необходимые ему для решения его задач, и избегать неоправданного расширения привилегий.
- Принцип эшелонированной защиты — нужно стараться обеспечить определенный уровень защиты при каждом переходе управления и/или данных между слоями/уровнями функциональности системы, не доверяя полностью защите каждого отдельного компонента или сервиса и защищаясь от возможного внедрения злоумышленника в любой из них.
  При этом понятно, что каждый слой защиты приводит к дополнительному снижению как производительности, так и удобства использования и сопровождения компонентов, поэтому организовывать защиту стоит не между каждыми двумя взаимодействующими компонентами, а между естественно возникающими в структуре системы их слоями.
- Принцип обеспечения безопасности по умолчанию — установленные по умолчанию настройки, значения данных и алгоритмы работы должны обеспечивать безопасное функционирование системы, даже если пользователи или администраторы никак не будут вмешиваться в ее работу.
- Принцип обеспечения безопасности сбоев — нужно стараться предусмотреть возможные сбои и ошибки в различных компонентах системы (включая возможные успешные атаки на них) и сделать возможным безопасное продолжение, или, при невозможности такового, безопасное завершение, работы системы при их возникновении, включая реакцию на такие события и аккуратное протоколирование происходящего для последующего анализа.
- Принцип прозрачности решений по защите — нужно стараться аккуратно определять возможные атаки и в полном и однозначном виде описывать принимаемые проектные решения для противодействия им, избегая неопределенных возможных угроз (от которых невозможно защититься достаточно надежно) и невнятных решений по защите (которая в этом случае не будет обладать значимым уровнем доверия).

3. Паттерны и антипаттерны безопасной архитектуры

Принципы не определяют проектные решения, но задают набор правил, которым стоит следовать при их выработке, а также позволяют оценить принимаемые решения как следующие им в достаточной мере, или, наоборот, имеющие потенциально небезопасные последствия.

Образцы проектирования, используемые при конструировании, нацеленном на обеспечение безопасности, задают более конкретные элементы решений по защите системы, хотя их доработка, конфигурирование и расширение остаются возможными [7-9].

Примером может служить образец монитор защиты (reference monitor) [10]. Монитор защиты определяет, как можно реализовать унифицированное применение единой политики безопасности во всех компонентах системы, при этом внесение изменений в политику может выполняться в одном месте и сразу становится актуальным для всей системы. Другим примером архитектурного образца, нацеленного на повышение безопасности, является виртуализация [11]. Она позволяет аккуратно реализовать принцип изоляции сразу для большого числа компонентов.

Важным подходом к обеспечению защиты процессов от нежелательного взаимного влияния является концепция ядра разделения [12]. Роль ядра безопасности заключается в воссоздании в рамках одной общей машины такого окружения, которое поддерживает различные компоненты системы и обеспечивает каналы связи между ними таким образом, что отдельные компоненты системы не могут отличить эту общую среду от физически распределенной.

Ядро с гарантиями изоляции [13] является своего рода ядром разделения для интегрированной модульной авионики и даёт гарантии изоляции в основном согласно схеме разделения ARINC 653 [14].

Ядра разделения в операционных системах предоставляют функции для обеспечения изоляции процессов и управления информационными потоками. Они должны быть достаточно компактными, чтобы обеспечить возможность формальной верификации корректности их работы. Функциональность может незначительно отличаться в зависимости от области применения системы, созданной с использованием такого ядра.

Ядро разделения не обязательно совпадает с ядром операционной системы.

Некоторые из низкоуровневых функций, реализованных ядром операционной системы, также могут поддерживать изоляцию процессов (разделение доменов), но, как правило, оно также предоставляет множество функций, отсутствующих в ядре разделения. С другой стороны, одной из задач ядра разделения может быть обеспечение выполнения политик управления информационными потоками. В настоящее время ядро разделения – это не “ядро” в общепринятом смысле, а комбинация аппаратных технологий и программных компонентов, которые гарантированно обеспечивают изоляцию процессов [15].

Ядра разделения обеспечивают пространственную и временную изоляцию процессов, но позволяют им взаимодействовать контролируемым образом. Степень изоляции, обеспечиваемая каждым из этих типов разделения, зависит от назначения системы и конкретного архитектурного решения.

4. Усиление защиты кода (hardening)

Методы, нацеленные на повышение безопасности за счет раннего обнаружения возможных ошибок в архитектуре, коде ПО, а также препятствующие техникам выполнения известных атак, принято объединять под термином hardening [16]. Одним из примеров такой техники является окаймление выделяемых участков памяти специальным образом заполненными массивами, что позволяет быстро выявить возможные выходы за границы буферов. В действительности hardening объединяет большое количество разнообразных приемов, нацеленных на предупреждение вероятных угроз. Набор таких угроз и методов их парирования постоянно обновляется, поэтому для практического использования нужны удобные технические средства для проведения защитных мероприятий в ходе разработки и эксплуатации систем на регулярной основе.

5. Инструментальная поддержка архитектурных моделей и каталоги уязвимостей

С ростом размера и сложности программной или программно-аппаратной системы возрастает потребность в инструментах работы с архитектурными моделями. Наиболее известным инструментом архитектурного моделирования, по-видимому, можно назвать SysML, который можно рассматривать как надъязык языка UML. Другим известным языком в этом классе является AADL (Architecture Analysis & Design Language) [17]. AADL в отличие от SysML имеет стандартизованное текстовое представление и строго описанную семантику, что делает его более удобным для разработчиков инструментов анализа архитектурных моделей.

Инструменты моделирования программных систем распространены не очень широко, еще меньше круг пользователей средств архитектурного моделирования. В связи с этим набор отработанных сценариев использования архитектурного моделирования и, в частности, моделирования, направленного на повышение информационной безопасности, невелик. Примерами видов архитектурного анализа и способов использования архитектурных моделей интересных с позиций конструктивной безопасности являются: поиск уязвимостей архитектурных решений, анализ отказоустойчивости и самовосстановления, автоматизированный анализ поверхности атаки, моделирование поведения системы в агрессивном окружении, моделирование управления конфигурациями, тестирование конфигураций и др.

Надо отметить, что большая часть инструментов архитектурного моделирования поддерживает лишь небольшое число сценариев использования, и нет ни одного, который поддерживал хотя бы перечисленный набор. Одной из причин, почему пока еще нет удобных и эффективных инструментов для конструирования архитектур безопасных программных систем, является отсутствие добротного каталога шаблонов безопасных архитектурных решений и уязвимостей архитектуры. Лучшим на сегодняшний день собранием архитектурных уязвимостей является каталог CAWE (Common Architectural Weakness Enumeration) [18]. В этом каталоге потенциальные уязвимости снабжены указанием архитектурной тактики и методов возможного устранения уязвимостей. Авторы разработали интерактивное веб-приложение для архитекторов и разработчиков для доступа к каталогу. Каталог CAWE содержит 224 уязвимости, связанные с безопасностью. Пока этот каталог далек от завершенности и логической стройности и, к сожалению, как минимум с 2020 года каталог CAWE не обновляется.

В качестве ориентира для развития CAWE можно использовать общую копилку уязвимостей в программном и аппаратном обеспечении CWE (Common Weakness Enumeration) [19] и список актуальных уязвимостей, который ведется в базе CVE [20].

В рамках Microsoft Security Development Lifecycle [21] составлен свой каталог потенциальных уязвимостей. Более того, в Microsoft разработали среду моделирования Microsoft Threat Modeling Tool [22] для описания дизайна и автоматической проверки его на угрозы из этого каталога. Следует обратить внимание работы, посвященные классификации потенциальных уязвимостей и техникам их формализованного описания для дальнейшего использования при анализе проектов [23-26].

6. Безопасные языки программирования

Еще одним инструментом конструктивного подхода к обеспечению безопасности программных систем являются так называемые языки безопасного программирования. Частным примером таких языков являются memory safe языки. В действительности в последнее время появилось много языков, обеспечивающих защиту от разнообразных угроз информационной безопасности. Такими угрозами могут быть гонки по данным (data race), некорректная работа со сложными, часто динамическими, типами данных. Сейчас вышли на

уровень промышленной зрелости некоторые функциональные языки, например, Haskell. Имеется совсем новое направление функционального программирования языки с так называемыми зависимыми данными (dependent types). Интересным представителем этого направления является Idris [27].

При промышленном применении, помимо прямого эффекта от использования безопасных языков, который состоит в предотвращении уязвимостей определенного рода, будет и дополнительный эффект, обусловленный отсутствием необходимости в соответствующих возможностях в инструментах статического и динамического анализа, а это в свою очередь, позволит выделять больше ресурсов на поиск дефектов и уязвимостей других видов.

7. Заключение

В данной статье мы концентрируемся на тех проблемах, которые в большей степени связаны с просчетами в архитектурных решениях, нежели с дефектами и уязвимостями, которые по многим другим причинам возникают в программном коде. Чего же не хватает для того, чтобы как позитивный, так и негативный опыт проектирования накапливался и эффективно использовался?

Во-первых, нужно отдавать себе отчет, что исследования в области конструктивной информационной безопасности и внедрение результатов этих исследований в практику требуют тщательно продуманных действий одновременно в разных направлениях. Нужны будут совместные усилия по исследованиям и разработкам, по созданию и распространению учебных курсов и учебно-методических материалов, созданию новых и корректировке имеющихся нормативно-правовых документов, проведению пилотных проектов внедрения технологий и процессов КИБ. Во-вторых, видно, что в настоящее время наименее исследованы методы и технологии поддержки ранних фаз жизненного цикла разработки ПО, в частности, методы анализа свойств и уязвимостей архитектурных решений еще на фазе проектирования, а также методы выявления связей проектных решений с проблемами информационной безопасности, которые возникают в процессе разработки и эксплуатации ПО.

Отставание в этой сфере (оно наблюдается как в нашей стране, так и за рубежом) объясняется не только отсутствием удобных и эффективных средств архитектурного моделирования, но и тем, что в практике на фазах разработки и особенно сопровождения ПО про архитектуру часто забывают – она была важна на фазе замысла, а когда код разработан, она, как бы, уже не нужна. Но это не так. В действительности, после того как ошибка проявилась, найдено место в программе, куда вносится исправление, «работа над ошибками» еще не доведена до конца. Полезно взглянуть на архитектуру системы еще раз и проанализировать возможные недоработки в структуре системы и интерфейсах ее компонентов. То есть, в ходе сопровождения и развития программной системы к анализу и, возможно, коррекции проекта, и, соответственно, к накоплению знаний о шаблонах «правильных» и «неправильных» архитектурных решений нужно возвращаться регулярно.

Нередко ошибки в программах обусловлены тем, что интерфейсы подталкивают разработчиков к небезопасному программированию. Например, если в интерфейсах не предусмотрена возможность проверки работоспособности того или иного компонента-сервера, многие компоненты-клиенты будут продолжать обращаться к нему в расчете, что соответствующие операции выполняются. В некоторых ситуациях это приводит к катастрофическим результатам, и основная ответственность здесь не на программистах-разработчиках, а на проектировщиках.

Заметим, что действенным инструментом повышения защищенности программ является организация баз данных уязвимостей. Такие коллекции служат нескольким целями. Во-первых, это форма накопления опыта для обучения и использования многочисленным сообществом программистов; во-вторых, это систематизированная база для создания check-

list'ов, используемых, например, при инспекциях кода; в-третьих, это основа для разработки различных инструментов анализа кода.

В широкой практике программной инженерии, к сожалению, нет аналогичных репозиториев для архитектурных ошибок, а они, очевидно, появятся, если работу над анализом и исправлением найденных ошибок в коде доводить до конца, то есть выявлять те архитектурные уязвимости, которые привели к ошибкам в программном коде или, по крайней мере, повысили риски, связанные с определенными программными решениями. В настоящее время инструментов, поддерживающих такой сценарий работы нам не известно. Самые близкие к этой теме работы опубликованы в [28-29]. С другой стороны, когда появятся такие репозитории, возникнет заинтересованность в программных инструментах для работы с архитектурными моделями и поиска архитектурных уязвимостей на всех фазах жизненного цикла. Это одно из важных направлений развития технологий конструктивной информационной безопасности.

Конструирование систем, нацеленное на обеспечение безопасности — один из множества подходов к обеспечению безопасности, занимающий в этом наборе достаточно важное место. Он развивается уже более 50 лет, однако широкие массы разработчиков не очень хорошо знакомы с его элементами, разве что отдельные техники и образцы используются достаточно активно. Такое положение дел не только не способствует эффективному развитию области, связанной с обеспечением информационной безопасности, но и приводит к тому, что многократно приходится возвращаться к преодолению проблем, которые, казалось бы, уже достаточно хорошо изучены. Исследование способов предотвращения проблем информационной безопасности и разработка методов, инструментов и технологий конструктивной информационной безопасности, это еще один путь для повышения защищенность программных и программно-аппаратных систем.

Список литературы / References

[1]. Cavoukin, A. and Dixon, M. Privacy and security by design: An enterprise architecture approach. Information and Privacy Commissioner, Canada, Tech. Rep., 9, 2013.

[2]. Johnsson, D.B., Deogun, D., and Sawano, D. Secure By Design. Manning Publications, 2019. ISBN: 9781617294358.

[3]. Dijkstra, E.W.A Discipline of Programming. Prentice Hall, Upper Saddle River, 1976. ISBN: 9780132158718/

[4]. Gries, D. The Science of Programming. Springer, Heidelberg, 1987. DOI: 10.1007/978-1-4612-5983-1.

[5]. Kourie, D.G., Watson, B.W. The Correctness-by-Construction Approach to Programming. Springer, Heidelberg, 2012. DOI: 10.1007/978-3-642-27919-5.

[6]. OWASP Application Security Verification Standard, v. 4.0.3. 2021.

[7]. Dougherty, C., Sayre, K., Seacord, R.C., Svoboda, D., and Togashi, K. Secure Design Patterns. Technical Report CMU/SEI-2009-TR-010, Software Engineering Institute, 2009. DOI: 10.1184/R1/6583640.v1.

[8]. Fernandez-Buglioni, E. Security Patterns in Practice: Designing Secure Architectures Using Software Patterns, 2013. ISBN: 9781119998945.

[9]. Washizaki, H., Xia, T., Kamata, N., Fukazawa, Y., Kanuka, H., and Yamaoto, D. Taxonomy and Literature Survey of Security Pattern Research. Proc. of 2018 IEEE Conference on Application, Information and Network Security (AINS), pp. 87-92, 2018. DOI: 10.1109/AINS.2018.8631465.

[10]. Jaeger, T. Operating System Security. Synthesis Lectures on Information Security, Privacy, and Trust. Springer, 2008. DOI: 10.1007/978-3-031-02333-0.

[11]. Pearce, M., Zeadally, S., and Hunt, R. Virtualization: Issues, security threats, and solutions. ACM Computing Surveys, 45(2), Article 17, 2013. DOI: 10.1145/2431211.2431216.

[12]. Rushby, J. Design and Verification of Secure Systems. 8-th ACM Symposium on Operating System Principles, pp. 12-21, Asilomar, CA, December 1981. DOI: 10.1145/800216.80658.

[13]. Rushby, J. Partitioning in Avionics Architectures: Requirements, Mechanisms, and Assurance. NASA Contractor Report CR-1999-209347, NASA Langley Research Center, June 1999.

[14]. Aeronautical Radio, Inc. (ARINC). Avionics Application Software Standard Interface, Part 0, Overview of ARINC 653, ARINC 653P0-2, August 2019.

[15]. DeLong, R.J., Rudina, E. MILS Architectural Approach Supporting Trustworthiness of the IIoT Solutions. Whitepaper, Industrial Internet Consortium, 2021.
[16]. Patra, P.K., and Pradhan, P.L. Hardening of UNIX Operating System. International Journal of Computer and Communication Technology 1(1), Article 9, 2010. DOI: 10.47893/ijcct.2010.1008.
[17]. Open AADL. URL: http://www.openaadl.org/ (доступ 05.12.2024).
[18]. Santos, J.C.S., Tarrit, K., and Mirakhorli, M. A catalog of security architecture weaknesses. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), pp. 220-223, 2017. DOI: 10.1109/ICSAW.2017.25.
[19]. MITRE. Common Weakness Enumeration, 2022. URL: https://cwe.mitre.org/index.html (доступ 01.11.2024).
[20]. MITRE. Common Vulnerabilities and Exposures, 2024. URL: https://www.cve.org (доступ 01.11.2024).
[21]. Microsoft Security Development Lifecycle.
URL: https://www.microsoft.com/en-us/securityengineering/sdl/ (доступ 01.11.2024).
[22]. Microsoft Threat Modeling Tool.
URL: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool (доступ 01.11.2024)
[23]. Almorsy, M., Grundy, J., and Ibrahim, A.S. Automated software architecture security risk analysis using formalized signatures. Proc. of 35-th International Conference on Software Engineering (ICSE), pp. 662-671, San Francisco, CA, USA, 2013. DOI: 10.1109/ICSE.2013.6606612.
[24]. Frydman, M., Ruiz, G., Heymann, E., César, E., and Miller, B.P. Automating Risk Analysis of Software Design Models. The Scientific World Journal, June 2014, pp. 248-259. DOI: 10.1155/2014/805856.
[25]. Nafees, T., Coull, N., Ferguson, I., and Sampson, A. Vulnerability Anti-Patterns: a Timeless Way to Capture Poor Software Practices (Vulnerabilities). Proc. of 24-th Conference on Pattern Languages of Programs. The Hillside Group, ACM, 23, 2017.
[26]. Seifermann, S., Heinrich, R., and Reussner, R. Data-Driven Software Architecture for Analyzing Confidentiality. Proc. of IEEE International Conference on Software Architecture (ICSA), pp. 1-10. Hamburg, Germany, 2019. DOI: 10.1109/ICSA.2019.00009.
[27]. IDRIS. URL: https://www.idris-lang.org/ (доступ 05.12.2024).
[28]. Siu, K., Moitra, A., Li, M., Durling, M., Herencia-Zapana, H., and Interrante, J. Architectural and Behavioral Analysis for Cyber Security. In 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 2019, pp. 1-10. DOI: 10.1109/DASC43569.2019.9081652.
[29]. VERDICT Project. URL: https://ge-high-assurance.github.io/VERDICT/ (доступ 05.12.2024).

Информация об авторах / Information about authors

Виктор Вячеславович КУЛЯМИН – кандидат физико-математических наук, доцент кафедры Системного программирования ВМК МГУ, ведущий научный сотрудник ИСП РАН. Сфера научных интересов: программная инженерия, тестирование на основе моделей, формальные методы программной инженерии, формальные методы обеспечения безопасности.

Victor Viatcheslavovitch KULIAMIN – Cand. Sci. (Phys.-Math.), Associate Professor of System Programming Department, Faculty of Computational Mathematics and Cybernetics Moscow State University, Leading Researcher of the Institute for System Programming, Russian Academy of Sciences. Research interests: software engineering, model-based testing, formal methods of software engineering, formal methods of software security assurance.

Александр Константинович ПЕТРЕНКО — доктор физико-математических наук, профессор кафедры Системного программирования ВМК МГУ, заведующий отделом Технологий программирования ИСП РАН. Его научные интересы включают формальные методы программной инженерии, языки спецификаций и моделирования, их применение для поддержки разработки и верификации программного обеспечения.

Alexander Konstantinovich PETRENKO — Dr. Sci. (Phys.-Math.), Professor of System Programming Department, Faculty of Computational Mathematics and Cybernetics Moscow State University, Head of Software Engineering department of the Institute for System Programming, Russian Academy of Sciences. His research interests include formal methods of software

engineering, specification and modeling languages, and their use in software development and verification.

Екатерина Александровна РУДИНА – кандидат технических наук в области компьютерной и сетевой безопасности, аналитик. Работает в Департаменте перспективных технологий «Лаборатории Касперского» в области исследования угроз и оценки рисков для систем критической инфраструктуры. Принимает участие в разработке национальных стандартов и стандартов ISO, IEEE, рекомендаций ITU и Консорциума промышленного Интернета вещей. Научные интересы включают методы моделирования угроз информационной безопасности, подходы к проектированию и реализации систем с заданными свойствами безопасности и устойчивости, методы обнаружения компьютерных атак.

Ekaterina Alexandrovna RUDINA – Cand. Sci. (Tech.) in Computer and Network Security, the analyst who works for the Kaspersky in the scope of threat research and risk assessment for the systems of critical information infrastructure. Ekaterina is a contributor to ISO, IEEE, ITU, and Industrial Internet Consortium documents and to national standards. Her research interests cover the methods of threat modeling, approaches to secure and safe systems engineering, and intrusion detection methods.

DOI: 10.15514/ISPRAS-2024-36(5)-2

CC BY

Идентификация реквизитов сборки через отслеживание системных вызовов

[1,3] *А.М. Гранат, ORCID: 0009-0007-6589-3347 <a.granat@ispras.ru>*
[1] *П.Д. Дунаев, ORCID: 0000-0002-9142-0945 <p.dunaev@ispras.ru>*
[1,2] *А.А. Синкевич, ORCID: 0009-0002-3364-6468 <artsin666@gmail.com>*
[2] *И.А. Батраева, ORCID: 0000-0002-6539-8473 <batraevaia@info.sgu.ru>*
[2,4] *Д.Ю. Петров, ORCID: 0000-0001-6364-2084 <iac_sstu@mail.ru>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*
[2] *Саратовский государственный университет имени Н.Г. Чернышевского, Россия, 410012, Саратов, ул. Астраханская, 83.*
[3] *Высшая школа экономики, Россия, 101000, г. Москва, ул. Мясницкая, д. 20*
[4] *Институт проблем точной механики и управления РАН, Россия, 410028, Саратов, ул. Рабочая 24*

Аннотация. В статье рассматриваются методы идентификации реквизитов сборки, описываются их сильные и слабые стороны. Представлен инструмент, обеспечивающий журналирование процесса сборки с помощью отслеживания системных вызовов. Приводится оценка временных затрат на сборку с использованием разработанного инструмента.

Ключевые слова: база данных компиляции; перехват сборки; системный вызов ptrace.

Для цитирования: Гранат А.М., Дунаев П.Д., Синкевич А.А., Батраева И.А., Петров Д.Ю. Идентификация реквизитов сборки через отслеживание системных вызовов. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 17–30. DOI: 10.15514/ISPRAS–2024–36(5)–2.

Identifying Build Requisites via System Call Tracing

[1,3] *A.M. Granat, ORCID: 0009-0007-6589-3347 <a.granat@ispras.ru>*
[1] *P.D. Dunaev, ORCID: 0000-0002-9142-0945 <p.dunaev@ispras.ru>*
[1,2] *A.A. Sinkevich, ORCID: 0009-0002-3364-6468 <artsin666@gmail.com>*
[2] *I.A. Batraeva, ORCID: 0000-0002-6539-8473 <batraevaia@info.sgu.ru>*
[2,4] *D.Yu. Petrov, ORCID: 0000-0001-6364-2084 <iac_sstu@mail.ru>*
[1] *Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
[2] *Saratov State University,*
83, Astrakhanskaya Street, Saratov, 410012, Russia.
[3] *Higher School of Economics,*
20, Myasnitskaya Street, Moscow, 101000, Russia.
[4] *Institute for Problems of Precision Mechanics and Control,*
24, Rabochaya Street, 410028.

Abstract. In this work, we discuss methods for identifying build requisites, and describe their strengths and weaknesses. The buildography tool is presented that provides logging of the build process by tracking system calls. An estimate of the time spent on build process using the buildography tool is given.

Keywords: compilation database; build interception; ptrace.

For citation: Granat A.M., Dunaev P.D., Sinkevich A.A., Batraeva I.A., Petrov D.Yu. Identifying Build Requisites via System Call Tracing. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024, pp. 17-30 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-2.

1. Введение

В условиях стремительного развития технологий и растущей сложности современных систем, безопасная разработка программного обеспечения приобретает ключевое значение. Организации все чаще сталкиваются с необходимостью обеспечения безопасности на всех этапах жизненного цикла разработки программного обеспечения, чтобы защитить свои системы и данные от потенциальных угроз.

1 апреля 2024 года был введён в действие государственный стандарт «ГОСТ Р 71206-2024 Защита информации. Разработка безопасного программного обеспечения. Безопасный компилятор языков С/С++. Общие требования» [1], направленный на установление единых требований к безопасному компилятору языков Си и С++. Одним из пунктов стандарта является функция журналирования процесса трансляции, с сохранением такой информации, как параметры компиляции и хэш-суммы входных и выходных файлов.

Несмотря на то, что стандарт ориентирован на языки Си и С++, функция журналирования полезна и для программ, написанных на других компилируемых языках. Это подчеркивает актуальность разработки инструмента для журналирования процесса трансляции, независимого от языка программирования. Такой инструмент может стать важным компонентом системы безопасности программного обеспечения, позволяя обнаруживать и устранять уязвимости на этапе компиляции.

Цель статьи – анализ методов идентификации реквизитов сборки, описание разработанного инструмента для генерации базы данных компиляции `buildography`, оценка времени работы процесса сборки с использованием инструмента и без него. В разделе 2 описывается проблема генерации базы данных компиляции, существующие инструменты и подходы к данной проблеме, в разделах 3 и 4 – описаны механизмы `ptrace` и `seccomp`, выбранные для использования в итоговом инструменте. Далее в разделе 5 описана архитектура и реализация `buildography`, алгоритм работы его компонентов. В разделе 6 подводятся итоги исследования.

2. Описание проблемы

2.1. База данных компиляции

База данных компиляции представляет собой структурированное хранилище данных, содержащее информацию о процессах компиляции программного кода. Такая база данных может включать детали о каждом шаге компиляции, включая используемые файлы, их версии, а также параметры и результаты работы компилятора.

База данных компиляции может быть полезна для проведения статического анализа кода. Статические анализаторы, такие как Clang Static Analyzer [2], Coverity [3] или Svace [4], используют информацию о процессе компиляции из соответствующего файла (например, Clang Static Analyzer по умолчанию читает информацию из файла `compile_commands.json` в рабочем каталоге проекта) для выполнения детального анализа исходного кода на предмет потенциальных ошибок и уязвимостей. С помощью базы данных компиляции анализаторы могут точно понять, как код был собран, и, следовательно, провести более точный анализ.

Из других случаев, когда генерация базы данных компиляции может принести пользу, можно выделить портирование программного обеспечения на другие платформы – при портировании программного обеспечения на новые платформы или архитектуры важно знать параметры и конфигурацию компиляции. База данных компиляции позволяет разработчикам понимать исходные параметры компиляции и адаптировать их под новые условия.

В соответствии с государственным стандартом «ГОСТ Р 71206-2024 Защита информации. Разработка безопасного программного обеспечения. Безопасный компилятор языков С/С++. Общие требования», для каждой единицы трансляции база данных компиляции должна хранить:

- рабочий каталог компиляции;
- имя и хэш-сумму файла, представляющего единицу трансляции и выходного файла, а также каждого файла, включаемого в процесс компиляции;
- выполняемую команду компиляции целиком или полный список аргументов компиляции [1].

2.2. Обзор существующих инструментов

Для сбора параметров компиляции и генерации файла, содержащего информацию о командах компиляции, применяемой в различных инструментах анализа кода, используется утилита `Bear` [5]. Она перехватывает вызовы компилятора во время сборки проекта и создаёт соответствующий файл. Команда `bear -- <build_command>` позволяет получить необходимую информацию для последующего анализа кода. Помимо Bear, также применяется, например, `compiledb` [6], подходящая для сборочных систем, основанных на Makefile. Особенность `compiledb` заключается в использовании флага `-n` утилиты `make`, который позволяет сгенерировать базу данных компиляции без запуска сборки.

Кроме того, инструменты сборки проектов, такие как `CMake`, `Ninja`, `Qt Build System` и `qmake`, также способны автоматически генерировать файлы конфигурации, содержащие информацию о параметрах компиляции.

`CDE` [7] решает смежную задачу: для транспортировки необходимого пакета вместе с окружением, в котором он был собран, третьему лицу, этот инструмент сохраняет информацию о файлах, открытых в процессе сборки, и включает их в состав пакета. Это позволяет при сборке на другой машине использовать файлы, сохраненные вместе с пакетом на исходной системе.

Еще одним подходом к генерации базы данных компиляции является встраивание механизма непосредственно в компилятор и его активация с помощью определённых флагов

компиляции. Например, в компиляторе Clang доступна функциональность генерации JSON-файла с информацией о каждом этапе компиляции при указании флага `-MJ` [8]. В безопасном компиляторе «SAFEC» [9], разработанном в Институте системного программирования РАН на основе GCC [10], также добавлена функциональность генерации базы данных компиляции с помощью флага компиляции `--dump-sbom <dump-dir>`.

Однако, одним из основных ограничений генерации базы данных с помощью внедрения дополнительной функциональности в компилятор является сложность отслеживания всех внешних зависимостей при использовании сложных сценариев сборки или нестандартных инструментов.

Кроме того, прямая работа с компилятором ограничивает доступность данных только той информацией, которую компилятор может предоставить, что может быть недостаточно для некоторых задач.

Для преодоления ограничений возможностей компилятора для генерации базы данных компиляции, существуют альтернативные подходы, которые могут быть использованы для сбора информации о процессе компиляции.

2.3. Альтернативные подходы к генерации базы данных компиляции

В качестве альтернативных подходов для операционных систем семейства Linux можно выделить:

1) `FUSE` (Filesystem in Userspace) – механизм, позволяющий пользователям создавать собственные файловые системы в пользовательском пространстве, что в свою очередь позволяет реализовывать собственные механизмы отслеживания и журналирования операций над файлами, связанных с процессом компиляции. Несмотря на возможности, `FUSE` может работать медленнее, чем файловая система, реализованная на уровне ядра ОС, поскольку все операции должны проходить через пользовательское пространство, что добавляет накладные расходы при переключении контекста [11]. Также механизм не обеспечивает необходимый инструмент информацией о командах компиляции, что приводит к зависимости от инструмента для их отслеживания.
2) `inotify` – это механизм ядра Linux, который позволяет отслеживать изменения в файловой системе. Он может быть использован для мониторинга действий, связанных с файлами и каталогами, с которыми происходит взаимодействие в процессе компиляции, и запуска определённых действий при обнаружении изменения или доступа к файлам, за которыми ведется наблюдение [12]. Плюсом данного подхода является отслеживание всех действий, связанных с работой с файлами, однако `inotify` имеет тот же недостаток, что и `FUSE`, связанный с отсутствием информации о командах компиляции. Другим минусом данного механизма является ограничение на количество отслеживаемых файлов, хранящееся в значении переменной `max_user_watches` конфигурации `inotify`, что приводит к невозможности работы с масштабными проектами;
3) `ptrace` – инструмент, позволяющий отслеживать и манипулировать процессами в операционной системе. Его механизм может быть использован для наблюдения за работой компилятора и сбора информации обо всех его действиях – открытие файлов и работа с ними, вызовы команд компиляции, создание новых процессов, окончание их работы. Из плюсов данного подхода можно выделить возможность отслеживать каждый системный вызов, полученный в процессе компиляции, что даёт полное представление о взаимодействии сборочной системы с ОС, а это, в свою очередь, решает проблему работы с внешними зависимостями и сложными системами сборки. Из минусов можно выделить необходимость учитывать зависимости от

процессорной архитектуры при разработке инструмента, например, имена регистров, в которых хранятся данные о системном вызове [13].

Наиболее релевантным подходом для разработки инструмента, решающего задачу сбора информации о процессе компиляции является `ptrace`, так как он обеспечивает полный контроль над исполнением процесса компиляции и собирает подробную информацию о его действиях.

3. Системный вызов ptrace

Основная сфера применения `ptrace` – инструменты для отладки с использованием точек останова (например, `gdb`) и трассировщики системных вызовов (`strace` и подобные инструменты).

Для языка Си интерфейс для работы с `ptrace` предоставлен в заголовочном файле `sys/ptrace.h` – саму функцию для системного вызова и перечисление возможных запросов `__ptrace_request`, необходимых для взаимодействия с трассируемым процессом, получением необходимой информации и управления его выполнением [14].

Начать отладку процесса можно двумя способами:

1) Присоединиться к процессу с помощью запросов `PTRACE_ATTACH` или `PTRACE_SEIZE`;
2) Сделать родителя текущего процесса трассировщиком с помощью запроса `PTRACE_TRACEME`.

Общий механизм работы `ptrace` имеет следующий вид:

1) Процесс, предназначенный для отслеживания, получает сигнал `SIGSTOP` и трассировщик подключается к нему;
2) Трассируемый процесс останавливается и трассировщик может получить информацию (запросы `PTRACE_PEEK*`, `PTRACE_GET*`) о его текущем состоянии или как-либо его изменить (`PTRACE_POKE*`, `PTRACE_SET*`) [15];
3) Трассировщик перезапускает отслеживаемый процесс одним из способов перезапуска (`PTRACE_CONT`, `PTRACE_SYSCALL`, `PTRACE_SINGLESTEP`).

Проверить, что процесс перешел в состояние остановки, можно с помощью макроса `WIFSTOPPED` (истинно, если процесс находится в состоянии остановки), передав ему статус процесса, полученный из системного вызова `wait`.

Существует несколько видов состояния остановки:

1) Остановка на системном вызове (syscall stop): вызывается непосредственно при системном вызове и сразу после его завершения в случае, если процесс перезапускается с помощью команды `PTRACE_SYSCALL`.
2) Остановка на событии (event stop): вызывается при событии, специфичном для `ptrace`, например:
   - `PTRACE_EVENT_FORK`, `PTRACE_EVENT_VFORK` или `PTRACE_EVENT_CLONE` – происходит при вызове `fork()`, `vfork()` или `clone()` отслеживаемым процессом;
   - `PTRACE_EVENT_EXEC` – срабатывает при выполнении `execve()`;
   - `PTRACE_EVENT_EXIT` – остановка перед завершением работы процесса.

   В основном, чтобы трассируемый процесс останавливался на подобных событиях, необходимо явно прописать для ядра ОС флаги, которые позволяют отслеживать их, например, чтобы происходила остановка на событии `PTRACE_EVENT_EXEC`, при

подключении к трассируемому процессу необходимо выставить флаг `PTRACE_O_TRACEEXEC`.

3) Остановка при доставке сигнала (signal-delivery-stop): вызывается при сигнале, полученном трассируемым процессом. Трассировщик получает информацию о полученном сигнале первым, и может модифицировать его обработку, либо перезапустить процесс. Исключение в этом случае – сигнал `SIGKILL`, так как в этом случае трассируемый процесс немедленно завершается.
4) Групповая остановка (group-stop): возникает, когда все потоки в группе процессов находятся в состоянии остановки из-за получения сигнала, который останавливает процесс – `SIGSTOP`, `SIGTSTP`, `SIGTTIN` или `SIGTTOU`. В случае, если трассировщик был подключен с помощью запроса `PTRACE_SEIZE`, такой вид остановки сопровождается событием `PTRACE_EVENT_STOP`. Когда такой сигнал отправляется группе процессов, все потоки в группе приостанавливают свою работу и переходят в состояние остановки. После того как процесс был остановлен, трассировщик может возобновить его выполнение с помощью запроса `PTRACE_CONT` или других аналогичных команд. Однако в случае групповой остановки процесс перестает реагировать на сигналы до тех пор, пока не получит `SIGCONT`. Если групповой процесс был переведен в состояние ожидания запросом `PTRACE_LISTEN`, он также не будет реагировать на команды `ptrace` до получения `SIGCONT` [16].

4. Механизмы seccomp и BPF

`seccomp` (от англ. Secure Computing Mode) – механизм ядра Linux, предназначенный для ограничения системных вызовов, которые может выполнять процесс. Технология повышает безопасность системы за счёт фильтрации доступных системных вызовов, что позволяет предотвратить множество видов атак, основанных на эксплуатации уязвимостей программного обеспечения.

В Linux `seccomp` реализуется ввиде двух режимов:

1) Строгий режим, который ограничивает процесс только четырьмя системными вызовами: `read()`, `write()`, `_exit()`, и `sigreturn()`. Любые другие системные вызовы в данном режиме приводят к вызову сигнала `SIGKILL`;
2) Фильтрационный режим, разрешающий настройку ограничений с помощью BPF. В этом режиме имеется возможность задать произвольный набор правил в коде программы, определяющих, какие системные вызовы разрешены, а какие блокируются.

BPF (Berkeley Packet Filters) – технология, разработанная в 1992 году для эффективной фильтрации сетевого трафика на уровне ядра операционной системы. В Linux 3.5 BPF был адаптирован для фильтрации системных вызовов через seccomp, что позволило использовать его в различных сценариях безопасности и мониторинга.

Виртуальная машина BPF предоставляет простой набор инструкций, который позволяет выполнять базовые операции, такие как загрузка, хранение, переходы между инструкциями на определенное количество шагов и арифметические операции. Чтобы предотвратить возможное “зависание” ядра операционной системы, в программах BPF запрещены циклы и наложены другие ограничения, включая лимит на 4096 инструкций для одной программы.

Фильтры BPF компилируются во время выполнения и загружаются в ядро для применения к вызовам системных функций. Это дает разработчикам возможность точно настроить поведение приложений в зависимости от их потребностей в системных ресурсах, что значительно повышает уровень безопасности.

В режиме `seccomp` BPF используется для создания детализированных политик безопасности, которые контролируют, какие системные вызовы могут выполняться. Когда процесс переходит в режим `seccomp` и выполняет системный вызов, ядро применяет соответствующий фильтр BPF к этому вызову.

Чтобы использовать режим `seccomp`, должно быть выполнено одно из условий:

1) Пользователь, инициирующий действие, должен иметь привилегии администратора (`CAP_SYS_ADMIN`);
2) Процесс должен сделать системный вызов `prctl` с параметрами, выставляющими бит `no_new_privs`, ограничивая процесс и все его дочерние процессы от получения новых привилегий.

Пример ситуации, когда получение процессом новых привилегий может привести к неожиданным последствиям при использовании `seccomp` – использование команды `sudo`. `seccomp` позволяет разработчику прописывать правила для системных вызовов, обрабатываемых механизмом, и одной из возможностей правил является пропуск системного вызова. В данном случае при вызове `sudo -u <user> <cmd>` ядро может повысить уровень привилегий процесса, а в момент смены пользователя выполнение системного вызова `setuid`, отвечающего за это действие, будет пропущено и при этом вернётся значение 0, таким образом пользователь получит привилегии суперпользователя вместо того чтобы выполнять действие от лица пользователя `<user>` с соответствующими привилегиями.

Для удобства работы с BPF в библиотеке `linux/filter.h` языка Си представлена структура `sock_filter`, хранящая в себе код операции, количество инструкций, на которое должен быть совершен переход в случае, если условие верно или неверно, и операнд, а также макросы `BPF_STMT` и `BPF_JUMP`, делающие более читаемым настройку фильтра BPF.

Для демонстрации связи механизмов `seccomp`, BPF и `ptrace` подойдет следующая настройка фильтра:

```
BPF_STMT(BPF_LD | BPF_W | BPF_ABS, (offsetof(struct seccomp_data, nr))),
BPF_JUMP(BPF_JMP | BPF_JEQ | BPF_K, SYS_openat, 1, 0),
BPF_STMT(BPF_RET | BPF_K, SECCOMP_RET_ALLOW),
BPF_STMT(BPF_RET | BPF_K, SECCOMP_RET_TRACE),
```

Программа с установленным фильтром при полученном системном вызове проверяет номер системного вызова (поле `nr` в структуре `seccomp_data`), далее он сравнивается с переданным значением. В случае, если значение совпадает, необходимо пропустить 1 инструкцию, если нет, продолжить выполнение.

В случае, если получен системный вызов `openat` (открытие файла), фильтр возвращает значение `SECCOMP_RET_TRACE`, и ядро выполняет попытку оповестить трассировщик, что получен соответствующий системный вызов перед его выполнением.

Важно, что для подключения фильтра к трассировщику необходимо передать на вход `ptrace` флаг `PTRACE_O_SECCOMP`, иначе трассировщик не сможет получить уведомления о полученных системных вызовах.

5. Инструмент buildography

5.1. Архитектура и используемые инструменты

Инструмент `buildography` состоит из двух частей: трассировщика и постпроцессора. Из-за специфики реализации, а именно из-за работы с системными вызовами, `ptrace` API, механизмами `seccomp` и `BPF`, инструмент поддерживает генерацию базы данных компиляции только для операционных систем семейства Linux.

Для реализации трассировщика был выбран язык Си из-за его высокой производительности и более глубокого управления работой с памятью, в то время как постпроцессор реализован на языке Python в силу простоты реализации скрипта и удобной работы с форматом JSON.

Зависимостями проекта являются следующие библиотеки:

1) `rhash` – библиотека для хэширования функцией, определенной государственным стандартом ГОСТ Р 34.11-2012;
2) `uthash` – библиотека для работы с хэш-таблицами.

Задача трассировщика заключается в генерации промежуточного JSON-файла, состоящего из следующих полей:

- `directory` – рабочий каталог сборки;
- `command` – полный список аргументов командной строки команды сборки;
- `component_commands` – массив записей, содержащих данные о каждой единице трансляции, состоит из:
  - `command` – список всех аргументов команды;
  - `dependencies` – список, состоящий из имён и хэш-сумм каждого включаемого файла;
  - `output` – список, состоящий из имён и хэш-сумм выходных файлов.

Постпроцессор получает на вход промежуточный файл, обрабатывает его и генерирует следующие поля:

- `input` – список, состоящий из имён и хэш-сумм каждой единицы трансляции;
- `utilities_info` – список, состоящий из инструментов, задействованных в трансляции исходных текстов в исполняемый код, а именно пути к их исполняемым файлам и их хэш-суммы.

5.2. Трассировщик

Для получения необходимой информации для базы данных компиляции трассировщик отслеживает системные вызовы, перечисленные в табл. 1.

Табл. 1. Отслеживаемые системные вызовы.
Table 1. Monitored system calls.

Системный вызов	Необходимая информация
`open/openat/openat2`	Путь к файлу и режим, в котором он был открыт
`execve`	Список аргументов команды
`fork/vfork/clone`	Идентификатор нового процесса

Для отслеживания системных вызовов семейства `open` используется `seccomp`-фильтр со следующими настройками:

```
BPF_STMT(BPF_LD | BPF_W | BPF_ABS, (offsetof(struct seccomp_data, nr))),
#ifdef SYS_openat2
        BPF_JUMP(BPF_JMP | BPF_JEQ | BPF_K, SYS_openat2, 3, 0),
#endif
    /* There is no open syscall on ARM64 architecture */
#ifdef SYS_open
        BPF_JUMP(BPF_JMP | BPF_JEQ | BPF_K, SYS_open, 2, 0),
#else
        BPF_JUMP(BPF_JMP | BPF_JA, 1, 0, 0);
#endif
        BPF_JUMP(BPF_JMP | BPF_JEQ | BPF_K, SYS_openat, 1, 0),
        BPF_STMT(BPF_RET | BPF_K, SECCOMP_RET_ALLOW),
        BPF_STMT(BPF_RET | BPF_K, SECCOMP_RET_TRACE),
```

Для остановки процесса на остальных необходимых вызовах используются специальные флаги `ptrace`:

- `PTRACE_O_TRACEEXEC`;
- `PTRACE_O_TRACEFORK`;
- `PTRACE_O_TRACEVFORK`;
- `PTRACE_O_TRACECLONE`.

Для работы трассировщика были реализованы следующие структуры данных:

- Хэш-таблица для хранения информации о файлах: время последнего изменения файла (позволяет отследить изменение файла, открытого на чтение, вне процесса сборки), его хэш-сумма и канонический путь. Ключом в данной хэш-таблице является комбинация номера устройства и индексного дескриптора (inode), позволяющие уникально идентифицировать файл.
- Хэш-таблица, хранящая в себе информацию о процессе: его идентификатор и указатель на данные об образе процесса.
- Данные об образе процесса, состоящие из команды, запустившей образ, списка файлов, открытых на чтение и на запись, а также счётчик ссылок, позволяющий следить за количеством процессов, имеющих одинаковый образ.
- Массивы для хранения хэш-сумм и имён файлов.

Механизм работы трассировщика заключается в следующем:

1) При остановке на системном вызове `execve`, для процесса, из которого был получен вызов, создаётся новая запись об образе процесса и заполняется соответствующими данными, для предыдущего образа процесса счётчик ссылок уменьшается на 1.
2) При получении информации о вызове `fork` или `clone` в хэш-таблицу процессов добавляется новая запись, хранящая в себе указатель на образ родительского процесса, счётчик ссылок для соответствующего образа увеличивается на 1;
3) При остановке на `PTRACE_EVENT_STOP`, соответствующему в данном случае запуску нового процесса, порожденный процесс останавливается, если соответствующий `PTRACE_EVENT_FORK` еще не был получен (и будет разблокирован, когда идентификатор его родителя станет известным).
4) В момент открытия файла трассировщику передаётся информация о пути файла и режиме открытия файла, и в зависимости от режима информация о файле добавляется в соответствующий список файлов для текущего образа процесса, подсчитывается его хэш-сумма;
5) В момент окончания работы процесса из хэш-таблицы процессов удаляется запись и декрементируется счётчик ссылок соответствующего образа;
6) Если счётчик ссылок образа процесса после его уменьшения равен нулю, то добавляется новая запись в базу данных компиляции в формате JSON.

Важно понимать, что трассировщик не имеет возможности отслеживать процессы, запускаемые вне сборки, и защита от вредоносных процессов не является целью его работы, поэтому в случае, например, подмены файлов в процессе сборки посторонним процессом трассировщик не сможет это отследить.

В рамках исследования были проведены замеры времени сборки библиотек `musl` и `qtbase` с помощью компиляторов `gcc` и `clang` в трех различных условиях: с использованием трассировщика с хэшированием, с использованием трассировщика без функции хэширования, а также без применения трассировщика. Библиотека `musl` была выбрана как

проект с большим количеством небольших по размеру единиц трансляции, а `qtbase` – как представитель класса крупных библиотек, написанных на языке C++. Эксперимент позволил оценить влияние процесса трассировки и хэширования на общую производительность сборки. Результаты представлены в табл. 2.

Для снижения процента замедления от использования трассировщика было принято решение оптимизировать процесс, реализовав многопоточный режим.

В многопоточном режиме вместо того, чтобы считать хэш-сумму файла сразу и хранить ее, в хэш-таблицу файлов передаётся порядковый номер хэша, подсчёт хэш-суммы запускается в новом побочном потоке, и работа трассировщика продолжается. После того, как хэш-сумма была подсчитана, она передаётся в общий для всех потоков массив хэш-сумм по индексу, равному порядковому номера хэша.

Табл. 2. Результаты измерения производительности однопоточного трассировщика.
Table 2. Performance evaluation results of a single-threaded tracer.

Библиотека, компилятор и количество потоков	**Без трассировщика, сек**	**Трассировка без хэширования, сек**	**Трассировка с хэшированием, Сек**
`musl`, `gcc`, 1 поток	20.03	24.7	27.23
`musl`, `clang`, 1 поток	33.14	39.38	41.66
`musl`, `gcc`, 8 потоков	4.01	7.35	9.21
`musl`, `clang`, 8 потоков	5.69	7.97	10.31
`qtbase`, `gcc`, 8 потоков	252.61	285.73	294.73

Так как текущая реализация трассировщика позволяет записывать данные в базу данных только в последовательном режиме, а в новом многопоточном режиме хэш-суммы могут быть неизвестны в момент записи информации в базу данных компиляции, было принято решение вместо хэш-сумм записывать в базу данные-заполнители в формате, соответствующем хэш-сумме, для последующего удобства подстановки в места, определенные заполнителями. Для сохранения мест, которые будут использованы для подстановки, был определен массив смещений в буфере, в котором для каждого файла хранится позиция заполнителя, соответствующего ему.

В момент, когда в базу данных компиляции выводится информация об открытых файлах для образа процесса, в общий массив смещений добавляется текущая позиция в буфере, хранящем в себе базу данных, и порядковый номер в формате, совпадающем с размером хэш-суммы. Полученный механизм позволяет после окончания трассировки подставить все хэш-суммы в необходимые места за один полный проход по массиву смещений.

Для оценки эффективности многопоточного режима были проведены повторные замеры времени сборки тех же библиотек в модифицированных условиях. Результаты представлены в таблице ниже.

Табл. 3. Результаты измерения производительности для многопоточного трассировщика
Table 3. Performance evaluation results of a multi-threaded tracer

Библиотека, компилятор и количество потоков	**Без трассировщика, сек**	**Трассировка с параллельным хэшированием, сек**
`musl`, `gcc`, 1 поток	20.03	25.28
`musl`, `clang`, 1 поток	33.14	39.59
`musl`, `gcc`, 8 потоков	4.01	8.45
`musl`, `clang`, 8 потоков	5.69	8.55
`qtbase`, `gcc`, 8 потоков	252.61	288.87

В результате внедрения многопоточного хэширования в процесс трассировки было достигнуто ускорение. Различия во времени выполнения с трассировщиком между однопоточной и многопоточной сборками демонстрируют, что многопоточность эффективно сокращает время компиляции, но преимущества этого снижаются из-за однопоточной обработки системных вызовов механизмом `ptrace`. На однопоточной сборке выбранных проектов разработанный инструмент демонстрирует замедление от 19% до 26%, в то время как на многопоточной сборке происходит замедление от 14% до 110% в худшем случае. Более глубокий анализ замедления трассировщика и оптимизаций выходит за рамки данной работы.

Одной из ключевых проблем, выявленных при реализации трассировщика, является преждевременный подсчёт хэш-сумм содержимого выходных файлов. Проблема заключается в том, что при остановке процесса на системном вызове семейства `open`, когда файл открывается на запись, его содержимое ещё не успело быть записано. В результате, если хэширование начнётся в этот момент, оно приведёт к некорректным результатам, поскольку содержимое файла ещё не сформировано.

Для решения этой проблемы был внедрён новый механизм вычисления хэш-сумм содержимого выходного файла. При получении информации об открытии выходного файла, его данные заносятся в таблицу файлов с флагом `hash_calculated`, установленным в значение `false`. Если позже тот же файл появляется как входной, это указывает на завершение всех операций над ним внутри сборки, и, соответственно, можно приступать к вычислению его хэша.

Хэш-суммы, которые не были рассчитаны в процессе, считаются отложенными и будут вычислены в функции `calculate_pending_hashes`. В этой функции обрабатываются две категории файлов: артефакты сборки (исполняемый файл, полученный в процессе сборки и побочные выходные файлы) и временные файлы, которые не были прочитаны и затем удалены (например, `.res` файл при компиляции программы на языках Си/C++ компилятором GCC без флага `-flto`). Поскольку функция `calculate_pending_hashes` вызывается после завершения трассировки, то хэш-суммы артефактов сборки могут быть безопасно рассчитаны, в то время как непрочитанные временные файлы могут быть проигнорированы.

5.3. Постпроцессор

Трассировщик в его текущей реализации не позволяет генерировать множество входных файлов во время работы, а также не предоставляет гибкости как для опционального множества входных файлов, так и для получения информации об используемых утилитах. В связи с этим было принято решение использовать дополнительный скрипт, который получает на вход промежуточный файл и обрабатывает его, генерируя оставшиеся поля.

Постпроцессор и трассировщик связаны между собой следующим образом: по завершению работы трассировщика создаётся канал (`popen()`), запускается скрипт постпроцессора, и полученный в процессе трассировки JSON-файл передаётся в созданный канал. Постпроцессор, в свою очередь, получает этот JSON-файл через стандартный ввод (`stdin`) и обрабатывает его.

Первым шагом обработки JSON-файла, полученного из трассировщика, является удаление выходных файлов с неподсчитанной хэш-суммой: на данном этапе удаляются записи о временных файлах, которые не были прочитаны в процессе сборки.

Далее происходит генерация множества входных файлов: по умолчанию множеством входных файлов считаются все файлы, которые были открыты на чтение и не были открыты на запись в процессе трассировки.

Последним шагом в работе постпроцессора является сбор информации об использованных инструментах: по умолчанию в базу данных компиляции вносится информация о пути к исполняемому файлу инструмента и хэш-сумма его содержимого, однако в программном коде скрипта представлены функции, в данный момент не содержащие в себе механизм извлечения какой-либо информации, но позволяющие в дальнейшем поместить в базу данных еще версию инструмента и информацию об его конфигурационных файлах.

6. Заключение

В рамках данной работы были исследованы методы идентификации реквизитов сборки и существующие инструменты для генерации базы данных компиляции. Был реализован инструмент для идентификации реквизитов сборки с помощью отслеживания системных вызовов. Инструмент состоит из двух компонентов: трассировщика системных вызовов и постпроцессора, обрабатывающего промежуточную базу данных компиляции, полученную из трассировщика. Чтобы снизить процент замедления процесса сборки при использовании трассировщика, был реализован механизм многопоточного подсчета хэш-сумм содержимого файлов. Были подобраны проекты для оценки времени работы их сборочного процесса в обычной ситуации и с использованием трассировщика.

Разработанный инструмент предоставляет информацию о каждом этапе компиляции, командах компиляции, зависимостях и выходных файлах, при этом на выбранных для эксперимента проектах замедляет процесс сборки в процентном соотношении от 19% до 26% в случае однопоточной сборки, а при многопоточной сборке – от 14% до 110% в зависимости от объема проекта.

Список литературы / References

[1]. ГОСТ Р 71206-2024 «Защита информации. Разработка безопасного программного обеспечения. Безопасный компилятор языков C/C++. Общие требования». М., Российский институт стандартизации, 2024, 20 с.
[2]. Clang Static Analyzer, Available at: https://clang-analyzer.llvm.org/, accessed 28.04.2024.
[3]. Baloglu B. How to find and fix software vulnerabilities with coverity static analysis //2016 IEEE Cybersecurity Development (SecDev). – IEEE, 2016. – С. 153-153.
[4]. А.А. Белеванцев, А.О. Избышев, Д.М. Журихин. Организация контролируемой сборки в статическом анализаторе Svace. Системный администратор, выпуск 6-7 (176-177), 2017, стр. 135-139.
[5]. Build EAR, Available at: https://github.com/rizsotto/Bear, accessed 28.04.2024.
[6]. Compilation Database Generator, Available at: https://github.com/nickdiego/compiledb, accessed 28.04.2024.
[7]. Guo P. CDE: A tool for creating portable experimental software packages //Computing in Science & Engineering. – 2012. – Т. 14. – №. 4. – С. 32-35.
[8]. JSON Compilation Database Format Specification, Available at: https://clang.llvm.org/docs/JSONCompilationDatabase.html, accessed 28.04.2024.
[9]. Баев Р.В., Скворцов Л.В., Кудряшов Е.А., Бучацкий Р.А., Жуйков Р.А. Предотвращение уязвимостей, возникающих в результате оптимизации кода с неопределенным поведением. Труды ИСП РАН, том 33, вып. 4, 2021 г., стр. 195-210. DOI: 10.15514/ISPRAS–2021–33(4)–14./ Baev R.V., Skvortsov L.V., Kudryashov E.A., Buchatskiy R.A., Zhuykov R.A. Prevention of vulnerabilities arising from optimization of code with Undefined Behavior. Trudy ISP RAN/Proc. ISP RAS, vol. 33, issue 4, 2021. pp. 195-210 (in Russian). DOI: 10.15514/ISPRAS–2021–33(4)–14.
[10]. GCC, Available at: https://gcc.gnu.org/, accessed 28.04.2024.
[11]. Vangoor B. K. R., Tarasov V., Zadok E. To FUSE or not to FUSE: Performance of User-Space file systems //15th USENIX Conference on File and Storage Technologies (FAST 17). – 2017. – С. 59-72.
[12]. Love R. Kernel korner: Intro to inotify //Linux Journal. – 2005. – Т. 2005. – №. 139. – С. 8.
[13]. Keniston J. et al. Ptrace, utrace, uprobes: Lightweight, dynamic tracing of user apps //Proceedings of the 2007 Linux symposium. – 2007. – Т. 1. – С. 215-224.
[14]. Padala P. Playing with ptrace, Part I //Linux Journal. – 2002. – Т. 103. – №. 5.

[15]. Padala P. Playing with ptrace, part II //Linux J. – 2002. – Т. 104. – С. 4.
[16]. Ptrace Notes, Available at: https://shachaf.net/tmp/ptrace-notes.txt, accessed 28.04.2024.

Информация об авторах / Information about authors

Артемий Максимович ГРАНАТ – Старший лаборант Института системного программирования им. В.П. Иванникова Российской академии наук, студент 1 курса магистратуры Высшей школы экономики. Сфера научных интересов: компиляторы, операционные системы, компьютерные сети.

Artemiy Maksimovich GRANAT – Senior Laboratory technician at Ivannikov Institute for System Programming of the Russian Academy of Sciences, 1st year Master's student at Higher School of Economics. Research interests: compilers, operating systems, computer networks.

Павел Дмитриевич ДУНАЕВ – Аспирант Института системного программирования им. В.П. Иванникова Российской академии наук. Сфера научных интересов: компиляторы, операционные системы, дискретная математика.

Pavel Dmitrievich DUNAEV – postgraduate student at Ivannikov Institute for System Programming of the Russian Academy of Sciences. Research interests: compilers, operating systems, discrete mathematics.

Артем Александрович СИНКЕВИЧ – Старший лаборант Института системного программирования им. В.П. Иванникова Российской академии наук, студент 2 курса магистратуры Саратовского государственного университета. Сфера научных интересов: компиляторы, дискретная математика, нейронные сети.

Artem Aleksandrovich SINKEVICH – Senior Laboratory technician at Ivannikov Institute for System Programming of the Russian Academy of Sciences, 2nd year Master's student at Saratov State University. Research interests: compilers, discrete mathematics, neural networks.

Инна Александровна БАТРАЕВА – кандидат физико-математических наук, доцент, заведующая кафедрой технологий программирования. Сфера научных интересов: дискретная математика, теория автоматов, теория формальных языков и грамматик, информационные системы в теоретической и прикладной лингвистике.

Inna Aleksandrovna BATRAEVA – Cand. Sci. (Phys.-Math.), Associate Professor, Head of the Department of Programming Technologies. Research interests: discrete mathematics, automata theory, theory of formal languages and grammars, information systems in theoretical and applied linguistics.

Дмитрий Юрьевич ПЕТРОВ – кандидат технических наук, старший научный сотрудник лаборатории системных проблем управления и автоматизации в машиностроении; доцент кафедры системного анализа и автоматического управления Саратовского национального исследовательского государственного университета имени Н.Г. Чернышевского. Сфера научных интересов: системный анализ, системотехника на основе моделей, функциональное и имитационное моделирование, авионика и автоматизированные системы управления.

Dmitriy Yurievich PETROV – Cand. Sci. (Tech.) in Engineering, Senior Researcher of the Laboratory of System Problems of Control and Automation in Mechanical Engineering; Associate Professor of the Department of System Analysis and Automatic, Saratov State University. Research interests: system analysis, model-based system engineering, functional and simulation modeling, avionics and automated control systems.

DOI: 10.15514/ISPRAS-2024-36(5)-3

CC BY

Открытое промежуточное представление специализированных потоковых вычислителей, основанное на MLIR

[1,2] *А.С. Камкин, ORCID: 0000-0001-6374-8575 <kamkin@ispras.ru>*
[2] *М.Ю. Литвинов, ORCID: 0009-0009-8926-139X <litvinov@ispras.ru>*
[2] *И.А. Григоров, ORCID: 0009-0001-6373-738X <grigorovia@ispras.ru>*

[1] *Российский экономический университет имени Г.В. Плеханова, 117997, Россия, г. Москва, Стремянный пер., д. 36.*
[2] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25.*

Аннотация. В последнее время для решения вычислительных задач с жесткими ограничениями на производительность (пропускную способность) и энергопотребление (потребляемую мощность) широко используются гетерогенные компьютерные системы. Как правило, такие системы состоят из микропроцессоров общего назначения и аппаратных ускорителей на базе ПЛИС, реализующих наиболее затратные операции (обычно отражающих специфику предметной области). Данная статья посвящена автоматизации проектирования аппаратных ускорителей, ориентированных на задачи потоковой обработки данных (streaming data computing). Особенностями ускорителей этого типа (и решаемых ими задач) являются: (1) непрерывные (на каждом такте работы) прием и выдача данных; (2) ограниченная (по времени и памяти) зависимость выходных данных от входных. Потоковая обработка охватывает широкий класс приложений, включая цифровую обработку сигналов, шифрование трафика, численное моделирование, биоинформатику и другие. В работе предлагается концепция языка DFCIR (DataFlow Computer Intermediate Representation), предназначенного для промежуточного представления моделей потоковых вычислителей. Язык DFCIR основан на открытой компиляторной инфраструктуре MLIR (Multi-Level Intermediate Representation). Для построения RTL-моделей ускорителей по DFCIR-описаниям используются средства CIRCT (Circuit IR Compilers and Tools) – подпроекта MLIR, объединяющего инструменты для работы с моделями аппаратуры.

Ключевые слова: гетерогенная компьютерная система; аппаратный ускоритель; специализированный потоковый вычислитель; программируемая логическая интегральная схема; автоматизация проектирования аппаратуры; язык промежуточного представления; инфраструктура MLIR; инструментарий CIRCT.

Для цитирования: Камкин А.С., Литвинов М.Ю., Григоров И.А. Открытое промежуточное представление специализированных потоковых вычислителей, основанное на MLIR. Труды ИСП РАН, 2024, том 36 вып. 5, с. 31-46. DOI: 10.15514/ISPRAS-2024-36(5)-3.

Благодарности: Исследование выполнено в РЭУ им. Г.В. Плеханова за счет гранта Российского научного фонда № 23-21-00313, https://rscf.ru/project/23-21-00313/. От лица ИСП РАН (участника университетской программы Maxeler MAX-UP) авторы благодарят компанию Maxeler Technologies за предоставление доступа к инструменту MaxCompiler.

Open-Source MLIR-Based Intermediate Representation for Application-Specific Streaming Computers

[1,2] *A.S. Kamkin, ORCID: 0000-0001-6374-8575 <kamkin@ispras.ru>*
[2] *M.Yu. Litvinov, ORCID: 0009-0009-8926-139X <litvinov@ispras.ru>*
[2] *I.A. Grigorov, ORCID: 0009-0001-6373-738X <grigorovia@ispras.ru>*
[1] *Plekhanov Russian University of Economics,*
36 Stremyanny lane, Moscow, 117997, Russia.
[2] *Ivannikov Institute for System Programming of the Russian Academy of Sciences,*
25 Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Abstract. Recently, heterogeneous computer systems have been widely used to solve computational tasks with strict constraints on performance (throughput) and power consumption. Typically, such systems consist of general-purpose microprocessors and FPGA-based hardware accelerators implementing the most expensive operations (which are usually application-specific ones). This article is devoted to the design automation of hardware accelerators for streaming data computing. The features of this type of accelerators (and the problems they solve) are as follows: (1) continuous (cycle-by-cycle) reception and production of data; (2) bounded (in time and memory) output-input dependence. Streaming data computing covers a wide range of applications, including digital signal processing, traffic encryption, numerical modeling, bioinformatics, etc. The paper introduces the concept of DFCIR (DataFlow Computer Intermediate Representation), a language for an intermediate representation of streaming data computing designs. The DFCIR language is based on the open compiler infrastructure MLIR (Multi-Level Intermediate Representation). RTL models of accelerators are built from DFCIR descriptions with the use of CIRCT (Circuit IR Compilers and Tools), a subproject of MLIR that combines tools for working with hardware designs.

Keywords: heterogeneous computer system; hardware accelerator; application-specific streaming computer; field-programmable gate array; electronic design automation; intermediate representation language; MLIR; CIRCT.

For citation: Kamkin A.S., Litvinov M.Yu., Grigorov I.A. Open-Source MLIR-Based Intermediate Representation for Application-Specific Streaming Computers. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024. pp. 15-19 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-3.

Acknowledgements: This research has been completed in Plekhanov Russian University of Economics and funded by Russian Science Foundation (grant № 23-21-00313), https://rscf.ru/project/23-21-00313/. On behalf of ISP RAS (a member of the Maxeler University Program MAX-UP), the authors thank Maxeler Technologies for providing access to MaxCompiler.

1. Введение

Для решения вычислительных задач при наличии жестких ограничений на производительность (пропускную способность) и энергопотребление (потребляемую мощность) широко используются гетерогенные компьютерные системы (ГКС), то есть системы, состоящие из разнородных вычислительных элементов [1, 2]. Типичная ГКС содержит микропроцессор общего назначения и несколько аппаратных ускорителей, обычно на базе ПЛИС, реализующих наиболее затратные операции (в зависимости от предметной области это могут быть матричные умножения, криптографические примитивы и т.п.). За более высокую эффективность ГКС приходится расплачиваться усложнением процессов разработки и верификации: создание ускорителей требует привлечения языков описания аппаратуры (hardware description languages, HDL) [3-5]; для интеграции вычислительных элементов нужна специализированная операционная среда (runtime); отладка предполагает использование средств ко-эмуляции программных и аппаратных компонентов. Таким образом, актуальной задачей является разработка технологий автоматизации проектирования как ГКС в целом, так и отдельных аппаратных ускорителей, применяемых в ГКС.

В работе рассматриваются аппаратные ускорители определенного вида, а именно специализированные вычислители, ориентированные на задачи потоковой обработки данных (application-specific streaming computers); далее в статье вычислители этого вида называются потоковыми вычислителями (ПВ). Потоковая обработка (streaming computing) имеет много приложений: цифровая обработка сигналов, шифрование трафика, численное моделирование, биоинформатика и многие другие. Можно выделить две основные особенности ПВ (и решаемых ими задач): (1) непрерывные (на каждом такте работы) прием и выдача данных; (2) ограниченная (по времени и памяти) зависимость выходных данных от входных. Структурно ПВ представляют собой параллельно-конвейерные устройства с большим числом стадий (иногда более 1000), что роднит их с систолическими массивами [6], однако в отличие от последних ПВ не обязательно имеют однородную структуру. Большая пропускная способность ПВ обеспечивается высокой частотой работы, что достигается специальными методами планирования вычислений (scheduling) [7] и оптимизации временных характеристик (retiming) [8].

Ключевым элементом технологии проектирования аппаратных ускорителей является язык промежуточного представления (intermediate representation, IR): помимо возможностей, которые язык дает для описания функциональности и структуры вычислителя, он так или иначе определяет набор трансформаций, которые можно совершать над описанием (моделью). В работе предлагается язык DFCIR (DataFlow Computer Intermediate Representation), нацеленный на описание специализированных ПВ. В основе данного языка лежит открытая компиляторная инфраструктура MLIR (Multi-Level Intermediate Representation) [9], разрабатываемая командой LLVM [10]. MLIR позволяет создавать специализированные языки (диалекты) и средства трансформации, позволяющие переходить от описания на одном диалекте к описанию на другом, в том числе понижать уровень абстракции описания (lowering). С использованием разработанных средств проектирование ПВ организуется как последовательность преобразований MLIR-представлений: (1) предметно-ориентированное представление (выходит за рамки работы); (2) DFCIR-описание (оптимизированное под производительность или энергопотребление); (3) представление на базе диалектов CIRCT (Circuit IR Compilers and Tools) [11]; (4) HDL-описание (результат обработки CIRCT-представления). Важно отметить, что инфраструктура MLIR дает возможность разрабатывать программные и аппаратные компоненты ГКС в рамках единой среды проектирования.

В статье представлены следующие результаты: (1) требования к языку промежуточного представления специализированных ПВ и средства для их реализации; (2) язык DFCIR промежуточного представления специализированных ПВ, основанный на компиляторной инфраструктуре MLIR; (3) средства трансляции из DFCIR в языки описания аппаратуры Verilog/SystemVerilog, основанные на инструментарии CIRCT.

Статья организована следующим образом. Раздел 2 содержит базовые сведения о проектировании цифровой аппаратуры и специализированных ПВ. В разделе 3 рассматриваются существующие разработки (языки и инструменты), используемые для проектирования ПВ. Раздел 4 посвящен предлагаемому языку DFCIR: описываются типы данных и конструкции языка; приводятся примеры описаний; определяется схема трансляции в FIRRTL (один из диалектов CIRCT). В разделе 5 приводятся результаты экспериментального анализа разработанных средств. Раздел 6 является заключением; в нем подводятся итоги работы и определяются направления дальнейших исследований.

2. Предметная область

Традиционный маршрут проектирования цифровой аппаратуры начинается с создания модели уровня регистровых передач (register-transfer level, RTL). RTL-модель описывает аппаратуру в терминах пересылок данных между регистрами (элементами памяти) и

функциональными элементами (выполняющими арифметико-логические преобразования данных). Для разработки RTL-моделей используют языки описания аппаратуры, наиболее распространенными из которых являются Verilog [3], SystemVerilog [4] и VHDL [5].

Разработка аппаратуры на уровне регистровых передач – это трудозатратный процесс, при этом изменение требований к целевым характеристикам устройства (частоте, площади и/или энергопотреблению) может потребовать существенной переработки RTL-модели (что чревато внесением ошибок). Актуальной задачей является создание средств автоматизированной разработки аппаратуры.

2.1. Подходы к автоматизации проектирования

Автоматизация создания RTL-моделей обеспечивается двумя основными способами [12]:

1. параметризация – вместо одной модели, создается семейство моделей, рассчитанных на разные случаи (режимы) работы; достижение целевых характеристик реализуется подбором значений параметров;
2. повышение уровня абстракции – при описании аппаратуры игнорируются детали реализации, упор делается на поведении; структура устройства конкретизируется в результате планирования вычислений.

В первом случае используются как стандартные HDL-языки (в современных редакциях этих языков есть средства параметризации и макрогенерации), так и специализированные языки, предназначенные для конструирования аппаратуры (hardware construction): Chisel [13], MyHDL [14], Bluespec SystemVerilog [15] и другие. Достоинством подхода является предсказуемый результат: сгенерированные модели по характеристикам сопоставимы с разработанными вручную. Недостаток – недостаточно высокий уровень автоматизации.

Во втором способе, известном как высокоуровневый синтез (high-level synthesis, HLS) [16], часто используются популярные языки программирования: C, C++ и другие. Примерами HLS-инструментов являются Vitis HLS [17], Bambu [18] и LegUp [19]. Следует отметить, что фон-неймановская модель вычислений, лежащая в основе императивных языков программирования, плохо согласуется с «неограниченным» параллелизмом аппаратуры, что затрудняет достижение высоких характеристик синтезируемых схем. Лучшие результаты достигаются при использовании языков, базирующихся на функциональных и потоковых парадигмах [12].

Предлагаемый подход занимает промежуточную нишу между конструированием аппаратуры и высокоуровневым синтезом: с одной стороны, он обеспечивает контролируемый результат синтеза; с другой стороны, позволяет абстрагироваться от некоторых деталей аппаратуры (прежде всего связанных с временны́ми характеристиками: задержкой устройства, периодичностью поступления данных и т.п.).

2.2. Потоковые вычислители

Настоящая работа фокусируется на синхронных статически конфигурируемых потоковых вычислителях, далее для краткости называемых просто потоковыми вычислителями (ПВ). Вычислители этого типа принимают и выдают данные непрерывно, на каждом такте работы; подразумевается, что зависимость выходных данных от входных ограничена по времени и требует для своего выражения небольшого объема памяти. ПВ можно представить в виде графа, узлами которого являются порты ввода/вывода, функциональные элементы (ФЭ) и блоки памяти, а дугами – каналы передачи данных. Все компоненты ПВ функционируют синхронно, при этом ФЭ являются конвейерными устройствами без блокировок (глубины конвейеров разных ФЭ могут различаться, но для каждого ФЭ это число фиксировано и известно на этапе планирования вычислений). При синтезе ПВ для выравнивания времен поступления данных по разным каналам на один ФЭ в схему вставляются вспомогательные

очереди (FIFO); также могут выполняться оптимизирующие преобразования, меняющие структуру вычислителя (изложение этого вопроса выходит за рамки статьи).

На рис. 1 приведен пример ПВ для вычисления скалярного произведения двух трехэлементных векторов. Вычислитель имеет шесть портов ввода и один порт вывода: x_1, x_2, x_3 – координаты первого вектора; y_1, y_2, y_3 – координаты второго вектора; z – результат. Предположим, что выполнение операции умножения занимает 5 тактов, а выполнение операции сложения – 2 такта. Для выравнивания времен поступления данных на сумматор **6** со стороны умножителя **3** вставляется двухэлементная очередь **5**. Для данного ПВ задержка вычислений (latency) составляет 9 тактов (сначала за 5 тактов параллельно выполняются три умножения, затем последовательно выполняются два сложения, каждое из которых занимает 2 такта); периодичность (periodicity) входных и выходных потоков (время между двумя последовательными пересылками операндов и результатов) – 1 такт; соответственно, производительность (throughput) – 1 операция за такт (при частоте 100 МГц – 100 млн операций в секунду).

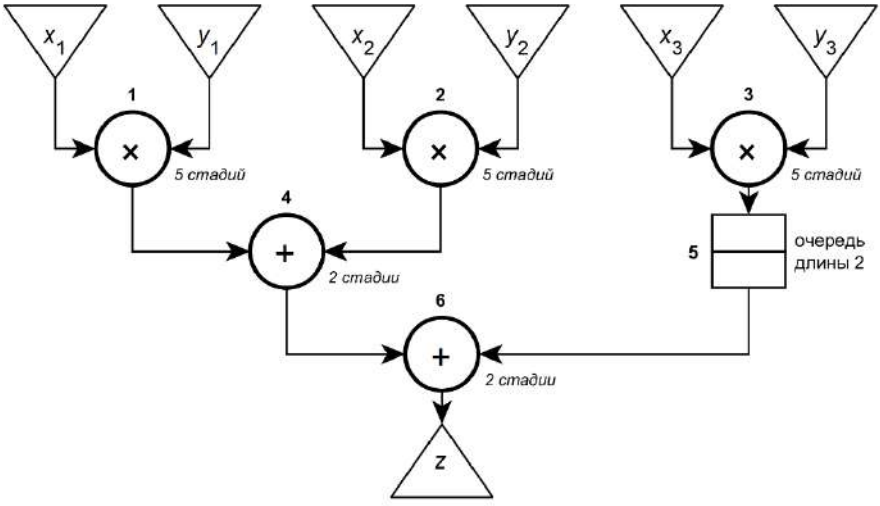


Рис. 1. Потоковый вычислитель, реализующий скалярное произведение векторов (вариант 1).
Fig. 1. Streaming computer implementing the dot product (case 1).

На рис. 2 показан другой ПВ, также вычисляющий скалярное произведение двух трехэлементных векторов. Данный ПВ имеет два порта ввода и один порт вывода: x_i и y_i – координаты первого и второго вектора соответственно (координаты подаются последовательно); z – результат. По сравнению с ПВ из первого примера здесь используется один умножитель вместо трех. Для определения границ векторов в потоках координат используется устройство **3** – счетчик по модулю 3. Компаратор **4** проверяет, что значение счетчика не превосходит 1; результат сравнения (один бит) подается в трехэлементную очередь **5**, выход которой соединяется с селектором мультиплексора **6**. Заметим, что очередь в данном случае реализована в форме сдвигового регистра; его разрядность (L_5) согласована с числом стадий выполнения операций умножения (L_1), сравнения (L_4) и мультиплексирования (L_6): $L_1 = L_4 + L_5 + L_6$. Таким образом, результат умножения первой пары координат векторов будет складываться с нулем, как и результат умножения второй пары. Здесь требуется пояснение: выполнение сложения занимает 2 такта, поэтому, когда умножитель выдает произведение очередной пары координат, сумма произведений предшествующих пар еще не готова; возникает эффект «расслоения» потока – на одних позициях потока вычисляются суммы произведений пар координат с нечетными номерами, на других (с противоположной четностью) – наоборот, с нечетными номерами. Для

«склейки» двух образовавшихся «подпотоков» используется регистр **7** и сумматор **8**. Оставшиеся элементы представленной схемы обеспечивают стробирование выходного потока (они являются опциональными): всякий раз когда значение счетчика становится равным 2, единичный бит помещается в младший разряд сдвигового регистра **10**, старший бит которого используется в качестве строба. Разрядность регистра (L_{10}) определяется условием: $L_9 + L_{10} = L_1 + L_2 + L_8$.

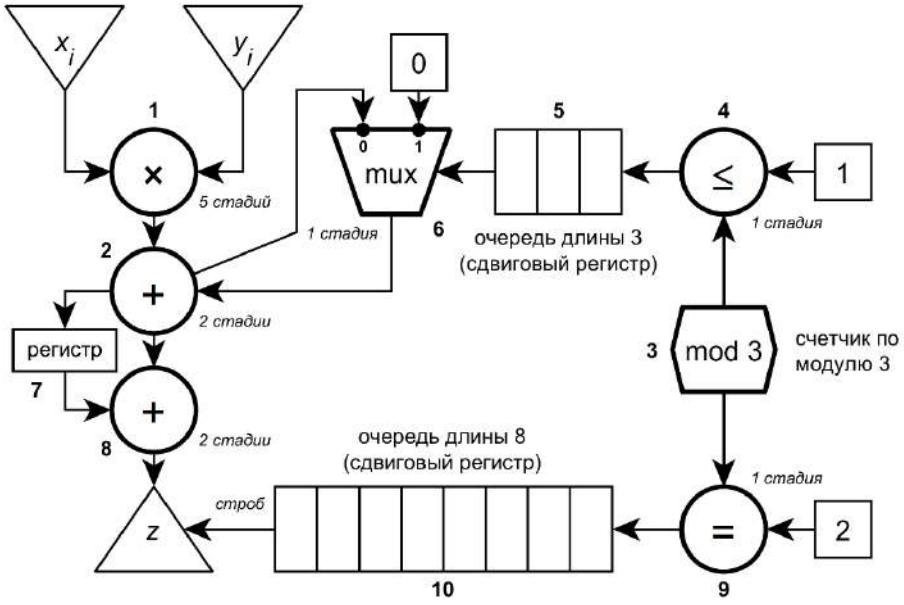


Рис. 2. Потоковый вычислитель, реализующий скалярное произведение векторов (вариант 2).
Fig. 2. Streaming computer implementing the dot product (case 2).

2.3. Требования к языку промежуточного представления

Центральным элементом технологии автоматизированного проектирования аппаратуры (конструирования или высокоуровневого синтеза) является язык промежуточного представления (ЯПП). ЯПП имеет ту же выразительную мощь, что и исходный язык, но при этом лишен избыточности («синтаксического сахара»), характерной для человеко-ориентированных языков. Минималистичность ЯПП упрощает формализацию его семантики и реализацию трансформаций.

К ЯПП ПВ были сформулированы следующие требования:

1. ориентация на потоковую обработку данных (наличие в языке такой сущности, как поток, и операций для работы с потоками);
2. возможность абстрагирования от временны́х характеристик (задержки, периодичности) отдельных ФЭ и ПВ в целом;
3. наличие механизмов конкретизации временны́х характеристик ФЭ и ПВ (фиксирования результатов планирования вычислений);
4. бесшовная интеграция с ЯПП выше- и нижележащих уровней (например, на основе инфраструктуры MLIR).

3. Существующие разработки

В настоящее время имеется широкий спектр языков, применяемых (или потенциально применимых) в проектировании ПВ. Их можно разделить на языки высокоуровневого описания и ЯПП. Краткая характеристика этих языков приведена в табл. 1.

Табл. 1. Краткая характеристика языков, используемых в проектировании потоковых вычислителей.
Table 1. Brief description of the languages used in the design of streaming computers.

Язык	Лицензия	Специализация	Уровень абстракции	Тип языка
MaxJ	Коммерческая	Потоковые вычислители	Поведенческий уровень	Язык высокоуровневого описания
DSLX	Apache License v2.0	Потоковые вычислители	Поведенческий уровень	Язык высокоуровневого описания
Пифагор	*Нет данных*	Потоковые вычислители	Поведенческий уровень	Язык высокоуровневого описания
HIR	Apache License v2.0	Аппаратные ускорители	Поведенческий уровень	Язык промежуточного представления
FIRRTL	Apache License v2.0	Аппаратура общего назначения	Уровень регистровых передач	Язык промежуточного представления
HW/Comb/ Seq	Apache License v2.0	Аппаратура общего назначения	Уровень регистровых передач и вентилей	Язык промежуточного представления
LLHD	Apache License v2.0	Аппаратура общего назначения	Все уровни	Язык промежуточного представления

Будучи ориентированными на проектирование аппаратуры, данные языки имеют схожие системы типов: битовые векторы, кортежи, структуры, массивы; высокоуровневые языки поддерживают числа с фиксированной и плавающей точкой произвольной точности.

3.1 Языки высокоуровневого описания

В данном разделе представлена информация по языкам высокоуровневого описания ПВ: MaxJ (проект MaxCompiler) [20], DSLX (проект XLS) [21] и Пифагор [22]. Ключевой особенностью этих языков является абстрагирование (в той или иной мере) от временны́х характеристик проектируемой аппаратуры: компиляторы (синтезаторы) сами планируют вычисления с учетом заданных проектных ограничений и проводят разные оптимизации, в том числе конвейеризацию (pipelining) и ретайминг (retiming).

Язык MaxJ разработан в рамках проекта MaxCompiler компании Maxeler Technologies (в 2022 году приобретена компанией Groq) и предназначен для построения высокопроизводительных ПВ на базе ПЛИС. В отличие от других средств, рассматриваемых в статье, MaxCompiler позволяет создавать как отдельные аппаратные ускорители, так и ГКС в целом. В основе MaxJ лежит язык программирования Java, расширенный следующими абстракциями: поток (stream), скаляр (scalar), счетчик (counter), смещение в потоке (offset), локальная память (fast memory), глобальная память (large memory). Поток есть последовательность данных одного типа, а скаляр – фиксированное значение (параметр времени исполнения). Над потоками

определены стандартные арифметико-логические операции (сложение, вычитание, умножение, деление и другие): результатом является поток, элементы которого получаются применением заданной операции к соответствующим элементам потоков-операндов. Счетчики – это потоки специального вида (целочисленные прогрессии), используемые для организации циклов; вложенные циклы реализуются с помощью цепей счетчиков (counter chains). Смещения дают возможность обращаться к «прошлым» и «будущим» элементам потока (например, смещение -1 дает доступ к элементу, полученному такт назад). Блоки памяти позволяют сохранять промежуточные результаты вычислений.

Язык DSLX является входным языком открытого инструмента XLS, ориентированного на построение ПВ. Выходом XLS является SystemVerilog-описание, не привязанное к конкретным моделям ПЛИС и платам-ускорителям, что позволяет использовать инструмент в разработке блоков заказных интегральных схем. DSLX похож на язык программирования Rust, но в отличие от последнего учитывает специфику аппаратуры: объекты имеют фиксированную разрядность, графы вызовов функций допускают полную развертку и т.п. Следует отметить такие особенности языка, как параметрические структуры и функции, вывод типов (type inference) и конструкции for (циклы с фиксированным числом итераций).

Язык Пифагор, разрабатываемый в Сибирском федеральном университете, предназначен для создания переносимых параллельных программ и аппаратных реализаций ПВ. Язык базируется на функционально-потоковой парадигме: вычисления представляются информационным графом (отражающим ФЭ и каналы передачи данных); на основе информационного графа строится управляющий граф, отвечающий за синхронизацию работы ФЭ (формирование сигналов готовности результатов). Отличительной чертой компилятора с языка Пифагор является возможность редукции параллелизма, что может быть использовано для построения ускорителей на базе ПЛИС с небольшим числом логических элементов.

3.2 Языки промежуточного представления

В данном разделе даются краткие сведения по ЯПП, применимых к проектированию ПВ: HIR [23], FIRRTL [24], HW/Comb/Seq [25-27] и LLHD [28]. Все перечисленные языки реализованы в форме MLIR-диалектов и, за исключением HIR, развиваются в рамках зонтичного проекта CIRCT.

Язык HIR разработан в Индийском научном институте (Indian Institute of Science) и предназначен для представления аппаратных ускорителей общего назначения. Основными средствами описания вычислений в нем являются циклы, функции и конструкции доступа к памяти. Следует отметить поддержку многомерных тензоров, широко применяемых в задачах машинного обучения. В отличие от языков, рассматриваемых ниже, HIR не фиксирует потактовое поведение устройства, предоставляя свободу для планирования вычислений и оптимизации. Результаты планирования задаются с помощью так называемых временны́х переменных (time variables) и используются для генерации управляющих автоматов.

Язык FIRRTL (Flexible Intermediate Representation for Register Transfer Level) создавался в Калифорнийском университете в Беркли (UC Berkeley) как средство промежуточного представления кода, сгенерированного по Chisel-описанию; сейчас язык развивается в рамках проекта CIRCT. FIRRTL предоставляет конструкции для унифицированного (независимого от HDL) описания цифровой аппаратуры на уровне регистровых передач: схема (circuit), модуль (module), вход (input), выход (output), тактовый сигнал (clock), провод (wire), регистр (reg), блок памяти (mem), условие (when), присваивание (<=), экземпляр модуля (inst) и другие (в том числе разные арифметико-логические операции).

Языки HW, Comb и Seq используются в связке для описания RTL-моделей: HW определяет общую структуру системы, а Comb и Seq – комбинационные и последовательностные схемы,

лежащие в основе модулей. Основными сущностями HW являются модули разных видов (hw.module, hw.module.extern, hw.module.generated) и их экземпляры (hw.instance); Comb определяет арифметико-логические операции (comb.add, comb.and, comb.concat и другие); Seq включает конструкции для работы с регистрами, очередями, блоками памяти (seq.firreg, seq.fifo, seq.firmem и другие).

Язык LLHD (Low-Level Hardware Description) – разработка Швейцарской высшей технической школы Цюриха (ETH Zürich) – задумывался как самодостаточное средство представления цифровой аппаратуры на поведенческом (behavioral), структурном (structural) и вентильном (netlist) уровнях. Несмотря на то, что изначально LLHD возник как отдельный проект, сейчас он развивается в рамках инфраструктуры CIRCT. Основными единицами LLHD являются функции (functions), процессы (processes) и сущности (entities): функции определяют вычисления без побочных эффектов (могут использоваться в разных частях описания); процессы описывают поведение системы (изменение состояния и выходов в зависимости от входов); сущности определяют структуру системы в форме иерархии (в них создаются экземпляры процессов и других сущностей). Среди особенностей языка следует отметить поддержку 9-значной логики, используемой в языке VHDL (IEEE 1164 [29]), и Тьюринг-полноту, что, например, необходимо для представления тестовых окружений.

3.3 Анализ существующих разработок

Результаты анализа существующих языков представлены в табл. 2. Как видно, ни один из них не удовлетворяет полностью сформулированным требованиям: языки MaxJ, DSLX и Пифагор не являются ЯПП (при этом они ориентированы на разработку ПВ и могут служить идейной основой для разработки ЯПП); языки HIR, FIRRTL, HW/Comb/Seq и LLHD не привязаны к потоковой обработке, однако, являясь MLIR-диалектами, они допускают интеграцию с ЯПП других уровней; из них выделяется язык HIR, который основан на более высоком уровне абстракции и имеет механизмы конкретизации временны́х характеристик (фиксирования результатов планирования вычислений).

Табл. 2. Анализ языков, используемых в проектировании потоковых вычислителей.
Table 2. Analysis of the languages used in the design of streaming computers.

Язык / **Требование**	**MaxJ, DSLX, Пифагор**	**HIR**	**FIRRTL, HW/Comb/Seq, LLHD**
1. потоковая обработка данных	+	−	−
2. временная абстракция	+	+	−
3. средства планирования	−	+	−
4. интеграция другими уровнями	−	+	+

ЯПП ПВ разумно строить путем сочетания сильных сторон языков разных типов. На наш взгляд, наиболее перспективная комбинация включает MaxJ (проработанный понятийный аппарат, большой промышленный опыт использования), HIR (механизм временны́х переменных) и диалекты CIRCT (трансформации представления и экспорт HDL-описания).

4. Язык промежуточного представления DFCIR

Предлагаемый ЯПП называется DFCIR (DataFlow Computer Intermediate Representation). Реализация DFCIR основана на инфраструктуре MLIR (DFCIR является MLIR-диалектом). Основные сущности языка заимствованы из MaxJ (см. раздел 3.1): поток, скаляр, счетчик, цепь счетчиков, смещение в потоке и другие.

Применение DFCIR в автоматизации проектирования ПВ описано ниже:

1. высокоуровневое описание ПВ на предметно-ориентированном языке (DSL, Domain-Specific Language) транслируется в DFCIR;
2. на уровне DFCIR осуществляется планирование вычислений и оптимизация под заданные целевые характеристики;
3. оптимизированное DFCIR-представление транслируется в FIRRTL, HW/Comb/Seq или иные ЯПП уровня RTL;
4. RTL-представление экспортируется в HDL (Verilog, SystemVerilog, VHDL).

4.1 Типы данных и операции

Типы данных языка показаны в табл. 3: базовые типы включают числа с фиксированной точкой (`fixed`), числа с плавающей точкой (`float`) и битовые векторы (`rawbits`); составные типы – комплексные числа (`complex`), структуры (`struct`) и векторы (`vector`).

Табл. 3. Типы данных DFCIR.
Table 3. DFCIR data types.

Тип данных	**Синтаксис**	**Описание**

Число с фиксированной точкой	**`fixed`**
	`  $sign,`
	`  $intBits,`
	`  $fracBits>`
		`$sign` – знаковость числа (*true* или *false*);
		`$intBits` – число бит в целой части;
		`$fracBits` – число бит в дробной части.

Число с плавающей точкой	**`float`**
	`  $expBits,`
	`  $fracBits>`
		`$expBits` – число бит в порядке;
		`$fracBits` – число бит в мантиссе.

Битовый вектор	**`rawbits`**`<$width>`	`$width` – длина битового вектора.
Комплексное число	**`complex`**`<$type>`	`$type` – тип вещественной и мнимой частей.
Структура	**`struct`** `  $name`$_1$`:$type`$_1$ `  [, $name`$_i$`:$type`$_i$`]*` >	`$name`$_i$ – имя *i*-ого поля; `$type`$_i$ – тип *i*-ого поля.

Вектор	**`vector`**
	`  $type`
	`  $size>`
		`$type` – тип элементов вектора;
		`$size` – размер вектора.

Над данными определены следующие арифметико-логические операции: получение противоположного значения (`neg`), сложение (`add`), вычитание (`sub`), умножение (`mul`), деление (`div`), остаток от деления (`rem`, `mod`), сравнение на равенство/неравенство (`eq`, `notEq`), сравнение на меньше/больше (`less`, `greater`), сравнение на меньше/больше или равно (`lessEq`, `greaterEq`), сдвиг влево/вправо (`shl`, `shr`), побитовые НЕ, И, ИЛИ, исключающее ИЛИ (`not`, `and`, `or`, `xor`), извлечение битов (`bits`), конкатенация битовых векторов (`cat`), приведение типа (`cast`).

4.2 Основные конструкции

Основные конструкции DFCIR представлены в табл. 4: имена параметров начинаются с символа `$`; запись `$arg*` (`*` в конце является сокращением от `$arg-name : $arg-type`); скобки ***[ ]*** выделяют необязательную часть; ***[ ]**** – повторение нуля или более раз.

Табл. 4. Конструкции DFCIR.
Table 4. DFCIR constructs.

Синтаксис	Описание
Объявления	
kernel $name { $body }	Ядро с именем $name и телом $body.
scalarInput<$type>($name)	Входной скаляр / поток типа $type с именем $name. В случае потока можно задать управляющий (стробирующий) поток $control.
input<$type>($name ***[***, $control****]***)	
scalarOutput<$type>($name) ***[***<= $stream****]***	Выходной скаляр / поток типа $type с именем $name и присоединяемым потоком $stream. В случае потока можно задать управляющий (стробирующий) поток $control.
output<$type>($name ***[***, $control****]*** ***[***<= $stream****]***	
constant<$type> $value	Константа типа $type со значением $value.
Арифметико-логические операции	
UNARY_OP($arg*) : $type	Унарная / бинарная операция над аргументом $arg / аргументами $lhs и $rhs (потоками или скалярами) с указанием типа результата $type.
BIN_OP($lhs*, $rhs*) : $type	
bits($arg*, $lhs, $rhs) : **rawbits**<$width>	Извлечение бит из диапазона [$lhs, $rhs] из скаляра / потока $arg и представление результата в битовом векторе длины $width.
cat($lhs*, $rhs*) : **rawbits**<$width>	Конкатенация битовых представлений $lhs и $rhs и представление результата в битовом векторе длины $width.
cast<$type>($arg*)	Приведение скаляра / потока $arg к типу $type.
Управляющие конструкции	
mux($selector*, $element$_1$ ***[***, $element$_i$****]*****) : $type	Мультиплексор с потоком-селектором $selector и входными потоками $element$_1$, … типа $type. Если $element$_i$ является константой или скаляром, для них должен быть указан их исходный тип.
offset($stream*, $offset*) : $type	Смещение $offset в потоке $stream.
simpleCounter<$type>($max*)	Счетчик со значениями типа $type с верхней границей $max.
createCc	Создание цепи счетчиков.
addCounter($chain, $step, $max*) : $type	Создание и добавление счетчика со значениями типа $type с шагом $step и верхней границей $max в цепь $chain.
Операция соединения	
connect($connecting*, $connectee*)	Соединение скаляра / потока $connecting со скаляром / потоком $connectee.
Операция задержки	
latency [$delay] $stream*	Задержка потока $stream на $delay тактов (вставляется при планировании вычислений).

4.3 Примеры описаний

Описание ядра на языке DFCIR начинается с объявления входов (потоков и скаляров), а заканчивается объявлением выходов. Каждый выход требуется соединить с потоком: это можно сделать прямо в операции **`output`** / **`outputScalar`** или с помощью отдельной операции **`connect`**. Для формирования выходных потоков могут использоваться операции, описанные выше. С потоками данных могут быть ассоциированы управляющие потоки (`control`): значение 1 говорит о готовности данных, 0 – о неготовности.

В листинге 1 приведен пример ядра, вычисляющего значения многочлена $x^2 + 2x$.

```
// Объявление псевдонимов используемых типов.
!ui32t = !dfcir.fixed<false, 32, 0>
!boolt = !dfcir.fixed<false, 1, 0>

// Объявление ядра, вычисляющего значение многочлена второго порядка.
dfcir.kernel "Polynomial2" {
  // Объявление входных потоков.
  %x       = dfcir.input<!ui32t>("x")
  %control = dfcir.input<!boolt>("control")

  // Вычисление значения выражения x*x + x.
  %square  = dfcir.mul(%x: !dfcir.stream<!ui32t>,
                       %x: !dfcir.stream<!ui32t>)
                         : !dfcir.stream<!ui32t>
  %midResult  = dfcir.add(%square: !dfcir.stream<!ui32t>,
                          %x     : !dfcir.stream<!ui32t>)
                                 : !dfcir.stream<!ui32t>

  %result = dfcir.add(%midResult: !dfcir.stream<!ui32t>,
                      %x        : !dfcir.stream<!ui32t>)
                                : !dfcir.stream<!ui32t>

  // Объявление выходного потока.
  %out = dfcir.output<!ui32t>("out", %control: !dfcir.stream<!boolt>)
  dfcir.connect(%out: !dfcir.stream<!ui32t>, %result:
!dfcir.stream<!ui32t>)
}
```

Листинг 1. Ядро, вычисляющее $x^2 + 2x$.
Listing 1. A kernel computing $x^2 + 2x$.

Ядро `Polynomial2` имеет два входных потока, `x` и `control`: по первому передаются 32-битные целые числа, по второму – булевы значения. Для значений потока `x` вычисляются выражения $x^2 + 2x$, которые формируют поток `result`. Выход ядра – поток `out` – соединяется с потоком `result` (данные) и потоком `control` (управление).

В листинге 2 приведен пример описания ядра, вычисляющего скользящее среднее.

Ядро `MovingAverage` имеет входной поток `x`, по которому передаются числа с плавающей точкой, и входной скаляр `size`, задающий число элементов в потоке `x`. Для каждых трех подряд идущих элементов рассчитывается среднее значение (для первого и последнего элементов потока среднее считается по двум значениям – элементы за пределами потока полагаются равными 0). В примере используются конструкция **`offset`** для обращения к элементам до и после текущего. Последний элемент в потоке определяется с помощью счетчика (**`simpleCounter`**). Условия реализованы на мультиплексорах (**`mux`**): например, если текущий элемент является граничным, используется делитель 2, иначе – 3.

```
// Объявление псевдонимов используемых типов.
!ui32t = !dfcir.fixed<false, 32, 0>
!ui32s = !dfcir.stream<!ui32t>
!ui32v = !dfcir.scalar<!ui32t>
!ui32c = !dfcir.const<!ui32t>
!boolt = !dfcir.fixed<false, 1, 0>
!bools = !dfcir.stream<!boolt>
!f32t  = !dfcir.float<8, 23>
!f32s  = !dfcir.stream<!f32t>
!f32c  = !dfcir.const<!f32t>

// Объявление ядра, вычисляющего скользящее среднее.
dfcir.kernel "MovingAverage" {
  // Объявление используемых констант и входных данных.
  %C0   = dfcir.constant<!ui32t> 0: i64
  %C0F  = dfcir.constant<!f32t> 0.0: f32
  %C1   = dfcir.constant<!ui32t> 1: i64
  %C2   = dfcir.constant<!ui32t> 2: i64
  %C3   = dfcir.constant<!ui32t> 3: i64
  %x    = dfcir.input<!f32t>("x")
  %size = dfcir.scalarInput<!ui32t>("size")

  %prevOrig = dfcir.offset(%x, -1: i64) : !f32s
  %nextOrig = dfcir.offset(%x,  1: i64) : !f32s

  // Объявление счетчика и вычисление предикатов.
  %cnt = dfcir.simpleCounter<!ui32t>(%size: !ui32v)
  %abvLowBnd = dfcir.greater(%cnt: !ui32s, %C0: !ui32c) : !bools
  %lessThanSize = dfcir.sub(%size: !ui32v, %C1: !ui32c) : !ui32s
  %blwUppBnd = dfcir.less(%cnt: !ui32s, %lessThanSize: !ui32s) : !bools
  %inBounds = dfcir.and(%abvLowBnd: !bools, %blwUppBnd: !bools) : !bools
  %prev = dfcir.mux(%abvLowBnd: !bools, %C0F: !f32c, %prevOrig) : !f32s
  %next = dfcir.mux(%blwUppBnd: !bools, %C0F: !f32c, %nextOrig) : !f32s

  // Вычисление суммы для подсчета скользящего среднего.
  %divisor = dfcir.mux(%inBounds: !bools, %C2: !ui32c, %C3: !ui32c) : !ui32s
  %sum1 = dfcir.add(%prev: !f32s, %x: !f32s) : !f32s
  %sum2 = dfcir.add(%sum1: !f32s, %next: !f32s) : !f32s
  %result = dfcir.div(%sum2: !f32s, %divisor: !ui32s) : !f32s

  // Объявление выходного потока.
  %y = dfcir.output<!f32t>("y") <= %result: !f32s
}
```

Листинг 2. Ядро, вычисляющее скользящее среднее.
Listing 2. A kernel computing a moving average.

4.4 Трансляция в FIRRTL

В табл. 4 представлено отображение типов DFCIR в типы FIRRTL.

Табл. 4. Трансляция типов DFCIR в типы FIRRTL.
Table 4. DFCIR-to-FIRRTL type translation.

DFCIR	**FIRRTL**
!**dfcir.float**<*m*, *n*>	!**firrtl.uint**<*m* + *n* + 1>
!**dfcir.fixed**<false, *m*, *n*>	!**firrtl.uint**<*m* + *n*>
!**dfcir.fixed**<true, *m*, *n*>	!**firrtl.int**<*m* + *n* + 1>
!**dfcir.rawbits**<*n*>	!**firrtl.uint**<*n*>
!**dfcir.complex**<*type*>	!**firrtl.bundle**<re: *type*, im: *type*>
!**dfcir.struct**<*name_i: type_i, ...*>	!**firrtl.bundle** <*name_i: type_i, ...*>
!**dfcir.vector**<*type*, *size*>	!**firrtl.vector**<*type*, *size*>

В табл. 5 описывается обобщенная схема трансляции конструкций DFCIR в конструкции FIRRTL.

Табл. 5. Трансляция конструкций DFCIR в конструкции FIRRTL.
Table 5. DFCIR-to-FIRRTL construct translation.

DFCIR	**FIRRTL**
dfcir.kernel "*Name*"	**firrtl.module** @*Name*(..., !**firrtl.clock**)
Ядро транслируется в одноименный модуль. Для каждого объявления входного / выходного скаляра / потока создается соответствующий порт в модуле. Интерфейс модуля дополняется портом тактового сигнала (!**firrtl.clock**). Создается одноименная схема (**firrtl.circuit**), в которую помещается созданный модуль.	
dfcir.constant *:*	**firrtl.constant** *:*
dfcir.op(...):(...)	**firrtl.instance**(..., !**firrtl.clock**):(...)
Для всех арифметико-логических операций (за исключением побитовых) объявляются внешние модули (**firrtl.extmodule**) и создаются их экземпляры (**firrtl.instance**); интерфейсы модулей расширены портами тактового сигнала. Побитовые операции DFCIR имеют прямые аналоги в FIRRTL. Выходы одних экземпляров соединяются со входами других с помощью операции **firrtl.connect**.	
dfcir.connect(...)	**firrtl.connect**(...)
dfcir.bits(...)	**firrtl.bits**(...)
dfcir.cat(...)	**firrtl.cat**(...)
dfcir.latency [*d*] ...	**firrtl.instance**(..., !**firrtl.clock**):(...)
Для всех задержек объявляются внешние модули (**firrtl.extmodule**) и создаются их экземпляры (**firrtl.instance**); интерфейсы модулей расширены портами тактового сигнала. Выходы одних экземпляров соединяются со входами других с помощью операции **firrtl.connect**.	

5. Экспериментальный анализ

Реализация DFCIR была внедрена в открытый инструмент синтеза Utopia HLS [30]. Для высокоуровневого описания ПВ в Utopia HLS используется язык DFCxx (оба языка разработаны в рамках одного проекта; рассмотрение DFCxx выходит за рамки статьи). Для экспериментального анализа использовался пример обратного дискретного косинусного преобразования (ОДКП) и методика сравнения, описанная в [12]. По DFCxx-описанию ОДКП 8x8 было построено промежуточное представление на DFCIR; на базе промежуточного представления было выполнено планирование вычислений, после чего было сгенерировано SystemVerilog-описание; далее с помощью инструмента Vivado 2017.4 синтезирована схема для ПЛИС Xilinx Virtex Ultrascale+ (XCVU9P-FLGB2104-2-E). Результаты, дополненные данными из [12], представлены в табл. 6.

Вычислитель синтезированный Utopia HLS, оказался близким по производительности к вычислителю, синтезированному XLS (31.6 и 31.3 млн операций в секунду соответственно), при этом его описание на языке DFCxx на 65 строк меньше. Лидером является инструмент MaxCompiler, однако в отличие от других сравниваемых инструментов он строит схему с интерфейсом PCIe, а не AXI-Stream.

6. Заключение

В работе предложен язык DFCIR, ориентированный на промежуточное представление специализированных ПВ. Система понятий DFCIR заимствована из языка MaxJ, реализация выполнена на основе инфраструктуры MLIR, синтаксис спроектирован с оглядкой на языки

проекта CIRCT, прежде всего FIRRTL. Реализация DFCIR внедрена в инструмент синтеза Utopia HLS [30]; проведено экспериментальное сравнение разработанных средств с другими инструментами (MaxCompiler, XLS, Vivado HLS, Bambu). Результаты показали потенциал предложенного подхода (инструмент уступил лишь MaxCompiler). Среди направлений дальнейшей работы следует выделить следующие: (1) реализация оптимизирующих преобразований моделей ПВ; (2) разработка методов имитационной и формальной верификации ПВ; (3) создание предметно-ориентированных библиотек функциональных элементов для конструирования ПВ.

Табл. 6. Результаты экспериментального анализа инструмента Utopia HLS.
Table 6. Utopia HLS experimental analysis results.

Язык / инструмент	**Лицензия**	**Специализация**	**Число строк в описании**	**Итоговая частота (МГц)**	**Пропускная способность (млн оп/с)**
Verilog	–	Аппаратура общего назначения	316	113.21	14.15
MaxJ / MaxCompiler	Проприетарная	Потоковые вычислители	163	403.13	44.79
DSLX / XLS	Apache License v2.0	Потоковые вычислители	243	250.50	31.31
C / Vivado HLS	Проприетарная	Аппаратура общего назначения	130	131.46	16.43
C / Bambu	GPL v3.0	Аппаратура общего назначения	191	257.33	1.39
DFCxx / Utopia HLS	Apache License v2.0	Потоковые вычислители	178	252.52	31.57

Список литературы / References

[1]. I. Ekmecic, I. Tartalja, V. Milutinovic. EM/sup 3/: a taxonomy of heterogeneous computing systems. Computer, vol. 28, no. 12, 1995. pp. 68-70. DOI: 10.1109/2.476202.

[2]. I. Ekmecic, I. Tartalja, V. Milutinovic. A survey of heterogeneous computing: concepts and systems. Proceedings of the IEEE, vol. 84, no. 8, 1996. pp. 1127-1144. DOI: 10.1109/5.533958.

[3]. IEEE Standard for Verilog Hardware Description Language. IEEE Std 1364-2005, 2006. DOI: 10.1109/IEEESTD.2006.99495.

[4]. IEEE Standard for SystemVerilog: Unified Hardware Design, Specification and Verification Language. IEEE Std 1800-2005, 2005. DOI: 10.1109/IEEESTD.2005.97972.

[5]. IEEE Standard VHDL Language Reference Manual. IEEE Std 1076-2019, 2019. DOI: 10.1109/IEEESTD.2019.8938196.

[6]. С.Ю. Кун. Матричные процессоры на СБИС. Мир, 1991. 672 с.

[7]. E. Lee, D. Messerschmitt. Static scheduling of synchronous data flow programs for digital signal processing. IEEE Transactions on Computers, C-36(1), 1987. pp. 24-35. DOI: 10.1109/TC.1987.5009446.

[8]. C. Leiserson, J. Saxe. Retiming synchronous circuitry. Algorithmica, vol. 6, 1991. pp. 5-35. DOI: 10.1007/BF01759032.

[9]. MLIR: Multi-level Intermediate Representation. URL: https://mlir.llvm.org/ (дата обращения: 15.08.2024).

[10]. The LLVM Compiler Infrastructure Project. URL: https://llvm.org/ (дата обращения: 15.08.2024).

[11]. CIRCT: Circuit IR Compilers and Tools. URL: https://circt.llvm.org/ (дата обращения: 15.08.2024).

[12]. A. Kamkin, M. Chupilko, M. Lebedev, S. Smolov, G. Gaydadjiev. High-level synthesis versus hardware construction. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2023. DOI: 10.23919/DATE56975.2023.10136904.

[13]. Chisel/FIRRTL Hardware Compiler Framework. URL: https://www.chisel-lang.org/

(дата обращения: 15.08.2024).
[14]. MyHDL. URL: https://www.myhdl.org/ (дата обращения: 15.08.2024).
[15]. Bluespec SystemVerilog Documentation. URL: https://web.ece.ucsb.edu/its/bluespec/ (дата обращения: 15.08.2024).
[16]. P. Coussy, M. Meredith, D. Gajski, A. Takach. An Introduction to High-level Synthesis. IEEE Design and Test of Computers, 2009. pp. 8-17. DOI: 10.1109/MDT.2009.69.
[17]. Vitis HLS. URL: https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0090-vitis-hls-hub.html/ (дата обращения: 15.08.2024).
[18]. Bambu: A Free Framework for the High-Level Synthesis of Complex Applications. URL: https://panda.deib.polimi.it/?page_id=31/ (дата обращения: 15.08.2024).
[19]. LegUp: Software IDE for Programming Microchip FPGAs. URL: http://lightsail.legupcomputing.com/ (дата обращения: 15.08.2024).
[20]. MaxCompiler. URL: https://www.maxeler.com/products/software/maxcompiler/ (дата обращения: 15.08.2024).
[21]. XLS: Accelerated HW Synthesis. URL: https://google.github.io/xls/ (дата обращения: 15.08.2024).
[22]. О.В. Непомнящий, А.И. Легалов, И.Н. Рыженко. Метод архитектурно-независимого высокоуровневого синтеза СБИС. Известия ЮФУ. Технические науки, №8 (202), 2018. с. 38-47. DOI: 10.23683/2311-3103-2018-8-38-47.
[23]. M. Kingshuk, B. Uday. HIR: An MLIR-based Intermediate Representation for Hardware Accelerator Description. ACM International Conference on Architectural Support for Programming Languages and Operating Systems (APLOS), vol. 4, 2023. pp. 189-201. DOI: 10.1145/3623278.3624767.
[24]. FIRRTL dialect. URL: https://circt.llvm.org/docs/Dialects/FIRRTL/(дата обращения: 15.08.2024).
[25]. HW dialect. URL: https://circt.llvm.org/docs/Dialects/HW/ (дата обращения: 15.08.2024).
[26]. Comb dialect. URL: https://circt.llvm.org/docs/Dialects/Comb/ (дата обращения: 15.08.2024).
[27]. Seq dialect. URL: https://circt.llvm.org/docs/Dialects/Seq/, (дата обращения: 15.08.2024).
[28]. LLHD dialect. URL: https://circt.llvm.org/docs/Dialects/LLHD/ (дата обращения: 15.08.2024).
[29]. IEEE Standard Multivalue Logic System for VHDL Model Interoperability. IEEE Std 1164-1993, 1993. DOI: 10.1109/IEEESTD.1993.115571.
[30]. Utopia: A High-Level Synthesis Framework. URL: https://github.com/ispras/utopia-hls/ (дата обращения: 15.08.2024).

Информация об авторах / Information about authors

Александр Сергеевич КАМКИН – кандидат физико-математических наук, ведущий научный сотрудник ИСП РАН и РЭУ им. Г.В. Плеханова; преподает в МГУ, МФТИ, НИУ ВШЭ и РЭУ. Научные интересы: формальные методы, синтез и верификация цифровой аппаратуры, гетерогенные компьютерные системы.

Alexander Sergeevich KAMKIN – Cand. Sci. (Phys.-Math.), leading researcher at ISP RAS and Plekhanov Russian University of Economics; teaches at MSU, MIPT, HSE, and PRUE. Research interests: formal methods, synthesis and verification of digital hardware, and heterogeneous computer systems.

Михаил Юрьевич ЛИТВИНОВ – лаборант Отдела технологий программирования ИСП РАН. Научные интересы: автоматизация проектирования гетерогенных компьютерных систем.

Mikhail Yurievich LITVINOV is a laboratory assistant at Software Engineering Department of ISP RAS. Research interests: automated design of heterogeneous computer systems.

Иван Александрович ГРИГОРОВ – инженер Отдела технологий программирования ИСП РАН. Научные интересы: высокоуровневый синтез, модели потоков данных.

Ivan Aleksandrovich GRIGOROV is an engineer assistant at Software Engineering Department of ISP RAS. Research interests: high-level synthesis and dataflow models.

DOI: 10.15514/ISPRAS-2024-36(5)-4

CC BY

Декларативный синтез графических интерфейсов пользователя с помощью реляционного решателя ограничений

Д.С. Косарев, ORCID: 0000-0002-6773-5322 <d.kosarev@spbu.ru>
П.А. Лозов, ORCID: 0000-0003-3563-2828 <lozov.peter@gmail.com>
Д.Ю. Булычев, ORCID: 0000-0001-8363-7143 <dboulytchev@math.spbu.ru>

Санкт-Петербургский государственный университет,
Россия, 198504, Санкт-Петербург, Университетский пр., д. 28.

Аннотация. Авторы представляют систему, которая по набору правил-ограничений на дизайн и по структурному описанию пользовательского интерфейса (GUI), порождает набор конкретных интерфейсов, каждый из которых по построению соблюдает эти ограничения. Задача, ставящаяся перед системой, описывается как проблема удовлетворения ограничений, после чего на основе реляционного подхода “решатель-из-верификатора” конструируется корректный и полный решатель. Также описывается набор улучшений, делающих предложенный решатель более эффективным.

Ключевые слова: проектирование интерфейсов; синтез программ; программирование в ограничениях; реляционное программирование; встраиваемый язык miniKanren.

Для цитирования: Косарев Д.С., Лозов П.А., Булычев Д.Ю. Декларативный синтез графических интерфейсов пользователя с помощью реляционного решателя ограничений. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 47–66. DOI: 10.15514/ISPRAS–2024–36(5)–4.

Declarative GUI Layout Synthesis with Relational Constraint Solvers

D.S. Kosarev, ORCID: 0000-0002-6773-5322 <d.kosarev@spbu.ru>
P.A. Lozov, ORCID: 0000-0003-3563-2828 <lozov.peter@gmail.com>
D.Yu. Boulytchev, ORCID: 0000-0001-8363-7143 <dboulytchev@math.spbu.ru>

St. Petersburg State University,
Universitetski pr., 28, St. Petersburg, 198504, Russia.

Abstract. Authors describe a system which, given a set of designer-specified layout constraints (guidelines) and a description of graphic user interface (GUI) logical structure generates a set of particular layouts. Each of these layouts comply with given guidelines by construction. Authors also give a formal treatment of the task as a constraint satisfiability problem and describe the construction of a sound and complete solver based on the utilization of relational verifier-to-solver approach. They also describe a number of refinements which make the solver more efficient and applicable.

Keywords: interface design; program synthesis; constraint programming; relational programming; embedded domain-specific language miniKanren.

For citation: Kosarev D.S., Lozov P.A., Boulytchev D.Yu. Declarative GUI Layout Synthesis with Relational Constraint Solvers. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024. pp. 47-66 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-4.

1. Введение

Графические интерфейсы пользователя (GUI) – один из самых распространенных способов взаимодействия с программами. Как на персональном компьютере, так и на ноутбуке, мобильном телефоне, банкомате, игровой консоли и т.д. пользователь встретится с некоторым набором графических элементов управления и интуитивно понятным смыслом. За такой низкий порог вхождения мы платим некоторую цену: графические интерфейсы управления не так просто разрабатывать [1, 2].

Обычные пользователи как правило не осознают, что делает графический интерфейс приятно выглядящим и удобным в использовании. Чтобы получить качественный интерфейс необходимо учесть множество эстетических, эргономических и психологических аспектов, а это требует экспертизы, которой программисты обладают не всегда. Из-за этого появляются должности не только по разработке интерфейсов (UI), но и по дизайну взаимодействия с пользователем (UX) [3], где специалисты разрабатывают правила (англ. guidelines) разработки хороших интерфейсов. Данные правила часто неполны, двусмысленны и неформальны, а следование им требует интуиции, которой разработчики могут и не обладать. Взаимодействие между разработчиками и дизайнерами часто осложняется разницей в экспертизе, подходах и складах мышления [4]. Более того, упомянутые правила не всегда естественно переносятся между разными устройствами и даже разрешениями экрана. Поэтому, чтобы поддерживать все желаемые платформы, разработчикам придется следовать нескольким наборам правил проектирования интерфейсов одновременно.

В данной работе представлен подход к синтезу расположения (англ. layout) элементов управления GUI с учетом набора правил проектирования интерфейсов. Наша модель GUI отделяет логическую структуру элементов управления (часть предельно понятная программистам), от конкретного расположения элементов, отступов и выравниваний (понятных дизайнеру). Мы разработали автоматический подход, который по логической структуре (предоставляемой программистом) и набору правил (предоставленных дизайнером) синтезирует раскладку элементов управления, удовлетворяющую этим правилам. В общем случае правила неоднозначны, поэтому несколько конкретных раскладок может им удовлетворять. Мы рассматриваем синтез интерфейсов как задачу удовлетворения

ограничений (англ. constraint satisfaction), и используем реляционное программирование [5] и подход верификатор-из-решателя [6] (англ. verifier-to-solver), чтобы получить решатель, располагающий элементы управления с помощью примитивов. В примитивной форме такой решатель неэффективен, и мы применили набор оптимизаций (в том числе специализацию), чтобы его ускорить. В результате получился решатель, написанный на смеси из реляционного и функционального кода. Затем набор примитивов расположения превращается в линейные ограничения на координаты и решается с помощью SMT решателя Z3 [7], что дает нам абсолютные координаты всех элементов управления. Этот подход позволяет получить *корректный* и *полный* решатель: по указанной логической структуре он синтезирует все расположения элементов, удовлетворяющие правилам дизайна.

При этом решатель достаточно эффективен, чтобы запускаться на обычном ноутбуке.

2. Модель GUI

В данном разделе мы изложим модель пользовательского интерфейса, которая учитывает все важные особенности предметной области. Для реального использования модель должна быть немного расширена, но это не предоставит сложности, так как расширения не вносят серьёзных изменений в алгоритм синтеза.

Мы можем выделить три важных понятия, связанных с интерфейсами пользователя: *структура*, *расположение* (англ. layout) и набор правил проектирования интерфейсов (англ. guideline), который для краткости далее будем называть *предписанием*. *Структура* описывает набор элементов управления и связи между ними, *расположение* уточняет их визуальные позиции друг к другу. Такое разделение согласуется с распространенной практикой отделения бизнес-логики от визуального представления [8]. Наконец, *предписание* отображает определенные структуры в конкретные примитивы расположения.

Продемонстрируем эти понятия на примере. Рассмотрим интерфейс с рис. 1 и опишем структуру следующим образом.

- Всего три элемента управления GUI: галочка (англ. check box), метка и поле со списком (англ. combo box).
- Присутствует зависимость между меткой и полем, так как первое *описывает* второе; следовательно, их можно объединить в одну сущность.
- Существует зависимость между галочкой и только что введенной композитной сущностью, так как, по-видимому, дизайнер интерфейса полагал, что важность элементов управления убывает сверху вниз.

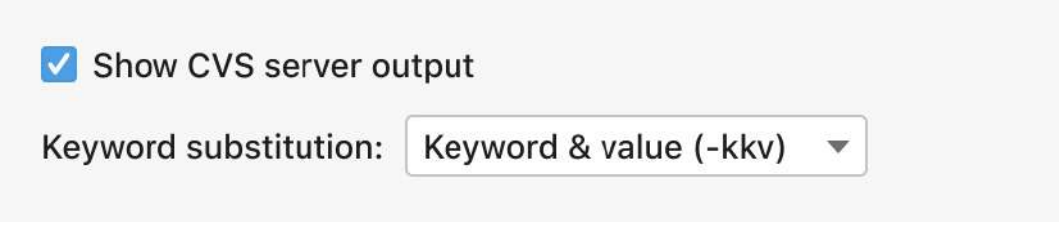


Рис. 1. Пример интерфейса GUI.
Pic. 1. Sample GUI form.

Мы схематично изобразили структуру на рис. 2. Теперь мы можем рассматривать её как набор именованных *отношений* между элементами управления, или даже над данными в них (строками, числами и т.п.). Наш подход разделяет все отношения на две категории: *абстрактные* и *конкретные*. Семантика абстрактных определяется предписанием, конкретных – нашей системой. Конкретные описывают типы элементов управления, различные их свойства (содержимое текстовых полей, размеры, и т.п.), а также позволяют

объединять элементы управления в упорядоченные или неупорядоченные группы, или в *виртуальные* – элементы управления, объединяющие несколько в один целый.

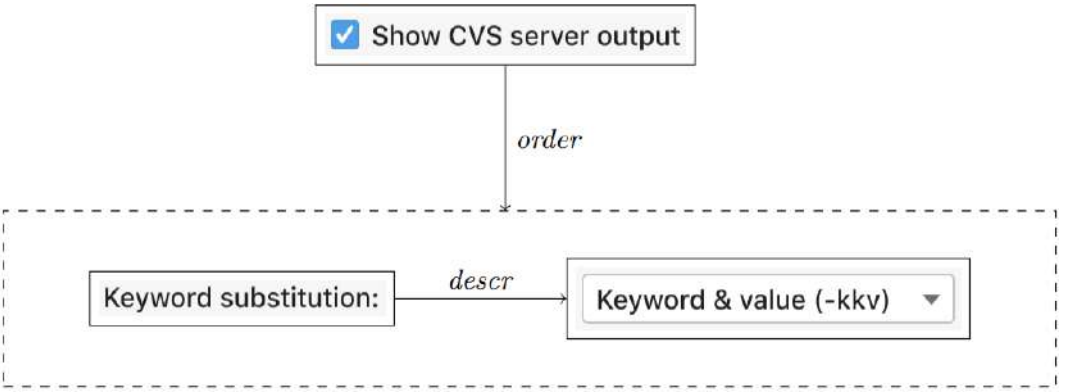


Рис. 2. Пример структуры GUI.
Pic. 2. Sample GUI form structure.

Множество *абстрактных* отношений включает: бинарное упорядочивание (один-за-другим), описание (один описывает другой, например, метка поле ввода или галочку), подчиненность (галочка делает активным/неактивным поле ввода). В реализации эти отношения – это просто удобные имена, используемые в структуре и предписании. Их семантика полностью определяется предписанием, а система никаким особым образом не обрабатывает эти отношения. Пользователи могут добавлять новые абстрактные отношения и определять их семантику в предписании для своих нужд.

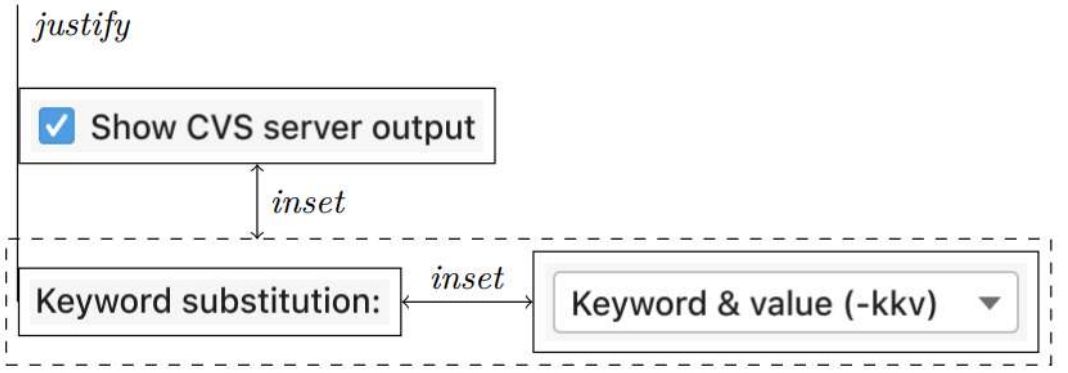


Рис. 3. Пример расположения элементов управления.
Pic. 3. Sample GUI form layout.

Для конкретной структуры её расположение (англ. layout) определяется набором примитивов расположения, выравниванием и другими подобными свойствами. В примере метка и поле со списком располагаются горизонтально с некоторым отступом, галочка располагается сверху от введённого виртуального элемента для пары метка-поле с другим отступом, и всё расположение выровнено влево, что изображено на рис. 3. В отличие от отношений, использованных при описании структуры, все примитивы расположения имеют фиксированную семантику, на которой основан процесс синтеза.

Наконец, предписание состоит из набора правил, описывающих какие примитивы расположения должны использоваться для соответствующих частей структуры и при каких условиях. На рис. 4 можно увидеть пример такого правила, взятого из правил [9]

проектирования интерфейсов компании JetBrains. Обычно, эти правила описываются неформально на примерах, в следующем разделе мы их формализуем.

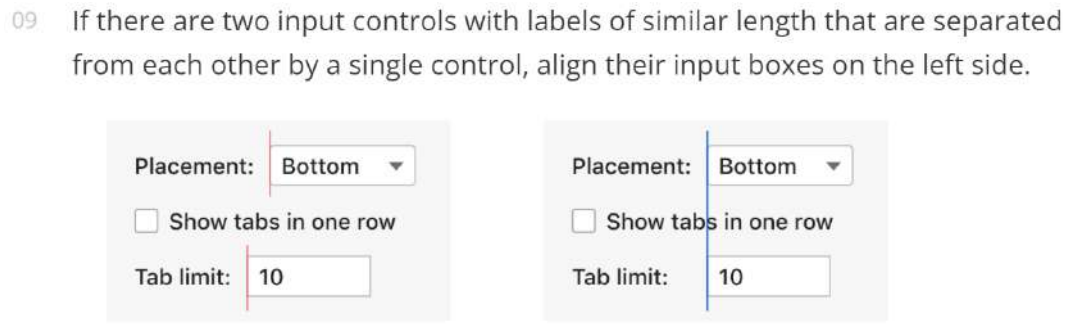


Рис. 4. Пример правила предписания.
Pic. 4. Sample GUI guideline rule.

3. Синтез расположения как задача удовлетворения ограничений

В нашей модели логическая структура, упомянутая в предыдущем разделе, описывается как набор отношений на некотором множестве элементов управления C. Обозначим нашу структуру $\Sigma = \{r_i^{n_i} \mid r_i^{n_i} \subseteq C^{n_i}\}$ как множество отношений, где $r_i^{n_i}$ является n_i-арным отношением над C; без потери общности мы не будем рассматривать такие отношения, как размер, текстовое содержимое элементов управления и т.п.

Реляционным шаблоном будем называть синтаксическую форму $r^n(X_1, \dots, X_n)$, где r^n – это n-арное отношение и X_j – это необязательно различные переменные.

Шаблоном расположения назовем синтаксическую форму $l(X_1, \dots, X_n)$, где l – это примитивы расположения, а X_j – переменные.

Как $s[X_i \leftarrow c_i]$ мы будем обозначать результат подстановки элементов управления $c_i \in C$ вместо переменных X_i в реляционный шаблон или в шаблон расположения s. Пусть $FV(s)$ – это множество всех переменных в реляционном шаблоне или в шаблоне расположения s, и будем называть *экземпляром расположения* шаблон без переменных.

Пусть $r^n(X_1, \dots, X_n)$ – это реляционный шаблон. Если для некоторого набора элементов управления $c_1, \dots, c_n \in C$ верно $(c_1, \dots, c_n) \in r^n$, то будем говорить, что результат подстановки $r^n(c_1, \dots, c_n)[X_i \leftarrow c_i]$ *встречается* в структуре Σ.

Правило (предписания) будем записывать как

$$p_1, \dots, p_k \mapsto l_1, \dots, l_m,$$

где p_i – реляционные шаблоны, l_j – шаблоны расположения. Также потребуем, чтобы любая переменная в правой части хотя бы раз встречалась в левой. Неформально, правило говорит, что если некоторые элементы управления некоторым образом соотносятся в структуре, то для них должны использоваться определенные примитивы расположения.

Предположим, нам дана структура Σ и множество экземпляров расположения S. Будем говорить, что это множество *подтверждается* набором правил R тогда и только тогда, когда для каждого элемента управления $l^* \in S$

(1) существует правило $p_1, \dots, p_n \mapsto l_1, \dots, l_i \dots, l_m$ в R, такое что $l^* = l_i[X_j \leftarrow c_j]$ для некоторых $\{c_j\} \subseteq C$, где $FV(l_i) = \{X_j\}$;

(2) для всех переменных $\{Y_1, \dots, Y_t\} = FV(p_1, \dots, p_n) \setminus FV(l_i)$ существуют элементы управления $c'_1, \dots, c'_t \in C$ такие, что для всех j подстановка $p_j[X_r \leftarrow c_r, \dots, Y_s \leftarrow c'_s]$ встречается в структуре Σ;

(3) $l_j[X_r \leftarrow c_r, \ldots, Y_s \leftarrow c'_s] \in S$ для всех j.

Неформально выражаясь, R подтверждает множество S тогда и только тогда, когда все элементы управления l^* могут быть корректно выведены из правил R. Это означает, что должно быть какое-то правило с шаблоном расположения l_i, который превратится в l^* с помощью некоторой подстановки переменных X_i. Но в этом правиле кроме X_i могут быть другие переменные Y_j. Если и для них существует такая подставка, которая совместно с подстановкой переменных X_i, заставит все реляционные шаблоны слева встречаться в Σ, тогда правило может быть применено и l^* выведено. К тому же, в правой части могут быть другие шаблоны расположения, и если и они окажутся тоже выведенными, то этот экземпляр расположения должен быть добавлен в S.

Подтверждённые множества содержат в себе *все* расположения, которые могут быть обоснованы правилами. Эти множества не содержат лишних расположений. Однако подтверждённое расположение элементов управления не делает его автоматически “хорошим” с точки зрения дизайнера. Пустое множество всегда подтверждается, но едва ли его можно назвать полезным расположением элементов. Из-за этого нам необходимо ввести ещё одно понятие: покрытие.

Будем говорить, что экземпляры расположения S *покрывают* структуру Σ тогда и только тогда, когда для каждого элемента управления $c \in C$ существует экземпляр расположения в S, который связывает c, т.е. содержит c как параметр. Покрытие неформально обозначает, что ни один элемент управления из структуры не остался не упомянутым в данном расположении.

Множество подтверждённых расположений в некотором смысле соответствует всем *корректным* расположениям с учетом правил, а покрытие – полноте. Однако мы воздержимся от введения этих понятий здесь, чтобы потом их употреблять в более общепринятом смысле.

Следующее требование навязано тем, что правила (предписания) должны применяться, а не игнорироваться. Будем называть подтвержденное и покрытое множество S *максимальным* тогда и только тогда, когда не существует подтвержденного и покрытого множества S', что $S \subset S'$.

Наконец, расположения могут содержать конфликты. Например, если элемент A должен быть расположен сверху B и наоборот, то, очевидно, что мы получили несовместные ограничения. Среди всех конфликтов мы можем выделить следующие:

- Не должно быть циклов, составленных только из примитивов расположения для вертикальной (горизонтальной) композиции.
- Не более одного элемента управления должно располагаться строго справа/снизу от другого элемента управления. Это и предыдущее требование проистекает из того, что отношения, составленные из этих примитивов, должны быть линейными порядками.
- Если два виртуальных элемента управления расположены горизонтально, то никакие два элемента управления внутри одного и другого не должны располагаться вертикально. Аналогично для вертикально расположенных виртуальных элементов управления.
- Никакие два элемента управления не могут быть расположены одновременно и вертикально, и горизонтально относительно друг друга.

Эти конфликты возникают из-за того, что мы располагаем элементы управления на двумерной плоскости (холсте). Кроме них бывают и другие конфликты, которые мы опустили для краткости изложения. Будем называть множества экземпляров без конфликтов *совместными*.

Итого, нам интересны подтвержденные, покрывающие структуру и совместные расположения с учетом выданного набора правил. Будем называть такие расположения *корректными*.

Стоит обсудить вопрос производительности подхода в худшем случае. Легко заметить, что нахождение корректного расположения – в общем случае NP-полная задача. Для подтверждения этого, рассмотрим решение NP-полной задачи поиска Гамильтонова пути в графе с помощью нашего подхода. Для произвольного ненаправленного графа мы можем построить структуру, в которой один элемент управления соответствует одной вершине графа, а некоторое бинарное отношение "E" – рёбрам. Затем рассмотрим такие правила расположения:

$$E\ (x, y) \mapsto \mathbf{hor}\ (x, y),$$
$$E\ (x, y) \mapsto \mathbf{hor}\ (y, x).$$

Любое корректное расположение с учётом правил по определению содержит все элементы управления (т.е. узлы графа) и только примитивы расположения вида "**hor**", где каждое вхождение примитива соответствует одному ребру в графе. Так как примитивы не могут образовывать цикл, и каждый элемент управления может быть связан ребром только с одним другим элементом, то ребра образуют Гамильтонов путь.

С другой стороны, можно заметить, что количество всех корректных расположений в худшем случае экспоненциально зависит от размера структуры. И выдача всех корректных расположений с помощью решателя, заставит его работать в худшем случае экспоненциально.

4. Реляционное программирование, верификаторы и решатели

Реализация синтеза интерфейсов использует реляционное программирование. В этом разделе мы кратко расскажем, что это такое, и как работает наш инструмент.

Реляционное программирование [5] – это подход, основанный на идее описания программ как отношений. Его можно рассматривать как вид логического программирования, в котором порицается использование всех не реляционных конструкций (эффекты, не логические конструкции). В узком смысле реляционное программирование – это написание программ на miniKanren – специально сконструированном для этих целей, встраиваемом предметно-ориентированном языке. В оригинале он разрабатывался для Scheme/Racket, позже miniKanren был портирован на другие языки[1]. Мы используем строго статически типизированную реализацию miniKanren, встроенную в OCaml, которая называется OCanren [10]. В miniKanren используется та же теория дизъюнктов Хорна, как и в Prolog, но с другим синтаксисом: явные унификации и дизъюнкции, конъюнкции, явное создание свежих (англ. fresh) переменных, а также используется стратегия *поиска с чередованием* (англ. interleaving search) [11], которая полна [12]. Кроме унификации с проверкой цикличности вхождений (англ. occurs-check) по умолчанию, miniKanren можно оснастить другими ограничениями, например: неравенства [13], конечные домены [14] или конструкции из номинальной логики [15].

В контексте нашей работы, самым важным свойством miniKanren является возможность выражения *обратимых вычислений*. Известно [16-17], что некоторые сложные программы могут быть построены как результат обращения некоторых других более простых программ. В частности, *решатель* задачи поиска можно рассматривать как обращение *верификатора* – программы, которая проверяет корректность ответа. Широко известно, что задача проверки корректности решения, как правило, гораздо проще задачи поиска корректного решения. Реляционная природа miniKanren позволяет осуществлять обратимые вычисления очень просто, что помогает в задачах синтеза программ [18–20].

[1] http://minikanren.org/#implementations

Другим компонентом нашего подхода является *реляционное преобразование* (англ. relational conversion). Во многих случаях (но не всегда) проще получить реляционную спецификацию из кода на функциональном языке, чем написать спецификацию вручную. Мы используем инструмент noCanren, который преобразует программы, написанные на подмножестве OCaml, в OCanren-спецификации, корректные по построению.

Продемонстрируем наш подход на следующем обозримом примере. Рассмотрим программу на рис. 5, которая складывает два натуральных числа в форме Пеано. Её реляционный образ, полученный с помощью реляционного преобразования (не буквально, но в эквивалентной записи) – на рис. 6. Сравнивая их, можно обратить внимание, что сопоставление с образцом было заменено на дизъюнкцию ($\vee$) и унификацию ($\equiv$); модификации потока данных – на конъюнкцию ($\wedge$); свежие переменные были добавлены, где это необходимо; а постфиксная нотация "^{o}" традиционно используется для обозначения определения реляций (вычислимых отношений). В отличие от функциональной реализации, у реляционной три аргумента (`x`, `y` и `z`) каждый из которых может содержать свежие переменные. Вычисление конструкции **`run`*** `{add`^{o} `x y z}` возвращает ленивый поток ответов, который содержит все подстановки этих трёх переменных, такие, что отношение $\{(x, y, z) \in \mathbb{N} \mid x + y = z\}$ верно. Этот поток можно исследовать в OCaml, чтобы получать ответы по одному. Одна и та же спецификация может использоваться и для сложения, и для вычитания, и для разбиения числа на два слагаемых.

```
let rec add x y =
  match x with
  | O → y
  | S xp → S (add xp y)
```

Рис. 5. Функциональная реализация сложения чисел Пеано.
Pic. 5. Functional implementation of Peano numbers addition.

```
let rec addᵒ x y z = ocanren {
  x ≡ O ∧ z ≡ y ∨
  fresh xp, zp in
    x ≡ S xp ∧
    z ≡ S zp ∧
    addᵒ xp y zp
}
```

Рис. 6. Реляционная реализация сложения чисел Пеано.
Pic. 6. Relational implementation of Peano numbers addition.

5. Синтез расположения GUI

В данном разделе мы сконструируем синтезатор расположений на основе предписания. Сделаем это за несколько шагов: начнём с простого функционального верификатора, сконвертируем его в реляционную форму, запустим в обратном направлении, проанализируем результат и применим некоторые оптимизации.

5.1 Функциональный верификатор

Устройство начального функционального верификатора на OCaml довольно прямолинейно. Воспользуется обычными для функционального программирования списками и алгебраическими типами данных, чтобы описать структуру и примитивы расположения. По

правилам предписания мы *синтезируем* верификатор, который в качестве аргументов принимает структуру и экземпляры расположения, и проводит прямолинейную проверку покрытия, совместности и подтверждения свойств, согласно определениям из раздела 3.

Покрытие можно легко получить, обойдя все примитивы расположения, собирая все невиртуальные элементы управления в множество, и проверяя, что оно соответствует множеству всех невиртуальных в структуре. Совместность проверяется аналогично, по определению.

Для подтверждения нам нужно обойти все примитивы расположения и подтвердить каждый их них. Это требует обращения правила предписания. Например, рассмотрим следующее правило:

$$\textbf{describes}\,(X,Y)\ \mapsto \textbf{vert}\,(X,Y),\ \textbf{halign}\,(X,Y).$$

$$\begin{aligned}
&\texttt{if}\ \ \textbf{vert}\,(X,Y)\in S\\
&\texttt{then}\ \ \textbf{describes}\,(X,Y)\in\Sigma\ \wedge\ \textbf{halign}\,(X,Y)\in S\\
&\texttt{else if}\ \ \textbf{halign}\,(X,Y)\in S\\
&\texttt{then}\ \ \textbf{describes}\,(X,Y)\in\Sigma\ \wedge\ \textbf{vert}\,(X,Y)\in S\,.
\end{aligned}$$

Оно определяет следующие случаи для процедуры подтверждения структуры Σ и набора экземпляров расположения S.

Мы *не проверяем максимальность* на данном шаге. Причина этому в особенности функционально-реляционного программирования. Чтобы проверить максимальность, нам нужно найти все остальные совместные экземпляры, которые и подтверждены, и покрыты. Но именно этой задачи мы хотим *избежать*, написав функциональный верификатор. Поэтому, реализация этого на функциональном языке подрывает всю идею использования реляционного программирования. Подчеркиваем, что это довольно частый случай для нашего подхода: для удобства программирования нам необходимо аккуратно провести границу между функциональным и реляционным кодом, чтобы не делать то, что мы получим позже за просто так.

Максимальность проверяется другим образом. Мы применяем реляционное преобразование для функционального верификатора, и запускаемся на конкретной структуре и на *свободной переменной* на месте конкретных экземпляров расположения. За счет полноты поиска miniKanren выдаст нам все множества, одновременно совместные, покрытые и подтверждённые. Затем мы просто отфильтруем не максимальные.

На этом первый шаг закончен, и мы получили полный неэффективный синтезатор. У наивной реализации присутствуют следующие недостатки:

- Синтезатор выдает (экспоненциально) большое количество эквивалентных ответов. Мы использовали списки для представления множеств, и синтезатор выдал нам все перестановки как уникальные ответы.
- Мы узнали, что представление структуры с помощью типов данных чрезмерно, и мы можем напрямую строить реляции miniKanren по структуре.
- Синтезатор выдает много частичных (не максимальных) ответов, что замедляет синтез и делает его непрактичным.

В следующих подразделах мы решим эти проблемы.

5.2 Представление с помощью бинарных гиперкубов

Для первой проблемы мы применили специальное представление расположения (*бинарный гиперкуб*), в котором любое расположение представляется однозначно. В сущности, это побитовое представление множеств и отношений, которое по-разному определяется для

каждой структуры. Например, пусть у нас два элемента управления a и b, и, для простоты, два примитива расположения **vert** и **hor**. Следующие объявления типов на языке OCaml определяют представление расположений:

```
type rels s  = {vert : bool; hor : bool}
type α roles = {a : α; b : α}
type layout  = rels roles roles
```

Этот подход нивелирует накладные расходы на поиск эквивалентных ответов с разными представлениями. Однако он требует специализации функционального верификатора под конкретную структуру. Это не такое уж и большое неудобство, к тому же, улучшения из следующего подраздела, потребуют сходной специализации.

5.3 Представление структуры как набора реляционных отношений

Как уже известно, структура – это набор отношений над элементами управления. В реляционном языке мы можем представлять отношения непосредственно как отношения. Предположим, мы работаем с фрагментом структуры с рис. 7. В оригинальном изложении она представлялась бы как данные, используя соответствующие типы для элементов управления и отношений. Однако, те же самые части структуры можно закодировать непосредственно в OCanren (рис. 8).

type $(A, \text{checkbox})$
type (B, label)
describes (B, A)

Рис. 7. Отношения структуры в абстрактной записи.
Pic. 7. Structure relations in abstract form.

```
let typeᵒ x y = ocanren {
  y ≡ Checkbox ∧ x ≡ A ∨
  y ≡ Label ∧ x ≡ B
}
let describesᵒ x y = ocanren {
  x ≡ B ∧ y ≡ A
}
```

Рис. 8. Реляционное представление отношений структуры.
Pic. 8. A relational representation of the structure relations.

После реляционного преобразования, шаблоны над структурой будут преобразованы в обычную реляционную форму. Это преобразование полностью убирает накладные расходы на интерпретацию для сопоставления структуры. Недостатком является то, что нам необходимо порождать эту часть системы для каждого изменения в структуре. Это естественный компромисс для подходов на основе специализации.

5.4 Жадный алгоритм разрешения конфликтов

Изначальная реализация порождала все возможные совместные множества экземпляров расположения. Но их число огромно, даже если мы рассматриваем только подтверждённые и покрытые. Чтобы сделать синтезатор применимым, нужно устранить эту неэффективность. Воспользуемся жадным подходом: вместо порождения всех подходящих и последующей

фильтрации, построим максимальное несовместное расположение, а затем найдем все конфликты и устраним их путём *недетерминированной* отмены минимального набора правил, которые образуют конфликты. Такой подход даст нам множество всех максимальных совместных расположений. Каждый шаг нами реализован как отдельная реляционная компонента.

На первом шаге мы недетерминировано применяем все правила предписания, чтобы получить максимальное (возможно, несовместное) множество экземпляров расположения, представленное гиперкубом.

На втором шаге воспользуемся особым реляционным верификатором, чтобы найти все конфликты. Для каждого вида конфликта применим специальный верификатор, который по гиперкубу и набору экземпляров расположения, проверяет, что

- множество содержится в гиперкубе;
- оно действительно содержит искомый конфликт.

Запустившись в обратном направлении на конкретном гиперкубе верификатор вернёт множество экземпляров расположения, которое образует конфликт соответствующего вида. Запуск в обратном направлении дизъюнкции верификаторов даст нам все конфликтующие экземпляры.

На третьем шаге мы для каждого конфликтующего расположения разрешаем конфликты, что даст нам (для каждого набора) множество максимальных совместных расположений. Сделаем это следующим образом. Обозначим множество конфликтующих экземпляров как S. Каждый найденный конфликт можно разрешить, убрав только один примитив расположения. Поэтому, будем по одному выкидывать экземпляры из S и смотреть какие конфликты разрешились. Заметим, что выкидывание одного примитива может разрешить сразу несколько конфликтов. Когда не останется ни одного конфликта, мы вернём максимальное совместное расположение. Однако, тут есть одна тонкость, осложняющая реализацию. Удаляя экземпляры, мы можем получить расположение, которое более не подтверждается предписанием, так как некоторые части расположения могут появляться и исчезать только одновременно. Поэтому, надо аккуратно реализовывать процедуру отмены правил. Весь шаг реализован реляционно, из-за многочисленного использования недетерминизма.

Обращаем внимание, что решение задействует три реляционные программы, которые передают результаты друг в друга. Эти программы нельзя объединить в одну, так как запуск следующей требует нахождения всех результатов предыдущей программы.

5.5 Вычисление абсолютных координат

Чтобы привести расположение к окончательному виду, необходимо посчитать координаты всех элементов управления. Для корректного множества экземпляров расположения эта задача сводится к решению задачи *в целочисленных линейных ограничениях*. Каждый примитив расположения добавляет некоторые неравенства в систему неравенств. Например, примитив $\mathbf{hor}\ (C_1, C_2)$ – горизонтальное расположение элементов управления C_1 и C_2 друг за другом – порождает следующие ограничения:

$$C_2.x - C_1.x \le i_H + C_1.width,$$
$$C_2.y - C_1.y = a_v,$$

где *переменные* $C_i.x$ и $C_i.y$ обозначают координаты x и y соответствующего элемента управления, целочисленные *константы* $C_i.width$ – ширину элемента C_i, i_H – горизонтальный отступ, a_v – отступ для вертикального выравнивания.

Также, для каждого виртуального элемента управления C, содержащего $C_1, \dots, C_n$, нужно добавить следующие ограничения на координаты, ширину и высоту:

$$C.x = \min\{C_1.x, \dots, C_n.x\},$$
$$C.y = \min\{C_1.y, \dots, C_n.y\},$$
$$C.x + C.width = \max\{C_i.x + C_i.width\}_{i=1}^{n},$$
$$C.y + C.height = \max\{C_i.y + C_i.height\}_{i=1}^{n}.$$

Дополнительно, нужно добавить неравенства, которые ограничивают максимальные значения переменных на основе размера холста, где мы располагаем элементы управления.

Существует множество способов разрешать линейные целочисленные неравенства, применение SMT решателя для теории линейной арифметики – один из них. Но заманчиво применить реляционный верификатор ещё раз. Этот решатель мог бы быть бесшовно интегрирован в существующую процедуру синтеза, что позволило бы проверять некоторые ограничения на более ранних фазах поиска.

Однако, текущий реляционный решатель показывает низкую производительность в присутствии виртуальных элементов управления. Как было упомянуто выше, размеры виртуальных элементов – не константы, что существенно увеличивает пространство поиска. Поэтому, на данный момент, мы используем Z3 [7], чтобы определять абсолютные координаты элементов управления, а также размеры виртуальных.

Заметим, что данный подход может повредить полноте решателя. Определение конфликтов, а также процедура их устранения из предыдущего раздела, никак не учитывают размеры холста и размеры виртуальных элементов (которые зависят только от размеров, входящих в них элементов). Таким образом, система целочисленных неравенств может быть несовместной даже для корректных расположений. Заметим, что отмена правил предписания влечет за собой отмену целочисленных ограничений, что может восстановить корректность системы. Другими словами, несовместность системы может быть решена также, как и конфликт – отменой правил. Отметим, что вся идея поиска конфликтов основана на предположении, что мы знаем заранее, что некоторые экземпляры расположения рано или поздно создадут несовместные ограничения на координаты. Но на стадии поиска и устранения конфликтов у нас недостаточно информации, чтобы решить эту задачу достаточно точно.

Восстановление полноты требует аккуратного взаимодействия с Z3. В случае невыполнимости системы S мы получаем *минимальное ядро невыполнимости* [21] (англ. unsatisfiable core). Это такое несовместное подмножество $UC \subseteq S$, что удаление одного ограничения из ядра делает систему выполнимой. На первый взгляд, достаточно недетерминировано удалять по одному ограничению из UC, и смотреть из каких частей расположения это ограничение появилось. Отменяя правило предписания, которое внесло это ограничение, мы сделаем систему выполнимой. К сожалению, не со всеми ограничениями можно так обойтись. Разделим UC на ограничения расположения L (порождённые экземплярами), и ограничения на размер Z: появившиеся из-за размера холста и размеров виртуальных элементов. Мы можем непосредственно отменять ограничения из L, но если не только они образуют ядро невыполнимости, то необходимо действовать более тонко.

Построим множество M, выбирая по одному ограничения c из L и запрашивая у Z3 решение системы $S \setminus \{c\}$ на каждом шаге. В итоге может получиться два результата:

(1) Множество $S \setminus \{c\}$ остается невыполнимым. Это означает, что c не влияет на невыполнимость и поиск продолжается.

(2) Множество $S \setminus \{c\}$ стало выполнимым. Значит c в конфликте с каким-то другим ограничением. Добавляем c в M и продолжаем дальше.

После окончания выполнения процедуры множество M будет содержать по построению все ограничения, такие что удаление одного из них восстанавливает выполнимость. Каждое из них появилось из-за какого-то экземпляра расположения. Будем недетерминировано отменять применение правил предписания, которые создали эти экземпляры. Из S мы

получим n подсистем, где $n = |M|$. Затем попробуем решить эти системы с помощью Z3 и для каждой повторим процедуру. В конце концов мы либо отменим достаточно правил, чтобы получить все корректные расположения, либо придём к состоянию, где система невыполнима, а ядро содержит только ограничения на размер, что означает, что элементы не помещаются на наш холст, как их не раскладывай. Это обосновывает полноту алгоритма синтеза. Псевдокод процедуры представлен на рис. 9.

```
solve (lay_part, size_part) {
    // Try to solve the full system
    if (Z3.solve (lay_part ∪ size_part)) {
        // Return a model
        return [ Z3.model ]
    }

    // If unsat we need to find and solve layout conflict
    // MUC is one of conflicts
    muc = Z3.minimal_unsat_core
    // A list of refined models
    answers = []

    for (muc_element ∈ muc) {
        // To solve the conflict we need to remove any layout
        // element of MUC from the system
        if (muc_element ∈ lay_part) {
            reduced_lay_part = lay_part \ {muc_element}
            answers ++= solve (reduced_lay_part, size_part)
        }
    }

    return answers
}
```

Рис. 9. Алгоритм решения конфликтов координат.
Pic. 9. Solving coordinate conflicts algorithm.

6. Реализация и эксперименты

Прототип синтезатора реализован как клиент-серверное приложение[2]. Со стороны клиента мы используем HTML5 и JavaScript для взаимодействия с сервером. На сервере исполняется OCaml приложение, оснащённое OCanren и Z3.

Чтобы облегчить пользователю представление структуры GUI и предписания, мы разработали два текстовых предметно-ориентированных языка. В реализации выразительность языка предписаний расширена: мы разрешаем в правилах использовать пользовательские ограничения, присутствуют конструкции для задания табличных расположений, и т.п. Ни одно из расширений существенно не изменяет подход, описанный в предыдущих главах, и все расширения могут быть учтены. Мы также параметризуем систему набором констант, которые определяют свойства окружения (например, размер холста, где мы располагаем элементы управления). Пример описания структуры для диалогового окна настроек поиска сервиса Google Drive представлена на рис. 10. Текстовое описание предписания слишком длинное, чтобы представить его здесь.

Для оценки нашего прототипа на промышленных примерах мы также реализовали отображение элементов управления с помощью Qt/QML. На рис. 11 можно увидеть два

[2] Прототип доступен по ссылке: https://se.math.spbu.ru/projects/genui (проверено: 26.08.2024).

расположения структуры диалогового окна Google Drive, синтезированного с учетом двух немного различных версий правил проектирования интерфейсов от JetBrains [9] (по которым мы составили два различных предписания), нарисованные с помощью QtQuick Controls[3] и темы оформления Material Design. Предписание регламентирует, что “слишком длинный” текст должен располагаться под меткой. На правом расположении константа, регламентирующая свойство “слишком длинный” была уменьшена. Синтез занял примерно 5 секунд. Некоторые другие примеры можно найти на рис. 12.

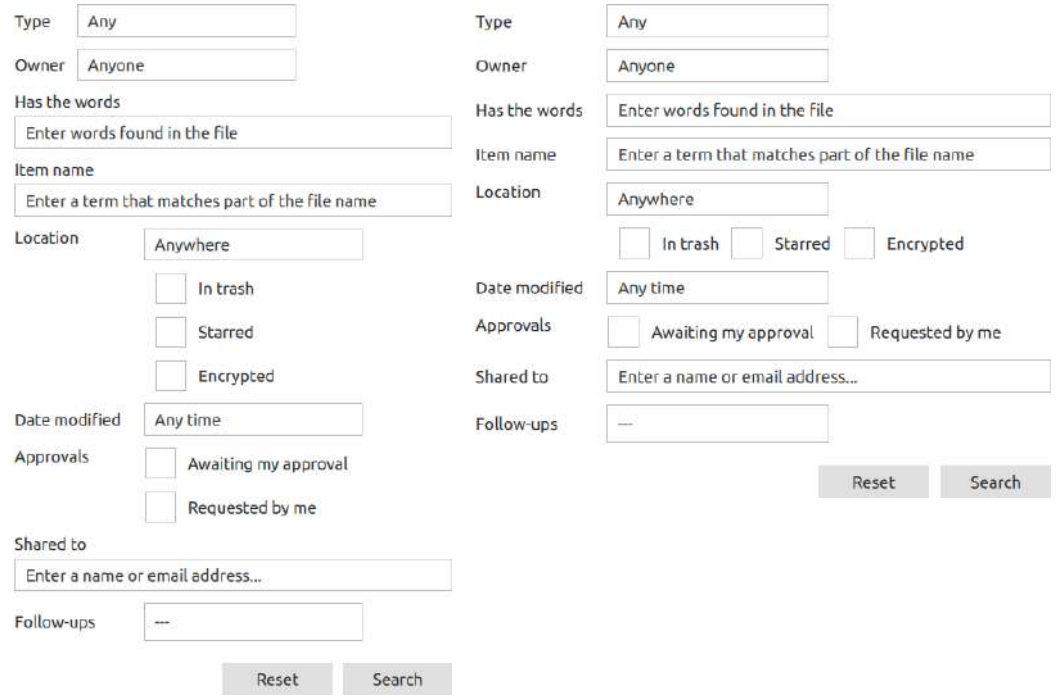
7. Обзор

Дизайн и реализация GUI – горячая тема уже много десятилетий, поэтому можно найти много инструментов, подходов, статей и отчётов по теме. Большая часть из них (если не все) предлагают декларативные и автоматические решения. Но если присмотреться, то их понятия “декларативности” и “автоматизации” отчаются от наших.

```
ordered main_group (
  Label "Type" describes TextEdit "Any" {width=200}
  Label "Owner" describes TextEdit "Anyone" {width=200}
  Label "Has the words" describes
    TextEdit "Enter words found in the file" {width=400}
  Label "Item name" describes
    TextEdit "Enter a term that matches part of the file name" {
      width = 400
    }
  (Label "Location" describes
    (TextEdit "Anywhere" { width = 200 }) dominates
    ordered location_checkboxes (
      Label "In trash" describes CheckBox in_trash_checkbox
      Label "Starred" describes CheckBox starred_checkbox
      Label "Encrypted" describes CheckBox encrypted
    )
  )
  Label "Date modified" describes TextEdit "Any time" {
    width=200
  }
  Label "Approvals" describes
    ordered approvals_checkboxes (
      Label "Awaiting my approval" describes
        CheckBox awaiting_checkbox
      Label "Requested by me" describes
        CheckBox requestred_checkbox
    )
  Label "Shared to" describes
    TextEdit "Enter a name or email address..." {width=400}
  Label "Follow-ups" describes
    TextEdit "---" {width=200, height=35}
)
Button "Search" approves^ main_group
Button "Reset" cancels^ main_group
```

Рис. 10. Описание структуры диалогового окна настроек поиска из сервиса Google Drive.
Pic. 10. Google Drive search settings dialog structure description.

[3] https://doc.qt.io/qt-5/qtquick-controls2-qmlmodule.html (проверено: 26.08.2024).



(a) Если метка описывает "слишком длинный" элемент управления, то располагать их вертикально
(a) If label describes controls which are "too long", then place them vertically

(b) То же самое, но константа для свойства "слишком длинный" уменьшена
(b) The same, but the constant for "too long" was decreased

Рис. 11. Синтезированные расположения элементов для диалогового окна из Google Drive.
Pic. 11. Google Drive search settings dialog layouts.

Во-первых, необходимо упомянуть некоторое количество инструментов для реализации GUI и визуализации данных, например, React [22], Jetpack Compose [23], SwiftUI [24], Streamlit [25], D3 [26] и другие. Они предоставляют пользователю наборы примитивов, которые позволяют отображать данные и пользовательский интерфейс. Например, Streamlit предоставляет набор встроенных примитивов отображения: "колонки", "контейнеры", "модальные диалоги" и т.п. [27], а также множество внешних компонент. Эти примитивы позволяют пользователям абстрагироваться от конкретного вычисления координат и относительного выравнивания. Также они предоставляют разумное поведение при изменении размера холста. Однако, выбор конкретного примитива остаётся на усмотрение разработчика, а не определяется системой. И если расположение элементов должно по какой-то причине измениться, то эти изменения нужно реализовывать вручную. В нашем случае, разработчики не задают конкретное расположение, а только логическую структуру пользовательского интерфейса. Правила проектирования интерфейсов определяют конкретное расположение элементов управления, в зависимости от внешних ограничений, таких как разрешение экрана или настройки региона (например, письмо справа-налево). Пока логическая структура не изменяется, не потребуется никакого взаимодействия с программистом для отображения интерфейса по-другому. С другой стороны, данные инструменты могут использоваться совместно с нашим как способ визуализации интерфейса, так как они предоставляют сходные примитивы расположения. В этой работе мы таким образом задействовали Qt/QML. Программирование в ограничениях уже использовалось для

размещения элементов управления GUI. Одним из примеров реактивного языка программирования в ограничениях является язык Wallingford [28] и система Cassowary [29]. Wallingford позволяет прикреплять ограничения различной силы к значениям в программе. Система реагирует на изменения времени и обновляет значения, не нарушая ограничения. Например, можно реализовать элемент управления с шириной равной синусу текущего времени. Система Cassowary и её потомки позволяют вычислять размеры и положения элементов динамически, например, при изменении размера холста. Она поддерживает множество ограничений, в том числе глубину по оси Z; арифметические операции (например, ширина одного элемента может быть половиной высоты другого); элементы, накладывающиеся друг на друга и т.д. Эта система предназначена для задач автоматической изменения размеров элементов при изменении общего размера холста. Также, она не предоставляет поддержки правил общего вида для корректного позиционирования элементов. Мы сомневаемся в выразимости неоднозначных расположений элементов в данных системах, например, если вертикально или горизонтальное взаимоположение определяется на основе размеров. С другой стороны, количество различных ограничений больше, чем у нас. Например, поддержка накладывающихся друг на друга элементов управления запрещена у нас с самого начала, и такие расположения не создаются вообще.

Рис. 12. Примеры структур (слева) и синтезированных расположений (справа) с учётом правил проектирования интерфейсов от JetBrains.
Pic. 12. Examples of structures (left) and synthesized layouts (right) w.r.t. JetBrains guidelines.

В последние годы появились методы на основе машинного обучения. Некоторые работы ставят гораздо более амбициозные цели, чем наши.

Существует направление исследования, занимающееся порождением кода UI на основе картинок [30]. По полученному от дизайнера рисунку интерфейса, инструмент распознает

элементы управления и их относительное местоположение, и создает код реализации для инструмента создания пользовательских интерфейсов. Подход ортогонален и не совместим с предлагаемым нами. Он требует взаимодействия с дизайнером во время получения каждой части интерфейса, у нас же дизайнер задействован только при описании предписания. Также наш подход позволяет создавать много интерфейсов автоматически. В инструменте [31] была поставлена несколько другая задача: по картинке интерфейса синтезируется его реализация в терминах примитивов Android GUI, масштабируемая между различными устройствами и без типичных ошибок. Для выполнения этой задачи из картинки извлекается элементы управления, их местоположения, а также некоторые ограничения. Эти ограничения напоминают наши примитивы расположения, но специализированы для семантики виджета `ConstraintLayout` [32] из Android. На основе этих ограничений и набору *свойств надёжности*, разработанному авторами, вероятностная модель, обученная на большем наборе существующих интерфейсов, порождает код. Эти реализации, более устойчивы к измерению размера и разрешения экрана, чем полученные только путём распознавания картинок. Любопытно, что авторы мотивируют свою работу, заявляя, что “одно и то же расположение должно отображаться более чем на 15000 устройствах Android с ≈100 различных плотностей пикселей; требовать от программистов разработки и поддержки всех сочетаний входа чрезвычайно нежелательно”. Именно это делает наша система за несколько минут со 100% точностью, поэтому мы считаем свой подход более общим.

Также решалась задача согласованного во всём приложении оформления GUI [33]. Подход основан на идее заполнения: в предположении, что уже есть набор согласованных расположений, решается проблема добавления ещё одного элемента так, чтобы новое расположение было согласовано со старым. Эта задача связана с нашей, так как мы можем рассматривать уже существующее расположение как неявно заданное предписание, но мы можем указать несколько потенциальных недостатков. Во-первых, учитывается только добавление элементов, без удаления. Во-вторых, добавление/удаление компонент не обязательно приводит к “монотонному” изменению расположения: добавление ещё одного поля ввода может существенно поменять расположение. Наконец, начальный согласованный дизайн редко появляется из воздуха, наверняка это результат следования уже существующему предписанию, которое было описано явно.

В системе [34] поставлена гораздо более широкая задача синтеза без предписания, а только на основе эстетических, эргономических и других метрик. Для синтеза используется генетический алгоритм, а качество вычисляется на основе отзывов пользователей. Сам синтез работает часами. Хотя подход потенциально позволяет создавать эстетически приятные интерфейсы без участия дизайнера, задача получения корректного расположения для существенно разных структур обходится стороной.

Встречается интересная задача [35] исследования различных расположений элементов. Она ставит своей целью помочь дизайнерам разрабатывать убедительные и разнообразные расположения элементов управления. Вводится набор ограничений, с помощью которого дизайнеры составляют требования к результату. Любопытно, что в наших терминах этот набор ограничений является смесью структурных ограничений и ограничений на расположение: например, можно указывать и порядок элементов, и выравнивание. По набору ограничений система порождает множество расположений, удовлетворяющих им. Изменяя ограничения дизайнер исследует возможные интерфейсы. Для решения ограничений используется метод ветвей и границ. С ростом числа возможных решений применяются эвристики для отсечения эстетически неподходящих ответов. Несмотря на то, что эта работа похожа на нашу, она нацелена на дизайнеров и может рассматриваться скорее, как средство разработки предписаний, чем как средство получения расположения элементов, удовлетворяющего предписанию.

8. Заключение

Данная работа посвящена синтезу GUI на основе предписания. Наш подход позволяет автоматически располагать элементы управления GUI так, что результат по построению соответствует набору правил, сформулированных дизайнером. Прототип реализации позволяет синтезировать реалистичные примеры промышленных интерфейсов, с учетом промышленных предписаний за разумное время.

Одним из интересных вопросов, является необходимость использования реляционного программирования для решения нашей задачи. Да, набор правил, регламентирующих систему переписывания, в принципе, может быть реализован без использования реляционного подхода. Однако мы считаем, что в этом случае большая часть работы по обоснованию корректности решения должна быть повторена заново. В нашем же случае обоснование тривиально следует из полноты поиска miniKanren и полноты по опровержению (англ. refutational completeness) нашего решения. Мы также предсказываем, что решение потребует изобретения заново недетерминизма и поиска с возвратами, которые уже есть в реляционном программировании. Также дуальность между экземплярами в структуре и реляционными отношениями, изначально не ожидаемая нами, по нашему мнению, демонстрирует, что реляционное программирование – это подходящий подход для решения поставленной задачи.

Список литературы / References

[1]. Haft M., Humm B., Siedersleben J. The Architect's Dilemma – Will Reference Architectures Help? Quality of Software Architectures and Software Quality, 2005, pp. 106-122.

[2]. Ingram S. (2016) The Thumb Zone: Designing for Mobile Users. Smashing Magazine (online). Available at: https://www.smashingmagazine.com/2016/09/the-thumb-zone-designing-for-mobile-users, accessed 30.08.2024.

[3]. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. ISO 9241-210:2019, International Organization for Standardization, 2019. Available at: https://www.iso.org/standard/77520.html.

[4]. Gerber E., Carroll M. The psychological experience of prototyping. Design Studies, vol. 33, isuue 1, 2012, pp 64-84. DOI: 10.1016/j.destud.2011.06.005.

[5]. Friedman D.P., William W.E., Kiselyov O., Hemann J. The Reasoned Schemer. The MIT Press, 2nd edition, Cambridge, USA, 2005, 224 p.

[6]. Lozov P., Verbitskaia E., Boulytchev D. Relational Interpreters for Search Problems. In miniKanren and Relational Programming Workshop, 2019.

[7]. Leonardo de Moura, Bjørner N. Z3: An Efficient SMT Solver. Tools and Algorithms for the Construction and Analysis of Systems, Springer Berlin Heidelberg, 2008, pp. 337-340.

[8]. Bengfort J. Thin vs. Thick vs. Zero Client: What's the Right Fit for Your Business? Online. Available at: https://biztechmagazine.com/article/2018/10/thin-vs-thick-vs-zero-client-whats-right-fit-your-business-perfcon, accessed: 30.08.2024.

[9]. Intellij platform UI guidelines: Layout (online). JetBrains s.r.o., 2000-2022. Available at: https://jetbrains.github.io/ui/principles/layout, accessed: 30.08.2024.

[10]. Kosarev D., Boulytchev D. Typed Embedding of a Relational Language in OCaml. Electronic Proceedings in Theoretical Computer Science, 2016, pp. 1-22.

[11]. Kiselyov O., Chung-chieh Shan, Friedman D.P., Amr S. Backtracking, Interleaving, and Terminating Monad Transformers: (Functional Pearl). In Proceedings of the Tenth ACM SIGPLAN International Conference on Functional Programming, New York, USA, 2005, pp. 192-203.

[12]. Rozplokhas D., Vyatkin A., Boulytchev D. Certified Semantics for Relational Programming. Programming Languages and Systems, APLAS 2020, Lecture Notes in Computer Science, vol. 12470, Springer, Cham, pp. 167-185.

[13]. Comon H. Disunification: A Survey. Computational Logic – Essays in Honor of Alan Robinson. MIT Press, 1991, 322-359.

[14]. Alvis C.E., Willcock J.J., Carter K.M., Byrd W.E., Friedman D.P. cKanren: miniKanren with Constraints. Proceedings of the 2011 Annual Workshop on Scheme and Functional Programming, 2011.

[15]. Byrd W.E., Friedman D.P. αKanren A Fresh Name in Nominal Logic Programming. In Scheme and Functional Programming, 2007.
[16]. Abramov S., Glück R. From Standard to Non-Standard Semantics by Semantics Modifiers. International Journal of Foundations of Computer Science, vol. 12, issue 2, 2001, pp. 171-211. DOI: 10.1142/S0129054101000448.
[17]. Abramov S., Glück R. Combining Semantics with Non-standard Interpreter Hierarchies. FST TCS 2000: Foundations of Software Technology and Theoretical Computer Science, Springer Berlin Heidelberg, 2000, pp. 201-213.
[18]. Byrd W.E., Holk E., Friedman D.P. MiniKanren, Live and Untagged: Quine Generation via Relational Interpreters (Programming Pearl). Proceedings of the Annual Workshop on Scheme and Functional Programming, Association for Computing Machinery, New York, USA, 2012, pp. 8-29.
[19]. Byrd W.E., Ballantyne M., Rosenblatt G., Might M. A Unified Approach to Solving Seven Programming Problems (Functional Pearl). Proceedings of ACM Program. Lang., Association for Computing Machinery, New York, USA, 2017, pp. 8:1-8:26.
[20]. Kosarev D., Lozov P., Boulytchev D. Relational Synthesis for Pattern Matching. Programming Languages and Systems, Springer International Publishing, Cham, 2020, pp.293-310.
[21]. Guthmann O., Strichman O., Trostanetski A. Minimal Unsatisfiable Core Extraction for SMT. 2016 Formal Methods in Computer-Aided Design (FMCAD), Mountain View, CA, USA, 2016, pp. 57-64. DOI: 10.1109/FMCAD.2016.7886661.
[22]. React: A JavaScript Library for Building User Interfaces. Meta Platforms, Inc. Available at: https://reactjs.org/, accessed: 30.08.2024.
[23]. Jetpack Compose. Android Developers. Available at: https://developer.android.com/jetpack/compose, accessed: 30.08.2024.
[24]. SwiftUI. Apple Inc. Available at: https://developer.apple.com/xcode/swiftui, accessed: 30.08.2024.
[25]. Streamlit framework site. Available at: https://docs.streamlit.io, accessed: 30.08.2024.
[26]. The JavaScript library for bespoke data visualization. Available at: https://d3js.org, accessed: 30.08.2024.
[27]. Streamlit layouts and containers. Available at: https://docs.streamlit.io/develop/api-reference/layout, accessed: 30.08.2024.
[28]. Borning A. Wallingford: Toward a Constraint Reactive Programming Language. Companion Proceedings of the 15th International Conference on Modularity, Association for Computing Machinery, New York, NY, USA, 2016, pp. 45-49. DOI: 10.1145/2892664.2892667.
[29]. Badros G.J., Borning A., Stuckey P.J. The Cassowary Linear Arithmetic Constraint Solving Algorithm. ACM Trans. Comput.-Hum. Interact., vol. 8, issue 4, Association for Computing Machinery, New York, NY, USA, 2001, pp. 267-306. DOI: 10.1145/504704.504705.
[30]. Bo Cai, Jian Luo, Zhen Feng. A novel code generator for graphical user interfaces. Scientific Reports, vol. 13, 2023. DOI: 10.1038/s41598-023-46500-6.
[31]. Bielik P., Fischer M., Vechev M. Robust relational layout synthesis from examples for Android. Proc. ACM Program. Lang., vol. 2, Association for Computing Machinery, New York, NY, USA, 2018. DOI: 10.1145/3276526.
[32]. Android ConstraintLayout widget. Accessed: 30.08.2024, available at: https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout.
[33]. Brückner L., Leiva L.A., Oulasvirta A. Learning GUI Completions with User-defined Constraints. ACM Trans. Interact. Intell. Syst., vol. 12, Association for Computing Machinery, New York, NY, USA, 2022. DOI: 10.1145/3490034.
[34]. Shiripour M., Dayama N.R., Oulasvirta A. Grid-based Genetic Operators for Graphical Layout Generation. Proc. ACM Hum.-Comput. Interact., vol. 5, Association for Computing Machinery, New York, NY, USA, 2021. DOI: 10.1145/3461730.
[35]. Swearngin A., Wang C., Oleson A., Fogarty J., Amy J. Ko. Scout: Rapid Exploration of Interface Layout Alternatives through High-Level Design Constraints. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020. Available at: https://api.semanticscholar.org/CorpusID:210177012, accessed: 30.08.2024.

Информация об авторах / Information about authors

Дмитрий Сергеевич КОСАРЕВ является ассистентом кафедры Системного программирования мат-мех. факультета СПбГУ. Его научные интересы включают функциональное программирование, компиляторы и реляционное программирование.

Dmitry Sergeevich KOSAREV is an assistant at the Chair of System Programming of Mathematics and Mechanics Faculty of SPBU. His research interests include functional programming, compilers, and relational programming.

Петр Алексеевич ЛОЗОВ является кандидатом физико-математических наук. Сфера научных интересов: статический анализ, трансляция языков программирования, функционально-реляционное программирование.

Petr Alekseevich LOZOV – Cand. Sci. (Phys.-Math.). Research interests: static analysis, translation of programming languages, functional-relational programming.

Дмитрий Юрьевич БУЛЫЧЕВ – доцент кафедры системного программирования мат.-мех. факультета СПбГУ, кандидат физико-математических наук. Область научных интересов - языки и инструменты программирования, компиляторы, функциональное, логическое и реляционное программирование, анализ и синтез программ.

Dmitry Yuryevich BOULYTCHEV – Cand. Sci. (Phys.-Math.), associate Professor of the System Programming Chair of the Faculty of Mathematics and Mechanics of SPBU. His research interests include programming languages and tools, compilers, functional, logical and relational programming, program analysis and synthesis.

DOI: 10.15514/ISPRAS-2024-36(5)-5

CC BY

Эффективность систем одинаково распределенных конкурирующих процессов при неограниченном и ограниченном параллелизме

П.А. Павлов, ORCID: 0009-0008-1233-7387 <pavlov.p@polessu.by>

Полесский государственный университет,
Республика Беларусь, 225710, г. Пинск, ул. Днепровской флотилии, д. 23.

Аннотация. В статье *с учетом ограниченного числа копий структурированного программного ресурса* проведен сравнительный анализ математических соотношений для вычисления общего времени выполнения множества одинаково распределенных конкурирующих процессов в асинхронном и двух синхронных режимах, в случае неограниченного и ограниченного параллелизма по числу процессоров многопроцессорной системы получено достаточное условие эффективности одинаково распределенной системы, доказано необходимое и достаточное условие существования эффективной системы одинаково распределенных конкурирующих процессов в зависимости от величины дополнительных системных расходов.

Ключевые слова: распределенный процесс; взаимодействующие процессы; программный ресурс; асинхронный (синхронный) режим; неограниченный (ограниченный) параллелизм; эффективность.

Для цитирования: Павлов П.А. Эффективность систем одинаково распределенных конкурирующих процессов при неограниченном и ограниченном параллелизме. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 67–80. DOI: 10.15514/ISPRAS–2024–36(5)–5.

Efficiency of Systems of Identically Distributed Competing Processes with Unlimited and Limited Parallelism

P.A. Pavlov, ORCID: 0009-0008-1233-7387 <pavlov.p@polessu.by>

Polessky State University,
23, Dneprovskoy flotilii st., Pinsk, 225710, Belarus.

Abstract. In the article, taking into account the limited number of copies of a structured software resource, a comparative analysis of mathematical relationships for calculating the total execution time of a set of identically distributed competing processes in asynchronous and two synchronous modes was carried out; in the case of unlimited and limited parallelism by the number of processors of a multiprocessor system, a sufficient condition for the efficiency of an identically distributed system was obtained , a necessary and sufficient condition for the existence of an efficient system of identically distributed competing processes has been proven depending on the amount of additional system costs.

Keywords: distributed process; interacting processes; software resource; asynchronous (synchronous) mode; unlimited (limited) parallelism; efficiency.

For citation: Pavlov P.A. Efficiency of systems of identically distributed competing processes with unlimited and limited parallelism. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024. pp. 67-80 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-5.

1. Введение

Быстрое развитие информационно-коммуникационных и сетевых технологий привело к интенсивному использованию географически распределенных вычислительных ресурсов и созданию на их основе динамически-масштабируемых высокопроизводительных *распределенных вычислительных систем* (РВС), одним из основных преимуществ которых является возможность параллельной обработки процессов [1-3]. При проектировании и создании РВС особую актуальность приобретают задачи построения и исследования математических моделей организации взаимодействия параллельных процессов, конкурирующих за программный ресурс (ПР). Данные задачи имеют как прямой, так и обратный характер. При постановке прямых задач условиями являются значения параметров распределенной вычислительной системы, а решением минимальное общее время реализации заданных объемов вычислений. Постановка обратных задач сводится к поиску критериев эффективности и оптимальности организации выполнения множества распределенных процессов. Случай, когда в общей памяти РВС имеется одна копия ПР, с различных точек зрения был изучен в работах [4-9]. В частности, были решены задачи нахождения минимального общего времени выполнения распределенных конкурирующих процессов, использующих структурированный на блоки ПР в различных режимах взаимодействия процессов, процессоров и блоков, проведен сравнительный анализ режимов взаимодействия процессов, процессоров и блоков, получены критерии эффективности и оптимальности структурирования программных ресурсов, решен ряд оптимизационных задач по расчету числа процессов, процессоров и др. Изучение задач, относящихся к оптимальной организации распределенных параллельных вычислений, приобретает особую актуальность в случае, когда в общей памяти РВС может быть одновременно размещено ограниченное число копий программного ресурса.

2. Математическая модель системы распределенных вычислений при ограниченном числе копий программного ресурса

Как и в работах [4-9] под *взаимодействующими процессами*, т. е. которые влияют на поведение друг друга путем обмена информацией, будем понимать выполнение последовательности наборов блоков $I_s = (1,\ 2,\ \ldots,\ s)$. Многократно выполняемую в многопроцессорной системе программу или ее часть будем называть *программным ресурсом* (ПР), а множество процессов его выполняющим – *конкурирующими.*

Математическая модель системы распределенной обработки взаимодействующих процессов, конкурирующих за использование ограниченного числа копий структурированного программного ресурса, включает в себя [10-12]: $p \geq 2$, процессоров многопроцессорной системы, которые имеют доступ к общей памяти; $n \geq 2$, распределенных взаимодействующих конкурирующих процессов; $s \geq 2$, блоков структурированного на блоки программного ресурса; матрицу $T = [t_{ij}]$, $i = \overline{1,n}$, $j = \overline{1,s}$, времен выполнения блоков программного ресурса распределенными взаимодействующими конкурирующими процессами; $2 \leq c \leq p$, число копий структурированного на блоки программного ресурса, которые могут одновременно находиться в оперативной памяти, доступной для всех p процессоров; $\theta > 0$ – параметр, характеризующий время дополнительных системных расходов, связанных с организацией конвейерного режима использования блоков

структурированного программного ресурса множеством взаимодействующих конкурирующих процессов при распределенной обработке.

Будем также предполагать, что число процессов n кратно числу копий c структурированного программного ресурса, т. е. $n = mc$, где $m = n/c \geq 2$, и что взаимодействие процессов, процессоров и блоков программного ресурса подчинено следующим условиям: 1) ни один из процессоров не может обрабатывать одновременно более одного блока; 2) процессы выполняются в параллельно–конвейерном режиме группами, т. е. осуществляется одновременное (параллельное) выполнение c копий каждого блока в сочетании с конвейеризацией группы из c копий Q_j–го блока, $j = \overline{1,s}$, по процессам и процессорам; 3) обработка каждого блока программного ресурса осуществляется без прерываний; 4) в случае, когда число блоков программного ресурса $s \leq \left[\frac{p}{c}\right]$, где $[x]$ – целая часть числа, для каждого i–го процесса, где $i = c(l-1)+q$, $l = \overline{1,m}$, $q = \overline{1,c}$, распределение блоков Q_j, $j = \overline{1,s}$, программного ресурса по процессорам осуществляется по правилу: блок с номером j распределяется на процессор с номером $c(j-1)+q$.

Далее, как и в случае одной копии программного ресурса [4-9], введем базовые режимы конвейерной реализации взаимодействия процессов, процессоров и блоков ПР, но с учетом наличия c копий программного ресурса, а также определение одинаково распределенной и стационарной систем.

Асинхронной режим взаимодействия процессов, процессоров и блоков структурированного программного ресурса предполагает, что начало выполнения копий очередного Q_j–го блока, $j = \overline{1,s}$, определяется наличием c процессоров и готовностью копий блока к выполнению, при этом программный блок считается готовым к выполнению, если он не выполняется ни на одном из процессоров.

Первый синхронный режим обеспечивает линейный порядок выполнения блоков программного ресурса внутри каждого из процессов без задержек, т. е. в случае, когда $2 \leq s \leq \left[\frac{p}{c}\right]$, момент завершения выполнения Q_j–го блока, $j = \overline{1,s-1}$, процессом с номером $i = (l-1)c+q$, $l = \overline{1,m}$, $q = \overline{1,c}$, на $((j-1)c+q)$–м процессоре совпадает с моментом начала выполнения следующего Q_{j+1}–го блока на процессоре с номером $(jc+q)$.

При **втором синхронном режиме** в случае, когда $2 \leq s \leq \left[\frac{p}{c}\right]$, момент завершения выполнения i–м процессом, где $i = (l-1)c+q$, $l = \overline{1,m-1}$, $q = \overline{1,c}$, j–го блока, $j = \overline{1,s}$, на $((j-1)c+q)$–м процессоре совпадает с моментом начала выполнения j–го блока процессом с номером $(i+c)$ на этом же процессоре, т.е. обеспечивается непрерывное выполнение каждого блока всеми процессами.

О п р е д е л е н и е 1. Систему распределенных конкурирующих процессов будем называть **одинаково распределенной**, если времена выполнения всех блоков ПР каждым из процессов совпадают и равны t_i^{θ}, т. е. справедлива цепочка равенств $t_{i1}^{\theta} = t_{i2}^{\theta} = \ldots = t_{is}^{\theta} = t_i^{\theta}$, для всех $i = \overline{1,n}$.

Будем рассматривать случаи *неограниченного* по числу процессоров распределенной вычислительной системы, т. е. когда $s \le \left[\frac{p}{c}\right]$, и *ограниченного*, когда $s > \left[\frac{p}{c}\right]$, параллелизма.

3. Анализ режимов функционирования одинаково распределенных систем конкурирующих процессов

В [10-12] подробно исследованы базовые асинхронный и два синхронных конвейерных режима взаимодействия распределенных процессов в условиях конкуренции за ограниченное число копий программного ресурса. Для вычисления общего времени выполнения множества распределенных процессов в рамках очерченных режимов с учетом $2 \le c \le p$ копий ПР получены различные математические соотношения. Определенный теоретический и практический интерес представляют задачи сравнительного анализа полученных соотношений. Проведем такой анализ для класса *одинаково распределенных* систем с учетом параметра $\theta > 0$, характеризующего время дополнительных системных расходов, связанных с организацией конвейерного режима использования блоков структурированного программного ресурса множеством конкурирующих процессов при распределенной обработке.

Для проведения сравнительного анализа множество из n процессов разобьем на c подмножеств по m процессов в каждом. В каждое q-е подмножество, $q = \overline{1,c}$, будут включены процессы с номерами $i = c(l-1)+q$, $l = \overline{1,m}$, блоки которых будут выполняться на $(c(j-1)+q)$-х процессорах, $j = \overline{1,s}$. Через $T_q^\theta = \sum_{i=1}^{m} t_{c(i-1)+q}^\theta$ обозначим суммарное время выполнения q-го подмножества процессов, а $t_{\max}^{q+} = \max\limits_{1 \le i \le m} t_{c(i-1)+q}^\theta$ – максимальное время выполнения блока из этого подмножества, $q = \overline{1,c}$.

Определение 2. **Характеристическим** набором одинаково распределенной системы конкурирующих процессов будем называть набор параметров вида $(t_q^\theta,\ t_{c+q}^\theta,\ t_{2c+q}^\theta,...,\ t_{c(m-1)+q}^\theta,\ T_q^\theta)$, $q = \overline{1,c}$.

Пусть $\beta = \left\{ (t_q^\theta,\ t_{c+q}^\theta,...,\ t_{c(m-1)+q}^\theta,\ T_q^\theta) \;\middle|\; T_q^\theta = \sum_{i=1}^{m} t_{c(i-1)+q}^\theta,\ q = \overline{1,c} \right\}$ – множество всех допустимых характеристических наборов системы одинаково распределенных конкурирующих процессов. Выделим из множества β подмножество характеристических наборов вида:

$$S(\beta) = \left\{ \begin{array}{l} (t_q^\theta,\ t_{c+q}^\theta,\ t_{2c+q}^\theta,...,\ t_{c(m-1)+q}^\theta,\ T_q^\theta) \in \beta,\ \textit{где} \\ t_q^\theta \le t_{c+q}^\theta \le t_{2c+q}^\theta \le ... \le t_{c(k-1)+q}^\theta \ge t_{ck+q}^\theta \ge ... \ge t_{c(m-1)+q}^\theta,\ k = \overline{1,m},\ q = \overline{1,c} \end{array} \right\}.$$

Для выделенного подмножества характеристических наборов $S(\beta)$ справедлива

Теорема 1. *В случае неограниченного параллелизма по числу процессоров РВС минимальные общие времена выполнения множества одинаково распределенных конкурирующих процессов в многопроцессорной системе с параметрами* $p \ge 2$, $n \ge 2$, $s \ge 2$, $2 \le c \le p$, $\theta > 0$ *в асинхронном и двух синхронных режимах совпадают, т. е.*

$$T_{ac}^{op}(p,n,s,c,\theta) = T_{1c}^{op}(p,n,s,c,\theta) = T_{2c}^{op}(p,n,s,c,\theta).$$

Доказательство. Для любого характеристического набора одинаково распределенной системы в случае неограниченного параллелизма для асинхронного режима и второго синхронного режима, обеспечивающего непрерывное выполнение каждого блока всеми процессами, минимальное общее время выполнения n одинаково распределенных процессов, конкурирующих за использование c копий структурированного на s блоков программного ресурса в многопроцессорной системе с p процессорами с учетом параметра $\theta > 0$, в том числе и для любого характеристического набора $\mu \in S(\beta)$, вычисляется по формуле [9,10]:

$$T_{ac}^{op}(p,n,s,c,\theta) = T_{2c}^{op}(p,n,s,c,\theta) = \max_{1\le q\le c}(T_q^{\theta} + (s-1)t_{\max}^{q+}).$$

Если же взаимодействие процессов, процессоров и блоков структурированного ПР осуществляется в первом синхронном режиме, при котором обеспечивается выполнение блоков программного ресурса внутри каждого из процессов без задержек, то для любого характеристического набора из β при $2 \le s \le \left[\frac{p}{c}\right]$ выполняется равенство [11]:

$$T_{1c}^{op}(p,n,s,c,\theta) = \max_{1\le q\le c}\left(T_q^{\theta} + (s-1)\left[t_{c(m-1)+q}^{\theta} + \sum_{i=2}^{m}\max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0)\right]\right).$$

Покажем, что $t_{c(m-1)+q}^{\theta} + \sum_{i=2}^{m}\max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) = t_{\max}^{q+}$. Так как $t_{\max}^{q+} = \max_{1\le i\le m} t_{c(i-1)+q}^{\theta}$, $q = \overline{1,c}$, то для всех номеров $1 \le i \le k \le m$ имеем $\sum_{i=2}^{k}\max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) = 0$, а для $1 \le k \le i \le m$ имеет место равенство $\sum_{i=k+1}^{m}\max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) = t_{c(k-1)+q}^{\theta} - t_{c(m-1)+q}^{\theta}$.

Следовательно, $t_{c(m-1)+q}^{\theta} + \sum_{i=2}^{m}\max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) = t_{c(k-1)+q}^{\theta} = t_{\max}^{q+}$. Теорема доказана.

Теорема 2. *В случае неограниченного параллелизма по числу процессоров многопроцессорной системы если допустимый характеристический набор μ одинаково распределенной системы с параметрами $p \ge 2$, $n \ge 2$, $s \ge 2$, $2 \le c \le p$, $\theta > 0$, не принадлежит подмножеству $S(\beta)$, то*

$$T_{1c}^{op}(p,n,s,c,\theta) > T_{ac}^{op}(p,n,s,c,\theta) = T_{2c}^{op}(p,n,s,c,\theta).$$

Доказательство. Так как для асинхронного и второго синхронного режимов минимальные общие времена выполнения одинаково распределенных конкурирующих процессов с учетом ограниченного числа копий программного ресурса равны и определяются по формуле

$$T_{ac}^{op}(p,n,s,c,\theta) = T_{2c}^{op}(p,n,s,c,\theta) = \max_{1\le q\le c}(T_q^{\theta} + (s-1)t_{\max}^{q+}),$$

а для первого синхронного режима по формуле

$$T_{1c}^{op}(p,n,s,c,\theta) = \max_{1\le q\le c}\left(T_q^{\theta} + (s-1)\left[t_{c(m-1)+q}^{\theta} + \sum_{i=2}^{m}\max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0)\right]\right),$$

то доказательство неравенства $T_{1c}^{op}(p,n,s,c,\theta) > T_{ac}^{op}(p,n,s,c,\theta)$ и будет доказательством теоремы.

Доказательство неравенства

$$t_{c(m-1)+q}^{\theta} + \sum_{i=2}^{m} \max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) - t_{\max}^{q+} > 0, \; q = \overline{1,c}, \qquad (1)$$

проведем индукцией по числу процессов $m \geq 2$.

Пусть $m = 2$. В этом случае множество всех допустимых характеристических наборов системы одинаково распределенных конкурирующих процессов $\beta = (t_q^{\theta}, \; t_{c+q}^{\theta}) \in S(\beta)$, $q = \overline{1,c}$.

Пусть неравенство (1) выполняется при $m = k$, покажем, что оно справедливо и при $m = k + 1$. При $m = k + 1$ получим:

$$t_{ck+q}^{\theta} + \sum_{i=2}^{k+1} \max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) - t_{\max}^{q+} > 0, \text{ где } t_{\max}^{q+} = \max_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta}, \; q = \overline{1,c}.$$

Последнее неравенство для всех $q = \overline{1,c}$ равносильно неравенству вида:

$$t_{ck+q}^{\theta} + \sum_{i=2}^{k} \max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) + \max(t_{c(k-1)+q}^{\theta} - t_{ck+q}^{\theta}, 0) - \max_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta} > 0.$$

Если $\max\limits_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta}$ достигается при $i = k+1$, т. е. $\max\limits_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta} = t_{ck+q}^{\theta}$, то

$$t_{ck+q}^{\theta} + \sum_{i=2}^{k} \max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) + \max(t_{c(k-1)+q}^{\theta} - t_{ck+q}^{\theta}, 0) - t_{ck+q}^{\theta} =$$

$$= \sum_{i=2}^{k} \max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) + \max(t_{c(k-1)+q}^{\theta} - t_{ck+q}^{\theta}, 0) > 0.$$

Здесь первое слагаемое больше нуля, ибо в противном случае $\mu \in S(\beta)$, что противоречит условию теоремы, а второе слагаемое равно нулю, так как $t_{ck+q}^{\theta} \geq t_{c(k-1)+q}^{\theta}$.

Если значение $\max\limits_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta}$ находится в промежутке $1 \leq i \leq k$, то имеем:

$$t_{ck+q}^{\theta} + \sum_{i=2}^{k} \max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) + \max(t_{c(k-1)+q}^{\theta} - t_{ck+q}^{\theta}, 0) - \max_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta} =$$

$$= t_{c(k-1)+q}^{\theta} + \sum_{i=2}^{k} \max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) - \max_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta} +$$

$$+ t_{ck+q}^{\theta} - t_{c(k-1)+q}^{\theta} + \max(t_{c(k-1)+q}^{\theta} - t_{ck+q}^{\theta}, 0).$$

Здесь $t_{c(k-1)+q}^{\theta} + \sum\limits_{i=2}^{k} \max(t_{c(i-2)+q}^{\theta} - t_{c(i-1)+q}^{\theta}, 0) - \max\limits_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta} > 0$ по индукционному предположению и в силу того, что $\max\limits_{1 \leq i \leq k+1} t_{c(i-1)+q}^{\theta} = \max\limits_{1 \leq i \leq k} t_{c(i-1)+q}^{\theta}$.

Покажем далее, что $t_{ck+q}^{\theta} - t_{c(k-1)+q}^{\theta} + \max(t_{c(k-1)+q}^{\theta} - t_{ck+q}^{\theta}, 0) \geq 0$:

– если $t^{\theta}_{c(k-1)+q} = t^{\theta}_{ck+q}$ равенство нулю очевидно;

– если $t^{\theta}_{c(k-1)+q} > t^{\theta}_{ck+q}$ получим, что

$$t^{\theta}_{ck+q} - t^{\theta}_{c(k-1)+q} + \max(t^{\theta}_{c(k-1)+q} - t^{\theta}_{ck+q}, 0) = t^{\theta}_{ck+q} - t^{\theta}_{c(k-1)+q} + t^{\theta}_{c(k-1)+q} - t^{\theta}_{ck+q} = 0;$$

– если $t^{\theta}_{c(k-1)+q} < t^{\theta}_{ck+q}$, имеем, что $\max(t^{\theta}_{c(k-1)+q} - t^{\theta}_{ck+q}, 0) = 0$, тогда

$$t^{\theta}_{ck+q} - t^{\theta}_{c(k-1)+q} + \max(t^{\theta}_{c(k-1)+q} - t^{\theta}_{ck+q}, 0) = t^{\theta}_{ck+q} - t^{\theta}_{c(k-1)+q} > 0,$$

что и требовалось доказать.

4. Эффективность одинаково распределенных систем конкурирующих процессов при неограниченном параллелизме

В п.3 доказано, что в случае неограниченного параллелизма по числу процессоров распределенной вычислительной системы минимальные общие времена выполнения n *одинаково распределенных* процессов, конкурирующих за использование c копий структурированного на s блоков программного ресурса в многопроцессорной системе с p процессорами с учетом параметра $\theta > 0$ в асинхронном и двух синхронных режимах совпадают и вычисляются по формуле (теорема 1):

$$T^{op}(p,n,s,c,\theta) = \max_{1 \le q \le c}(T^{\theta}_{q} + (s-1)t^{q+}_{\max}), \tag{2}$$

где $T^{\theta}_{q} = \sum_{i=1}^{n/c} t^{\theta}_{c(i-1)+q}$ – время выполнения q–го подмножества процессов, $t^{q+}_{\max} = \max_{1 \le i \le n/c} t^{\theta}_{c(i-1)+q}$ – максимальное время выполнения блока из этого подмножества с учетом системных расходов θ, $q = \overline{1,c}$.

Определение 3. Одинаково распределенная система конкурирующих процессов называется **стационарной**, если $t^{\theta}_1 = t^{\theta}_2 = \ldots = t^{\theta}_n = t^{\theta}$.

В случае *стационарной* одинаково распределенной системы конкурирующих процессов минимальные общие времена выполнения определяются равенством [9-11]:

$$T^{cc}(p,n,s,c,\theta) = \left(\frac{n}{c} + s - 1\right)t^{\theta}, \text{ где } t^{\theta} = t + \theta.$$

Определение 4. Одинаково распределенную систему конкурирующих процессов будем называть **эффективной** при фиксированных $p \ge 2$, $s \ge 2$, $c \ge 2$, $\theta > 0$, если выполняется соотношение $\Delta_{op} = \max_{1 \le q \le c} sT_q - T^{op}(p,n,s,c,\theta) \ge 0$, где sT_q – время выполнения s блоков программного ресурса всеми процессами q-го подмножества в последовательном режиме, а $T_q = \sum_{i=1}^{n/c} t_{c(i-1)+q}$.

При наличии двух эффективных одинаково распределенных систем конкурирующих процессов будем считать, что первая не менее эффективная, чем вторая, если $\Delta^1_{op} \ge \Delta^2_{op}$.

Для введенного подмножества одинаково распределенных систем справедливо следующее утверждение.

Теорема 3. *Для любой эффективной одинаково распределенной системы конкурирующих процессов при* $s \le \left[\frac{p}{c}\right]$ *и* $\theta > 0$ *существует более эффективная стационарная одинаково распределенная система.*

Доказательство. Рассмотрим любую эффективную одинаково распределенную систему. Согласно определению 4 условие ее эффективности с учетом (2) можно записать в виде следующего неравенства:

$$\Delta_{op} = \max_{1\le q\le c} sT_q - T^{op}(p,n,s,c,\theta) = \max_{1\le q\le c}\left(sT_q - T_q^{\theta} - (s-1)t_{\max}^{q+}\right) =$$

$$= (s-1)\max_{1\le q\le c}(T_q - t_{\max}^q) - \left(\frac{n}{c} + s - 1\right)\theta \ge 0, \text{ где } t_{\max}^q = \max_{1\le i\le n/c} t_{c(i-1)+q}. \qquad (3)$$

Для любой *стационарной* одинаково распределенной системы условие эффективности запишется следующим образом:

$$\Delta_{cc} = sT^m - T^{cc}(p,n,s,c,\theta) = sT^m - \left(\frac{n}{c} + s - 1\right)t^{\theta} =$$

$$= (s-1)(T^m - t) - \left(\frac{n}{c} + s - 1\right)\theta \ge 0, \text{ где } T^m = \frac{n}{c}t. \qquad (4)$$

Для доказательства теоремы достаточно доказать выполнение неравенства $\Delta_{op} \le \Delta_{cc}$.

Подставив вместо Δ_{op} и Δ_{cc} выражения из (3) и (4), получим:

$$\max_{1\le q\le c}(T_q - t_{\max}^q) \le \left(\frac{n}{c} - 1\right)t.$$

Докажем справедливость последнего неравенства. Рассмотрим стационарную одинаково распределенную систему конкурирующих процессов, в которой $t = \max_{1\le i\le n} t_i = t_{\max}^q$. Пусть для определенности $t_{\max}^q = t_k$, тогда справедлива цепочка соотношений:

$$\max_{1\le q\le c}(T_q - t_{\max}^q) = \max_{1\le q\le c}\left(\sum_{i=1}^{k-1} t_{c(i-1)+q} + \sum_{i=k+1}^{n/c} t_{c(i-1)+q}\right) \le \left(\frac{n}{c} - 1\right)t_{\max}^q = \left(\frac{n}{c} - 1\right)t.$$

Теорема доказана.

Следующее утверждение в случае неограниченного параллелизма по числу процессоров распределенной вычислительной системы с учетом ограниченного числа копий структурированного программного ресурса определяет достаточное условие эффективности одинаково распределенной системы.

Теорема 4. *Система одинаково распределенных конкурирующих процессов с параметрами* p, n, s, c, θ, *удовлетворяющими соотношениям*

$$3 \le s \le \left[\frac{p}{c}\right],\ s = \frac{n}{c} \ne 3,\ ns \ge 2[n + c(s-1)],\ 0 < \theta \le t_{\min} = \min_{1\le i\le n} t_i,$$

является эффективной.

Доказательство. Согласно формуле (3), условие эффективности равносильно неравенству

$$\max_{1\le q\le c}\frac{T_q - t_{\max}^q}{\theta} \ge \frac{n + c(s-1)}{c(s-1)}. \qquad (5)$$

Следовательно, для доказательства теоремы достаточно убедиться в справедливости неравенства (5). В силу выбора $0 < \theta \le t_{\min}$ будет выполняться неравенство $t_{\min}/\theta \ge 1$, тогда справедлива цепочка

$$\max_{1\le q\le c} \frac{T_q - t_{\max}^q}{\theta} \ge \left(\frac{n}{c} - 1\right)\frac{t_{\min}}{\theta} \ge \frac{n}{c} - 1. \tag{6}$$

Далее, из $ns \ge 2[n + c(s-1)]$ следует справедливость неравенства

$$\frac{n}{c} - 1 \ge \frac{n + c(s-1)}{c(s-1)}. \tag{7}$$

Следствием неравенств (6) и (7) является неравенство (5). Таким образом, теорема доказана.

Ниже формулируется и доказывается необходимое и достаточное условие существования эффективной системы одинаково распределенных конкурирующих процессов при неограниченном параллелизме с учетом c копий программного ресурса в зависимости от величины дополнительных системных расходов θ.

Т е о р е м а 5. *Для существования эффективной одинаково распределенной системы конкурирующих процессов с заданными параметрами* $\left[\frac{p}{c}\right] \ge 3$, $2 \le s \le \left[\frac{p}{c}\right]$, $2 \le c \le p$ *и* $\theta > 0$ *необходимо и достаточно выполнение следующих условий:*

$$\theta \le \begin{cases} \varphi[c(1+\sqrt{s})], & \text{если } \sqrt{s} \text{ целое,} \\ \max\{\varphi[c(1+\sqrt{s})],\ \varphi[c(2+\sqrt{s})]\}, & \text{если } \sqrt{s} \text{ нецелое,} \end{cases} \tag{8}$$

где $\varphi(x) = \dfrac{c(s-1)T^m(x-c)}{x^2 + xc(s-1)}$, *а* $[x]$ *– наибольшее целое, не превосходящее* x.

Д о к а з а т е л ь с т в о. Согласно (4), условие эффективности любой стационарной одинаково распределенной системы конкурирующих процессов определяется соотношением

$$\Delta_{cc} = (s-1)(T^m - t) - \left(\frac{n}{c} + s - 1\right)\theta \ge 0, \text{ где } T^m = \frac{n}{c}t,$$

которое равносильно выполнению неравенства

$$\theta \le \frac{c(s-1)T^m(n-c)}{n^2 + nc(s-1)}. \tag{9}$$

Введем в рассмотрение функцию $\varphi(x) = \dfrac{c(s-1)T^m(x-c)}{x^2 + xc(s-1)}$. В силу того, что $\varphi'(x) = \dfrac{c(s-1)T^m[-x^2 + 2cx + c^2(s-1)]}{[x^2 + xc(s-1)]^2}$, то функция φ при $x > 0$ достигает своего максимального значения в точке $x = c(1+\sqrt{s})$. Положим

$$n_э = \begin{cases} c(1+\sqrt{s}), & \text{если } \sqrt{s} \text{ — целое,} \\ \max\{c(1+[\sqrt{s}]),\ c(2+[\sqrt{s}])\}, & \text{если } \sqrt{s} \text{ — нецелое.} \end{cases} \tag{10}$$

Необходимость условий (8) будет доказана, если будет установлена невозможность существования эффективной одинаково распределенной системы n конкурирующих процессов, для которой выполнялось бы неравенство вида:

$$\theta > \frac{c(s-1)T^m(n-c)}{n^2+nc(s-1)}. \quad (11)$$

Очевидно, такой системы нет при $n = n_э$, так как в силу определения функции φ, для такого n выполняется неравенство (11) и, следовательно, такая система не может быть эффективной.

Если все же предположить существование такой системы с n процессами, то должно выполняться соотношение $n \neq n_э$.

Выше установлено, что одинаково распределенная система с $n_э$ процессами эффективна, следовательно, для нее имеет место неравенство $\theta \le \frac{c(s-1)T^m(n_э-c)}{n_э^2+n_эc(s-1)}$. В то время как для гипотетической системы с n процессами в силу предположения должно выполняться неравенство $\theta > \frac{c(s-1)T^m(n-c)}{n^2+nc(s-1)}$. Очевидным следствием полученных неравенств, является неравенство $\theta > \theta$.

В случае $n < n_э$ в силу (11) должна выполняться цепочка неравенств

$$\frac{c(s-1)T^m(n_э-c)}{n_э^2+n_эc(s-1)} < \frac{c(s-1)T^m(n-c)}{n^2+nc(s-1)} < \theta,$$

из которой в силу (9) следует неэффективность одинаково распределенной системы с $n_э$ процессами.

Наконец, если $n > n_э$, то $\frac{c(s-1)T^m(n_э-c)}{n_э^2+n_эc(s-1)} > \frac{c(s-1)T^m(n-c)}{n^2+nc(s-1)} \ge \theta$, что указывает на эффективность гипотетической одинаково распределенной системы n конкурирующих процессов. Полученные противоречия во всех возможных случаях доказывают необходимость условий (8).

Достаточность условий (8) непосредственно следует из наличия функции φ со свойством (9). Действительно, в этом случае требуемой эффективной одинаково распределенной системы является система с $n = n_э$ конкурирующими процессами, где $n_э$ определяется формулой (10). Теорема доказана.

З а м е ч а н и е. При $\left[\frac{p}{c}\right] = s = 2$ одинаково распределенная система конкурирующих процессов будет эффективной, если выполняется неравенство $\frac{\theta}{T_m} \le \frac{c(n-c)}{n(n+c)}$.

5. Эффективность одинаково распределенных систем в условиях ограниченного параллелизма

В случае ограниченного параллелизма для вычисления минимального общего времени выполнения одинаково распределенных конкурирующих процессов в асинхронном и втором синхронном режимах имеют место формулы [10-12]:

$$T_{ac}^{op}(p,n,s,c,\theta) = T_{2c}^{op}(p,n,s,c,\theta) =$$

$$= \max_{1\le q\le c}\begin{cases} kT_q^{\theta} + \left(\left[\frac{p}{c}\right]-1\right)t_{\max}^{q+}, \ \textit{при}\ s = k\left[\frac{p}{c}\right],\ k>1,\ T_q^{\theta} > \left[\frac{p}{c}\right]t_{\max}^{q+}, \\ (k+1)T_q^{\theta} + (r-1)t_{\max}^{q+}, \ \textit{при}\ s = k\left[\frac{p}{c}\right]+r,\ k\ge 1,\ 1\le r < \left[\frac{p}{c}\right],\ T_q^{\theta} > \left[\frac{p}{c}\right]t_{\max}^{q+}. \end{cases}$$

Здесь $T_q^{\theta} = \sum_{i=1}^{n/c} t_{c(i-1)+q}^{\theta}$, $t_{\max}^{q+} = \max\limits_{1\le i\le n/c} t_{c(i-1)+q}^{\theta}$, $q = \overline{1,c}$.

Т е о р е м а 6. *Если параметры системы* $\frac{n}{c}\ge 3$ *одинаково распределенных конкурирующих процессов в многопроцессорной системе с* p *процессорами удовлетворяют соотношениям* $s\ge 3$, $s=\frac{n}{c}\ne 3$, $2\le c\le p$, $0<\theta\le t_{\min} = \min\limits_{1\le i\le n} t_i$ *и* $T_q^{\theta} > \left[\frac{p}{c}\right]t_{\max}^{q+}$, *то в случае ограниченного параллелизма рассматриваемая система будет эффективной, если выполняются условия:*

$$k\frac{n}{c}\left[\frac{p}{c}\right] \ge \begin{cases} 2\left(k\frac{n}{c}+\left[\frac{p}{c}\right]-1\right), & s = k\left[\frac{p}{c}\right],\ k>1, \\ \frac{n}{c}[2(k+1)-r]+2(r-1), & s = k\left[\frac{p}{c}\right]+r,\ k\ge 1,\ 1\le r<\left[\frac{p}{c}\right]. \end{cases}$$

Д о к а з а т е л ь с т в о. Для случая $s = k\left[\frac{p}{c}\right]$, $k>1$, условие эффективности одинаково распределенной системы конкурирующих процессов запишется в виде:

$$\Delta_{op}^{ac,2c}\left(s = k\left[\frac{p}{c}\right]\right) = \max_{1\le q\le c} k\left[\frac{p}{c}\right]T_q - T_{ac,2c}^{op}\left(p,n,k\left[\frac{p}{c}\right],c,\theta\right) =$$

$$= \left(\left[\frac{p}{c}\right]-1\right)\max_{1\le q\le c}(kT_q - t_{\max}^{q}) - \left(k\frac{n}{c}+\left[\frac{p}{c}\right]-1\right)\theta \ge 0,\ \text{где}\ T_q = \sum_{i=1}^{n/c} t_{c(i-1)+q},\ t_{\max}^{q} = \max_{1\le i\le n/c} t_{c(i-1)+q}$$

Или

$$\left(\left[\frac{p}{c}\right]-1\right)\max_{1\le q\le c}\frac{kT_q - t_{\max}^{q}}{\theta} \ge k\frac{n}{c}+\left[\frac{p}{c}\right]-1. \qquad (12)$$

В силу того, что $0<\theta\le t_{\min}$, получим:

$$\max_{1\le q\le c}\frac{kT_q - t_{\max}^{m}}{\theta} \ge \left(k\frac{n}{c}-1\right)\frac{t_{\min}}{\theta} \ge k\frac{n}{c}-1. \qquad (13)$$

Из (13) и (12) следует первая формула теоремы.

В случае, когда $s = k\left[\frac{p}{c}\right]+r$, $k\ge 1$, $1\le r<\left[\frac{p}{c}\right]$, условие эффективности будет иметь вид:

$$k\left(\left[\frac{p}{c}\right]-1\right)\max_{1\le q\le c}\frac{T_q}{\theta} + (r-1)\max_{1\le q\le c}\frac{T_q - t_{\max}^{q}}{\theta} \ge (k+1)\frac{n}{c}+r-1. \qquad (14)$$

Так как, $t_{\min}/\theta\ge 1$, то

$$k\left(\left[\frac{p}{c}\right]-1\right)\max_{1\le q\le c}\frac{T_q}{\theta}+(r-1)\max_{1\le q\le c}\frac{T_q-t_{\max}^q}{\theta}\ge k\frac{n}{c}\left(\left[\frac{p}{c}\right]-1\right)+\left(\frac{n}{c}-1\right)(r-1). \qquad (15)$$

Из (15) и (14) следует вторая формула теоремы. Теорема доказана.

Сформулируем и докажем необходимое и достаточное условие существования эффективной системы одинаково распределенных конкурирующих процессов в зависимости от величины накладных расходов θ в случае ограниченного параллелизма по числу процессоров РВС.

Теорема 7. *Для существования эффективной одинаково распределенной системы конкурирующих процессов с заданными параметрами* $\left[\frac{p}{c}\right]\ge 3$, $T^m=\frac{n}{c}t$, $2\le c\le p$, $\theta>0$ *необходимо и достаточно, чтобы выполнялись следующие условия:*

1) *при* $s=k\left[\frac{p}{c}\right]$, $k>1$,

$$\theta\le\begin{cases}\varphi_1\left[\frac{c}{k}\left(1+\sqrt{\left[\frac{p}{c}\right]}\right)\right], & \text{если } \frac{c}{k}\left(1+\sqrt{\left[\frac{p}{c}\right]}\right) \text{ – целое,}\\ \max\left\{\varphi_1\left(\left[\frac{c}{k}\left(1+\sqrt{\left[\frac{p}{c}\right]}\right)\right]\right),\ \varphi_1\left(\left[\frac{c}{k}\left(1+\sqrt{\left[\frac{p}{c}\right]}\right)\right]+1\right)\right\}, & \text{в противном случае,}\end{cases}$$

где $\varphi_1(x)=\frac{aT^m c(kx-c)}{kx^2+acx}$, $a=\left[\frac{p}{c}\right]-1$;

2) *при* $s=k\left[\frac{p}{c}\right]+r$, $k\ge 1$, $1\le r<\left[\frac{p}{c}\right]$,

$$\theta\le\begin{cases}\varphi_2(x), & \text{если } x \text{ – целое,}\\ \max\{\varphi_2([x]),\ \varphi_2([x]+1)\}, & \text{если } x \text{ – нецелое,}\end{cases}$$

где $\varphi_2(x)=\frac{cT^m[akx+b(x-c)]}{(k+1)x^2+bcx}$, $x=\frac{bc}{ak+b}\left(1+\sqrt{1+\frac{ak+b}{k+1}}\right)$, $b=r-1$.

Доказательство. В случае *стационарной* одинаково распределенной системы конкурирующих процессов во всех трех режимах минимальное общее время с учетом параметра $\theta>0$ определяется по формулам:

$$T^{cc}(p,n,s,c,\theta)=$$

$$=\begin{cases}\left(k\frac{n}{c}+\left[\frac{p}{c}\right]-1\right)t^\theta, & \text{если } \min(m,s)>\left[\frac{p}{c}\right] \text{ и } s=k\left[\frac{p}{c}\right],\ k>1,\\ \left((k+1)\frac{n}{c}+r-1\right)t^\theta & \text{если } \min(m,s)>\left[\frac{p}{c}\right] \text{ и } s=k\left[\frac{p}{c}\right]+r,\ k\ge 1,\ 1\le r<\left[\frac{p}{c}\right].\end{cases}$$

Условие эффективности стационарной одинаково распределенной системы конкурирующих процессов в случае $s=k\left[\frac{p}{c}\right]$, $k>1$, определяется соотношением

$$\Delta_{cc}\left(s=k\left[\frac{p}{c}\right]\right)=k\left[\frac{p}{c}\right]T^m-T^{cc}\left(p,n,k\left[\frac{p}{c}\right],c,\theta\right)=$$

$$=\left(\left[\frac{p}{c}\right]-1\right)(kT^m-t)-\left(k\frac{n}{c}+\left[\frac{p}{c}\right]-1\right)\theta\geq 0,$$

которое равносильно выполнению неравенства $\theta\leq\frac{aT^m c(kn-c)}{kn^2+acn}$, где $a=\left[\frac{p}{c}\right]-1$.

Введем в рассмотрение функцию $\varphi_1(x)=\frac{aT^m c(kx-c)}{kx^2+acx}$, которая при $x>0$ достигает своего максимума в точке $x=\frac{c}{k}\left(1+\sqrt{\left[\frac{p}{c}\right]}\right)$.

В случае $s=k\left[\frac{p}{c}\right]+r$, $k\geq 1$, $1\leq r<\left[\frac{p}{c}\right]$, условие эффективности стационарной одинаково распределенной системы конкурирующих процессов определяется неравенством:

$$\Delta_{cc}\left(s=k\left[\frac{p}{c}\right]+r\right)=akT^m+b(T^m-t)-\left((k+1)\frac{n}{c}+b\right)\theta\geq 0,\text{ где } b=r-1.$$

С учетом того, что $t=\frac{c}{n}T^m$, это равносильно $\theta\leq\frac{cT^m[akn+b(n-c)]}{(k+1)n^2+bcn}$.

Рассмотрим функцию $\varphi_2(x)=\frac{cT^m[akx+b(x-c)]}{(k+1)x^2+bcx}$. В силу того, что $\varphi_2'(x)=\frac{cT^m[(-ak-b)(k+1)x^2+2bc(k+1)x+b^2c^2]}{[(k+1)x^2+bcx]^2}=0$, функция $\varphi_2(x)$ при $x>0$ достигает своего максимума в точке $x=\frac{bc}{ak+b}\left(1+\sqrt{1+\frac{ak+b}{k+1}}\right)$. Теорема доказана.

6. Заключение

В данной работе для класса *одинаково распределенных систем конкурирующих процессов с учетом ограниченного числа копий программного ресурса* проведен сравнительный анализ математических соотношений времен выполнения множества процессов в асинхронном и двух синхронных режимах, в случаях неограниченного и ограниченного параллелизма по числу процессоров PBC получены достаточные условия эффективности одинаково распределенных систем, доказываются необходимые и достаточные условия существования эффективных систем одинаково распределенных конкурирующих процессов в зависимости от величины дополнительных системных расходов. Полученные условия эффективности имеют многочисленные области применения, в частности, они могут быть использованы при проектировании системного и прикладного программного обеспечения, ориентированного на многопроцессорные вычислительные системы и комплексы, а также при решении проблем *оптимального* использования вычислительных ресурсов.

Список литературы / References

[1]. Andrew S. Tanenbaum, Maarten Van Steen. Distributed Systems. Amazon Digital Services LLC, 2023. 684 p.

[2]. Robey R., Zamora Y. Parallel and High Performance Computing. Manning, 2021. 800 p.

[3]. Бабичев С.Л., Коньков К.А. Распределенные системы. М.: Юрайт, 2019. 507 с. / Babichev S., Konkov K. Distributed Systems. Moscow, Juright, 2019, 507 p. (in Russian).

[4]. Павлов П.А. Оптимальность систем одинаково распределенных конкурирующих процессов. Вестник БГУ. Серия 1: Физика. Математика. Информатика. 2009, №3. С. 114-118. / Pavlov P. Optimality of systems of identically distributed competing processes. BSU Bulletin. Series 1: Physics. Mathematics. Computer Science. 2009, №3. pp. 114-118 (in Russian).

[5]. Павлов П.А., Коваленко Н.С. Математическое моделирование параллллельных процессов. Germany: Lambert Academic Publishing, 2011. 246 с. / Pavlov P., Kovalenko N. Mathematical modeling of parallel processes. Germany, Lambert Academic Publishing, 2011, 246 p. (in Russian).

[6]. Павлов П.А. Оптимальность структурирования программных ресурсов при конвейерной распределенной обработке. Программные продукты и системы. 2010, №3. С. 79-85. / Pavlov P. Optimal structuring of software resources during pipeline distributed processing. Software products and systems. 2010, №3. pp. 79-85 (in Russian).

[7]. Павлов П.А. Задача оптимизации числа процессоров при распределенной обработке. Вестник государственного Самарского аэрокосмического университета имени академика С.П. Королева. 2011, №4. С. 230-240. / Pavlov P. The problem of optimizing the number of processors in distributed processing. Bulletin of the State Samara Aerospace University named after Academician S.P. Koroleva. 2011, №4. pp. 230-240 (in Russian).

[8]. Pavlov P.A. The optimality of software resources structuring through the pipeline distributed processing of competitive cooperative processes. International Journal of Multimedia Technology (IJMT). 2012, Vol.2, №1. pp. 5–10.

[9]. Kovalenko N.S., Pavlov P.A. Optimal Grouping Algorithm of Identically Distributed Systems. Programming and Computer Software. 2012, №3. pp. 143-150.

[10]. Павлов П.А., Коваленко Н.С. Распределенные вычисления при ограниченном числе копий программного ресурса. Программные продукты и системы. 2011, №4. С. 155-163. / Pavlov P., Kovalenko N. Distributed computing with a limited number of copies of a software resource. Software products and systems. 2011, №4. pp. 155-163 (in Russian).

[11]. Kovalenko N.S., Pavlov P.A., Ovseec M.I. Asynchronous distributed computations with a limited number of copies of a structured program resource. Cybernetics and systems analysis. 2012, №1. pp. 86-98.

[12]. Павлов П.А., Коваленко Н.С. Синхронный режим распределенных вычислений при непрерывном выполнении блоков ограниченного числа копий программного ресурса. Программные продукты и системы. 2024, №1. С. 43-53. / Pavlov P., Kovalenko N. Synchronous mode of distributed computing with continuous execution of blocks of a limited number of copies of a software resource. Software products and systems. 2024, №1. pp. 43-53 (in Russian).

Информация об авторах / Information about authors

Павел Александрович ПАВЛОВ – кандидат физико-математических наук, доцент, доцент кафедры информационных технологий и интеллектуальных систем Полесского государственного университета. Сфера научных интересов: математическое моделирование распределенных вычислительных систем конкурирующих процессов, исследование операций, математическое программирование.

Pavel Alexsandrovich PAVLOV – Cand. Sci. (Phys.-Math.), Associate Professor of the department of information technologies and intelligent systems (Polessky State University). Research interests: mathematical modeling of distributed computing systems of competing processes, operations research, mathematical programming.

DOI: 10.15514/ISPRAS-2024-36(5)-6

CC BY

Программная среда выполнения методических прикладных тестов для численного исследования параметров высокопроизводительных вычислительных систем

А.О. Игнатьев, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>
С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>

ФГУП «Российский Федеральный Ядерный Центр –
ВНИИ технической физики имени академика Е.И. Забабахина»,
456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13.

Аннотация. Неотъемлемой частью процесса создания высокопроизводительных вычислительных систем, предназначенных для решения задач численного моделирования различных физических процессов, является проверка их соответствия на заявленные при их проектировании характеристики. В статье рассматривается разработанная авторами программная среда выполнения методических прикладных тестов для численного исследования параметров высокопроизводительных вычислительных систем, позволяющая эффективно анализировать результаты выполнения прикладных тестов и выполнять оценку производительности и надежности вычислительных систем.

Ключевые слова: вычислительная система; высокопроизводительные вычисления; математическое моделирование; тестирование высокопроизводительных вычислительных систем.

Для цитирования: Игнатьев А.О., Мокшин С.Ю. Программная среда выполнения методических прикладных тестов для численного исследования параметров высокопроизводительных вычислительных систем. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 81–92. DOI: 10.15514/ISPRAS–2024–36(5)–6.

Runtime Environment for Conducting Methodical Applied Tests to Numerically Investigate of HPC Systems Parameters

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <*a.o.ignatyev@vniitf.ru*>
S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <*s.yu.mokshin@vniitf.ru*>

Russian Federal Nuclear Center –
Zababakhin All-Russian Research Institute of Technical Physics,
456770, Russia, Snezhinsk, Chelyabinsk region, Vasilieva street, 13.

Abstract. An integral part of the process of creating high-performance computing systems designed to solve problems of numerical modeling of various physical processes is to check their compliance with the characteristics stated during their design. The article discusses a script-based runtime environment developed by the authors for using methodical applied tests to numerically investigate parameters of high-performance computing systems. This environment enables efficient analysis of the results of applying tests and provides an assessment of the performance and reliability of high-performance computing systems.

Keywords: computing system; high-performance computing simulation; mathematical modeling; high-performance computing system testing, HPC.

For citation: Ignatyev A.O., Mokshin S.Yu. The environment suite for methodological applied tests for numerical research of HPC systems parameters. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024. pp. 81-92 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-6.

1.Введение

При решении задач численного моделирования физических процессов требуется огромный объем вычислений, обеспечиваемых высокопроизводительными вычислительными системами (ВВС). Современные ВВС, предназначенные для решения задач численного моделирования, представляют собой множество вычислительных узлов, систему хранения с параллельным доступом и сервисные подсистемы, объединенных высокопроизводительной коммуникационной средой [1]. При проектировании таких ВВС закладываются ожидаемые характеристики по производительности вычислений, основанные на физических показателях используемых элементов и используемой архитектуре ВВС. Однако, как показывает практика, переход от одной ВВС к другой с большей заявленной производительностью не гарантирует кратного сокращения сроков решаемых задач. Поэтому во всем мире разрабатываются специализированные пакеты тестовых программ [2, 3], реализующих различные численные алгоритмы и позволяющие оценить производительность ВВС применительно к данным алгоритмам. В России такие пакеты разрабатываются на предприятиях Госкорпорации «Росатом» [4, 5].

Процедура проведения испытаний, создаваемых в РФ ВВС узаконена Государственным стандартом [6], в котором методические прикладные тесты (МПТ) используются для определения производительности ВВС, эффективности распараллеливания и проверки технологического процесса счета задач.

Использование МПТ на испытаниях ВВС требует проведения многочисленных серий расчетов с вариацией ряда параметров запуска тестов (объем обрабатываемых тестовой программой данных, размещение процессов задачи на вычислительных узлах, используемые компиляторы и библиотеки и т.д.), поэтому их применение предполагает использование средств автоматизации проведения расчетов и сравнения полученных результатов.

Автоматизированные среды выполнения тестов (тестовые мейнфреймы) давно уже используются при отладке программного обеспечения [7, 8]. В данной статье рассматривается среда выполнения методических прикладных тестов (СВ МПТ), разработанная и успешно применяемая в РФЯЦ-ВНИИТФ. Принципиальным отличием рассматриваемой СВ МПТ является то, что она предназначена для оценки качества наладки

оборудования и получения характеристик производительности и надежности исполнения вычислительных систем, облегчая труд разработчиков и изготовителей ВВС.

2. Описание среды выполнения методических прикладных тестов

2.1 Назначение и область применения

Практически все современные ВВС, предназначенные для решения задач численного моделирования различных физических процессов, имеют однотипную архитектуру: вычислительная подсистема ВВС представляет собой множество многопроцессорных вычислительных узлов (ВУ), объединенных высокоскоростным коммуникационным оборудованием, к которому также подключены система хранения с параллельным доступом, подсистема управления и сервисные подсистемы [1]. Все узлы ВВС работают под управлением операционной системы (ОС) семейства Linux [9]. Для распараллеливания на общей памяти используется стандарт OpenMP [10] или системная библиотека libpthread, для распараллеливания между ВУ используется MPI [11]. Такая архитектура ВВС определяет состав и выбор программных средств, используемых в СВ МПТ.

Состав вычислительной подсистемы ВВС: количество ВУ, характеристики установленных в них процессоров, используемые компиляторы и библиотеки передачи сообщений условно назовем программно-аппаратной архитектурой ВВС. СВ МПТ предполагает наличие описания разных программных архитектур, обеспечивая для них унифицированную подготовку тестовых программ к выполнению, запуску и формированию отчетов в виде таблиц. Тем самым облегчается тестирование ВВС при наладке и проведении испытаний.

2.2 Назначение и область применения

Структура СВ МПТ представлена на рис. 1. Корневой каталог СВ МПТ содержит:

- описания тестируемых программно-аппаратных архитектур (ПА);
- каталоги с МПТ;
- общие программные компоненты, используемые для сборки, запуска МПТ и для вывода полученных результатов, включая функции для генерации последовательностей узлов или потоков, разложения числа процессов по направлениям сетки задачи, запуска задания на счет, привязки процессов к ядрам;
- каталог для выполнения комплексного теста (КТ).

2.3 Описание общих компонент СВ МПТ

2.3.1 Описание программной архитектуры ВВС

Каждая программно-аппаратная архитектура ВВС содержит ряд атрибутов, характеризующих как саму тестируемую ВВС, так и используемое для прогона тестов программное обеспечение. Описание ПА ВВС снабжается идентификатором и включается в общее описание всех тестируемых ПА ВВС.

Описание ПА ВВС включает в себя:

- количество вычислительных узлов (ВУ);
- количество и тип процессоров на ВУ,
- количество ядер в процессоре;
- тактовую частоту процессора;
- пути доступа к компиляторам и библиотекам MPI.

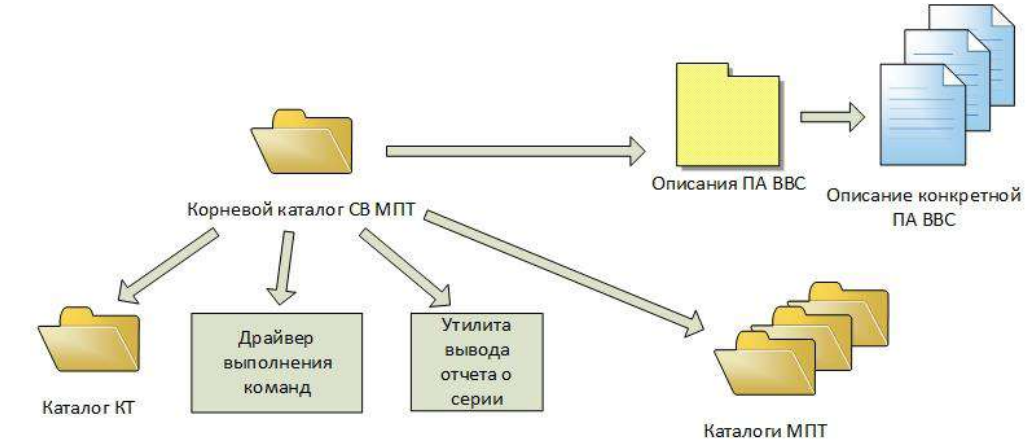


Рис. 1.Структура СВ МПТ.
Fig. 1. Program environment structure.

2.3.2 Каталоги с МПТ

Среда выполнения методических прикладных тестов содержит множество тестовых программ. Каждая тестовая программа располагается в отдельном каталоге и снабжается своим идентификатором.

2.3.3 Запуск задания на выполнение работы

Управляющий модуль (скрипт Execute) предназначен для выполнения работы по запуску других скриптов. В его параметрах указывается:

- идентификатор ПА ВВС;
- идентификатор МПТ;
- вид выполняемой работы (сборка исполнимого кода, запуск серии тестовых расчетов, вывод отчета);
- дополнительные параметры, специфичные для конкретной работы.

Этот скрипт считывает описание указанной в параметрах ПА ВВС, формирует на его основе переменные окружения, используемые при выполнении работы, и, в общем случае, переходит в каталог указанной в параметрах программы, запуская один из скриптов, выполняющих требуемую работу.

2.3.4 Формирование отчета о серии тестов

Утилита вывода отчета о серии тестов (ReportSeria.py) вызывается непосредственно из управляющего модуля. Для выбора отчетных показателей конкретной тестовой программы используется утилита ReportTest.py из каталога МПТ. Утилита просматривает все файлы в каталоге результатов указанной серии и выводит отчет в виде таблицы. В заголовке таблицы указывается название МПТ, используемые для ее сборки компилятор и библиотека MPI, а также опции компиляции. Эта таблица содержит информацию об использованном количестве узлов и ядер, времени выполнения теста, достигнутой производительности, эффективности распараллеливания и признак верификации расчета.

2.4 Описание МПТ

Структура каталога МПТ представлена на рис. 2.

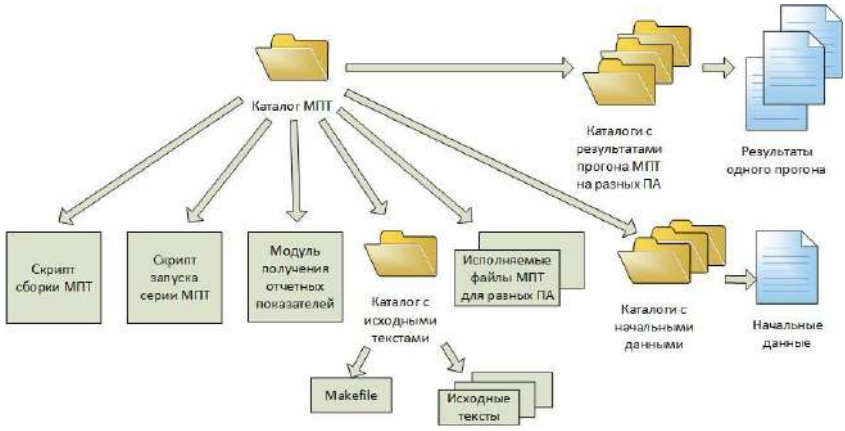


Рис. 2. Структура каталога МПТ.
Fig. 2. Tests directory structure.

Каталог с МПТ содержит:

- подкаталог с исходными текстами тестовой программы и файлом, используемым для сборки программы (makefile);
- скрипт сборки программы под используемую ПА;
- скрипт запуска серии расчетов для используемой ПА;
- модуль получения отчетных показателей для вывода результатов расчета серии;
- исполнимые коды программы для различных ПА;
- каталоги с начальными данными для расчетов серии (исходные данные генерируются автоматически для каждой серии расчетов);
- каталоги, содержащие результаты прогона тестовой программы на выбранных ПА ВВС.

2.4.1 Скрипт сборки МПТ

Скрипт Compile вызывается из управляющего модуля и используется для компиляции тестовой программы из исходных текстов и сборки исполнимого файла программы. Перед вызовом этого скрипта производится считывание описания указанной ПА ВВС и формируются переменные окружения, указывающие пути доступа к компиляторам и библиотекам, необходимым для сборки программы, а также опции компиляции. Полученный в процессе сборки исполняемый код программы помечается идентификатором ПА и, вместе с перечнем используемых при компиляции и линковке опций, помещается в каталог тестовой программы.

2.4.2 Скрипт запуска серии расчетов

Скрипт RunSeria вызывается из управляющего модуля и выполняет запуск серии расчетов одного из двух типов:

- распараллеливание внутри ВУ методом деления с использованием OpenMP;
- распараллеливание на множестве ВУ методом умножения с использованием коммуникационной библиотеки MPI.

Распараллеливание внутри ВУ методом деления выполняется на разном числе ядер, указанном в описании ПА ВВС. Как правило, это значения 1, 2, 4 и до максимально возможного значения.

Распараллеливание на множестве ВУ выполняется на разном числе ВУ. Для большинства тестов это число изменяется по степеням двойки от 1 до предельного числа ВУ, указанного в описании ПА ВВС, хотя есть тесты, в которых задается другой алгоритм изменения числа используемых ВУ.

Тип серии расчетов указывается в параметре скрипта. Для распараллеливания на множестве ВУ также может быть указано число MPI-процессов на одном ВУ (по умолчанию – один). Число используемых ядер для MPI-процесса задачи рассчитывается исходя из полного числа ядер на ВУ и числа MPI-процессов на ВУ, обеспечивая тем самым полную загрузку всего ВУ. Для каждого из расчетов в серии формируются необходимые начальные данные, после этого задание на расчет запускается с помощью системы управления задачами (используется менеджер ресурсов SLURM) [12]. Для указанной серии расчетов создается отдельный каталог в каталоге тестовой программы, в который помещаются все файлы с полученными результатами.

2.4.3 Модуль получения отчетных показателей

Модуль получения отчетных показателей состоит из утилиты ReportTest.py и вызывается из другой утилиты ReportSesia.py, при этом для каждого выводного файла серии выбираются соответствующие отчетные показатели.

2.5 Каталог комплексного теста

Комплексный тест (КТ) предназначен для проверки технологического процесса счета задач и согласно [6] должен обеспечивать непрерывное выполнение большого количества МПТ в течение длительного времени. Во время комплексного теста проверяется как будет работать ВВС в условиях, близких к реальным. Для прохождения КТ используются МПТ из состава СВ МПТ и программа – генератор заданий. Структура каталога КТ представлена на рис. 3.

Рис. 3. Структура каталога КТ.
Fig. 3. Structure of the complex test directory.

Задания, используемые в КТ, содержатся в отдельном подкаталоге и представляют собой скрипты, содержащие обращения к управляющему модулю СВ МПТ с параметрами, определяющими запуск одной конкретной тестовой программы на заданном числе ВУ. Каждому заданию соответствует свой скрипт верификации результатов.

Генератору заданий в параметрах указывается время выполнения КТ (обычно, не менее 8 часов). В процессе работы генератор в течение заданного времени генерирует и запускает задачи из каталога заданий, отслеживает их завершение и анализирует полученные

результаты. По окончании работы формируется отчет, содержащий времена начала и завершения КТ, информацию о запускаемых и завершаемых задачах, общее количество запущенных задач, количество задач завершенных успешно и с ошибками, а также разброс времен выполнения задач каждого типа.

3. Использование СВ МПТ

3.1 Формирование ПА ВВС

Множество ПА ВВС описывается в файле TstEnv в корневом каталоге СВ МПТ. Этот файл является shell-скриптом. Выбор конкретной ПА производится оператором case по идентификатору ПА. Поэтому описание ПА сводится к формированию строк вида, приведенным на листинге 1.

```
1 case $ARCH in
2    GNU11_MV237)
3      # Описание оборудования
4      export VNODE=4608
5      export SOCKETS=2
6      export CORpSOC=64
7      export THRpCOR=1
8      export CORES=$[ $SOCKETS * $CORpSOC * $THRpCOR ]
9      export NTMAX=128
10     export NHMAX=64
11     # Используемое для сборки ПО и опции
12       module load mpi/gnu
13     export MPIF90=mpifort
14     export MPICXX=mpicxx
15     module load gcc/11.2.0
16     FFOPTS="-ffree-line-length-none"
17     export FFLAGS="-fopenmp -O3 -I${MPI}/include $FFOPTS"
18     export CFLAGS="-fopenmp -O3 -I${MPI}/include"
19     # Переменные окружения при запуске тестов
20     export PART=VKADMIN
21     export TST_AFFINITY=GOMP_CPU_AFFINITY
22     ;;
23    ICF_MV237)
      ...
30      module load mpi/intel

      ...
40     module load intel/2020
      ...
50     ;;
51    *) BREAK No such arch $ARCH
52 esac
```

Листинг 1. Пример описания программных архитектур.
Listing 1. Program architecture description example.

В приведенном примере описаны две ПА, используемые при тестировании, описывающие ВВС из 64 ВУ, содержащих по 2 процессора. Первая ПА предназначена для программ, собираемых с использованием компиляторов семейства GNU, вторая – Intel.

Описание ПА состоит из трех частей:

- описание используемых ВУ;
- описание используемого программного обеспечения;

- особенности исполнения (партиция или резервация системы управления задачами, тип привязки процессов к ядрам процессоров и т.д.).

В описании второй ПА приведены только строки, отличающиеся от первой. В начале управляющего модуля устанавливается соответствие первого параметра описанным ПА, как показано на листинге 2.

```
1 case $1 in
2    -a1g) ARCH=GNU11_MV235
3       shift
4       ;;
5    -a1i) ARCH=ICF_MV235
6       shift
7       ;;
8    -a2) ...
9       ;;
10    *)
11 esac
```

Листинг 2. Пример выбора тестируемой ПА в управляющем модуле.
Listing 2. Example of selecting the software architecture in the control module.

3.2 Компиляция тестовой программы

Компиляция тестовой программы на выбранную ПА (параметр –a1g определяет ПА с идентификатором GNU11_MV235) осуществляется следующим образом:

```
# ./Execute -a1g <test_name>
```

Компиляция всех тестовых программ может быть выполнена командой:

```
# ./Execute -a1g all
```

3.3 Запуск серии расчетов

ГОСТ [6] предписывает проведение двух серий расчетов на определение эффективности распараллеливания для каждой МПТ:

- распараллеливание на общей памяти ВУ средствами OpenMP или libpthread;
- распараллеливание на распределенной памяти множества узлов средствами MPI.

В первом случае запускается серия расчетов с использованием одинакового объема пространства данных задачи, вся вычислительная нагрузка распределяется между потоками выполнения программы (метод деления). Серия запускается на одном ВУ и разном числе потоков от 1 до максимально возможного (параметр NTMAX в описании ПА).

Запуск выполняется командой:

```
# ./Execute -a1g <test_name> r1
```

Во втором случае запускается серия расчетов на разном числе узлов от 1 до максимально возможного (параметр NHMAX в описании ПА) с увеличением числа ВУ по степеням двойки (или иным, предусмотренном в тестовой программе способом).

Запуск выполняется командой:

```
# ./Execute -a1g <test_name> rN
```

где N – число MPI-процессов на ВУ (1,2,4 или 8).

При запуске серии в каталоге тестовой программы создается подкаталог, помеченный идентификатором ПА, в который будут записаны файлы результатов расчетов.

3.4 Вывод результатов серии расчетов

Вывод результатов серии на одном ВУ производится командой:

```
# ./Execute -a1g <test_name> R1
```

В результате исполнения получается табличный файл вида, представленного на рис. 4.

```
=== sPPM GNU11MV235 ===
/home/s4145/TestsRosatom/SPPM-OMP-MPI
MPIF90=/opt/mpi/mvapich2-2.3.5/gcc/10.2.0/bin/mpifort
FFLAGS=-fopenmp -O3 -I/opt/mpi/mvapich2-2.3.5/gcc/10.2.0/include -ffree-
line-length-none  -fdefault-real-8  -fdefault-double-8  -std=legacy  -
fopenmp
mpifort for MVAPICH2 version 2.3.5 gcc version 11.2.0 (GCC)
/home/s4145/TestsRosatom/SPPM-OMP-MPI
One Node Seria
NP  DProc    T        Esp  Mem/Proc(MB) Ver V(Gflops)   Ecp   Size(cell)
1     1x1x001  5.899e+01   100.00  8.493e+02    PASS  4.895e+00    13.60
192x192x192
1     1x1x004  1.486e+01    99.26  8.493e+02    PASS  1.944e+01    13.50
192x192x192
1     1x1x008  7.676e+00    96.07  8.493e+02    PASS  3.762e+01    13.06
192x192x192
1     1x1x016  3.967e+00    92.94  8.493e+02    PASS  7.280e+01    12.64
192x192x192
1     1x1x032  2.078e+00    88.72  8.493e+02    PASS  1.390e+02    12.06
192x192x192
1     1x1x064  1.263e+00    72.98  8.493e+02    PASS  2.286e+02     9.92
192x192x192
1     1x1x128  1.532e+00    30.08  8.493e+02    PASS  1.885e+02     4.09
192x192x192
```

Рис. 4. Таблица результатов одноузловой серии расчетов.
Fig. 4. Single-node calculation results.

В заголовке этого табличного файла приведены имя тестовой программы, идентификатор ПА, используемые компиляторы и опции компиляции. В столбцах таблицы приводятся:

- число процессов MPI;
- конфигурация распараллеливания запуска в виде <число ВУ>×<число MPI-процессов на ВУ>×<число потоков на MPI-процессе>;
- время выполнения;
- объем используемой на одном процессе MPI оперативной памяти;
- признак верификации результатов;
- производительность в гигафлопах (если программа умеет ее рассчитывать);
- эффективность использования процессора (в случае, если тестовая программа умеет рассчитывать производительность в операциях в секунду);
- размер задачи в ячейках.

Эффективность использования процессора рассчитывается как отношение полученной на тесте производительности к пиковой производительности ВУ (в процентах).

Вывод результатов многоузловой серии производится командой:

```
# ./Execute-alg <test_name> RS
```

В результате получается табличный файл, содержимое которого представлено на рис. 5. Обозначения в табличном виде те же самые.

3.5 Проверка технологического процесса счета задач

Проверка выполнения технологического процесса счета задач выполняется путем запуска комплексного теста. Традиционно в состав КТ входят задачи, разработанные в РФЯЦ-ВНИИТФ и РФЯЦ-ВНИИЭФ, охватывающие основные этапы численного моделирования, производимого в этих институтах, а также три задачи из NAS Parallel Benchmark (Block Tri-

diagonal solver, Lower-Upper Gauss-Seidel solver, calar Penta-diagonal solver). Параметры тестовых задач подобраны так, чтобы время счета каждой составляло не более нескольких минут. Таким образом, за время выполнения КТ (обычно, не менее 8 часов) просчитывается несколько тысяч задач, создавая нагрузку как на оборудование ВВС, так и на компоненты операционной системы.

```
=== sPPM GNU11MV235 ===
/home/4145/TestsRosatom/SPPM-OMP-MPI
MPIF90=/opt/mpi/mvapich2-2.3.5/gcc/10.2.0/bin/mpifort
FFLAGS=-fopenmp -O3 -I/opt/mpi/mvapich2-2.3.5/gcc/10.2.0/include -ffree-
line-length-none  -fdefault-real-8  -fdefault-double-8  -std=legacy  -
fopenmp
mpifort for MVAPICH2 version 2.3.5 gcc version 11.2.0 (GCC)
/home/s4145/TestsRosatom/SPPM-OMP-MPI
Scalable Nodes Seria
NP    DProc      T             Esp   Mem/Proc(MB) Ver V(Gflops)     Ecp
Size(cell)
  4   1x4x032  2.422e+00   100.00  8.493e+02     PASS  4.769e+02    10.35
192x192x192
  8   2x4x032  2.514e+00    96.34  8.493e+02     PASS  9.189e+02     9.97
192x192x192
 16   4x4x032  2.526e+00    95.88  8.493e+02     PASS  1.829e+03     9.92
192x192x192
 32   8x4x032  2.595e+00    93.33  8.493e+02     PASS  3.561e+03     9.66
192x192x192
 64  16x4x032  2.650e+00    91.40  8.493e+02     PASS  6.974e+03     9.46
192x192x192
128  32x4x032  2.663e+00    90.95  8.493e+02     PASS  1.388e+04     9.41
192x192x192
256  64x4x032  2.667e+00    90.81  8.493e+02     PASS  2.772e+04     9.40
192x192x192
```

Рис. 5. Таблица результатов многоузловой серии рачетов.
Fig. 5. Multi-node calculation results.

Он содержит следующую информацию:

- дата и время начала и конца выполнения теста;
- информацию о запуске и завершении задач;
- результат проверки правильности расчета каждой задачи;
- общее количество запущенных задач, полное количество завершенных задач и количество задач, завершенных с ошибками или аварийно.

При завершении КТ для каждого типа задач выдается:

- имя задачи (test);
- количество запущенных задач этого типа (start);
- количество завершенных задач (fin);
- количество успешно завершенных задач (ver);
- минимально, максимальное и среднее времена выполнения успешно завершенных задач этого типа (tmin, tmax, tave);
- величины разброса между минимальным и максимальным (dtM), а также между минимальным и средним временами выполнения (dtA) в процентах.

Последние значения позволяют оценить стабильность работы оборудования ВВС в условиях многозадачного режима.

Исполнимые коды МПТ берутся те же, что и при проведении серий расчетов на определение эффективности распараллеливания.

Запуск КТ производится скриптом RunAutoLauncher, в котором указывается необходимое время выполнения теста.

Журнал результатов работы КТ имеет вид, представленный на рис. 6.

```
2023-11-22 16:04:47 s4145@f1:/home/s4145/TestsRosatom/Complex pid=51531
запущен на 8.0 часов из Tasks
2023-11-22 16:04:47 В очереди 0 задач, в счете 0. Всего запущено 0,
завершено 0, с ошибками 0
Запуск md-16        Submitted batch job 322730
...
Завершено хорошо mht-16-336700.out , переносится в GOOD
2023-11-23 00:10:44 В очереди 0 задач, в счете 0. Всего запущено 13957,
завершено 13957, с ошибками 0 Не запускаем
2023-11-23 00-11-04 Завершение работы, время выполнения 4.86e+02 минут
-------------------------------
task       start   fin   ver tmin      tmax      tave        dtM%   dtA%
lu-512       821   821   821 4.651e+01 5.572e+01 4.717e+01  19.80   1.43
bt-225       821   821   821 1.230e+02 1.356e+02 1.246e+02  10.25   1.32
md-16        821   821   821 4.775e+00 5.446e+00 4.877e+00  14.05   2.14
bt-1089      821   821   821 3.086e+01 4.291e+01 3.431e+01  39.05  11.18
bt-625       821   821   821 5.098e+01 6.178e+01 5.206e+01  21.18   2.12
egida-16     821   821   821 1.994e+00 2.336e+00 2.084e+00  17.14   4.48
lu-256       821   821   821 8.654e+01 9.882e+01 8.775e+01  14.19   1.40
slon-16      821   821   821 2.135e+01 2.455e+01 2.163e+01  14.96   1.32
sppm-16      821   821   821 1.567e+00 2.759e+00 1.662e+00  76.07   6.04
lu-1024      821   821   821 2.461e+01 3.365e+01 2.530e+01  36.73   2.81
sp-625       821   821   821 6.622e+01 8.664e+01 6.784e+01  30.84   2.44
sp-225       821   821   821 1.711e+02 2.116e+02 1.740e+02  23.70   1.70
hctest-16    821   821   821 1.581e+02 1.838e+02 1.638e+02  16.26   3.63
sp-1089      821   821   821 3.902e+01 5.431e+01 4.096e+01  39.19   4.98
pauk-16      821   821   821 6.556e+01 8.369e+01 6.693e+01  27.64   2.09
mht-16       821   821   821 2.802e+02 2.845e+02 2.831e+02   1.51   1.03
machete-8    821   821   821 1.689e+02 1.898e+02 1.729e+02  12.41   2.36
```

Рис. 6. Пример журнала результатов работы комплексного теста
Fig. 6. Complex test results

4. Заключение

Представленная в статье программная среда выполнения методических прикладных тестов, разработанная в РФЯЦ-ВНИИТФ более трех лет назад, позволила автоматизировать процесс тестирования ВВС при их наладке и проведении испытаний, и тем самым значительно сократить сроки проведения этих работ.

СВ МПТ полностью учитывает требования ГОСТ по проведению приемочных испытаний и успешно применяется на создаваемых РФЯЦ-ВНИИТФ ВВС.

Авторы выражают надежду, что изложенная в статье информация о среде выполнения методических прикладных тестов окажется полезной другим специалистам, занимающимся разработкой ВВС.

Список литературы/References

[1]. Игнатьев А.О., Мокшин С.Ю. Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования, Препринт РФЯЦ-ВНИИТФ № 265, Снежинск, 2020 г., 21 с.

[2]. NAS Parallel Benchmarks. Available at: https://www.nas.nasa.gov/software/npb.html, accessed 07.07.2024.

[3]. Benchmarks and Performance Analysis. Available at https://www.lanl.gov/projects/crossroads/benchmarks-performance-analysis.php, accessed 19.07.2024.

[4]. Алексеев А.В., Беляев С.П., Бочков А.И. и др. Методические прикладные тесты РФЯЦ-ВНИИЭФ для численного исследования параметров высокопроизводительных вычислительных систем. ВАНТ, сер. Математическое моделирование физических процессов. 2020. Вып. 2., с. 86-100.

[5]. Игнатьев А.О., Мокшин С.Ю., Романова Е.М. и др. Набор методических тестовых программ для численного исследования параметров высокопроизводительных вычислительных систем. Труды ИСП РАН, том 36, вып. 2, 2024 г., с. 109-126.
[6]. ГОСТ Р 55700.26-2020. Высокопроизводительные вычислительные системы. Требования приемочных испытаний.
[7]. Linux Test Project. Available at http://github.com/Linux-test-project/releases/latest, accessed 18.07.2024.
[8]. Robot Framework. Available at http://robotframework.org, accessed 18.07.2024.
[9]. «СПО Супер-ЭВМ». Available at http://vniitf.ru/article/spo-super-evm, accessed 18.05.2024.
[10]. OpenMP. Available at: https://www.openmp.org/, accessed 07.05.2024.
[11]. Gropp W., Lusk E., Skjellum A. Using MPI: Portable parallel programming with the message-passing interface. – Second edition. – The MIT Press, 1999.
[12]. Slurm workload manager, Available at: https://slurm.schedmd.com/documentation.html, accessed 08.07.2024.

Информация об авторах / Information about authors

Алексей Олегович ИГНАТЬЕВ – начальник лаборатории Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 1998 года. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – Head of the Laboratory of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 1998. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

Сергей Юрьевич МОКШИН – начальник отдела Федерального государственного унитарного предприятия «Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина» с 2016 года. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – Head of the Department of Russian Federal Nuclear Center E. I. Zababakhin «All-Russian Scientific Research Institute of Technical Physics» since 2016. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means for protecting information.

DOI: 10.15514/ISPRAS-2024-36(5)-7

CC BY

Эффективный метод с компрессией для распределенных и федеративных кокоэрсивных вариационных неравенств

[1,2] *Д.О. Медяков, ORCID 0009-0003-2007-8446 <mediakov.do@phystech.edu>*
[1,2] *Г.Л. Молодцов, ORCID 0009-0004-3516-1848 <molodtsov.gl@phystech.edu>*
[1,2,3] *А.Н. Безносиков, ORCID 0000-0002-3217-3614 <anbeznosikov@gmail.com>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*
[2] *Московский физико-технический институт (НИУ), Россия, 117303, г. Москва, ул. Керченская, д. 1А, корп. 1.*
[3] *Университет Иннополис, Россия, 420500, г. Иннополис, ул. Университетская, д. 1.*

Аннотация. Вариационные неравенства как эффективный инструмент для решения прикладных задач, в том числе задач машинного обучения, в последние годы привлекают всё больше внимания исследователей. Области применения вариационных неравенств охватывают широкий спектр направлений – от обучения с подкреплением и генеративных моделей до традиционных приложений в экономике и теории игр. В то же время, невозможно представить современный мир машинного обучения без подходов распределенной оптимизации, которые позволяют значительно ускорить процесс обучения на огромных объемах данных. Однако, сталкиваясь с большими затратами на коммуникации между устройствами в вычислительной сети, научное сообщество стремится к разработке подходов, делающих вычисления дешевыми и стабильными. В этой работе исследуется техника сжатия передаваемой информации применительно к задаче распределенных вариационных неравенств. В частности, предлагается метод на основе продвинутых техник, исконно разработанных для задач минимизации. Для нового метода приводится исчерпывающий теоретический анализ сходимости для кокоэрсивных сильно монотонных вариационных неравенств. Проведенные эксперименты подчеркивают высокую производительность представленной техники и подтверждают практическую применимость.

Ключевые слова: Вариационные неравенства; распределенная оптимизация; федеративное обучение; техники сжатия информации.

Для цитирования: Медяков Д.О., Молодцов Г.Л., Безносиков А.Н. Эффективный метод с компрессией для распределенных и федеративных кокоэрсивных вариационных неравенств. Труды ИСП РАН, 2024, том 36, вып. 5, с. 93-108. Doi: 10.15514/ISPRAS-2024-36(5)-7.

Благодарности: Работа выполнена в Лаборатории проблем федеративного обучения ИСП РАН при поддержке Минобрнауки России (доп. соглашение № 2 от «19» апреля 2024 г. к Соглашению № 075-03-2024-214 от «18» января 2024 года).

Effective Method with Compression for Distributed and Federated Cocoercive Variational Inequalities

[1,2] *D.O. Medyakov, ORCID 0009-0003-2007-8446 <mediakov.do@phystech.edu>*
[1,2] *G.L. Molodtsov, ORCID 0009-0004-3516-1848 <molodtsov.gl@phystech.edu>*
[1,2,3] *A.N. Beznosikov, ORCID 0000-0002-3217-3614 <anbeznosikov@gmail.com>*

[1] *Ivannikov Institute for System Programming of the RAS,*
25, Alexander Solzhenitsyn str., Moscow, 109004, Russia.
[2] *Moscow Institute of Physics and Technology,*
1A, Kerchenskaya Str., Moscow, 117303, Russia.
[3] *Innopolis University,*
1, Universitetskaya str., Innopolis, 420500, Russia.

Abstract. Variational inequalities as an effective tool for solving applied problems, including machine learning tasks, have been attracting more and more attention from researchers in recent years. The use of variational inequalities covers a wide range of areas – from reinforcement learning and generative models to traditional applications in economics and game theory. At the same time, it is impossible to imagine the modern world of machine learning without distributed optimization approaches that can significantly speed up the training process on large amounts of data. However, faced with the high costs of communication between devices in a computing network, the scientific community is striving to develop approaches that make computations cheap and stable. In this paper, we investigate the compression technique of transmitted information and its application to the distributed variational inequalities problem. In particular, we present a method based on advanced techniques originally developed for minimization problems. For the new method, we provide an exhaustive theoretical convergence analysis for cocoersive strongly monotone variational inequalities. We conduct experiments that emphasize the high performance of the presented technique and confirm its practical applicability.

Keywords: Variational inequalities; distributed optimization; federated learning; compression techniques.

For citation: Medyakov D.O., Molodtsov G.L., Beznosikov A.N. Effective Method with Compression for Distributed and Federated Cocoercive Variational Inequalities. *Trudy ISP RAN/Proc. ISP RAS*, 2024, vol. 36, issue 5, pp. 93-108. Doi: 10.15514/ISPRAS-2024-36(5)-7.

Acknowledgements. The work was carried out in the Laboratory for Problems of Federal Education of the ISP RAS with the support of the Ministry of Education and Science of Russia (additional agreement No 2 dated April 19, 2024 to Agreement No 075-03-2024-214 dated January 18, 2024).

1. Введение

Вариационные неравенства (ВН) привлекают внимание исследователей в различных областях уже более полувека [1]. В данной работе рассматривается общая постановка задачи вариационных неравенств на неограниченном множестве:

$$\text{Найти } z^* \in \mathbb{R}^d \text{ такое, что } F(z^*) = 0, \tag{1}$$

где $F: R^d \to R^d$ – заданный оператор. Такая общая постановка покрывает множество известных задач. Приведем несколько примеров.

Пример 1. *Классическая задача минимизации:*

$$\min_{z \in R^d} f(z),$$

где f(z) – некоторая целевая функция. Для того, чтобы свести задачу 1 к задаче минимизации, достаточно рассмотреть оператор вида $F(z) = \nabla f(z)$. Более того, для выпуклых функций f(z) точка z^* будет решением задачи 1 тогда и только тогда, когда она будет решением этой задачи.

Пример 2. *Задача поиска седловой точки (минимаксная задача):*

$$\min_{x\in R^d}\max_{y\in R^d} g(x,y),$$

где $g(z)=g(x,y)$ *– некоторая целевая функция. Для того, чтобы свести задачу 1 к минимаксной задаче, достаточно рассмотреть оператор вида* $F(z)=F(x,y)=\left[\nabla_x g(x,y), -\nabla_y g(x,y)\right]$. *Более того, для выпуклых-вогнутых функций* $g(x,y)$ *точка* $z^*=(x^*,y^*)$ *будет решением задачи 1 тогда и только тогда, когда она будет решением минимаксной задачи.*

Тем не менее, несмотря на общность постановки, она практически неприменима в современных прикладных задачах, в первую очередь в задачах обучения. Дело в том, что считать полное значение оператора на одном вычислительном устройстве слишком затратно по времени из-за огромных объемов тренировочных данных. Поэтому на практике используют распределенные подходы [2-4]. В такой парадигме используется сеть из нескольких устройств, каждое из которых содержит часть тренировочной выборки. Эти устройства обычно объединены в звездную топологию, где крайние узлы вычисляют значение оператора, исходя из хранящейся локально информации, а сервер агрегирует полученные данные и производит обновление оптимизационных переменных. Отдельно выделяют постановку федеративного обучения [5-7]. Она подразумевает наличие на каждом устройстве уникальных тренировочных выборок, причем выборки на разных устройствах необязательно одинаково распределены и, как правило, содержат приватную информацию.

Таким образом, чтобы формально перейти к общей распределенной постановке, представим оператор F в виде конечной суммы:

$$F(z)=\frac{1}{n}\sum_{i=1}^{n} F_i(z), \tag{2}$$

где n – количество устройств в вычислительной сети, а $F_i(\cdot)$ – локальный оператор, вычисляемый на основе данных на i-ом устройстве. Комбинация (1) и (2) отражает всевозможный спектр задач распределенного машинного обучения от простых регрессий до нейронных сетей [8]. Несмотря на то, что такой подход применим к классической задаче минимизации (Пример 1), больший интерес он вызывает для поиска седловых точек (Пример 2) [9]. На практике это нашло особенно яркое применение в так называемом состязательном подходе, будь то обучение генеративных сетей (GAN) [10], или робастное обучение моделей [11-12]. Кроме того, вариационные неравенства также находят широкое применение в различных классических задачах, включая эффективное матричное разложение [13], устранение зернения в изображениях [14-15], робастную оптимизацию [16], постановки из экономики, теории игр [17] и оптимального управления [18].

Для решения такой задачи ВН (1) + (2) необходима адаптация классических методов оптимизации, например, таких как градиентный спуск. Ее можно осуществить по аналогии с Примером 1, рассмотрев $\nabla f(\cdot)\to F(\cdot)$. Однако, из-за распределенной постановки 2, чтобы собрать полное значение оператора $F(\cdot)$, вычислительным устройствам необходимо "общаться" друг с другом, как правило, посредством пересылки данных на сервер. Подобные затраты на коммуникацию представляют собой серьезное ограничение распределенных и федеративных подходов. Передача информации часто занимает много времени и нарушает процесс обучения. В некоторых случаях временные затраты могут значительно превышать сложность локальных вычислений, что делает архитектурные решения неэффективными для практического применения.

Для снижения издержек на обмен информацией между узлами были предложены различные методы уменьшения количества передаваемых данных [19-20]. В данной работе исследуется применимость техники компрессии к распределенным вариационным неравенствам, сочетая

преимущества сжатия и эффективной обработки данных для минимизации коммуникационных затрат и повышения устойчивости алгоритмов.

2. Обзор литературы

2.1 Методы со сжатием

Для преодоления возникающих проблем с коммуникационными затратами сообщество применяет различные методики. Идея уменьшения количества передаваемой информации в векторах градиентов была исследована в работе [21]. Авторы предлагают модификацию градиентного спуска, при которой на каждом шаге обновляются только некоторые случайные координаты. Такие операторы впоследствии стали называть RAND-K [22]. Через несколько лет метод координатного градиентного спуска был адаптирован для распределенной оптимизации [23]. Альтернативной методикой борьбы с коммуникационными издержками является техника сжатия передаваемой информации, основанная на использовании квантизации градиентов или параметров моделей [24]. Данные методы направлены на уменьшение объема данных, за счет ограничения количества бит, необходимых для представления чисел с плавающей точкой [25-27]. Обобщение техники сжатия (будь то с помощью выбора координат или квантизации) было сделано в работе [28] посредством введения общего определения оператора сжатия. Авторы предлагают добавить оператор в обычный градиентный спуск. Однако этот метод не сходится к истинному решению, а лишь к некоторой окрестности, зависящей от дисперсии квантизованных оценок градиентов [29].

Принимая во внимание данную проблему, сообщество исследователей в области оптимизации и машинного обучения активно разрабатывает распределенные алгоритмы, применяющие технику уменьшения дисперсии. В стандартной нераспределенной задаче стохастической оптимизации она была использована в методах SVRG [30], SAG [31], SAGA [32], SARAH [33-34], а затем адаптирована для распределенной оптимизации в методе DIANA [35] путем сжатия не самого градиента, а разности градиентов. В дальнейшем, идея была развита и использована в более продвинутых алгоритмах: VR-DIANA [36], FEDCOMGATE [37], FEDSTEPH [38]. Одной из наиболее продвинутых методик применения данных техник в невыпуклой распределенной оптимизации является MARINA [39]. В отличие от всех известных подходов, использующих несмещенные операторы сжатия, MARINA базируется на смещенной оценке градиента. Тем не менее, доказано, что данный метод обеспечивает гарантии сходимости, превосходящие все ранее известные техники.

2.2 Методы решения вариационных неравенств

Применение различных методов для решения задач вариационных неравенств и задач седловых точек является предметом обширных исследований [40-48]. Упомянутая выше техника редукции дисперсии была также перенесена и на вариационные неравенства [49-57]. Большинство из этих методов основаны на подходе SVRG, однако некоторые исследования были проведены в анализе более привлекательного с практической точки зрения для задач минимизации метода SARAH [58-59].

Методы борьбы за эффективность процесса общения также исследовались в общности вариационных неравенств [9, 60-65], исследовались и алгоритмы с компрессией. Для липшицевых операторов это было сделано в работах [66-68] на основе редукции дисперсии из работы [52].

Говоря о стандартных предположениях, которые используются в анализе методов решения вариационных неравенств, кокоэрсивный режим является одним из наиболее часто встречающихся, несмотря на то что данное требование является более строгим, нежели липшицевость операторов [69]. В частности, для кокоэрсивных вариационных неравенств существует общий анализ стохастических методов [70], включающий и методы со сжатием.

В разрезе работ, использующих предположение кокоэрсивности, стоит выделить основанный на алгоритме SARAH метод [59]. В то же время метод MARINA, как раз и базирующийся на SARAH, еще не был исследован в контексте вариационных неравенств. Шаг MARINA по факту является шагом SARAH с добавлением сжатия. Как уже отмечалось ранее, метод MARINA является наиболее привлекательным с точки зрения уменьшения коммуникационных издержек в парадигме распределенного обучения. Цель данной статьи – исследовать теоретическую и практическую применимость алгоритма MARINA к кокоэрсивным вариационным неравенствам.

3. Основные результаты

- **Новый метод.** В рамках исследования распределенной постановки представляется новый метод, который использует метод с компрессией MARINA для решения задач вариационных неравенств.
- **Теоретическая значимость.** В работе представлен полный теоретический анализ предложенного метода для кокоэрсивных монотонных вариационных неравенств.
- **Адаптивный анализ.** Данное исследование предлагает несколько возможных расширений. Например, полученные результаты могут быть обобщены для случая произвольного оператора квантования и различных размеров подмножеств данных, используемых клиентами.
- **Экспериментальная валидация.** Проведены численные эксперименты, которые подчеркивают применимость нашего метода с различными техниками сжатия (квантизацией, выбором координат).

4. Постановка

Приведем нотацию, которая используется в работе. Будем обозначать множество векторов размерности d с элементами, являющимися вещественными числами, как R^d, положительные вещественные коэффициенты греческими и латинскими буквами α, γ, μ и ℓ, Евклидову норму вектора $u \in R^d$ как $\| u \| = \sqrt{\langle u, u \rangle}$, математическое ожидание случайной величины ξ как $E[\xi]$. Напомним, что рассматривается задача 1, где оператор F принимает вид (2). Введем следующие предположения на оператор.

Предположение 1 (Кокоэрсивность). *Каждый оператор F_i является ℓ-кокоэрсивным, если для любых $u, v \in R^d$ выполняется*

$$\| F_i(u) - F_i(v) \|^2 \leq \ell \langle F_i(u) - F_i(v), u - v \rangle.$$

Это предположение является более сильным аналогом предположения на липшицевость F_i. Для задач выпуклой минимизации ℓ-липшицевость и ℓ-кокоэрсивность эквивалентны. Сравнение этих предположений для задачи вариационных неравенств приведено в работе [69].

Предположение 2 (Сильная монотонность). *Оператор F является -сильно монотонным, то есть для любых $u, v \in R^d$ выполняется*

$$\langle F(u) - F(v), u - v \rangle \geq \mu \| u - v \|^2.$$

Для задач минимизации это свойство означает сильную выпуклость, а для задач поиска седловой точки – сильную выпуклость-сильную вогнутость.

Предположение 3. *Отображение $C: R^d \to R^d$ является несмещенным компрессором, если для любого $u \in R^d$ существует $\alpha > 0$ такое, что выполняется*

$$E[C(u)] = u,$$
$$E\|C(u) - u\|^2 \leq \alpha \| u \|^2.$$

Это классическое предположение на компрессоры, используемые как в статье MARINA [39], так и в других работах [28, 71].

Предположение 4. *Компрессор* $C(\cdot)$ *оставляет долю информации равную* $\delta \in (0,1]$.

Например, для компрессоров, которые производят выбор только части координат, это предположение означает, что из d координат, они оставят только $d\delta$.

5. Метод и анализ сходимости

Для решения общей постановки задачи вариационных неравенств с липшицевыми операторами обычно используют методы, основанные на подходе EXTRAGRADIENT [40]. Однако в этой работе рассматриваются кокоэрсивные вариационные неравенства, поэтому достаточно использовать классический подход SGD [72-73]. Комбинируя его с продвинутой техникой сжатия MARINA, представляем формальное описание рассматриваемого алгоритма (Алгоритм 1). Отметим, что наш подход несколько отличается от предложенного в оригинальной статье [39]. Там предлагается производить рестарт рекурсивных обновлений аппроксимаций локальных градиентов с некоторой заранее выбранной вероятностью, здесь же это делается раз в фиксированное число итераций, которое называется эпохой (строка 6 в Алгоритме 1). Далее приведен теоретический анализ сходимости Алгоритма 1.

Алгоритм 1 MARINA для кокоэрсивных вариационных неравенств

1: **Параметры:** Шаг обучения $\gamma > 0$, количество итераций обучения K.
2: **Инициализация:** $z^0 \in \mathbb{R}^d$.
3: **for** $s = 1, 2, \ldots, S$ **do**
4: $\quad z^0 = \tilde{z}^{s-1}$
5: $\quad g^0 = F(z^0)$
6: $\quad z^1 = z^0 - \gamma g^0$
7: $\quad$ **for** $k = 1, 2, \ldots, K-1$ **do**
8: $\quad\quad$ Отправить g^{k-1} на каждое устройство
9: $\quad\quad$ **for** $i = 1, 2, \ldots, n$ **do**
10: $\quad\quad\quad g_i^k = g^{k-1} + \mathcal{C}\left(F_i(z^k) - F_i(z^{k-1})\right)$
11: $\quad\quad\quad$ Отправить g_i^k на сервер
12: $\quad\quad$ **end for**
13: $\quad\quad g^k = \frac{1}{n}\sum_{i=1}^{n} g_i^k$
14: $\quad\quad z^{k+1} = z^k - \gamma g^k$
15: $\quad$ **end for**
16: $\quad \tilde{z}^s = z^K$
17: **end for**

Лемма 1. *Пусть выполнены Предположения 1, 2, 3. Тогда для Алгоритма 1 с* $\gamma \leq \frac{1}{2\ell\left(1+\frac{\alpha}{n}\right)}$ *верна следующая оценка:*

$$E \parallel g^K \parallel^2 \leq \left(1 - \frac{2\gamma\mu}{3}\right)^K E \parallel F(z^0) \parallel^2.$$

Доказательство. Зафиксируем любое s в Алгоритме 1 и рассмотрим обновление аппроксимаций градиента g^k:

$$\|g^k\|^2 = \left\|\frac{1}{n}\sum_{i=1}^{n} g_i^k\right\|^2$$

$$
\begin{aligned}
&= \left\| g^{k-1} + \frac{1}{n}\sum_{i=1}^{n} \mathcal{C}\left(F_i(z^k) - F_i(z^{k-1})\right)\right\|^2 \\
&= \left\| g^{k-1} + \frac{1}{n}\sum_{i=1}^{n} \left(F_i(z^k) - F_i(z^{k-1})\right)\right\|^2 \\
&\quad + \left\| \frac{1}{n}\sum_{i=1}^{n} \left(\mathcal{C}\left(F_i(z^k) - F_i(z^{k-1})\right) - \left(F_i(z^k) - F_i(z^{k-1})\right)\right)\right\|^2 \\
&\quad + 2\Big\langle g^{k-1} + \frac{1}{n}\sum_{i=1}^{n} \left(F_i(z^k) - F_i(z^{k-1})\right), \\
&\qquad\qquad \frac{1}{n}\sum_{i=1}^{n} \left(\mathcal{C}\left(F_i(z^k) - F_i(z^{k-1})\right) - \left(F_i(z^k) - F_i(z^{k-1})\right)\right)\Big\rangle.
\end{aligned}
$$

Пользуясь Предположением 3, получим, что математическое ожидание правой части скалярного произведения равно нулю. Отсюда следует, что и скалярное произведение становится равным нулю. Более того, используя Предположение 3 для второй нормы, получаем

$$
\begin{aligned}
&\left\|\frac{1}{n}\sum_{i=1}^{n}\Big(\mathcal{C}\big(F_i(z^k) - F_i(z^{k-1})\big) - \big(F_i(z^k) - F_i(z^{k-1})\big)\Big)\right\|^2 \\
&= \tfrac{1}{n^2}\textstyle\sum_{i=1}^{n}\left\|\mathcal{C}\big(F_i(z^k) - F_i(z^{k-1})\big) - \big(F_i(z^k) - F_i(z^{k-1})\big)\right\|^2 \\
&\leq \frac{\alpha}{n^2}\sum_{i=1}^{n}\|F_i(z^k) - F_i(z^{k-1})\|^2,
\end{aligned}
$$

так как все попарные скалярные произведения равны нулю. Таким образом,

$$
\begin{aligned}
\mathbb{E}\|g^k\|^2 &\leqslant \mathbb{E}\|g^{k-1}\|^2 + \left\|F(z^k) - F(z^{k-1})\right\|^2 \\
&\quad + 2\left\langle g^{k-1}, F(z^k) - F(z^{k-1})\right\rangle \\
&\quad + \frac{\alpha}{n^2}\sum_{i=1}^{n}\left\|F_i(z^k) - F_i(z^{k-1})\right\|^2 \\
&= \mathbb{E}\|g^{k-1}\|^2 + \left\|F(z^k) - F(z^{k-1})\right\|^2 \\
&\quad - \frac{2}{\gamma}\left\langle z^k - z^{k-1}, F(z^k) - F(z^{k-1})\right\rangle \\
&\quad + \frac{\alpha}{n^2}\sum_{i=1}^{n}\left\|F_i(z^k) - F_i(z^{k-1})\right\|^2 \\
&= \mathbb{E}\|g^{k-1}\|^2 + \left\|F(z^k) - F(z^{k-1})\right\|^2 \\
&\quad + \frac{\alpha}{n^2}\sum_{i=1}^{n}\left\|F_i(z^k) - F_i(z^{k-1})\right\|^2 \\
&\quad - \frac{2}{3\gamma}\frac{1}{n}\sum_{i=1}^{n}\left\langle z^k - z^{k-1}, F_i(z^k) - F_i(z^{k-1})\right\rangle
\end{aligned}
$$

$$-\frac{2}{3\gamma}\left\langle z^k - z^{k-1}, F(z^k) - F(z^{k-1})\right\rangle$$
$$-\frac{2}{3\gamma}\left\langle z^k - z^{k-1}, F(z^k) - F(z^{k-1})\right\rangle.$$

Для первого и второго скалярного произведений используется Предположение 1, а именно $\langle z^k - z^{k-1}, F_i(z^k) - F_i(z^{k-1})\rangle \geq \frac{1}{\ell}\|F_i(z^k) - F_i(z^{k-1})\|^2$, а для третьего – Предположение 2, а именно $\langle z^k - z^{k-1}, F(z^k) - F(z^{k-1})\rangle \geq \mu\|F(z^k) - F(z^{k-1})\|^2$. Тогда

$$\begin{aligned}\mathbb{E}\|g^k\|^2 \quad &\leqslant \quad \mathbb{E}\|g^{k-1}\|^2 + \left(1 - \frac{2}{3\gamma\ell}\right)\left\|F(z^k) - F(z^{k-1})\right\|^2 \\ &+ \left(\frac{\alpha}{n} - \frac{2}{3\gamma\ell}\right)\frac{1}{n}\sum_{i=1}^{n}\left\|F_i(z^k) - F_i(z^{k-1})\right\|^2 \\ &-\frac{2\mu}{3\gamma}\|z^k - z^{k-1}\|^2.\end{aligned}$$

Выберем $\gamma \leq \frac{1}{2\ell\left(1+\frac{\alpha}{n}\right)}$. С учетом этого, получим

$$E \parallel g^k \parallel^2 \quad \leq \quad \left(1 - \frac{2\gamma\mu}{3}\right)E \parallel g^{k-1} \parallel^2.$$

Запустим рекурсию до первой итерации в эпохе и учтем, что $g^0 = F(z^0)$:

$$E \parallel g^K \parallel^2 \leq \left(1 - \frac{2\gamma\mu}{3}\right)^K E \parallel F(z^0) \parallel^2.$$

□

Следующая лемма дает нам представление о разнице между полным оператором $F(\cdot)$ и его сжатой версией g в течение внутреннего цикла Алгоритма 1.

Лемма 2. *Пусть выполнены Предположения 1, 2, 3. Тогда для Алгоритма 1 верна следующая оценка:* $E\|F(z^K) - g^K\|^2 \leq \frac{\gamma\ell\left(1+\frac{\alpha}{n}\right)}{1-\gamma\ell\left(1+\frac{\alpha}{n}\right)}E \parallel F(z^0) \parallel^2$.

Доказательство. Рассмотрим следующую цепочку:

$$\begin{aligned}\mathbb{E}\left\|F(z^k) - g^k\right\|^2 \quad &= \quad \mathbb{E}\left\|\left[F(z^{k-1}) - g^{k-1}\right] + \left[F(z^k) - F(z^{k-1})\right] - \left[g^k - g^{k-1}\right]\right\|^2 \\ &= \quad \mathbb{E}\left\|F(z^{k-1}) - g^{k-1}\right\|^2 + \mathbb{E}\left\|F(z^k) - F(z^{k-1})\right\|^2 \\ &\quad +\mathbb{E}\left\|g^k - g^{k-1}\right\|^2 + 2\mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, F(z^k) - F(z^{k-1})\right\rangle \\ &\quad -2\mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, g^k - g^{k-1}\right\rangle \\ &\quad -2\mathbb{E}\left\langle F(z^k) - F(z^{k-1}), g^k - g^{k-1}\right\rangle. \qquad (3)\end{aligned}$$

Теперь отдельно оценим второе скалярное произведение:

$$\begin{aligned}&\mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, g^k - g^{k-1}\right\rangle \\ &\quad = \mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, \frac{1}{n}\sum_{i=1}^{n}\mathcal{C}\left(F_i(z^k) - F_i(z^{k-1})\right)\right\rangle\end{aligned}$$

$$= \mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, \frac{1}{n}\sum_{i=1}^{n}\left(F_i(z^k) - F_i(z^{k-1})\right)\right\rangle$$
$$+\mathbb{E}\Big\langle F(z^{k-1}) - g^{k-1},$$
$$\frac{1}{n}\sum_{i=1}^{n}\left(\mathcal{C}\left(F_i(z^k) - F_i(z^{k-1})\right) - \left(F_i(z^k) - F_i(z^{k-1})\right)\right)\Big\rangle.$$

Так как второе скалярное произведение равно нулю ввиду Предположения 3, получим

$$\mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, g^k - g^{k-1}\right\rangle = \mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, F(z^k) - F(z^{k-1})\right\rangle. \quad (4)$$

Делая аналогичные выкладки, можно оценить третье скалярное произведение в (3), как

$$\begin{aligned}\mathbb{E}\left\langle F(z^k) - F(z^{k-1}), g^k - g^{k-1}\right\rangle &= \mathbb{E}\left\langle F(z^k) - F(z^{k-1}), F(z^k) - F(z^{k-1})\right\rangle \\ &= \left\|F(z^k) - F(z^{k-1})\right\|^2. \end{aligned} \quad (5)$$

Подставляя (4) и (5) в (3), получим

$$\begin{aligned}\mathbb{E}\left\|F(z^k) - g^k\right\|^2 &= \mathbb{E}\left\|F(z^{k-1}) - g^{k-1}\right\|^2 + \mathbb{E}\left\|F(z^k) - F(z^{k-1})\right\|^2 \\ &\quad +\mathbb{E}\left\|g^k - g^{k-1}\right\|^2 \\ &\quad +2\mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, F(z^k) - F(z^{k-1})\right\rangle \\ &\quad -2\mathbb{E}\left\langle F(z^{k-1}) - g^{k-1}, F(z^k) - F(z^{k-1})\right\rangle \\ &\quad -2\mathbb{E}\left\|F(z^k) - F(z^{k-1})\right\|^2 \\ &\leqslant \mathbb{E}\left\|F(z^{k-1}) - g^{k-1}\right\|^2 + \mathbb{E}\left\|g^k - g^{k-1}\right\|^2.\end{aligned}$$

Запустим рекурсию до первой итерации в эпохе и учтем, что $g^0 = F(z^0)$*:*

$$\mathbb{E}\left\|F(z^K) - g^K\right\|^2 \leqslant \sum_{k=1}^{K}\mathbb{E}\|g^k - g^{k-1}\|^2. \quad (6)$$

Пользуясь Предположением 3,

$$\begin{aligned}\mathbb{E}\|g^k - g^{k-1}\|^2 &= \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\mathcal{C}\left(F_i(z^k) - F_i(z^{k-1})\right)\right\|^2 \\ &\leqslant \frac{1+\frac{\alpha}{n}}{n}\sum_{i=1}^{n}\left\|F_i(z^k) - F_i(z^{k-1})\right\|^2. \end{aligned} \quad (7)$$

Далее, аналогично доказательству Леммы 1:

$$\begin{aligned}\mathbb{E}\|g^k\|^2 &= \mathbb{E}\left\|g^{k-1} + \frac{1}{n}\sum_{i=1}^{n}\mathcal{C}\left(F_i(z^k) - F_i(z^{k-1})\right)\right\|^2 \\ &\leqslant \mathbb{E}\left\|g^{k-1} + \frac{1}{n}\sum_{i=1}^{n}\left(F_i(z^k) - F_i(z^{k-1})\right)\right\|^2\end{aligned}$$

$$
\begin{aligned}
&+\frac{\alpha}{n^2}\sum_{i=1}^{n}\left\|F_i(z^k)-F_i(z^{k-1})\right\|^2\\
\leqslant\ & \mathbb{E}\|g^{k-1}\|^2+\frac{1+\frac{\alpha}{n}}{n}\sum_{i=1}^{n}\left\|F_i(z^k)-F_i(z^{k-1})\right\|^2\\
&+2\left\langle g^{k-1},F(z^k)-F(z^{k-1})\right\rangle\\
=\ & \mathbb{E}\|g^{k-1}\|^2+\frac{1+\frac{\alpha}{n}}{n}\sum_{i=1}^{n}\left\|F_i(z^k)-F_i(z^{k-1})\right\|^2\\
&-\frac{1}{\gamma}\frac{1}{n}\sum_{i=1}^{n}\left\langle z^k-z^{k-1},F_i(z^k)-F_i(z^{k-1})\right\rangle\\
&-\frac{1}{\gamma}\left\langle z^k-z^{k-1},F(z^k)-F(z^{k-1})\right\rangle\\
\leqslant\ & (1-\gamma\mu)\mathbb{E}\|g^{k-1}\|^2\\
&+\left(1-\frac{1}{\gamma\ell\left(1+\frac{\alpha}{n}\right)}\right)\frac{1+\frac{\alpha}{n}}{n}\sum_{i=1}^{n}\left\|F_i(z^k)-F_i(z^{k-1})\right\|^2.
\end{aligned}
$$

Выражая сумму,

$$
\begin{aligned}
&\frac{1+\frac{\alpha}{n}}{n}\sum_{i=1}^{n}\|F_i(z^k)-F_i(z^{k-1})\|^2\\
\leq&\frac{\gamma\ell\left(1+\frac{\alpha}{n}\right)}{1-\gamma\ell\left(1+\frac{\alpha}{n}\right)}[(1-\gamma\mu)E\parallel g^{k-1}\parallel^2-E\parallel g^{k-1}\parallel^2]\\
\leq&\frac{\gamma\ell\left(1+\frac{\alpha}{n}\right)}{1-\gamma\ell\left(1+\frac{\alpha}{n}\right)}[E\parallel g^{k-1}\parallel^2-E\parallel g^{k}\parallel^2].
\end{aligned}
$$

Комбинируя полученную оценку с (7) и (6), получаем

$$
E\|F(z^K)-g^K\|^2\leq\frac{\gamma\ell\left(1+\frac{\alpha}{n}\right)}{1-\gamma\ell\left(1+\frac{\alpha}{n}\right)}E\parallel g^0\parallel^2=\frac{\gamma\ell\left(1+\frac{\alpha}{n}\right)}{1-\gamma\ell\left(1+\frac{\alpha}{n}\right)}E\parallel F(z^0)\parallel^2.
$$

□

Теперь объединим результаты Леммы 1 и 2 и получим основную теорему данной статьи.

Теорема 1. *Пусть выполнены Предположения 1, 2, 3. Тогда для Алгоритма 1 с* $\gamma=\frac{1}{8\ell\left(1+\frac{\alpha}{n}\right)}$ *и* $K=\frac{30\ell\left(1+\frac{\alpha}{n}\right)}{\mu}$ *верна следующая оценка:*

$$
E\parallel F(z^s)\parallel^2\leq\frac{1}{2}E\parallel F(z^{s-1})\parallel^2.
$$

Доказательство. Начнем с

$$
E\|F(z^K)\|^2\leq 2E\|F(z^K)-g^K\|^2+2E\|g^K\|^2.
$$

Далее применим Леммы 1, 2:

$$\mathbb{E}\left\|F(z^K)\right\|^2 \leqslant \left[2\frac{\gamma\ell\left(1+\frac{\alpha}{n}\right)}{1-\gamma\ell\left(1+\frac{\alpha}{n}\right)}+2\left(1-\frac{2\gamma\mu}{3}\right)^K\right]\mathbb{E}\left\|F(z^0)\right\|^2$$
$$\leqslant \left[2\frac{\gamma\ell\left(1+\frac{\alpha}{n}\right)}{1-\gamma\ell\left(1+\frac{\alpha}{n}\right)}+2\exp\left(-\frac{2}{3}\gamma\mu K\right)\right]\mathbb{E}\left\|F(z^0)\right\|^2.$$

Здесь использовалось, что $\gamma\mu \in (0;1)$ и $(1-\gamma\mu) \leq exp(-\gamma\mu)$. Подставив $\gamma = \frac{1}{8\ell\left(1+\frac{\alpha}{n}\right)}$ и $K = \frac{30\ell\left(1+\frac{\alpha}{n}\right)}{\mu}$, получаем оценку

$$E\|F(z^K)\|^2 \leq \frac{1}{2}E\|F(z^0)\|^2.$$

Так как $z^0 = \mathbb{z}^{S-1}$ и $z^K = \mathbb{z}^S$, была получена сходимость за одну эпоху

$$E\|F(\mathbb{z}^S)\|^2 \leq \frac{1}{2}E\|F(\mathbb{z}^{S-1})\|^2.$$

□

Следствие 1. *Пусть выполнены Предположения 1, 2, 3, 4. Тогда Алгоритм 1 с* $\gamma = \frac{1}{8\ell\left(1+\frac{\alpha}{n}\right)}$ *и* $K = \frac{30\ell\left(1+\frac{\alpha}{n}\right)}{\mu}$ *для достижения -точности, где* $\varepsilon^2 \sim E\|F(\mathbb{z}^S)\|^2$, *требует*

$$O\left(\left[1+\delta\frac{\ell}{\mu}\left(1+\frac{\alpha}{n}\right)\right]log_2\frac{\|F(z^0)\|^2}{\varepsilon^2}\right)$$

пересылок градиента для каждого устройства.

Доказательство. Из Теоремы 1:

$$E\|F(\mathbb{z}^S)\|^2 \leq \left(\frac{1}{2}\right)^S E\|F(z^0)\|^2.$$

Тогда для достижения точности $\varepsilon^2 \sim E\|F(\mathbb{z}^S)\|^2$ нам нужно следующее число внешних итераций (эпох) S:

$$S = O\left(log_2\frac{\|F(z^0)\|^2}{\varepsilon^2}\right).$$

На каждой внешней итерации полный оператор пересылается только один раз, а в течении остальных $K-1$ внутренних итераций используются сжатые версии размера δ (Предположение 4). Тогда каждое устройство пересылает следующее количество градиентов:

$$S\times\left(1+\delta\times(K-1)\right) = O\left(\left[1+\delta\frac{\ell}{\mu}\left(1+\frac{\alpha}{n}\right)\right]log_2\frac{\|F(z^0)\|^2}{\varepsilon^2}\right).$$

□

Замечание 1. *Выбирая* $\delta \leq \frac{1}{\alpha}$ *и* $\alpha = n$, *Следствие 1 дает следующую оценку сложности коммуникаций:*

$$O\left(\left[1+\frac{\ell}{\mu n}\right]log_2\frac{\|F(z^0)\|^2}{\varepsilon^2}\right).$$

Сравнивая полученный результат с другими методами, отметим, что в работе [70] была получена такая же оценка сходимости метода DIANA для кокоэрсивных вариационных неравенств.

6. Эксперименты

В данном разделе демонстрируется практическое применение предложенного алгоритма. Наша основная цель – оценить эффективность различных методов для решения вариационных неравенств с кокоэрсивными свойствами. В частности, проводится сравнение производительности метода с несмещенной компрессией, метода с квантизацией [74] и метода без компрессии для алгоритма MARINA.

Рассмотрим задачу седловой точки конечной суммы, определяемую следующим образом:

$$g(x,y) = \frac{1}{n}\sum_{i=1}^{n}\left[x^\top A_i y + a_i^\top x + b_i^\top y + \frac{\lambda}{2}\| x \|^2 - \frac{\lambda}{2}\| y \|^2\right],$$

где $A_i \in R^{d\times d}$, $a_i, b_i \in R^d$. Данная задача является -сильно выпуклой по x и λ-сильно вогнутой по y, а также L-гладкой с $L = \| A \|_2$, где $A = \frac{1}{n}\sum_{i=1}^{n} A_i$.

Положим $n = 10$, $d = 100$, и $\lambda = 1$. Матрицы A_i и векторы a_i, b_i генерируются случайным образом. Примечательно, что константа кокоэрсивности для данной задачи определяется как $\ell = \frac{\|A\|_2^2}{\lambda}$. Эти параметры позволяют исследовать поведение алгоритмов в различных условиях.

В качестве критерия возьмем квадрат нормы градиента на k −ой итерации, отнесенного к квадрату нормы функционала на первой итерации: $\left\|\frac{F(z^k)}{F(z^0)}\right\|^2$. Обозначим $\delta = \frac{K}{d}$ – отношение числа координат в компрессии, которые выбираются, к общей размерности задачи. Для экспериментов использовалась квантизация для перевода чисел из формата fp-32 в формат int-8, что позволяет значительно оптимизировать модели за счет уменьшения размера весов модели в 4 раза и того, что многие процессоры эффективнее обрабатывают 8-битные данные [75-76]. Метод квантизации для удобства на графиках обозначаем Q-MARINA. По оси абсцисс отображается объем передаваемой информации в килобитах, что позволяет оценить эффективность алгоритмов с учетом ограничений на передачу данных. Для комплексной оценки методов проектируются три экспериментальных сценария с разными уровнями кокоэрсивности: низким ($\ell \approx 10^2$), средним ($\ell \approx 10^3$) и высоким ($\ell \approx 10^4$).

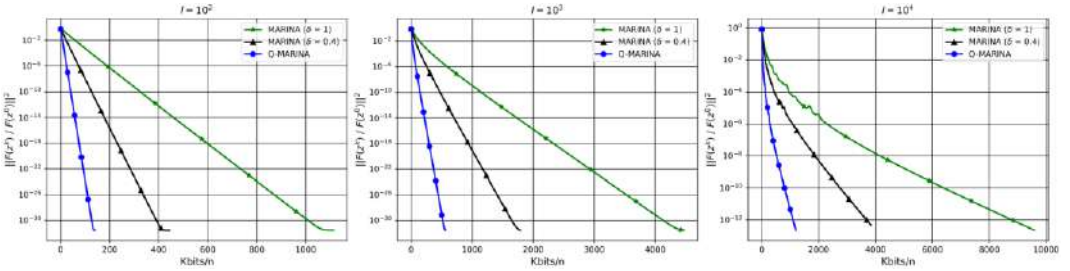


Рис. 1. Сравнение производительности методов на основе метода MARINA для кокоэрсивных вариационных неравенств на билинейной задаче седловой точки.
Fig. 1. Performance comparison of MARINA-based methods for cocoersive variational inequalities on the bilinear saddle point problem.

Результаты экспериментов представлены на рис. 1. Как видно из графиков, методы MARINA с компрессией стабильно превосходит методы без нее во всех сценариях. В то время как сжатие RAND-K демонстрирует приемлемую производительность, оно уступает квантизации. В свою очередь, метод без сжатия показывает значительно более медленную сходимость с точки зрения объемов передаваемой информации, что подчеркивает его ограничения при решении задач больших размерностей.

Таким образом, эксперименты подтверждают эффективность предложенного подхода к решению вариационных неравенств. Метод MARINA с компрессией обеспечивает

значительное сокращение объемов передаваемой информации при сохранении скорости сходимости. Это делает его перспективным инструментом для распределенных систем, где ограниченные ресурсы передачи данных являются критическим фактором.

Список литературы / References

[1]. F. E. Browder, “Nonexpansive nonlinear operators in a banach space,” Proceedings of the National Academy of Sciences, vol. 54, no. 4, pp. 1041–1044, 1965.

[2]. P. Kairouz et al., “Advances and open problems in federated learning,” Foundations and trends in machine learning, vol. 14, no. 1–2, pp. 1–210, 2021.

[3]. T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” IEEE signal processing magazine, vol. 37, no. 3, pp. 50–60, 2020.

[4]. J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, “A survey on distributed machine learning,” Acm computing surveys (csur), vol. 53, no. 2, pp. 1–33, 2020.

[5]. J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” arXiv preprint arXiv:1610.02527, 2016.

[6]. V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning,” Advances in neural information processing systems, vol. 30, 2017.

[7]. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.

[8]. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436–444, 2015.

[9]. A. Beznosikov, V. Samokhin, and A. Gasnikov, “Distributed saddle-point problems: Lower bounds, near-optimal and robust algorithms,” arXiv preprint arXiv:2010.13112, 2020.

[10]. I. Goodfellow et al., “Generative adversarial networks,” Communications of the ACM, vol. 63, no. 11, pp. 139–144, 2020.

[11]. X. Liu et al., “Adversarial training for large neural language models,” arXiv preprint arXiv:2004.08994, 2020.

[12]. C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, “Freelb: Enhanced adversarial training for natural language understanding,” arXiv preprint arXiv:1909.11764, 2019.

[13]. F. Bach, J. Mairal, and J. Ponce, “Convex sparse matrix factorizations,” arXiv preprint arXiv:0812.1869, 2008.

[14]. E. Esser, X. Zhang, and T. F. Chan, “A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science,” SIAM Journal on Imaging Sciences, vol. 3, no. 4, pp. 1015–1046, 2010.

[15]. A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” Journal of mathematical imaging and vision, vol. 40, pp. 120–145, 2011.

[16]. A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, Robust optimization, vol. 28. Princeton university press, 2009.

[17]. J. Von Neumann and O. Morgenstern, Theory of games and economic behavior: By j. Von neumann and o. morgenstern. Princeton university press, 1953.

[18]. F. Facchinei and J.-S. Pang, Finite-dimensional variational inequalities and complementarity problems. Springer, 2003.

[19]. D. Kovalev, A. Beznosikov, E. Borodich, A. Gasnikov, and G. Scutari, “Optimal gradient sliding and its application to optimal distributed optimization under similarity,” Advances in Neural Information Processing Systems, vol. 35, pp. 33494–33507, 2022.

[20]. D. Medyakov, G. Molodtsov, A. Beznosikov, and A. Gasnikov, “Optimal data splitting in distributed optimization for machine learning,” in Doklady mathematics, Pleiades Publishing Moscow, 2023, pp. S465–S475.

[21]. Y. Nesterov, “Efficiency of coordinate descent methods on huge-scale optimization problems,” SIAM Journal on Optimization, vol. 22, no. 2, pp. 341–362, 2012.

[22]. A. Beznosikov, S. Horváth, P. Richtárik, and M. Safaryan, “On biased compression for distributed learning,” Journal of Machine Learning Research, vol. 24, no. 276, pp. 1–50, 2023.

[23]. P. Richtárik and M. Takáč, “Distributed coordinate descent method for learning with big data,” Journal of Machine Learning Research, vol. 17, no. 75, pp. 1–25, 2016.

[24]. A. T. Suresh, X. Y. Felix, S. Kumar, and H. B. McMahan, “Distributed mean estimation with limited communication,” in International conference on machine learning, PMLR, 2017, pp. 3329–3337.

[25]. J. Wu, W. Huang, J. Huang, and T. Zhang, “Error compensated quantized SGD and its applications to large-scale distributed optimization,” in International conference on machine learning, PMLR, 2018, pp. 5325–5333.
[26]. J. Wangni, J. Wang, J. Liu, and T. Zhang, “Gradient sparsification for communication-efficient distributed optimization,” Advances in Neural Information Processing Systems, vol. 31, 2018.
[27]. W. Wen et al., “Terngrad: Ternary gradients to reduce communication in distributed deep learning,” Advances in neural information processing systems, vol. 30, 2017.
[28]. D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-efficient SGD via gradient quantization and encoding,” Advances in neural information processing systems, vol. 30, 2017.
[29]. E. Gorbunov, “Distributed and stochastic optimization methods with gradient compression and local steps,” arXiv preprint arXiv:2112.10645, 2021.
[30]. R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” Advances in neural information processing systems, vol. 26, 2013.
[31]. N. Roux, M. Schmidt, and F. Bach, “A stochastic gradient method with an exponential convergence _rate for finite training sets,” Advances in neural information processing systems, vol. 25, 2012.
[32]. A. Defazio, F. Bach, and S. Lacoste-Julien, “SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives,” Advances in neural information processing systems, vol. 27, 2014.
[33]. L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, “SARAH: A novel method for machine learning problems using stochastic recursive gradient,” in International conference on machine learning, PMLR, 2017, pp. 2613–2621.
[34]. A. Beznosikov and M. Takáč, “Random-reshuffled SARAH does not need full gradient computations,” Optimization Letters, vol. 18, no. 3, pp. 727–749, 2024.
[35]. K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik, “Distributed learning with compressed gradient differences,” Optimization Methods and Software, pp. 1–16, 2024.
[36]. S. Horvóth, C.-Y. Ho, L. Horvath, A. N. Sahu, M. Canini, and P. Richtárik, “Natural compression for distributed deep learning,” in Mathematical and scientific machine learning, PMLR, 2022, pp. 129–141.
[37]. F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, “Federated learning with compression: Unified analysis and sharp guarantees,” in International conference on artificial intelligence and statistics, PMLR, 2021, pp. 2350–2358.
[38]. R. Das, A. Acharya, A. Hashemi, S. Sanghavi, I. S. Dhillon, and U. Topcu, “Faster non-convex federated learning via global and local momentum,” in Uncertainty in artificial intelligence, PMLR, 2022, pp. 496–506.
[39]. E. Gorbunov, K. P. Burlachenko, Z. Li, and P. Richtárik, “MARINA: Faster non-convex distributed learning with compression,” in International conference on machine learning, PMLR, 2021, pp. 3788–3798.
[40]. A. Juditsky, A. Nemirovski, and C. Tauvel, “Solving variational inequalities with stochastic mirror-prox algorithm,” Stochastic Systems, vol. 1, no. 1, pp. 17–58, 2011.
[41]. G. Gidel, H. Berard, G. Vignoud, P. Vincent, and S. Lacoste-Julien, “A variational inequality perspective on generative adversarial networks,” arXiv preprint arXiv:1802.10551, 2018.
[42]. Y.-G. Hsieh, F. Iutzeler, J. Malick, and P. Mertikopoulos, “On the convergence of single-call stochastic extra-gradient methods,” in Advances in neural information processing systems, Curran Associates, Inc., 2019, pp. 6938–6948.
[43]. K. Mishchenko, D. Kovalev, E. Shulgin, P. Richtárik, and Y. Malitsky, “Revisiting stochastic extragradient,” in International conference on artificial intelligence and statistics, PMLR, 2020, pp. 4573–4582.
[44]. Y.-G. Hsieh, F. Iutzeler, J. Malick, and P. Mertikopoulos, “Explore aggressively, update conservatively: Stochastic extragradient methods with variable stepsize scaling,” in Advances in Neural Information Processing Systems, 2020, pp. 16223–16234.
[45]. E. Gorbunov, H. Berard, G. Gidel, and N. Loizou, “Stochastic extragradient: General analysis and improved rates,” in International conference on artificial intelligence and statistics, PMLR, 2022, pp. 7865–7901.
[46]. A. Beznosikov, B. Polyak, E. Gorbunov, D. Kovalev, and A. Gasnikov, “Smooth monotone stochastic variational inequalities and saddle point problems: A survey,” European Mathematical Society Magazine, no. 127, pp. 15–28, 2023.
[47]. A. Beznosikov, S. Samsonov, M. Sheshukova, A. Gasnikov, A. Naumov, and E. Moulines, “First order methods with markovian noise: From acceleration to variational inequalities,” Advances in Neural Information Processing Systems, vol. 36, 2024.

[48]. V. Solodkin, A. Veprikov, and A. Beznosikov, “Methods for optimization problems with markovian stochasticity and non-euclidean geometry,” arXiv preprint arXiv:2408.01848, 2024.
[49]. B. Palaniappan and F. Bach, “Stochastic variance reduction methods for saddle-point problems,” in Advances in neural information processing systems, 2016, pp. 1416–1424.
[50]. T. Chavdarova, G. Gidel, F. Fleuret, and S. Lacoste-Julien, “Reducing noise in gan training with variance reduced extragradient,” in Advances in Neural Information Processing Systems, 2019, pp. 393–403.
[51]. A. Alacaoglu, Y. Malitsky, and V. Cevher, “Forward-reflected-backward method with variance reduction,” Computational Optimization and Applications, vol. 80, Nov. 2021, doi: 10.1007/s10589-021-00305- 3.
[52]. A. Alacaoglu and Y. Malitsky, “Stochastic variance reduction for variational inequality methods,” arXiv preprint arXiv:2102.08352, 2021.
[53]. D. Kovalev, A. Beznosikov, A. Sadiev, M. Persiianov, P. Richtárik, and A. Gasnikov, “Optimal algorithms for decentralized stochastic variational inequalities,” Advances in Neural Information Processing Systems, vol. 35, pp. 31073–31088, 2022.
[54]. A. Beznosikov, A. Gasnikov, K. Zainulina, A. Maslovskiy, and D. Pasechnyuk, “A unified analysis of variational inequality methods: Variance reduction, sampling, quantization andCoordinate descent,” arXiv preprint arXiv:2201.12206, 2022.
[55]. A. Pichugin, M. Pechin, A. Beznosikov, A. Savchenko, and A. Gasnikov, “Optimal analysis of method with batching for monotone stochastic finite-sum variational inequalities,” in Doklady mathematics, Springer, 2023, pp. S348–S359.
[56]. A. Pichugin, M. Pechin, A. Beznosikov, V. Novitskii, and A. Gasnikov, “Method with batching for stochastic finite-sum variational inequalities in non-euclidean setting,” Chaos, Solitons & Fractals, vol. 187, p. 115396, 2024.
[57]. D. Medyakov, G. Molodtsov, E. Grigoriy, E. Petrov, and A. Beznosikov, “Shuffling heuristic in variational inequalities: Establishing new convergence guarantees,” in International conference on computational optimization,
[58]. L. Chen, B. Yao, and L. Luo, “Faster stochastic algorithms for minimax optimization under polyak-{\l} ojasiewicz condition,” Advances in Neural Information Processing Systems, vol. 35, pp. 13921–13932, 2022.
[59]. A. Beznosikov and A. Gasnikov, “Sarah-based variance-reduced algorithm for stochastic finite-sum cocoercive variational inequalities,” in Data analysis and optimization: In honor of boris mirkin’s 80th birthday, Springer, 2023, pp. 47–57.
[60]. W. Liu, A. Mokhtari, A. Ozdaglar, S. Pattathil, Z. Shen, and N. Zheng, “A decentralized proximal point-type method for saddle point problems,” arXiv preprint arXiv:1910.14380, 2019.
[61]. M. Liu et al., “A decentralized parallel algorithm for training generative adversarial nets,” Advances in Neural Information Processing Systems, vol. 33, pp. 11056–11070, 2020.
[62]. I. Tsaknakis, M. Hong, and S. Liu, “Decentralized min-max optimization: Formulations, algorithms and applications in network poisoning attack,” in ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2020, pp. 5755–5759.
[63]. Y. Deng and M. Mahdavi, “Local stochastic gradient descent ascent: Convergence analysis and communication efficiency,” in International conference on artificial intelligence and statistics, PMLR, 2021, pp. 1387–1395.
[64]. A. Beznosikov, G. Scutari, A. Rogozin, and A. Gasnikov, “Distributed saddle-point problems under data similarity,” Advances in Neural Information Processing Systems, vol. 34, pp. 8172–8184, 2021.
[65]. A. Beznosikov, P. Dvurechenskii, A. Koloskova, V. Samokhin, S. U. Stich, and A. Gasnikov, “Decentralized local stochastic extra-gradient for variational inequalities,” Advances in Neural Information Processing Systems, vol. 35, pp. 38116–38133, 2022.
[66]. A. Beznosikov, P. Richtárik, M. Diskin, M. Ryabinin, and A. Gasnikov, “Distributed methods with compressed communication for solving variational inequalities, with theoretical guarantees,” Advances in Neural Information Processing Systems, vol. 35, pp. 14013–14029, 2022.
[67]. A. Beznosikov and A. Gasnikov, “Compression and data similarity: Combination of two techniques for communication-efficient solving of distributed variational inequalities,” in International conference on optimization and applications, Springer, 2022, pp. 151–162.
[68]. A. Beznosikov, M. Takác, and A. Gasnikov, “Similarity, compression and local steps: Three pillars of efficient communications for distributed variational inequalities,” Advances in Neural Information Processing Systems, vol. 36, 2024.

[69]. N. Loizou, H. Berard, G. Gidel, I. Mitliagkas, and S. Lacoste-Julien, “Stochastic gradient descent-ascent and consensus optimization for smooth games: Convergence analysis under expected co-coercivity,” Advances in Neural Information Processing Systems, vol. 34, pp. 19095–19108, 2021.
[70]. A. Beznosikov, E. Gorbunov, H. Berard, and N. Loizou, “Stochastic gradient descent-ascent: Unified theory and new efficient methods,” in International conference on artificial intelligence and statistics, PMLR, 2023, pp. 172–235.
[71]. E. Gorbunov, F. Hanzely, and P. Richtárik, “A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent,” in International conference on artificial intelligence and statistics, PMLR, 2020, pp. 680–690.
[72]. H. Robbins and S. Monro, “A stochastic approximation method,” The annals of mathematical statistics, pp. 400–407, 1951.
[73]. E. Moulines and F. Bach, “Non-asymptotic analysis of stochastic approximation algorithms for machine learning,” Advances in neural information processing systems, vol. 24, 2011.
[74]. B. Jacob et al., “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2704–2713.
[75]. R. Krishnamoorthi, “Quantizing deep convolutional networks for efficient inference: A whitepaper,” arXiv preprint arXiv:1806.08342, 2018.
[76]. H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, “Integer quantization for deep learning inference: Principles and empirical evaluation,” arXiv preprint arXiv:2004.09602, 2020.

Информация об авторах / Information about authors

Даниил Олегович МЕДЯКОВ – магистрант Московского физико-технического института, лаборант лаборатории проблем федеративного обучения, старший лаборант исследовательского центра доверенного искусственного интеллекта Института системного программирования РАН. Сфера научных интересов: стохастическая оптимизация, распределенное и федеративное обучение.

Daniil Olegovich MEDYAKOV – master student at Moscow Institute of Physics and Technology, laboratory assistant at the Laboratory of Federated Learning Problems, senior laboratory assistant at the Research Center of Trusted Artificial Intelligence of the Institute for System Programming of the Russian Academy of Sciences. Research interests: stochastic optimization, distributed and federated learning.

Глеб Львович МОЛОДЦОВ – магистрант Московского физико-технического института, лаборант лаборатории проблем федеративного обучения Института системного программирования РАН. Сфера научных интересов: стохастическая оптимизация, распределенное и федеративное обучение.

Gleb Lvovich MOLODTSOV – master student at Moscow Institute of Physics and Technology, laboratory assistant at the Laboratory of Federated Learning Problems of the Institute for System Programming of the Russian Academy of Sciences. Research interests: stochastic optimization, distributed and federated learning.

Александр Николаевич БЕЗНОСИКОВ – кандидат физико-математических наук, заведующий лабораторией проблем федеративного обучения Института системного программирования РАН, доцент кафедры математических основ управления Московского физико-технического института. Сфера научных интересов: стохастическая оптимизация, распределенное и федеративное обучение.

Aleksandr Nikolaevich BEZNOSIKOV – Cand. Sci. (Phys.-Math.), Head of the Laboratory of Federated Learning Problems of the Institute for System Programming of the Russian Academy of Sciences, Associate Professor of the Department of Mathematical Foundations of Management for the Moscow Institute of Physics and Technology. Research interests: stochastic optimization, distributed and federated learning.

DOI: 10.15514/ISPRAS-2024-36(5)-8

CC BY

Дилемма защитника: совместимы ли методы защиты от разных атак на модели машинного обучения?

[1,2] *Г.В. Сазонов, ORCID: 0009-0003-0905-534X <sazonovg@ispras.ru>*
[1, 3, 4] *К.С. Лукьянов, ORCID: 0009-0009-5235-2175 <lukyanov.k@ispras.ru>*
[2] *И.Н. Мелешин, ORCID: 0009-0009-1541-3418 <igor.meleshin@graphics.cs.msu.ru>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*
[2] *Московский государственный университет имени М.В. Ломоносова, Россия, 119991, Москва, Ленинские горы, д. 1.*
[3] *Московский физико-технический институт (НИУ), Россия 117303, Москва, ул. Керченская, д.1 А, корп. 1*
[4] *Исследовательский центр доверенного искусственного интеллекта ИСП РАН, Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*

Аннотация. В условиях растущего применения моделей искусственного интеллекта (ИИ) всё больше внимания уделяется вопросам доверия и безопасности систем, использующих ИИ от разных типов угроз (атаки уклонения, отравления, вывод о членстве и т.д.). В этой работе мы сосредотачиваемся на задаче классификации вершин графов, выделяя ее как одну из самых сложных. Эта работа является первой, насколько нам известно, в которой исследуется взаимосвязь методов защиты моделей ИИ от разных типов угроз на графовых данных. Наши эксперименты проводятся на наборах данных: цитирования и графов покупок. Мы показываем, что в общем случае нельзя просто использовать комбинации методов защит от разных типов угроз и, что это может иметь серьезные негативные последствия вплоть до полной потери эффективности модели. Мы также приводим теоретическое доказательство противоречия класса методов защиты от атак отравления на графах и состязательного обучения.

Ключевые слова: Атаки на модели искусственного интеллекта; защищенность; классификация вершин графов; доверенный искусственный интеллект.

Для цитирования: Сазонов Г.В., Лукьянов К.С., Мелешин И.Н. Дилемма защитника: совместимы ли методы защиты от разных атак на модели машинного обучения? Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 109–126. DOI: 10.15514/ISPRAS–2024–36(5)–8.

Благодарности: Работа выполнена при поддержке грантом для исследовательских центров в области искусственного интеллекта, представленным Аналитическим центром в соответствии с договором о предоставлении субсидии (идентификатор договора 000000D730321P5Q0002) и договором с Институтом системного программирования им. В. П. Иванникова от 02 ноября 2021 г. № 70-2021-00142.

The Defender's Dilemma: Are Defense Methods Against Different Attacks on Machine Learning Models Compatible?

[1, 2] *G.V. Sazonov ORCID: 0009-0003-0905-534X <sazonovg@ispras.ru>*
[1, 3, 4] *K.S. Lukyanov ORCID: 0009-0009-5235-2175 <lukyanov.k@ispras.ru>*
[2] *I.N. Meleshin ORCID: 0009-0009-1541-3418 <igor.meleshin@graphics.cs.msu.ru>*
[1] *Ivannikov Institute of System Programming of the Russian Academy of Sciences, 25, A. Solzhenitsyn str., Moscow, 109004, Russia.*
[2] *Lomonosov Moscow State University, Leninskie Gory, 1, Moscow, 119991, Russia.*
[3] *Moscow Institute of Physics and Technology (National Research University), building 1, 1 A, Kerchenskaya str., Moscow, 117303, Russia.*
[4] *Research Center for Trusted Artificial Intelligence ISP RAS, 25, A. Solzhenitsyn str., Moscow, 109004, Russia.*

Abstract. With the increasing use of artificial intelligence (AI) models, more attention is being paid to issues of trust and security in AI systems against various types of threats (evasion attacks, poisoning, membership inference, etc.). In this work, we focus on the task of graph node classification, highlighting it as one of the most complex. To the best of our knowledge, this is the first study exploring the relationship between defense methods for AI models against different types of threats on graph data. Our experiments are conducted on citation and purchase graph datasets. We demonstrate that, in general, it is not advisable to simply combine defense methods for different types of threats, as this can lead to severe negative consequences, including a complete loss of model effectiveness. Furthermore, we provide theoretical proof of the contradiction between defense methods against poisoning attacks on graphs and adversarial training

Keywords: AI model attacks; security; graph node classification; trusted AI.

For citation: G.V. Sazonov, Lukyanov K.S., Meleshin I.N. The Defender's Dilemma: Are Defense Methods Against Different Attacks on Machine Learning Models Compatible? *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024, pp. 109-126 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-8.

Acknowledgements: The work was supported by a grant for research centers in the field of artificial intelligence, presented by the Analytical Center in accordance with the subsidy agreement (000000D730321P5Q0002 agreement identifier) and the agreement with the Ivannikov Institute for System Programming dated November 02, 2021, No. 70-2021-00142.

1. Введение

Устойчивость глубоких нейронных сетей к входным возмущениям является важнейшим свойством для их интеграции в различные отрасли, требующие безопасности, такие как беспилотные автомобили, медицинская диагностика и финансы. Хотя нейронные сети должны выдавать схожие результаты для схожих входных данных, давно известно, что они уязвимы для состязательных возмущений [1] – небольших, вычисленных преобразований входных данных, которые не изменяют семантику входного объекта, но заставляют модель выдавать предопределенное решение. Наиболее популярный класс атак – это атаки уклонения [2-4] направленные на обман моделей во время эксплуатации. Большинство методов изучения состязательной устойчивости нейронных сетей направлены на создание состязательных возмущений, которые указывают на то, что в целом прогнозы нейронной сети ненадежны.

Также, известны атаки отравления [5-6] направленные на модификацию наборов данных, обучение на котором может приводить к общему снижению качества модели или появлению у нее так называемых бэкдоров, триггеров, появление которого в данных провоцирует модель сделать определенный неверный прогноз выгодный нарушителю. Против атак отравления в литературе предложены свои методы защиты [7-8], который позволяет выполнить особую

предобработку данных перед обучением, чтобы избавиться от подозрительных данных и/или модифицировать данные.

Несмотря на эффективность методов защиты от различных атак определенного класса, против которого они были разработаны, на данный момент нет четких рекомендаций как эффективно защитится от нескольких атак одновременно. В качестве наиболее простого и очевидного решения можно предложить просто комбинировать лучшие методы защиты от классов угроз, против которых хочется иметь защиту. Однако, в этом исследовании мы показываем, что такой подход может быть малоэффективным и приводить к серьезным негативным последствиям. Причем такой подход помимо того, что не позволяет защититься от атак различной природы, но и разрушает защиту от атак, которым модель с одним типом защит успешно противостояла, а в худшем случае модель вообще становится непригодной для использования даже без каких-либо угроз и атак.

Наш вклад можно представить следующим образом:

1. Насколько нам известно, мы первые кто исследует совместимость методов защит от разных типов угроз на графовых данных;
2. Мы показываем, что в общем случае комбинирование методов защит от разных типов атак приводит к противоречию методов защиты, ухудшая модель и не позволяя эффективно защититься ни от одной из угроз;
3. Мы приводим теоретическое доказательство противоречия класса методов защиты от атак отравления на графах и состязательного обучения;
4. Мы экспериментально подтверждаем все выкладки на двух графовых доменах.

Структура статьи организована следующим образом. Во втором разделе представлен обзор литературы, в котором рассматриваются основные подходы и результаты, достигнутые в данной области исследования. Третий раздел посвящен постановке задач, где формулируются основные определения и цели исследования, которые оно призвано решить. В четвертом разделе описывается методология, применяемая для достижения поставленных целей. Пятый раздел содержит описание проведенных экспериментов и анализ полученных результатов. В шестом разделе рассматриваются ограничения предложенных подходов и их потенциальное влияние на результаты. Наконец, седьмой раздел содержит заключение, в котором подводятся итоги работы, а также намечаются возможные направления для будущих исследований.

2. Обзор литературы

В этом разделе приводится краткий обзор существующих методов интерпретации, атак черного ящика и защит от состязательных атак применительно к моделям, работающим с графовыми данными.

Существуют разные подходы к интерпретации данных. Наиболее распространенным в литературе является апостериорный подход к интерпретации, когда модель сначала обучается, а уже только после этого применяются методы интерпретации. Методы апостериорной интерпретации для моделей, работающих с графовыми данными, можно разделить на три основные группы: методы, основанные на внимании, распространении значимости и градиентные методы. Каждый подход по-своему объясняет предсказания графовых моделей, уделяя особое внимание топологии и структуре графов.

Все атаки можно разделить на разные классы угроз, которые различаются моментом атаки в большом цикле разработки и эксплуатации моделей машинного обучения, а также, по задачи атаки. Так атаки уклонения [9-11] направлены на нарушение работы модели путём манипуляции входными данными на этапе эксплуатации модели. Атаки отравления модифицируют данные до обучения существуют атаки, нацеленные на общее снижение эффективности модели после обучения [5-6] и есть атаки отравления, нацеленные на

внедрения триггеров [12-13]. Также, есть атаки, направленные на нарушение приватности, например, класс атак вывода о членстве [14-15], цель которых понять какие данные использовались для обучения модели. Есть атаки похищения модели [16], цель которых извлечь примерные веса закрытой модели посредством запросов или обучить свою модель затратив значительно меньшее вычислительные мощности, чем требуется для обучения похищаемой модели. Есть и другие классы атак [17-18]. Однако, в этой работе мы сосредоточились на исследовании атак отравления, понижающих общую эффективность после обучения и атаках уклонения.

Целевые и нецелевые атаки на узлы и подграфы – эти методы нацелены на изменение конкретных узлов и рёбер, чтобы повлиять на предсказания. Nettack [10] – пример целевой атаки, которая изменяет рёбра и признаки отдельных узлов для нарушения предсказаний. FGSM (Fast Gradient Sign Method) [9], является примером целевой атаки, применяющей градиентные оценки для создания искажений на уровне вершин графов. Переносимые атаки – атаки, направленные на создание искажений, которые будут эффективны для нескольких моделей одновременно. Атаки чёрного ящика – атаки, где злоумышленник не имеет доступа к параметрам модели и полагается на изменения в предсказаниях, чтобы провести атаку. RL-S2V [11] – это метод, использующий обучение с подкреплением для атаки на графовые модели без знания их внутренних параметров, полагаясь только на доступ к предсказаниям модели.

Эти подходы к построению различных типов атак уклонения показывают, как изменения в графовой структуре могут нарушить предсказания, подчёркивая необходимость в разработке надёжных защитных механизмов, способных противостоять таким атакам.

Атаки CLGA [6] и Mettack [5] являются примерами атак отравления, направленных на графовые данные, с различными стратегиями для достижения максимального воздействия на модели графового обучения. CLGA – метод ориентирован на использование контрастивных потерь для внесения искажений в графовые данные на этапе обучения. Этот подход основан на том, чтобы злоумышленник изменял рёбра графа, минимизируя качество представлений, получаемых в ходе обучения методом без учителя. Это достигается с помощью обратного распространения градиентов через функцию контрастивных потерь, что делает искажения более целенаправленными и эффективными, даже в отсутствии меток. Mettack – атака отравления, основанная на метаобучении, предложенная в работе [5]. Основная идея состоит в том, чтобы оптимизировать структуру графа или признаки его узлов с учетом предстоящего обучения модели. Mettack моделирует процесс обучения целевой модели как вложенную оптимизационную задачу, где внешняя оптимизация направлена на внесение искажений, ухудшающих итоговые предсказания. Этот метод поддерживает как целевые, так и нецелевые атаки и может быть применен к различным архитектурам графовых нейронных сетей, что делает его переносимым.

Эти подходы к построению различных типов атак отравления показывают, как изменения в графовой структуре могут снижать ее эффективность при обучении после применения этих типов атак на сырые данные.

Причем, методы, которые модифицируют данные после обучения модели с целью обмануть модель являются атаками уклонения, а методы атаки, применяемые к исходным данным до обучения модели, являются атаками отравления.

Методы защиты от атак на графы можно классифицировать по их подходам к устойчивости: защита на уровне данных, включающая фильтрацию и корректировку графа, архитектурные методы, повышающие устойчивость моделей к искажениям, и регуляризация, предотвращающая зависимость от отдельных узлов или рёбер. Основные стратегии включают фильтрацию, добавление шума и устойчивые к атакам архитектуры. Основное внимание в этой работе будет уделено методом, позволяющим противостоять состязательным атакам.

Фильтрация и корректировка данных – метод, который на уровне данных предотвращает добавление искажений, фильтруя подозрительные ребра и узлы. Jaccard Similarity Filtering [8] выполняет фильтрацию рёбер на основе схожести признаков узлов, препятствуя внедрению вредоносных рёбер и относится к защите от атак отравления. GNN Guard [7] использует механизмы для обнаружения подозрительных рёбер и снижает их влияние на предсказания, что защищает от атак и также относится к методам защиты от атак отравления. Архитектурные изменения – изменение модели для повышения устойчивости к атакам. Robust GCN [19] добавляет регуляризацию и шум на уровне узлов, снижая чувствительность модели к мелким искажениям. Регуляризация и шум – методы, предотвращающие переобучение на отдельных элементах графа, что делает модель менее восприимчивой к атакам. Также к регуляризации можно отнести Adversarial training [20] и Gradient Regularization [21], которые меняют процесс обучения добавляя данные применением атаки во время обучения и ограничивая градиенты, защищая модель от атак уклонения.

Эти защитные подходы позволяют значительно снизить уязвимость графовых моделей от соответствующих типов угроз, сохраняя высокую точность предсказаний и надежность модели при использовании графов.

Соответственно методы защиты, который выполняет модификацию данных до обучения является методами защиты от атак отравления. Методы защит, которые модифицируют поведение/архитектуру модели и/или меняют процедуру обучения, являются методами защиты от атак уклонения.

На основании обзора литературы видно, что существует множество методов атак и защит. В работе [22] было проведено исследования о противоречии методов защит, для моделей классификации изображений. Однако, на данный момент в литературе не представлено как одновременно защищаться от атак разного типа при этом сохранив высокую эффективность, а также не известно характерна ли проблема несовместимости методов защит от разных типов данных для моделей, работающих с другими типами данных.

3. Постановка задач

В этом разделе формально вводится постановка задач, вводятся обозначения, используемые в статье, и формулируются вопросы исследования. Введем формальные определения атак уклонения, отравления, защищенных моделей от соответствующих атак и частично защищенной модели.

3.1 Атаки на модель классификации

Атака уклонения (англ. *evasion attack*) для моделей классификации – это процесс, при котором злоумышленник стремится изменить входные данные $x \in R\hat{}d$ таким образом, чтобы модифицированные данные $x' = x + \delta$ обошли классификацию модели $f: R\hat{}d \to Y$, где Y – множество классов.

Целью атаки уклонения является нахождение возмущения $\delta \in R\hat{}d$, удовлетворяющего следующим условиям:

1. $arg⟦max⟧\_(y \in Y)\ f(x') \neq arg⟦max⟧\_(y \in Y)\ f(x)$, то есть предсказанный класс изменяется после применения возмущения.
2. $\| \delta \| \_p \leq \epsilon$, где $\|\cdot\| \_p$ – L_p норма (например, с $p = 2$ или $p = \infty$), а ϵ – заданное ограничение на величину возмущения, чтобы сохранялась правдоподобность измененного входа.

Таким образом, задача атаки уклонения может быть формализована как задача оптимизации: $arg⟦max⟧\_(y \in Y)\ f(x') \neq arg⟦max⟧\_(y \in Y)\ f(x)$, при условии $\| \delta \| \_p \leq \epsilon$

Атаки уклонения делятся на два основных типа:

1. Белый ящик (white-box): Злоумышленник обладает полным доступом к модели f, включая её параметры и градиенты.
2. Чёрный ящик (black-box): Злоумышленник имеет ограниченный доступ к модели f, например, может наблюдать только выходы модели в ответ на определенные входы.

Стоит отметить, что в случае графовых данных атака может модифицировать вместо матрицы признаков матрицу ребер графа $G(V, E)$, где V – количество вершин, E – количество ребер или обе матрицы одновременно.

Атака отравления (англ. *poisoning attack*) – это процесс, при котором злоумышленник модифицирует обучающие данные модели машинного обучения с целью ухудшить её производительность или добиться определённого поведения на новых данных.

Пусть $D\_train = \{(x\_i, y\_i)\}\_(i = 1)\^n$ – исходный набор обучающих данных, где $x\_i \in R\^d$ – входные данные, а $y\_i \in Y$ – метки классов. Атака отравления предполагает создание нового набора данных $D\_poisoned = \{(x\_i', y\_i')\}\_(i = 1)\^n$, такого что:

Обученная на $D\_poisoned$ модель $f\^{}$ демонстрирует ухудшение качества на проверочных данных D_test:

$$L(f\^{}, D\_test) > L(f, D\_test),$$

где L – функция ошибки, а f – модель, обученная на чистых данных D_train.

1. Либо модель $f\^{}$ демонстрирует целенаправленное поведение, выгодное злоумышленнику, например, ошибочную классификацию целевого входа x_target:

$$arg\ (max)\┬(y \in Y)\ f\^{}\ (x\_target) = y\_adversarial,$$

2. где $y\_adversarial$ – целевой класс, заданный злоумышленником.

Задача атаки отравления может быть формализована как задача оптимизации: найти $L(f\^{}, D\_test) > \tau$, или $arg\ (max)\┬(y \in Y)\ f\^{}\ (x\_target) = y\_adversarial$, где τ – порог допустимой ошибки на проверочных данных.

Атаки отравления классифицируются на два основных типа:

- **Целенаправленные (targeted):** Злоумышленник модифицирует данные с целью добиться определенного поведения модели на конкретных входах.
- **Общие (untargeted):** Злоумышленник стремится ухудшить обобщающую способность модели на всех проверочных данных.

3.2 Защищенная модель

Модель $f: R\^d \to Y$ называется защищенной от атак уклонения, если она сохраняет свою устойчивость к малым возмущениям входных данных. Формально, для любого входа $x \in R\^d$ и любого допустимого возмущения $\delta \in R\^d$, удовлетворяющего $\| \delta \| \_p \leq \epsilon$, модель удовлетворяет следующему условию:

$$arg\ (max)\┬(y \in Y)\ f(x + \delta) = arg\ (max)\┬(y \in Y)\ f(x),$$

где $\epsilon > 0$ – заданное ограничение на норму возмущения, а $\|\cdot\| \_p$ – норма p (например, $p = 2$ или $p = \infty$).

Таким образом, защищённая модель f остается инвариантной к возмущениям, которые находятся в пределах заданного ограничения ϵ, обеспечивая корректную классификацию даже в условиях возможного противодействия злоумышленника.

Модель $f: R\^d \to Y$ называется защищенной от атак отравления, если она сохраняет свою способность к корректной классификации проверочных данных $D\_test = \{(x\_i, y\_i)\}\_(i = 1)\^m$, даже если обучающие данные D_train содержат модифицированные элементы, созданные злоумышленником.

Формально, пусть $D\_poisoned = D\_train \cup D\_adv$, где $D\_adv = \{(x\_i', y\_i')\}\_(i = 1)\^k$ – добавленные или измененные злоумышленником данные. Модель f считается защищенной, если:

$$L(f, D\_test) \leq L\_clean + \delta,$$

где L – функция ошибки, L_clean – ошибка модели, обученной на чистых данных, а $\delta \geq 0$ – допустимое отклонение, ограничивающее влияние атакующего.

Кроме того, для целенаправленных атак защищённая модель должна обеспечивать корректную классификацию целевых данных x_target:

$$arg\ (max)\_(y \in Y)\ f(x\_target) = y\_true,$$

где y_true – истинная метка класса.

Стоит отметить, что полной защиты удается добиться в редких случаях, тогда можно говорить, что модель является **частично защищенной**, если:

$$(i)\ \ Q(f, X) - Q(f', X) \leq \alpha,$$
$$(ii)\ \ Q(f', X') - Q(f, X') \geq \beta,$$

где $Q(\cdot,\cdot)$ – функция качества модели на входных данных, α – максимальное допустимое снижение качества на чистых данных, а β – минимально допустимое увеличение качества на атакованных данных относительно незащищенной модели. f – исходная модель, а f' – её защищённая версия. Обозначим $X \in R\^d$ как чистые входные данные, а X' – атакованные данные. Эти условия обеспечивают частичную устойчивость модели к атакам, сохраняя приемлемый уровень производительности.

Вопросы исследования

- Можно ли взять произвольные качественные методы защиты в своих категориях (от атак уклонения или от атак отравления) и получить защищенную модель от обоих типов угроз?
- Какие негативные последствия могут возникать при комбинировании методов защит?

В следующем разделе мы отвечаем на эти вопросы и подкрепляем это соответствующими экспериментами.

4. Методология

В данном разделе представлена методика исследования, направленная на достижение поставленных целей и решение заявленных задач.

4.1 Математическое противоречие методов защиты

В этом подразделе математически показано, что добавление защиты от атак отравления математически противоречит защите состязательного обучения от атаки уклонения на структуру графа.

Пусть задан граф $G(V, E)$, где V – вершины в графе, E - ребра в графе , определена атака уклонения Ev посредством удаления и добавления ребер с параметром $\epsilon; 0 < \epsilon \ll 1$, то есть атака Ev может удалить $k\_1 < E \vee ; G(V, E)$ ребер и добавить $k\_2 < E\_full/E \vee ; G\_full\ ((V, E\_full))/G\ (V, E)$ (где G_full клика построенная на вершинах V, а E_full количество ребер в клике) ребер так, чтобы $0 < k\_1 + k\_2 = k < \epsilon * E$. Состязательная защита AT использует для генерации состязательных примеров ту же атаку Ev с тем же параметром ϵ и определен метод защиты от атак отравления PD с параметром удаления $t; 0 < t < 1$. Далее для простоты количество ребер будет обозначаться как E, а не |E|.

Теорема. При одновременном использовании метода защит от атак отравления PD и состязательного обучения AT от атак уклонения типа Ev верны два утверждения: 1) с увеличением параметра t метода защиты от атак отравления PD уменьшается количество возможных состязательных примеров, которые могут быть найдены во время выполнения состязательного обучения AT; 2) с увеличением параметра t метода защиты от атак отравления PD уменьшается корреляция между распределением количества ребер возмущенных графов во время применения состязательного обучения AT и распределением количества ребер возмущенных графов полученных методом Ev во время выполнения атаки уклонения .

Доказательство 1 утверждения. Параметр t говорит о том, что метод защиты от атак отравления модифицирует граф G посредством удаления из него до $t * E$ ребер. Обозначим количество удаленных ребер как $0 < m * E \leq t * E; 0 < m \leq t < 1$. В результате получается граф $G'(V; (1-m)E)$. Тогда общее количество атак N_attack (модификаций графа отличных от изначального графа G), можно вычислить по следующей формуле:

$$N_{attack} = \sum_{k_1=0}^{k} \left(\binom{E}{k_1} * \sum_{k_2=0}^{k-k_1} \binom{\frac{V*(V-1)}{2} - E}{k_2} \right),$$

где функция $\binom{n}{k} = C_k^n$ и соответствует биномиальному коэффициенту, определяя количества сочетаний из n по k.

В каждом слагаемом первой суммы фиксируется значение k_1 пробегая значения от 0 до k. При каждом фиксированном k_1 надо выбрать k_1 ребер из E для удаления, что определяет биномиальный коэффициент $\binom{E}{k_1}$ и это число умножается на все допустимые варианты добавить k_2 ребер выбранных из дополнения графа G до клики G_full, так как в клике на V вершин $(V * (V-1))/2$ ребер, из которых надо вычесть существующие ребра E, то общее количество таких вариантов при фиксированном k_2 равно $\binom{\frac{V*(V-1)}{2} - E}{k_2}$. k_2 в свою очередь может принимать любое значение от 0 до $k - k\_1$ при заранее определенном значении k_1.

При добавлении метода защиты от атак отравления PD бюджет атаки Ev используемой состязательной защитой AT станет меньше, обозначим бюджет атаки Ev при условии применения метода защиты от атаки отравления PD как $k' = \epsilon * (1-m) * E = \epsilon * E' < k = \epsilon * E$. Теперь заметим, что формула для вычисления N_attack зависит от трех параметров: $V; E; \epsilon$, в случае атаки уклонения, обозначим эту величину за $N\_attack\text{^}evasion$. И от четырех: $V; E; \epsilon; m$, в случае одновременного использования двух методов защит: $AT; PD$, обозначим эту величину как $N\_attack\text{^}(full-defense)$. Тогда можно заметить, что $N\_attack\text{^}evasion > N\_attack\text{^}(full-defense)$. Распишем явно обе формулы:

$$N_{attack}^{evasion}(V; E; \epsilon) = \sum_{k_1=0}^{\epsilon*E} \left(\binom{E}{k_1} * \sum_{k_2=0}^{\epsilon*E-k_1} \binom{\frac{V*(V-1)}{2} - (1-\epsilon)*E}{k_2} \right)$$

$$N_{attack}^{full-defense}(V; E; \epsilon; m) = \sum_{k_1=0}^{\epsilon*(1-m)*E} \left(\binom{E}{k_1} * \sum_{k_2=0}^{\epsilon*(1-m)*E-k_1} \binom{\frac{V*(V-1)}{2} - (1-\epsilon)*E}{k_2} \right)$$

Функция N_attack является убывающей функцией при уменьшении E. Так как $0 < m \leq t < 1$, то $N\_attack\hat{}evasion > N\_attack\hat{}(full - defense)$, что требовалось доказать в первой части теоремы.

Доказательство 2 утверждения.

Пусть вероятность добавить ребро во время генерации состязательного примера методом Ev равна p, вероятность удалить ребро q и вероятность не изменить граф 1-p-q. Тогда случайная величина θ равная изменению количеству ребер в состязательном примере G_attack, полученного в ходе применения метода Ev как атаки уклонения, по отношению к количеству ребер в изначальном графе G имеет среднее и дисперсию:

$$E(\theta) = ((-1) * p + 1 * q + 0 * (1 - p - q))\epsilon E = E\epsilon(q - p)$$
$$E(\theta\hat{}2) = (1 * p + 1 * q + 0 * (1 - p - q))\epsilon E = E\epsilon(q + p)$$
$$V(\theta) = E\epsilon(q + p) + E\epsilon$$

При применении метода защиты PD количество ребер уменьшится на $0 < mE < tE < E$. Тогда перед применением метода Ev в состязательном обучении количество ребер в графе станет равно $E(1 - m)$. Тогда случайная величина ξ равная изменению количеству ребер в состязательном примере G_at, полученного в ходе применения метода Ev в процессе выполнения состязательного обучения AT, по отношению к количеству ребер в изначальном графе после применения метода защиты от атак отравления PD имеет среднее и дисперсию:

$$E(\xi) = ((-1) * p + 1 * q + 0 * (1 - p - q))\epsilon E(1 - m) = E\epsilon(1 - m)(q - p)$$
$$V(\xi) = E\epsilon(1 - m)(q + p) + E\epsilon(1 - m)$$

Случайная величина ξ складывается из $\epsilon(1 - m)E$ случайных реализаций выбора, добавления или пропуску действия, а θ из ϵE. Между ними есть пересечение, равное $\epsilon(1 - mE)$, т.е. ξ полностью входит в θ. Остальная часть θ содержит ϵmE уникальных членов. Ковариация Cov определяется как сумма ковариаций всех пересекающихся членов, так как ковариация между непересекающимися членами равна нулю. Посчитаем ковариацию и корреляцию случайных величин ξ и θ

$$Cov(\xi; \theta) = V(\xi)$$

$$r(\xi; \theta) = \frac{Cov(\xi;\theta)}{\sqrt{V(\xi) * V(\theta)}} = \frac{E\epsilon(1-m)\left(p+q-p^2+2pq-q^2\right)}{\sqrt{E\epsilon(1-m)\left(p+q-p^2+2pq-q^2\right)E\epsilon\left(p+q-p^2+2pq-q^2\right)}} = \frac{1-m}{\sqrt{1-m}} = \sqrt{1 - m}$$

Корреляция $r(\xi; \theta)$ является убывающей функцией с ростом m. При стремлении m к 1 получается полное отсутствие корреляции, а при m стремящимся к 0 получается корреляция близкая к 1, что и требовалось доказать.

Следствие из теоремы 1. Различие структур между возмущенными графами, получаемыми в результате применения состязательного обучения и возмущениями, генерируемыми во время атаки уклонения может привести к тому, что состязательное обучение обеспечит защиту от атак уклонения, но не в той области, в которой необходимо. Это следует из основного принципа состязательного обучения, что модель дообучается на примерах из того же распределения, что и примеры, которые может найти атака. Однако, как было показано выше, добавление защиты от атак отравления построенной на принципе удаления ребер приводит к смещению распределения и к уменьшению количества примеров, которые могут быть найдены.

4.2 Методика комбинирования методов защиты от разных типов угроз

В этой работе были рассмотрены методы атак отравления: CLGA [6] и Mettack [5]; метод атаки уклонения: FGSM [9]; метод защиты от атак отравления: Jaccard Similarity Filtering [8];

методы защиты от атак уклонения: Adversarial training [20] и Gradient Regularization [21]. Все методы, которые были предложены для других типов данных, были адаптированы для работы с графовыми моделями решающие задачу классификации вершин. Однако, на нашей реализации атаки отравления Mettack не удалось воспроизвести удовлетворительные результаты из оригинальной статьи, поэтому из экспериментов он был исключен.

Рассмотрим в какой момент цикла разработки моделей машинного обучения появляются угрозы атак отравления и уклонения, а также, в какой момент необходимо добавлять методы защиты от соответствующих типов угроз. Атака происходит перед обучением модели, после этого должен идти метод защиты от атак отравления, который выполняет функцию очистки/модификаций данных от потенциальных некачественных данных. Атака уклонения, как упоминалось ранее, происходит на этапе эксплуатации модели, что в рамках лабораторных исследований равносильно атаки на тестовый набор данных. Чтобы защититься от атак уклонения необходимо модифицировать процесс обучения. Например, состязательное обучение (Adversarial training) расширяет обучающую выборку самостоятельно сгенерированными атаками, которые генерируются по ходу обучения модели.

Чтобы оценить, как методы защиты влияют друг на друга, сначала измерялось качество модели на исходных данных, затем при наличии только атаки одного типа. Далее измерялось качество при наличии защиты на исходных данных. После чего проводилось обучение с атакой и использованием защиты от атак выбранного типа. И последнее обучение было при наличии обоих типов атак (атак отравления и атак уклонения) и обоих методов защиты соответственно. Эксперименты с атакой и защитой только от другого типа атак не проводились, так как если данные были модифицированы атакой отравления, то защита от атак уклонения не сможет их исправить из-за того, что эти методы предполагают, что данные для обучения качественные. Аналогично наличие защиты от атаки отравления не поможет защититься от атак уклонения, так как атаки уклонения используют градиенты обученной модели, чтобы вычислить шумовую маску для обмана модели на тесте, а так как защита от атак отравления модифицирует данные, то это изменит конечную модель, но не сделает ее устойчивой к шумовой маске, вычисленной на основании градиентов.

5. Эксперименты

В этом разделе описаны эксперименты и все необходимое для их воспроизведения.

5.1 Математическое противоречие методов защиты

5.1.1 Датасеты и обучение моделей

В экспериментах использовались датасеты: Cora – относящийся к домену цитирования [23] и представленных в библиотеке torch-geometric в группе датасетов Planetoid и Photo – домена графов покупок [24], также представленные в torch-geometric в группе датасетов Amazon.

Статистика наборов данных:

- Cora: 2708 вершин, 10556 ребер, 7 классов, 1433 признаков
- Photo: 7650 вершин, 238162 ребер, 7 классов, 745 признаков

В качестве модели обучалась двухслойная модель GCN (GCN-2l) и двухслойная модель GIN (GIN-2l).

```
GCN-2l(
  Sequential(
    (0): GCNConv(input_size, 16)
```

```
            (1): ReLU(inplace)
            (2): GCNConv(16, output_size)
            (3): LogSoftmax(inplace)
        )
    )



    GIN-2l(
        Sequential(
            (0): GINConv(
                Sequential(
                    (0): Linear(input_size, 16)
                    (1): BatchNorm1d(16, eps=1e-05)
                    (2): ReLU(inplace)
                    (3): Linear(16, 16)
                    (4): BatchNorm1d(16, eps=1e-05)
                    (5): ReLU(inplace)
                )
            )
            (1): ReLU(inplace)
            (2): GINConv(
                Sequential(
                        (0): Linear(16, 16) (1):
                        (1): BatchNorm1d(16, eps=1e-05)
                        (2): ReLU(inplace)
                        (3): Linear(16, output_size)
                    )
                )
            (3): LogSoftmax(inplace)
        )
    )
```

Для обучения использовался оптимизатор Adam с параметрами по умолчанию и функция ошибки NLLLoss из библиотеки PyTorch, разделение на батчи не используется. Количество эпох обучения равно 200 для достижения высокой точности классификации на всех наборах данных. Данные были разделены в соотношении 80 к 20 для обучения и тестирования соответственно.

5.1.2 Параметры проведения экспериментов

В экспериментах использовались: метод атаки отравления: CLGA [6]; метод атаки уклонения: FGSM [9]; метод защиты от атак отравления: Jaccard Similarity Filtering [8]; методы защиты от атак уклонения: Adversarial training [20] и Gradient Regularization [21]. Гиперпараметры соответствующих методов представлены в табл. 1.

5.1.3 Протокол оценки

В качестве основной метрики оценки моделей использовалась метрика точности (Accuracy). Для оценки эффективности атак и защит использовалась разница точности моделей при наличии или отсутствии соответствующего метода. Также, в случае неоднозначности выводов при использовании атак по одной метрики использовалась метрика ASR (Average Success Rate; средняя вероятность успеха). Для методов защиты смотрелось как на изменение метрики точности как на исходных данных, так и на атакованных для комплексной оценки

влияния метода защиты на модель.

Табл. 1. Гиперпараметры методов атак и защит.
Table 1. Hyperparameters of attack and defense methods.

Метод	Гиперпараметры
CLGA	learning_rate = 0.01 num_hidden = 256 num_proj_hidden = 32 activation = *prelu* drop_edge_rate_1 = 0.3 drop_edge_rate_2 = 0.4 tau = 0.4 num_epochs = 3000 weight_decay = $1e-5$ drop_scheme = *degree*
FGSM	$\epsilon = 0.005$
Jaccard	threshold = 0.4
Adversarial training	attack_name = *FGSM* $\epsilon = 0.01$
Gradient Regularization	regularization_strength = 50

5.2 Результаты экспериментов

При наличии атаки отравления результат экспериментов усредняются по 5 запускам (так как обучение одной модели с применением атаки отравления занимает до 10 часов), в остальных случаях по 15.

В табл. 2, 3, 4 и 5 представлены результаты экспериментов с использованием различных комбинаций методов атак и защит. Сокращенные названия методов: AdvTrain – защита состязательного обучения; GradReg – защита регуляризацией градиентов. В табл. 2 и 3 представлены результаты на наборе данных Cora, домен цитирования. В табл. 4 и 5 представлены результаты на наборе данных Photo, домен граф покупок. В табл. 2 и 4 представлены результаты модели GCN-2l. В табл. 3 и 5 представлены результаты модели GIN-2l.

Можно заметить, что комбинирования защит на исходных данных не вызывает никаких особых негативных последствий (верхний левый квадрат результатов каждой таблицы), а в некоторых случаях комбинирование защит даже приводит к повышению точности относительно результатов при отдельном использовании каждого из методов защиты. При этом отметим, что каждый из методов защит по отдельности приводит к повышению эффективности при противодействии от своей профильной угрозы. Также заметим, что как только появляются угрозы, то результаты моделей значимо падают при использовании комбинации защит относительно случаев, когда от угрозы защищаются профильным методом защиты (левый нижний и правый верхний квадраты результатов в каждой таблице). Также, видно, что при комбинации угроз качество значимо падает и в 3-х случаях из 4-х становится еще ниже, чем при наличии только одной угрозы (правый нижний квадраты результатов в каждой таблице).

В табл. 2 показана точность модели GCN-2l на наборе данных Cora при различных комбинациях методов атак и защит. Пропуски, обозначенные символом "\textemdash" означают, что эксперимент с соответствующей конфигурацией не проводился. В первой строке указан метод атаки уклонения, его отсутствие подписано как "No attack". В первом столбце указан метод атаки отравление, его отсутствие подписано как "No attack". Во второй

строке указан метод защиты от атак уклонения или его отсутствие, которое помечено как "No defense". Во втором столбце указан метод защиты от атак отравления или его отсутствие, которое помечено как "No defense". В левом верхнем углу указан набор данных и модель.

Табл. 2. Точность модели GCN -2l на наборе данных Cora при различных комбинациях методов атак и защит.
Table 2. Accuracy of the GCN -2l model on the Cora dataset under various attack and defense methods combinations.

Cora	GCN-2l	No attack			FGSM		
		No defense	AdvTrain	GradReg	No defense	AdvTrain	GradReg
No attack	No defense	90 ± 0	88 ± 2	91 ± 1	57 ± 3	78 ± 3	70 ± 4
	Jaccard	77 ± 5	81 ± 5	80 ± 5	46 ± 4	69 ± 5	63 ± 2
CLGA	No defense	71 ± 2	—	—	—	—	—
	Jaccard	76 ± 6	67 ± 3	61 ± 2	—	57 ± 4	42 ± 1

В табл. 3 показана точность модели GIN-2l на наборе данных Cora при различных комбинациях методов атак и защит. Пропуски, обозначенные символом "\textemdash" означают, что эксперимент с соответствующей конфигурацией не проводился. В первой строке указан метод атаки уклонения, его отсутствие подписано как "No attack". В первом столбце указан метод атаки отравление, его отсутствие подписано как "No attack". Во второй строке указан метод защиты от атак уклонения или его отсутствие, которое помечено как "No defense". Во втором столбце указан метод защиты от атак отравления или его отсутствие, которое помечено как "No defense". В левом верхнем углу указан набор данных и модель.

Табл. 3. Точность модели GIN-2l на наборе данных Cora при различных комбинациях методов атак и защит.
Table 3. Accuracy of the GIN-2l model on the Cora dataset under various attack and defense methods combinations.

Cora	GIN-2l	No attack			FGSM		
		No defense	AdvTrain	GradReg	No defense	AdvTrain	GradReg
No attack	No defense	90 ± 3	86 ± 2	85 ± 3	43 ± 3	76 ± 1	59 ± 4
	Jaccard	65 ± 4	80 ± 6	75 ± 4	29 ± 3	66 ± 3	53 ± 4
CLGA	No defense	66 ± 5	—	—	—	—	—
	Jaccard	64 ± 3	65 ± 2	39 ± 7	—	52 ± 2	37 ± 3

В табл. 4 показана точность модели GCN-2l на наборе данных Photo при различных комбинациях методов атак и защит. Пропуски, обозначенные символом "\textemdash" означают, что эксперимент с соответствующей конфигурацией не проводился. В первой строке указан метод атаки уклонения, его отсутствие подписано как "No attack". В первом столбце указан метод атаки отравление, его отсутствие подписано как "No attack". Во второй строке указан метод защиты от атак уклонения или его отсутствие, которое помечено как "No defense". Во втором столбце указан метод защиты от атак отравления или его отсутствие, которое помечено как "No defense". В левом верхнем углу указан набор данных и модель.

В табл. 5 показана точность модели GIN-2l на наборе данных Photo при различных комбинациях методов атак и защит. Пропуски, обозначенные символом "\textemdash" означают, что эксперимент с соответствующей конфигурацией не проводился. В первой строке указан метод атаки уклонения, его отсутствие подписано как "No attack". В первом столбце указан метод атаки отравление, его отсутствие подписано как "No attack". Во второй строке указан метод защиты от атак уклонения или его отсутствие, которое помечено как "No defense". Во втором столбце указан метод защиты от атак отравления или его отсутствие, которое помечено как "No defense". В левом верхнем углу указан набор данных и модель.

Табл. 4. Точность модели GCN-2l на наборе данных Photo при различных комбинациях методов атак и защит.
Table 4. Accuracy of the GCN-2l model on the Photo dataset under various attack and defense methods combinations.

Photo	GCN-2l	No attack			FGSM		
		No defense	AdvTrain	GradReg	No defense	AdvTrain	GradReg
No attack	No defense	94 ± 1	91 ± 1	92 ± 2	90 ± 2	92 ± 2	92 ± 1
	Jaccard	92 ± 2	92 ± 3	92 ± 2	89 ± 1	88 ± 2	87 ± 1
CLGA	No defense	88 ± 2	—	—	—	—	—
	Jaccard	90 ± 1	88 ± 1	87 ± 1	—	85 ± 1	84 ± 1

Табл. 5. Точность модели GIN-2l на наборе данных Photo при различных комбинациях методов атак и защит.
Table 5. Accuracy of the GIN-2l model on the Photo dataset under various attack and defense methods combinations.

Photo	GIN-2l	No attack			FGSM		
		No defense	AdvTrain	GradReg	No defense	AdvTrain	GradReg
No attack	No defense	83 ± 2	85 ± 2	78 ± 7	62 ± 13	81 ± 3	73 ± 3
	Jaccard	65 ± 5	66 ± 7	60 ± 4	56 ± 4	60 ± 10	53 ± 7
CLGA	No defense	58 ± 6	—	—	—	—	—
	Jaccard	64 ± 3	70 ± 8	58 ± 5	—	78 ± 3	65 ± 2

5.3 Обсуждение результатов

Из результатов экспериментов можно сделать выводы, что комбинирование произвольных методов защиты может не просто не помогать защититься от двух и более угроз, а приводить к ухудшению качества и разрушению защиты даже от одной атаки, против которой без дополнительного метода защиты модель успешно справлялась.

Также можно отметить специфические результаты на датасете Photo. В случае модели GCN-2l падение качества было не слишком большим. Это можно объяснить тем, что гиперпараметры всех атак и защит подбирались на обучающей выборке датасета Cora и оставались неизменными для набора данных Photo. Для получения более существенных изменений качества моделей необходимо увеличить возмущения для FGSM атак и увеличить количество итераций для атаки отравления CLGA (так как количество итераций определяет, сколько возмущений в данные может внести этот метод, корректнее вносить изменения в процентном отношении от общего числа элементов графа, а не в абсолютном). А в случае модели GIN-2l можно заметить, что при нескольких угрозах качество растет по сравнению с тем, когда угроза только одна. Это можно объяснить тем, что в силу структурных особенностей графов покупок удаление ребер не позволяет найти уникальные паттерны, так как свертка GIN работает на принципе поиска изоморфных подграфов и в отличии от случая с датасетом Cora атака, скорее всего, приводит к невозможности выделения уникальных изоморфных графов.

6. Ограничения и обсуждение

Стоит отметить, что доказанная в этой работе теорема имеет ограниченное применение и не учитывает возможность атаки не только структуры графа, но и атаки на признаки. Также, класс защиты от отравления ограничен теми, которые только удаляют подозрительные данные. Однако, теории, которая бы однозначно могла сказать, как непротиворечиво надо комбинировать методы защит, на данный момент нет, и предложенная теорема – это первый шаг к созданию рекомендаций, подкрепленных не только экспериментами, но и

математической теорией.

Хочется также отметить, что возможным способом решить проблему смещения распределения при комбинировании методов защит может быть увеличение допустимых возмущений во время состязательного обучения. С практической точки зрения это означает, что если мы считаем атаку незаметной, если ее бюджет возмущений был 5% от всего графа, то можно, например, разрешить возмущения в 10% во время состязательного обучения, таким образом распределение генерируемых возмущенных графов с учетом смещения при комбинировании будет включать в себя и распределение с допустимым возмущением в 5%, но которое не было смещено.

Отдельно отметим, что основной вывод этой работы, что комбинирование методов защиты, который по отдельности были эффективные не гарантирует защиту при комбинации. При этом этот вывод не говорит, что невозможно защитится от нескольких угроз, а говорит, что следует исследовать эту проблематику в будущих работах и аккуратно подходить к построению систем, требующих наличие защищенности от нескольких различных типов угроз.

7. Заключение и будущая работа

В этой статье было впервые показано, что комбинирование произвольных методов защит от разных типов угроз не позволяет защититься от них на графовых данных. Причем наивное комбинирование не просто не позволяет получить модель, защищенную от обеих угроз одновременно, но и может приводить к серьезным последствиям разрушая защиту от угрозы, от которой ранее была защищена. Мы показываем верность этих утверждений для нескольких различных архитектур и нескольких доменов графов (графах цитирования и графах покупок). А также, приведено теоретическое доказательство как класса методов защиты от атак отравления на графах мешает обеспечить защиту посредством состязательного обучения от атак уклонения.

В качестве направлений будущей работы можно выделить исследования большего количества методов и разработку практических рекомендаций эффективных комбинаций, принципы обеспечения защит которых не разрушают защиты друг друга, что может быть оформлено в большой бенчмарк. Более того, можно расширить бенчмарки исследованиями противодействия трем и более типам угроз. Также, можно проверить наличие этой проблемы на других типах данных: текстовые данные, временные ряды и видео. Еще одним направлением может быть создание методов защит, которые изначально были бы способны противостоять сразу нескольким типам угроз, при этом не имеющих внутренних противоречий принципов обеспечения защиты от разных типов угроз. А также, стоит развивать теоретическую базу анализирующую возможность комбинировать методы защит, на основании которой можно будет формировать рекомендации позволяющие обеспечивать одновременную защиту от атак разного типа.

Список литературы / References

[1]. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. In *2nd International Conference on Learning Representations*.

[2]. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. *CoRR*, abs/1412.6572.

[3]. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

[4]. Moosavi-Dezfooli, S.-M., Fawzi, A., & Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2574–2582.

[5]. Zügner, D., & Günnemann, S. (2019). Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations, Workshop Track*. Available at: https://arxiv.org/abs/1902.08412.
[6]. Zhang, S., Chen, H., Sun, X., Li, Y., & Xu, G. (2022). Unsupervised graph poisoning attack via contrastive loss back-propagation. In *Proceedings of the ACM Web Conference 2022*, 1322–1330.
[7]. Zhang, X., & Zitnik, M. (2020). Gnnguard: Defending graph neural networks against adversarial attacks. In *Advances in neural information processing systems*, 33, 9263–9275.
[8]. Wu, H., Wang, C., Tyshetskiy, Y., Docherty, A., Lu, K., & Zhu, L. (2019). Adversarial examples on graph data: Deep insights into attack and defense. *arXiv preprint arXiv:1903.01610*.
[9]. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
[10]. Zügner, D., Akbarnejad, A., & Günnemann, S. (2018). Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2847–2856.
[11]. Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., & Song, L. (2018). Adversarial attack on graph structured data. In *International conference on machine learning*, 1115–1124.
[12]. Zhang, Z., Jia, J., Wang, B., & Gong, N. Z. (2021). Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, 15–26.
[13]. Zheng, H., Xiong, H., Chen, J., Ma, H., & Huang, G. (2022). Motif-Backdoor: Rethinking the Backdoor Attack on Graph Neural Networks via Motifs. *arXiv preprint arXiv:2210.13710*.
[14]. Shaikhelislamov, D., Lukyanov, K., Severin, N., Drobyshevskiy, M., Makarov, I., & Turdakov, D. (2024). A study of graph neural networks for link prediction on vulnerability to membership attacks. *Journal of Mathematical Sciences*, 1–11.
[15]. Conti, M., Li, J., Picek, S., & Xu, J. (2022). Label-Only Membership Inference Attack against Node-Level Graph Neural Networks. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*, 1–12.
[16]. Yuan, X., Ding, L., Zhang, L., Li, X., & Wu, D. O. (2022). Es attack: Model stealing against deep neural networks without data hurdles. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5), 1258–1270.
[17]. Wang, S., & Gong, Y. (2022). Adversarial example detection based on saliency map features. *Applied Intelligence*, 52(6), 6262–6275.
[18]. Ma, J., Deng, J., & Mei, Q. (2022). Adversarial attack on graph neural networks as an influence maximization problem. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 675–685.
[19]. Zhu, D., Zhang, Z., Cui, P., & Zhu, W. (2019). Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1399–1407.
[20]. Feng, F., He, X., Tang, J., & Chua, T.-S. (2019). Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, 33(6), 2493–2504.
[21]. Finlay, C., & Oberman, A. M. (2019). Scaleable input gradient regularization for adversarial robustness. *arXiv preprint arXiv:1905.11468*.
[22]. Szyller, S., & Asokan, N. (2023). Conflicting interactions among protection mechanisms for machine learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12), 15179–15187.
[23]. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, 29(3), 93–93.
[24]. McAuley, J., Targett, C., Shi, Q., & Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 43–52.

Информация об авторах / Information about authors

Георгий Владимирович САЗОНОВ – сотрудник отдела информационных систем института системного программирования им. В.П. Иванникова Российской академии наук; студент магистратуры МГУ.

Georgii Vladimirovich SAZONOV – an employee of the Information Systems Department of the

Ivannikov Institute for System Programming of the Russian Academy of Sciences; master's student at Moscow State University.

Кирилл Сергеевич ЛУКЬЯНОВ – исследователь центра доверенного искусственного интеллекта ИСП РАН; аспирант МФТИ.

Kirill Sergeevich LUKYANOV – Researcher at the Center for Trusted Artificial Intelligence of the Ivannikov Institute for System Programming of the Russian Academy of Sciences; postgraduate student at Moscow Institute of Physics and Technology.

Игорь Николаевич МЕЛЕШИН – сотрудник лаборатории компьютерной графики и мультимедиа; студент бакалавриата МГУ.

Igor Nikolaevich MELESHIN – employee of the Laboratory of Computer Graphics and Multimedia Moscow State University; undergraduate student of Moscow State University.

DOI: 10.15514/ISPRAS-2024-36(5)-9

Так ли безопасна интерпретируемость ИИ: взаимосвязь интерпретируемости и защищенности моделей машинного обучения

[1,2] *Г.В. Сазонов, ORCID: 0009-0003-0905-534X <sazonovg@ispras.ru>*
[1,3,4] *К.С. Лукьянов, ORCID: 0009-0009-5235-2175 <lukyanov.k@ispras.ru>*
[5] *С.К. Боярский, ORCID: 0000-0003-0093-3719 <serafimboyarsky@yandex.ru>*
[4,6] *И.А. Макаров, ORCID: 0000-0002-3308-8825 <iamakarov@hse.ru>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*
[2] *Московский государственный университет имени М.В. Ломоносова, Россия, 119991, Москва, Ленинские горы, д. 1.*
[3] *Московский физико-технический институт (НИУ), Россия 117303, Москва, ул. Керченская, д.1 А, корп. 1.*
[4] *Исследовательский центр доверенного искусственного интеллекта ИСП РАН, Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*
[5] *Школа анализа данных Яндекса, Россия, 119021, Москва, ул. Тимура Фрунзе, 11, корп. 2.*
[6] *Институт искусственного интеллекта AIRI, Россия, 105064, г. Москва, Нижний Сусальный пер., 5 стр. 19.*

Аннотация. В условиях растущего применения интерпретируемых моделей искусственного интеллекта (ИИ) всё больше внимания уделяется вопросам доверия и безопасности для всех типов данных. В этой работе мы сосредотачиваемся на задаче классификации вершин графов, выделяя ее как одну из самых сложных. Эта работа является первой, насколько нам известно, в которой комплексно исследуется взаимосвязь интерпретируемости и защищенности. Наши эксперименты проводятся на наборах данных: цитирования и графов покупок. Мы предлагаем методики построения атак черного ящика графовых моделей на основании результатов интерпретации, показываем, как добавление защиты влияет на интерпретируемость моделей ИИ.

Ключевые слова: интерпретируемость; защищенность; атаки на модели искусственного интеллекта; атаки черного ящика; классификация вершин графов; доверенный искусственный интеллект.

Для цитирования: Сазонов Г.В., Лукьянов К.С., Боярский С.К., Макаров И.А. Так ли безопасна интерпретируемость ИИ: взаимосвязь интерпретируемости и защищенности моделей машинного обучения. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 127–142. DOI: 10.15514/ISPRAS–2024–36(5)–9.

Is AI Interpretability Safe: the Relationship between Interpretability and Security of Machine Learning Models

[1, 2] *G.V. Sazonov, ORCID: 0009-0003-0905-534X <sazonovg@ispras.ru>*
[1, 3, 4] *K.S. Lukyanov, ORCID: 0009-0009-5235-2175 <lukyanov.k@ispras.ru>*
[5] *S.K. Boyarsky, ORCID: 0000-0003-0093-3719 <serafimboyarsky@yandex.ru>*
[4,6] *I.A. Makarov, ORCID: 0000-0002-3308-8825 <iamakarov@hse.ru>*

[1] *Ivannikov Institute of System Programming of the Russian Academy of Sciences, 25, A. Solzhenitsyn st., Moscow, 109004, Russia.*
[2] *Lomonosov Moscow State University, Leninskie Gory, 1, Moscow, 119991, Russia.*
[3] *Moscow Institute of Physics and Technology (National Research University), 1 A, building 1, Kerchenskaya st., Moscow, 117303, Russia.*
[4] *Research Center for Trusted Artificial Intelligence ISP RAS, 25, A. Solzhenitsyn st., Moscow, 109004, Russia.*
[5] *Yandex School of Data Analysis, bld. 2, 11, Timur Frunze st., Moscow, 119021, Russia.*
[6] *AIRI, bld. 19, 5, Nizhnij Susal'nij per., Moscow, 105064, Russia.*

Abstract. With the growing application of interpretable artificial intelligence (AI) models, increasing attention is being paid to issues of trust and security across all types of data. In this work, we focus on the task of graph node classification, highlighting it as one of the most challenging. To the best of our knowledge, this is the first study to comprehensively explore the relationship between interpretability and robustness. Our experiments are conducted on datasets of citation and purchase graphs. We propose methodologies for constructing black-box attacks on graph models based on interpretation results and demonstrate how adding protection impacts the interpretability of AI models.

Keywords: interpretability; robustness; attacks on AI models; Black-box attacks; graph node classification; trusted AI.

For citation: Sazonov G.V., Lukyanov K.S., Boyarsky S.K., Makarov I.A. Is AI interpretability safe: the relationship between interpretability and security of machine learning models. *Trudy ISP RAN/Proc. ISP RAS,* vol. 36, issue 5, 2024. pp. 127-142 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-9.

1. Введение

Устойчивость глубоких нейронных сетей к входным возмущениям является важнейшим свойством для их интеграции в различные области машинного обучения, требующие безопасности, такие как беспилотные автомобили, медицинская диагностика и финансы. Хотя нейронные сети должны выдавать схожие результаты для схожих входных данных, давно известно, что они уязвимы для состязательных возмущений [1] – небольших, тщательно продуманных преобразований входных данных, которые не изменяют семантику входного объекта, но заставляют модель выдавать предопределенное решение. Большинство методов изучения состязательной устойчивости нейронных сетей направлены на создание состязательных возмущений, которые указывают на то, что в целом прогнозы нейронной сети ненадежны. Наиболее эффективные и скрытные атаки требуют доступа к градиентам модели и, следовательно, сами по себе имеют мало практического применения [2-4].

Также в последние годы возросло использование интерпретируемых моделей искусственного интеллекта (ИИ) [5-7], что сопровождается возрастающим вниманием к вопросам доверия и безопасности. Методы интерпретации, направленные на повышение прозрачности и объяснимость решений моделей, считаются ключевыми компонентами, укрепляющими доверие пользователей и регуляторов.

Однако, наряду с очевидными преимуществами, интерпретируемость может привести к непреднамеренному раскрытию уязвимостей модели, что открывает возможности для

злоумышленников реализовывать атаки, эксплуатирующие эти слабые места, в том числе в сценарии черного ящика. С другой стороны, попытки защитить модели от таких атак могут нарушить корректность или снизить качество интерпретации, создавая проблему одновременного выполнения требований интерпретации и безопасности. Настоящее исследование фокусируется на анализе этой взаимосвязи, рассматривая, как методы интерпретации могут быть использованы для выявления слабых мест модели, и как добавление механизмов защиты сказывается на качестве интерпретаций. Кроме того, обсуждаются потенциальные направления для разработки систем, сбалансированных по критериям прозрачности и устойчивости к атакам.

Наш вклад можно представить следующим образом:

1. Насколько нам известно, мы первые кто комплексно исследует взаимосвязь интерпретируемости и защищенности моделей ИИ.
2. Мы предлагаем методики построения атак черного ящика основываясь на результатах работы методов интерпретации.
3. Мы предлагаем методику оценки совместимости методов интерпретации с защищенными методами.
4. Мы экспериментально демонстрируем эффективность предлагаемых методик на двух графовых доменах.

Структура статьи организована следующим образом. Во второй главе представлен обзор литературы, в котором рассматриваются основные подходы и результаты, достигнутые в данной области исследования. Третья глава посвящена постановке задач, где формулируются основные определения и цели исследования, которые оно призвано решить. В четвертой главе описывается методология, включающая разработанный методы и методики, применяемые для достижения поставленных целей. Пятая глава содержит описание проведенных экспериментов и анализ полученных результатов. В шестой главе рассматриваются ограничения предложенных подходов и их потенциальное влияние на результаты. Наконец, седьмая глава содержит заключение, в котором подводятся итоги работы, а также намечаются возможные направления для будущих исследований.

2. Обзор литературы

В этом разделе приводится краткий обзор существующих методов интерпретации, атак черного ящика и защит от состязательных атак применительно к моделям, работающим с графовыми данными.

Существуют разные подходы к интерпретации данных. Наиболее распространенным в литературе является апостериорный подход к интерпретации, когда модель сначала обучается, а уже только после этого применяются методы интерпретации. Методы апостериорной интерпретации для моделей, работающих с графовыми данными, можно разделить на три основные группы: методы, основанные на внимании, распространении значимости и градиентные методы. Каждый подход по-своему объясняет предсказания графовых моделей, уделяя особое внимание топологии и структуре графов.

Примеры методы на основе внимания: PGExplainer [8], GraphMask [9]. PGExplainer – метод, который строит вероятностные маски для рёбер, указывая значимость каждого соединения в графе. GraphMask – это метод, который использует обучаемые маски для оценивания значимости ребер и узлов в графе. Примеры градиентных методах: Integrated Gradients [10], GNNExplainer [7]. Integrated Gradients для GNN – это метод, который оценивает вклад каждого узла и ребра через интегрирование градиентов. Он позволяет вычислить вклад каждого элемента графа в итоговое предсказание, расширяя базовый метод Integrated Gradients на графовые структуры. GNNExplainer – работает, оптимизируя маску для рёбер и признаков узлов, чтобы минимизировать разницу между предсказанием модели на полном

графе и на выделенном подграфе. Это позволяет определить, какие рёбра и признаки наиболее значимы для конкретного предсказания. При этом GNNExplainer можно применять для интерпретации на разных уровнях (узла, рёбра или графа). Примеры основанные на распространении значимости: GraphLIME [11], SubgraphX [12].

Эти методы могут помогать находить неточности в работе моделей. Но также эти методы могут, с одной стороны, использоваться для построения атак на модели машинного обучения. С другой стороны, их можно использовать, чтобы оценить уязвимости графовых моделей, закладывая основу для анализа защиты от атак, которые используют уязвимые узлы и связи в графе.

Состязательные атаки на модели обработки графов можно классифицировать по степени доступа к модели (атаки белого и черного ящика), цели атаки (на узлы, подграфы, граф в целом) и переносимости. Основные категории включают целевые и нецелевые атаки, атакующие конкретные узлы или подграфы, и переносимые атаки, которые остаются эффективными даже при изменении модели. Эти подходы направлены на манипулирование предсказаниями модели посредством малозаметных изменений в графе. В данной работе основной акцент будет на целевые атаки черного ящика.

Целевые и нецелевые атаки на узлы и подграфы – эти методы нацелены на изменение конкретных узлов и рёбер, чтобы повлиять на предсказания. Nettack [13] – пример целевой атаки, которая изменяет рёбра и признаки отдельных узлов для нарушения предсказаний. FGSM (Fast Gradient Sign Method) [14], адаптированный для графов, является примером целевой атаки, применяющей градиентные оценки для создания искажений на уровне узлов. Переносимые атаки – атаки, направленные на создание искажений, которые будут эффективны для нескольких моделей одновременно. Mettack [15], основанный на метаобучении, позволяет создавать искажения, сохраняющие свою силу против различных архитектур. Атаки черного ящика – атаки, где злоумышленник не имеет доступа к параметрам модели и полагается на изменения в предсказаниях, чтобы провести атаку. RL-S2V [16] – это метод, использующий обучение с подкреплением для атаки на графовые модели без знания их внутренних параметров, полагаясь только на доступ к предсказаниям модели.

Эти подходы к построению различных типов атак показывают, как изменения в графовой структуре могут нарушить предсказания, подчёркивая необходимость в разработке надёжных защитных механизмов, способных противостоять таким атакам.

Методы защиты от атак на графы можно классифицировать по их подходам к устойчивости: защита на уровне данных, включающая фильтрацию и корректировку графа, архитектурные методы, повышающие устойчивость моделей к искажениям, и регуляризация, предотвращающая зависимость от отдельных узлов или рёбер. Основные стратегии включают фильтрацию, добавление шума и устойчивые к атакам архитектуры. Основное внимание в этой работе будет уделено методам, позволяющим противостоять состязательным атакам.

Фильтрация и корректировка данных – метод, который на уровне данных предотвращает добавление искажений, фильтруя подозрительные ребра и узлы. Jaccard Similarity Filtering [17] выполняет фильтрацию рёбер на основе схожести признаков узлов, препятствуя внедрению вредоносных рёбер. Архитектурные изменения – изменение модели для повышения устойчивости к атакам. Robust GCN [18] добавляет регуляризацию и шум на уровне узлов, снижая чувствительность модели к мелким искажениям. Регуляризация и шум – методы, предотвращающие переобучение на отдельных элементах графа, что делает модель менее восприимчивой к атакам. Также к регуляризации можно отнести Adversarial training [19] и Gradient Regularization [20], который меняют процесс обучения добавляя данные применением атаки во время обучения и ограничивая градиенты. GNNGuard [21] использует

механизмы для обнаружения подозрительных рёбер и снижает их влияние на предсказания, что защищает от атак.

Эти защитные подходы позволяют значительно снизить уязвимость графовых моделей, сохраняя высокую точность предсказаний и надежность модели при использовании графов в критических приложениях. Однако, нет исследований, которые оценивали бы насколько меняется качество интерпретации применительно к защищенным моделям.

Хочется также отметить некоторые постановки, которые встречаются в литературе, где одновременно встречаются вопросы, связанные с парами интерпретация + атаки; интерпретация + защита и сказать, чем они отличаются от постановок, предложенных в этой работе. В работе "Adversarial Detection with Model Interpretation"[22] авторы показывают, как результаты работы метода интерпретации могут использоваться для повышения безопасности за счет того, что интерпретация предоставляет больше информации о процессе принятия решений. Результаты работы демонстрируют эффективность при отсутствии других факторов, но не учитываю, что из-за атак сама интерпретация может ухудшаться. В частности, есть направление атак на ухудшение интерпретации [23-24]. Отличие этой постановки в том, что цель построить атаку в первую очередь на методы интерпретации, например, сделав интерпретацию нестабильной, а не ухудшить прогноз модели на тестовых данных как в постановке атак уклонения. И наиболее близкая постановка представлена в статье "Adversarial Attack on Graph Neural Networks as An Influence Maximization Problem" [24], где авторы анализируют данные основываясь на различных показателях распространения информации в графе и на основании этого формируют атаку. Однако, для анализа распространения информации в графах требуется доступ ко всему графу сразу, что не отвечает сценарию атаки черного ящика, по этой причине сравнения с результатами этой работой не представлено в разделе экспериментов.

На основании обзора литература видно, что существует множество методов интерпретации, атак и защит. Однако, на данный момент не так много вопросов изучено на границе пересечения исследований по обеспечению требований безопасности и интерпретируемости в области доверенного ИИ. Далее подробнее описаны предлагаемые новые постановки задач на стыке обеспечения двух требований доверия и их взаимодействия.

3. Постановка задач

В этом разделе формально вводится постановка задач, вводятся обозначения, используемые в статье, и формулируются вопросы исследования.

3.1 Состязательная атака на нейронную сеть решающую задачу классификации

Предположим, что $f: R^d \to \Delta^K$ – это нейронная сеть, решающая задачу классификации, которая сопоставляет входной объект $X \in R^d$ с вектором $f(X) \in \Delta^K$ вероятностей для K классов, а

$$h(f, X) = arg \max_{i \in [1,\dots,K]} f(X)_i$$

соответствующее правило классификации. Начнем с формального определения состязательного примера для данной нейронной сети и переносимости состязательного примера между двумя сетями.

Definition 3.1 (Состязательный пример). Предположим, что $X \in R^d$ – это входной объект, правильно сопоставленный классу $y \in [1, \dots, K]$ сетью f, а именно, $h(f, X) = y$. Пусть $\delta > 0$ – фиксированная константа. Тогда объект $X' \in R^d: \| X - X' \|_p \leq \delta$ – это нецелевой состязательный пример для f в точке X, если $h(f, X') \neq h(f, X)$, где p, может быть, например, равно $p = 2$ или $p = \infty$.

Если $h(f, X') = t$ для некоторого предопределенного индекса класса t, то X' называется targeted состязательным примером.

Definition 3.2 (Переносимый состязательный пример). Пусть X' будет состязательным примером, вычисленным для сети f в точке X, и пусть $b: R^d \to \Delta^K$ будет некоторой другой сетью. Тогда X' переносится с f на b, если $h(f, X) = h(b, X)$ и $h(f, X') = h(b, X')$.

3.2 Интерпретация моделей

Definition 3.3 (Результат интерпретации). Пусть X – входные данные для модели f, а $M(X) \in [0,1]^d$ – маска, определяющая важность каждого признака X_i во входе X. Здесь $M(X)_i = 1$ указывает на полный вклад i-го признака, $M(X)_i = 0$ означает, что признак не вносит вклада, а значения между 0 и 1 соответствуют частичной значимости признака. Тогда важное подмножество X определяется как $X^{int} = X \odot M(X)$, где $\odot$ обозначает поэлементное умножение. Подмножество X^{int} сохраняет только значимые признаки исходного входа X.

Метрики интерпретации:

- Корректность объяснения (Fidelity). Fidelity измеряет, насколько точно результаты метода интерпретации отражают поведение исходной модели.

$$Fidelity = 1/N \sum\_(i = 1)\^N |f(X\_i\^int) - f(X\_i)|$$

- Разреженность (Sparsity). Sparsity оценивает, насколько просто объяснение, то есть какой процент признаков не участвуют в интерпретации прогноза:

$$Sparsity = (\sum\_(j = 1)\^m 1\{M(X)\_j \neq 0\})/m$$

  где $M(X)_j$ – значение маски интерпретации показывающую вклад j-го элемента, а m – общее количество признаков.

- Стабильность объяснений (Stability). Stability оценивает, насколько похожи объяснения для схожих входных данных. Изменения в объяснениях измеряется посредством добавляя к исходным данным небольшой шум и вычислениями отклонений:

$$Stability = 1/n \sum\_(i = 1)\^n \| M(X\_i) - M(X\_i\^noise) \| \_2$$

  где X_i – исходные данные, X_i^{noise} – данные с небольшими изменениями, $\|\cdot\|_2$ – евклидова норма.

- Согласованность объяснений (Consistency). Consistency измеряет, насколько похожи объяснения для одного и того же примера при разных запусках модели или методов интерпретации:

$$Consistency = 1/n \sum\_(i = 1)\^n cos(M(X)\_i, M(X)\_(i + 1))$$

  где $M(X)_i$ и $M(X)_{i+1}$ – объяснения для одного и того же примера, полученные при разных запусках; cos – косинусное расстояние.

Стоит отметить, что помимо рассматриваемых метрик интерпретации существуют и другие. Однако, выбраны были именно эти метрики на основании рекомендаций и исследований, представленных в литературе. Согласно исследованиям [25], объективные количественные метрики позволяют избежать зависимости от субъективной оценки экспертов и повышают воспроизводимость результатов. Выбранные метрики широко используются и дают количественную оценку интерпретируемости, поскольку они не требуют субъективной оценки и могут быть автоматически рассчитаны на основе выходных данных модели. В литературе отмечено, что вовлечение доменных экспертов усложняет оценку интерпретируемости, так как мнение экспертов может варьироваться в зависимости от их

опыта и восприятия [26]. Выбранные метрики не требуют построения дополнительных интерпретируемых моделей или обучения дополнительных классификаторов. Это существенно упрощает процесс и снижает вычислительную сложность. В исследовании [27] отмечается, что такие метрики полезны для оперативной интерпретации, особенно в приложениях с высокими вычислительными затратами. Выбранные метрики охватывают разные аспекты интерпретируемости: Fidelity позволяет измерить, насколько корректно объяснение отражает работу модели, Stability и Consistency оценивают устойчивость и согласованность объяснений, а Sparsity снижает когнитивную нагрузку, минимизируя количество признаков в интерпретации. Это соотносится с рекомендациями по многомерному анализу интерпретируемости, предложенными в работах [28, 29].

3.3 Защищенная модель

Definition 4 (Защищенная модель). Пусть f – исходная модель, а f' – её защищённая версия. Обозначим $X \in R^d$ как чистые входные данные, а X' – атакованные данные. Модель f' называется защищенной, если выполняются следующие условия:

$$(i) \quad Q(f, X) - Q(f', X) \leq \alpha,$$
$$(ii) \quad Q(f', X') - Q(f, X') \geq \beta,$$

где $Q(\cdot,\cdot)$ – функция качества модели на входных данных, α – максимальное допустимое снижение качества на чистых данных, а β – минимально допустимое увеличение качества на атакованных данных относительно незащищенной модели. Эти условия обеспечивают устойчивость защищенной модели к атакам, сохраняя приемлемый уровень производительности.

Вопросы исследования

- Как, используя результаты работы методов интерпретации, построить состязательный пример в сценарии черного ящика? Для оценки эффективности атак оценивается среднее количество запросов к модели и средний процент успешности атаки.
- Как добавление защитных механизмов влияет на метрики интерпретируемости, указанные в этом разделе?

В следующем разделе мы отвечаем на эти вопросы и предлагаем соответствующие методики.

4. Методология

В данном разделе представлена методика исследования, направленная на достижение поставленных целей и решение заявленных задач. Вначале описана методика построения атак основываясь на результатах работы методов интерпретации. Далее показано как можно оценить влияние методов, обеспечивающих защиту на интерпретируемость моделей.

4.1 Методика построения атаки черного ящика на базе интерпретации

Так как работа делает основной фокус на работу с графовыми данными, то принцип построения атаки может быть направлен на изменения признаков или структуры, или на обе составляющие данных сразу. Также стоит отметить, что вектор направления атаки зависит от того какой метод интерпретации применяется и какой результат этот метод выдает. Так, например, GNNExplainer выдает ранжированную важность ребер и признаков, SubgraphX выдает только важные ребра, а Zorro [30] выдает ранжированную важность вершин и признаков.

4.1.1 Атака на признаки

При атаке на признаки рассмотрим вариант действия: изменение признаков.

Необходимо выбрать некоторый процент признаков относительно общего количество $p\_feature$, который можно модифицировать. Этот параметр играет роль бюджета атаки на признаки. Если важных признаков меньше, чем заданный процент от общего числа, то выбираем исключительно важные, а остальные не модифицируются.

В случае изменения признаков необходимо добавить случайное возмущение ϵ, если признак категориальный, то просто поменять его класс. ϵ играет роль максимального допустимого возмущения.

4.1.2 Атака на структуру

При атаке на признаки рассмотрим три варианта действий: переключение ребер, удаление и добавление.

Аналогично атаке на признаки необходимо выбрать некоторый процент ребер относительно общего количество p_edges, который можно модифицировать. Этот параметр играет роль бюджета атаки на структуру.

В случае удаления важные ребра удаляются. Стоит обратить внимание, что чем ближе ребро к целевой вершине (вершина, которую атакуют, и цель поменять класс именно этой вершины), тем больше информации будет потеряно и атака будет более заметной. В этом смысле вариант с переключением является более предпочтительным с точки зрения незаметности атаки. В случае добавления ребер необходимо выбрать вершину не являющуюся важной (или не являющийся вершиной, из которой выходит или входит важное ребро) и связать ее с важной вершиной (или вершиной, из которой выходит важное ребро). В случае переключения выбираются важное ребро, которое удаляется и добавляется ребро связывающая вершину, из которой выходило важное ребро и связывается с целевой вершиной.

Основная цель методик при построении атакующих возмущений, в том, чтобы удалить/поменять важные элементы в окрестности целевой вершины, так как по определению интерпретации выделяет элементы данных оказавший наибольшее влияние на конечный прогноз модели, при условии высокого значения метрики корректности объяснения. Либо, в случае атаки на структуру, спровоцировать размытие информации при агрегации данных по структуре графа, так как в научных работах было показано это является значительной уязвимостью графовых нейросетей [31-32].

4.2 Методика оценки влияние добавления защиты к модели на качество интерпретации

Основная задача методики, описанной в этом подразделе, оценить, насколько добавление защиты влияет на интерпретируемость моделей.

4.2.1 Метод

1. Построить модель;
2. Обучить;
3. Оценить значения метрик интерпретации для этой модели (Корректность, Стабильность, Согласованность и Разреженность);
4. Добавить защиту применительно к модели;
5. Обучить защищенную модель;
6. Оценить значения метрик интерпретации для защищенного аналога;
7. С помощью метода доверительных интервалов оценить значимость изменений метрик интерпретации.

Также, отметим, что основной задачей предложенной методики является показать возможный принцип качественного анализа взаимосвязи различных аспектов доверия. И при

необходимости методика может быть дополнена новыми метриками оценки интерпретируемости моделей машинного обучения.

5. Эксперименты

В этом разделе описаны эксперименты и все необходимое для их воспроизведения. Первый эксперимент проводился для оценки эффективности атак, основанных на интерпретации, и результаты сравниваются с базовыми методами. Второй эксперимент показывает, как добавление методов защиты влияет на метрики интерпретации.

5.1 Методология экспериментов

5.1.1 Наборы данных и обучение моделей

В экспериментах использовались наборы данных: Cora, CiteSeer, PubMed – относящиеся к домену цитирования [33] и представленных в библиотеке torch-geometric в группе наборов данных Planetoid и Computers, Photo – домена графов покупок [34], также представленные в torch-geometric в группе наборов данных Amazon.

Статистика наборов данных:

- Cora: 2708 вершин, 10556 ребер, 7 классов, 1433 признаков;
- CiteSeer: 3327 вершин, 9104 ребер, 6 классов, 3703 признаков;
- PubMed: 19717 вершин, 88648 ребер, 3 классов, 500 признаков;
- Computers: 13752 вершин, 491722 ребер, 10 классов, 767 признаков;
- Photo: 7650 вершин, 238162 ребер, 7 классов, 745 признаков.

В качестве модели обучалась двухслойная модель GCN (GCN-2l).

GCN-2l

```
(
  Sequential(
    (0):  GCNConv(input_size, 16)
    (1):  ReLU(inplace)
    (3):  GCNConv(16, output_size)
    (4):  LogSoftmax(inplace)
  )
)
```

Для обучения использовался оптимизатор Adam с параметрами по умолчанию и функция ошибки NLLLoss из библиотеки PyTorch, разделение на пакеты данных не используется. Количество эпох обучения равно 200 для достижения высокой точности классификации на всех наборах данных.

5.1.2 Параметры проведения экспериментов

Для всех экспериментов использовался метод интерпретации GnnExplainer на базе реализации библиотеки torch-geometric. Все настройки метода интерпретации взяты по умолчанию из библиотеки, кроме node_mask_type, который был взят со значением "attributes", чтобы метод интерпретации возвращал не только важные ребра, но и важные признаки. Основные эксперименты усредняются на основании 100 запусков на одном и том же наборе вершин. Набор вершин для каждого набора данных и эксперимента выбирался различный из тестового набора вершин. Для расчета метрик интерпретации, требующие добавление шума или требующих внесения дополнительных случайных величин в эксперимент, проводилось дополнительные внутренние 20 запусков для каждого из 100 запусков на фиксированном наборе вершин. Так, например, для оценки метрики Stability на каждой итерации метрика считалась не только на 100 вершинах, но и для каждой отдельной вершины считалось на 20 различных добавлениях случайного шума в данные.

Для методов атаки максимально допустимая величина варьировалась и бралась равной 5, 10 и 15 % соответственно для того, чтобы оценить не только качество атаки, но и изучить как изменения допустимого возмущения влияет на ее эффективность.

В качестве методов защиты использовались методы: Adversarial Training в качестве метода атаки в Adversarial Training использовался метод FGSM с параметром возмущения $\epsilon = 0.01$, GnnGuard и метод Jaccard Defense. Последние два метода использовали все настройки по умолчанию в соответствии с приведенными в оригинальных статьях. GNNGuard: $lr = 0.01, attention = True, drop = True, train\_iters = 50$; Jaccard Defense: $threshold = 0.35$; Методы защиты Jaccard и GNNGuard относятся к защитам от атак отравления, а метод AdvTraining – к защитам от атак уклонения.

5.1.3 Базовый метод и методы для сравнения

По причине, что подход построения атаки основываясь на результатах интерпретации, насколько нам известно, является абсолютно новым, то сравнение предложенных методов проводится с базовым методом вариантами, основанными на случайных изменениях. Также, стоит отметить, что предложенный подход атаки работает за один запрос к модели черного ящика получая только класс объекта в результате работы модели и результаты интерпретации модели на заданном объекте, что не позволяет сравниться с другими подходами атак на модели черного ящика. Это происходит, так как часть подходов изначально предполагает доступ к обучающим данным, чего нет в нашем случае и, что значительно упрощает проведение атаки посредством обучения качественной суррогатной модели. Другие же подходы атакуют черный ящик, которые работают без какой-либо дополнительной информации, что ставит их в заведомо проигрышную позицию по метрике среднего количества запросов (AQN), так как в нашем случае она всегда равна 1.

Все базовые методы использовали аналогичный предлагаемым атакам бюджет в рамках экспериментов, то есть 5, 10 и 15 % соответственно. Базовые методы были случайные атаки, которые переключали, удаляли или добавляли заданный процент ребер или в случае атаки на признаки добавляли случайный шум на заданный процент признаков.

5.1.4 Протокол оценки

Для иллюстрации эффективности предлагаемого подхода построения атак мы приводим значение метрики ASR (средний показатель успешности атаки) и эффективности атаки, которая считается как снижение точности на оригинальных и атакованных данных. Также, отдельно сравниваются между собой варианты построения атак.

Для методики оценки влияние добавления защиты к модели на качество интерпретации считались метрики интерпретации для незащищенной и защищенных моделей. После с помощью методов доверительных интервалов проводится оценка значимости влияния добавления методов защиты на качество интерпретации.

5.2 Результаты экспериментов

В табл. 1 представлено сравнение метрик интерпретации для модели без защиты (Unprotected) и для моделей с использованием 3 защит (Jaccard, AdvTraining и GNNGuard соответственно). Строки представляют наборы данных и метрики интерпретации, наборы данных скомпонованы по доменам. Направление стрелок рядом с метриками соответствует в каком направлении улучшается значение метрики, стрелка вверх означает больше лучше, стрелка вниз – меньше лучше. На основании метода доверительных интервалов с уровнем значимости 5% было выявлено, что на наборе данных PubMed произошло значимое улучшение всех метрик, на наборе данных CiteSeer значимо ухудшилась метрика стабильности для моделей с защитой AdvTraining и GNNGuard и для набора данных Photo произошло значимое улучшение стабильности для всех методов защиты. Также, стоит отметить, что при использовании метода защиты Jaccard в среднем получаются лучше

значение метрик стабильности и разреженности, чем при использовании других методов. Основной причиной может быть то, что этот метод защиты удаляет ребра, которые могут приводить к некорректному распространению информации и могли быть добавлены во время атаки, что положительно сказывается на указанные выше метрики.

Табл. 1. Сравнение метрик интерпретации для незащищенной модели и для модели с различными защитами. Стрелка вверх рядом с метриками показывает, что большее значение соответствует лучшему значению метрики, стрелка вниз – показывает, что меньшее значение соответствует лучшему значению метрики.
Table 1. Comparison of interpretation metrics for the unprotected model and for the model with various defenses. An upward arrow next to the metrics indicates that a higher value corresponds to a better metric value, while a downward arrow indicates that a lower value corresponds to a better metric value.

Набор данных	Метрика	Без защиты	Jaccard	AdvTraining	GNNGuard
Cora	Корректность (↑)	0.97 ± 0.03	0.90 ± 0.09	0.97 ± 0.03	0.97 ± 0.03
	Согласованность (↑)	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
	Стабильность (↓)	1.55 ± 0.87	1.55 ± 0.82	1.90 ± 0.49	1.88 ± 0.46
	Разреженность (↓)	0.04 ± 0.00	0.03 ± 0.00	0.04 ± 0.00	0.04 ± 0.00
CiteSeer	Корректность (↑)	0.87 ± 0.12	0.90 ± 0.09	0.90 ± 0.09	0.90 ± 0.09
	Согласованность (↑)	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Стабильность (↓)	0.75 ± 0.39	0.68 ± 0.27	2.38 ± 1.11	2.39 ± 1.14
	Разреженность (↓)	0.03 ± 0.00	0.00 ± 0.00	0.03 ± 0.00	0.03 ± 0.00
PubMed	Корректность (↑)	0.83 ± 0.14	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Согласованность (↑)	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Стабильность (↓)	2.04 ± 1.19	0.15 ± 0.04	0.22 ± 0.07	0.21 ± 0.07
	Разреженность (↓)	0.03 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
Photo	Корректность (↑)	0.87 ± 0.12	0.80 ± 0.16	0.83 ± 0.14	0.83 ± 0.14
	Согласованность (↑)	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Стабильность (↓)	1.37 ± 0.55	0.65 ± 0.38	0.56 ± 0.26	0.56 ± 0.29
	Разреженность (↓)	0.60 ± 0.69	0.12 ± 0.06	0.15 ± 0.08	0.15 ± 0.08
Computers	Корректность (↑)	0.77 ± 0.18	0.80 ± 0.16	0.83 ± 0.14	0.83 ± 0.14
	Согласованность (↑)	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Стабильность (↓)	1.31 ± 0.52	1.33 ± 0.51	1.18 ± 0.42	1.16 ± 0.42
	Разреженность (↓)	0.22 ± 0.12	0.2 ± 0.1	0.22 ± 0.12	0.22 ± 0.12

В табл. 2 и табл. 3 представлены результаты предложенной атаки построенной на основании работы метода интерпретации в сравнении со случайными атаками, которые были выбраны в качестве базовых методов по причине отсутствия рассматриваемой подхода атаки уклонения в сценарии черного ящика. В таблицах представлено значение метрики среднего процента успешности атаки (ASR), то есть больше значение метрики соответствует большей вероятности успеха атаки.

Табл. 2. Сравнение результатов работы предложенных атак, построенных на основании работы метода интерпретации в сравнении со случайными атаками, которые были выбраны в качестве базовых методов по причине отсутствия рассматриваемой атаки уклонения в сценарии черного ящика. В качестве допустимых модификаций данных рассматривается добавление ребер и изменение признаков. В таблице представлено значение метрики среднего процента успешности атаки (ASR) в процентах, больше значение метрики соответствует большей вероятности успеха атаки.
Table 2. Comparison of the results of the proposed attacks based on the interpretation method against random attacks, which were chosen as baselines due to the absence of the considered evasion attack approach in the black-box scenario. Acceptable data modifications include the addition of edges and changes to features. The table presents the value of the average success rate of the attack (ASR) in percentage, where a higher metric value corresponds to a greater probability of attack success.

Набор данных	Тип атаки	Изменение признаков	Добавление ребер
		$\epsilon = 5\%, 10\%, 15\%$	$\epsilon = 5\%, 10\%, 15\%$
Cora	На основе интерпретации	57±2, 66±2, 69±3	**7±1, 8±1,** 6±1
	Случайная атака	56±3, 66±3, 69±4	2±1, 5±3, 7±1
CiteSeer	На основе интерпретации	**71±7**, 51±5, 45±5	**6±2, 7±2, 8±0**
	Случайная атака	46±4, 56±5, 62±6	2±1, 2±1, 3±2
PubMed	На основе интерпретации	45±1, 44±2, 49±1	**2±1**, 3±1, 3±2
	Случайная атака	68±1, 61±2, 76±2	0±1, 4±1, 3±1
Photo	На основе интерпретации	1±1, 3±2, 7±3	15±2, 16±2, 15±1
	Случайная атака	10±1, 10±2, 14±2	16±1, 17±1, 17±1
Computers	На основе интерпретации	3±1, 5±2, 4±3	**20±1, 22±1, 22±1**
	Случайная атака	7±1, 6±1, 11±2	14±, 15±2, 16±2

Отметим, что метрика среднее число запросов не была проведена, так как во всех случаях она равнялась 1, что лучше любой существующей атаки черного ящика, но прямое сравнение не корректно, так как другие методы не используют результаты интерпретации из-за отличной от рассматриваемой в этой статье постановки.

Из таблиц можно сделать выводы, что атака за счет удаления ребер оказывается неэффективной, атака за счет переключения ребер оказалась значимо эффективнее только на наборе данных Computers, атака за счет добавления ребер оказалась наиболее эффективной и побила базовый метод или была сопоставима на всех наборах данных и атака на основе изменения признаков оказалась эффективнее только на наборе данных CiteSeer.

6. Ограничения

Среди четырех предложенный атак только на одному удалось добиться значимой эффективности. Основной причиной можно выделить то, что подходы на основе переключения и удаления ребер слабо влияют на результаты работы, так как на всех графах при удалении даже всех ребер достигается качество на уровне 60%. В тоже время, добавление ребер является более эффективной стратегией, так как на основании интерпретации. Еще стоит отметить, что увеличение бюджета не приводит к эффективности атаки, а в некоторых случаях приводит к ухудшению.

Табл. 3. Сравнение результатов работы предложенных атак, построенных на основании работы метода интерпретации в сравнении со случайными атаками, которые были выбраны в качестве базовых методов по причине отсутствия рассматриваемой атаки уклонения в сценарии черного ящика. В качестве допустимых модификаций данных рассматривается удаление и переключение ребер. В таблице представлено значение метрики среднего процента успешности атаки (ASR) в процентах, больше значение метрики соответствует большей вероятности успеха атаки.
Table 3. Comparison of the results of the proposed attacks based on the interpretation method against random attacks, which were chosen as baselines due to the absence of the considered evasion attack approach in the black-box scenario. Acceptable data modifications include the deleting and switching edges. The table presents the value of the average success rate of the attack (ASR) in percentage, where a higher metric value corresponds to a greater probability of attack success.

Набор данных	Тип атаки	Удаление ребер	Переключение ребер
		$\epsilon = 5\%, 10\%, 15\%$	$\epsilon = 5\%, 10\%, 15\%$
Cora	На основе интерпретации	1±1, 1±1, 1±1	0±0, 4±1, 6±2
	Случайная атака	0±0, 0±1, 1±1	3±2, 4±1, 6±1
CiteSeer	На основе интерпретации	3±1, 1±1, 3±1	0±0, 1±0, 0±0
	Случайная атака	0±0, 0±0, 0±1	1±0, 3±1, 2±1
PubMed	На основе интерпретации	0±0, 0±1, 0±0	2±1, 1±1, 1±1
	Случайная атака	0±1, 0±1, 0±1	1±1, 3±1, 3±1
Photo	На основе интерпретации	1±1, 0±1, 1±1	12±1, 13±0, 12±1
	Случайная атака	9±1, 8±1, 9±1	17±1, 16±1, 17±1
Computers	На основе интерпретации	0±0, 0±0, 0±0	18±1, 19±2, 19±3
	Случайная атака	0±1, 0±1, 0±1	8±1, 16±2, 12±2

С одной стороны, это говорит об ограниченных возможностях атаки, с другой, о том, что атаке для достижения своей пиковой эффективности требуется очень небольшой бюджет.

Основным ограничением в методики оценки влияния на качество интерпретации методов защиты можно выделить высокую с2тоимость этих оценок. В то же время, подобное сравнение можно позиционировать как бенчмарк, для которого всегда требуются большие вычислительные ресурсы.

Также, хочется отметить, что проводились эксперименты с другими моделями такие как двухслойные модели со сверточными слоями GAT и SAGE, а также, трехслойные модели с этими же типами сверточных слоев. К сожалению, в рамках ограничений на объем работы все эксперименты нет возможности добавить, но все сформулированные выводы корректны и для других рассмотренных архитектур моделей.

7. Заключение и будущая работа

В этой статье был предложен новый подход и соответствующая методика построения атак черного ящика на основании результатов работы методов интерпретации в приложении к графовым данным. Была показана эффективность атаки при добавлении ребер на наборах данных различных доменов: цитирования, графа покупок. При это предложенный подход

требует буквально единичное обращение к модели черного ящика. Также, была предложена методика оценки влияния добавления защиты на качество интерпретации. Эксперименты на популярных методах защиты графовых нейросетей от атак отравления и уклонения показали, что использование методов защиты чаще повышает качество интерпретации, чем ухудшает его.

В качестве направлений будущей работы можно выделить улучшение переносимости атак (ASR) за счет увеличения количества запросов (AQN), так как на практике не обязательно добиваться снижение запросов до 1. Современные атаки в сценариях черного ящика требуют порядка 300-500 запросов для формирования одной атаки и снижения количества запросов даже до 50 будет значительным прорывом, но в тоже время это может позволить значительно улучшить показатель успешной переносимости.

Методику оценки влияния добавление методов защит на качество интерпретации можно развивать до полноценного бенчмарка и оценить влияние методов защиты не только использую другие методы интерпретации, но и на других данных. В будущем мы планируем провести комплексное сравнение значительно большего количества различных методов защит, изучить причины этих результатов и дать практические рекомендации выделяя наиболее эффективные комбинации методов защит и интерпретаций, которые позволят эффективнее внедрять модели ИИ в критическую инфраструктуру.

Список литературы / References

[1]. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, Rob Fergus. Intriguing properties of neural networks. 2nd International Conference on Learning Representations, 2014.

[2]. Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy. Explaining and Harnessing Adversarial Examples. CoRR, 2014, abs/1412.6572.

[3]. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

[4]. Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 2574–2582.

[5]. Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, Cheekong Lee. Protgnn: Towards self-explaining graph neural networks. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(8): 9127–9135.

[6]. Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, Pietro Liò. Global concept-based interpretability for graph neural networks via neuron analysis. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(9): 10675–10683

[7]. Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. Advances in Neural Information Processing Systems, 2019.

[8]. Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, Xiang Zhang. Parameterized explainer for graph neural network. Advances in Neural Information Processing Systems, 2020, 33: 19620–19631.

[9]. Michael Sejr Schlichtkrull, Nicola De Cao, Ivan Titov. Interpreting graph neural networks for NLP with differentiable edge masking. arXiv preprint arXiv:2010.00577, 2020.

[10]. Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T. Schütt, Klaus-Robert Müller, Grégoire Montavon. Higher-order explanations of graph neural networks via relevant walks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(11): 7581–7596.

[11]. Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(7): 6968–6972.

[12]. Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. International Conference on Machine Learning, 2021, 12241–12252.

[13]. Daniel Zügner, Amir Akbarnejad, Stephan Günnemann. Adversarial attacks on neural networks for graph data. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, 2847–2856.

[14]. Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy. Explaining and Harnessing Adversarial Examples. CoRR, 2014, abs/1412.6572.
[15]. Daniel Zügner, Stephan Günnemann. Adversarial Attacks on Graph Neural Networks via Meta Learning. International Conference on Learning Representations, Workshop Track, 2019, https://arxiv.org/abs/1902.08412.
[16]. Xiang Zhang, Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. Advances in Neural Information Processing Systems, 2020, 33: 9263–9275.
[17]. Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, Liming Zhu. Adversarial examples on graph data: Deep insights into attack and defense. arXiv preprint arXiv:1903.01610, 2019.
[18]. Dingyuan Zhu, Ziwei Zhang, Peng Cui, Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, 1399–1407.
[19]. Fuli Feng, Xiangnan He, Jie Tang, Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. IEEE Transactions on Knowledge and Data Engineering, 2019, 33(6): 2493–2504.
[20]. Chris Finlay, Adam M. Oberman. Scaleable input gradient regularization for adversarial robustness. arXiv preprint arXiv:1905.11468, 2019.
[21]. Xiang Zhang, Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. Advances in Neural Information Processing Systems, 2020, 33: 9263–9275.
[22]. Ninghao Liu, Hongxia Yang, Xia Hu. Adversarial Detection with Model Interpretation. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, 1803–1811, https://doi.org/10.1145/3219819.3220027.
[23]. Shen Wang, Yuxin Gong. Adversarial example detection based on saliency map features. Applied Intelligence, 2022, 52(6): 6262–6275.
[24]. Jiaqi Ma, Junwei Deng, Qiaozhu Mei. Adversarial attack on graph neural networks as an influence maximization problem. Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, 675–685.
[25]. Finale Doshi-Velez, Been Kim. Towards a Rigorous Science of Interpretable Machine Learning. arXiv preprint arXiv:1702.08608, 2017.
[26]. Zachary C. Lipton. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue, 2018, 16(3): 31–57
[27]. Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016, 1135–1144.
[28]. Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys (CSUR), 2018, 51(5): 1–42.
[29]. Tim Miller. Explanation in Artificial Intelligence: Insights from the Social Sciences. Artificial Intelligence, 2019, 267: 1–38.
[30]. Thorben Funke, Megha Khosla, Mandeep Rathee, Avishek Anand. Zorro: Valid, sparse, and stable explanations in graph neural networks. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(8): 8687–8698.
[31]. Yiqing Xie, Sha Li, Carl Yang, Raymond Chi-Wing Wong, Jiawei Han. When do GNNs work: Understanding and improving neighborhood aggregation. IJCAI'20: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020.
[32]. Xie Y. et al. When do gnns work: Understanding and improving neighborhood aggregation //IJCAI'20: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, {IJCAI} 2020. – 2020. – Т. 2020. – №. 1.
[33]. Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, Tina Eliassi-Rad. Collective classification in network data. AI Magazine, 2008, 29(3): 93–93.
[34]. Julian McAuley, Christopher Targett, Qinfeng Shi, Anton Van Den Hengel. Image-based recommendations on styles and substitutes. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, 43–52.

Информация об авторах / Information about authors

Георгий Владимирович САЗОНОВ – сотрудник отдела информационных систем института

системного программирования им. В.П. Иванникова Российской академии наук; студент магистратуры МГУ.

Georgii Vladimirovich SAZONOV – employee of the Information Systems Department of the Ivannikov Institute for System Programming of the Russian Academy of Sciences; master's student at Moscow State University.

Кирилл Сергеевич ЛУКЬЯНОВ – исследователь центра доверенного искусственного интеллекта ИСП РАН; аспирант МФТИ.

Kirill Sergeevich LUKYANOV – Researcher at the Center for Trusted Artificial Intelligence of the Ivannikov Institute for System Programming of the Russian Academy of Sciences; postgraduate student at Moscow Institute of Physics and Technology.

Серафим Константинович БОЯРСКИЙ – студент школы анализа данных Яндекса; студент университета ИТМО.

Serafim Konstantinovich BOYARSKY – student of Yandex School of Data Analysis, student of ITMO University.

Илья Андреевич МАКАРОВ – старший научный сотрудник научно-исследовательского института искусственного интеллекта (AIRI), Москва, Россия, где руководит исследованиями в области промышленного ИИ.

Ilya Andreevich MAKAROV – Senior Research Fellow position at Artificial Intelligence Research Institute (AIRI), Moscow, Russia, where he leads the research in Industrial AI.

DOI: 10.15514/ISPRAS-2024-36(5)-10

CC BY

Разработка вредоносного набора данных для защиты больших языковых моделей от атак

[1] *И.С. Алексеевская, ORCID: 0009-0006-8833-441X <alekseevskaia@ispras.ru>*
[1, 2] *К.В. Архипенко, ORCID: 0000-0002-8699-889X <arkhipenko@ispras.ru>*
[1, 2] *Д.Ю. Турдаков, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>*

[1] *Институт системного программирования РАН,*
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.
[2] *Московский государственный университет имени М.В. Ломоносова,*
Россия, 119991, Москва, Ленинские горы, д. 1.

Аннотация. Современные большие языковые модели представляют собой огромные системы со сложным внутренними механизмами, реализующие генерацию ответа на основе черного ящика. Несмотря на то, что выровненные большие языковые модели имеют встроенные механизмы защиты от атак, последние исследования демонстрируют уязвимость больших языковых моделей к атакам. В данном исследовании мы стремимся расширить существующие вредоносные наборы данных, полученные в результате атак, чтобы в будущем можно было устранить подобные уязвимости в больших языковых моделях путем процедуры выравнивания. Кроме того, мы проводим эксперименты с современными большими языковыми моделями на нашем вредоносном наборе данных, что демонстрирует существующие недостатки в моделях.

Ключевые слова: большие языковые модели; атаки “побег из тюрьмы”; наборы данных красной команды; доверенный искусственный интеллект.

Для цитирования: Алексеевская И.С., Архипенко К.В., Турдаков Д.Ю. Разработка вредоносного набора данных для защиты больших языковых моделей от атак. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 143–152. DOI: 10.15514/ISPRAS–2024–36(5)–10.

Благодарности: Авторы благодарят Институт системного программирования им. В.П. Иванникова Российской академии наук за поддержку проводимых исследований.

Development of a Red-Teaming Dataset for Defending Large Language Models against Attacks

[1] *I.S. Alekseevskaia, ORCID: 0009-0006-8833-441X <alekseevskaia@ispras.ru>*
[1, 2] *K.V. Arkhipenko, ORCID: 0000-0002-8699-889X <arkhipenko@ispras.ru>*
[1, 2] *D.Y. Turdakov, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>*

[1] *Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*
[2] *Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia.*

Abstract. Modern large language models are huge systems with complex internal mechanisms implementing black-box response generation. Although aligned large language models have built-in defense mechanisms against attacks, recent studies demonstrate the vulnerability of large language models to attacks. In this study, we aim to expand the existing malicious datasets obtained from attacks so that similar vulnerabilities in large language models can be addressed in the future through the alignment procedure. In addition, we conduct experiments with modern large language models on our malicious dataset, which demonstrates the existing weaknesses in the models.

Keywords: large language models; jailbreak attacks, red-teaming datasets; trustworthy artificial intelligence.

For citation: Alekseevskaia I.S., Arkhipenko K.V., Turdakov D.Y. Development of a red-teaming dataset for defending large language models from attacks. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024, pp. 143-152 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-10.

Acknowledgements: Authors are thankful to the Ivannikov Institute for System Programming of the Russian Academy of Sciences for supporting their research efforts.

1. Введение

Последнее время активно развиваются системы искусственного интеллекта (ИИ) и применяются в различных областях, как в качестве помощников ChatGPT [1] для генерации текстовых ответов на вопросы, так и для более прикладных задач CodeLlama [2] для генерации программного кода. Также применение ИИ затрагивает и более критические области, например, Med-PaLM [3] для выявления заболеваний.

Однако, исследователями было выявлено, что большие языковые модели уязвимы к состязательным атакам [4-6], бэкдор атакам [7-9], утечка данных [10-12], что потенциально может привести к проблемам с безопасностью и вопросом доверия системам с ИИ. Более того, было обнаружено, что модели могут дискриминировать определенные расы людей [13], говорить последовательность шагов по изготовлению нелегальных веществ [14], выдавать конфиденциальные данные [15]. В связи с этим, в научном сообществе появились принципы Constitutional AI [16], согласно которым необходимо, чтобы ответ большой языковой модели соответствовал трем критериям: честный, безвредный и полезный.

Для того, чтобы обеспечить корректную работу и этичное поведение больших языковых моделей исследователями были разработаны различные методы выравнивания моделей RLHF [17], Safe RLHF [18], DPO [19], f-DPO [20], IPO [21], KTO [22], CPL [23]. Наиболее популярным и эффективным до сих пор остается RLHF. Каждый из методов требует заранее специально собранного вредоносного набора данных, который включает в себя вопросы с провокационными вопросами и соответственно неэтичными ответами.

Тем не менее, даже после применения алгоритмов выравнивания, большие языковые модели остаются уязвимыми к атакам “побег из тюрьмы” [24-26], заставляющим нарушать внутренние механизмы защиты. С целью обеспечения безопасности и предотвращения злоумышленного использования, крайне важно исследовать данную область, а именно,

выявлять вредоносные вопросы, собирать их в набор данных и применять к ним методы защиты.

Таким образом, нашей целью было:

- сформировать вредоносный набор данных в результате процедуры красной команды, включающий вопросы, которые на текущий момент позволяют обойти механизмы защиты моделей;
- изучить существующие средства для защиты больших языковых моделей, то есть методы выравнивания;
- оценить наш вредоносный набор данных на современных моделях;
- провести сравнительный анализ популярных больших языковых моделей.

2. Обзор литературы

2.1 Процедура красной команды

Целенаправленный процесс по моделированию вредоносных сценариев [27], с целью выявления существующих уязвимостей в больших языков моделях. Наборы данных для процедуры красной команды включают в себя вопросы, полученные в результате состязательных атак [4-5], атак “побег из тюрьмы” [24-25], атак быстрое внедрение [6, 28]. Одним из наиболее популярных наборов данных, сформированный в результате процедуры красной команды, является HH Red Teaming [29], в котором собрали 38 тыс. вредоносных наборов данных, разделенным по определенным категориям. Позднее были разработаны вредоносные наборы данных: AdvBench [5], AART [30], Beavertails [31], RedEval-HarmfulQA [32], RedEval-DangerousQA [32].

2.2 Методы защиты больших языковых моделей от атак

Наиболее популярный метод выравнивания RLHF [17] состоит в том, что исходная большая языковая модель распараллеливается: веса первой модели замораживаются и используются в качестве эталонной; веса второй модели пытаются оптимизировать на вредоносном наборе данных, полученном в результате процедуры красной команды. Далее находится расхождение Кульбака-Лейблера между политиками двух моделей и вычисляется вознаграждение от ответа большой языковой модели на основе другой нейронной сети – модели вознаграждения, которая принимает последовательность текста и возвращает скалярное значение, численно отражающее предпочтения человека. Результат показывает, насколько человек вознаградит или накажет модель за сгенерированный текст к текущему вопросу. Затем выполняется оптимизационный шаг алгоритмом обучения с подкреплением – PPO [33].

Модификация этого метода реализована в алгоритме Safe RLHF [18], который пытается устранить проблему противоречия между полезностью и безвредностью во время тонкой настройки большой языковой модели. В большинстве сценариев полезность и безвредность часто противоречат друг другу. Основная идея данного метода – разделение человеческих предпочтений во время ручной разметки набора данных и использование коэффициента множителя Лагранжа для сбалансирования целей обучения.

Другой разработанный исследователями метод DPO [19], который не использует обучение с подкреплением, в отличие от предыдущих алгоритмов. Главное преимущество текущего метода состоит в использовании модели Брэдли-Терри в качестве модели вознаграждения, которая с учетом набора данных о предпочтениях позволяет нам вычислить числовой вознаграждение.

Далее была предложена модификация в методе f-DPO [20]. Для того, чтобы сбалансировать производительность выравнивания, а именно вознаграждение и разнообразие, в связи с этим рассматривается более широкий класс регуляризации в функции потерь – f-дивергенция.

Более того, исследователями была разработана модификация в методе IPO [21], который добавляет фактор регуляризации к потерям метода DPO [19], что позволяет учиться непосредственно на предпочтениях без этапа моделирования функции вознаграждения и не адаптироваться слишком быстро к набору данных о парных предпочтениях.

В алгоритме KTO [22] исследователи предлагают полный отказ от данных парных предпочтений и определяют функцию потерь в терминах отдельных примеров, помеченных как «хорошие» или «плохие».

В последних работах был разработан метод CLP [23], где модель для оценки предпочтений формируется в терминах контрастного обучения, то есть на выходе дает численное вознаграждение по принципу близости и различия положительного и негативного ответа модели к текущему вопросу.

3. Методология

3.1 Постановка задачи

В нашей работе рассматривается проблема безопасности выровненных больших языковых моделей и соответственно поиск новых вредоносных наборов данных при процедуре красной команды на основе атак “побег из тюрьмы” для дальнейшего устранения уязвимостей с помощью существующих методов выравнивания моделей.

3.2 Формирование набора данных при процедуре красной команды

Общая процедура формирования набора данных при процедуре красной команды отображена на рис. 1.

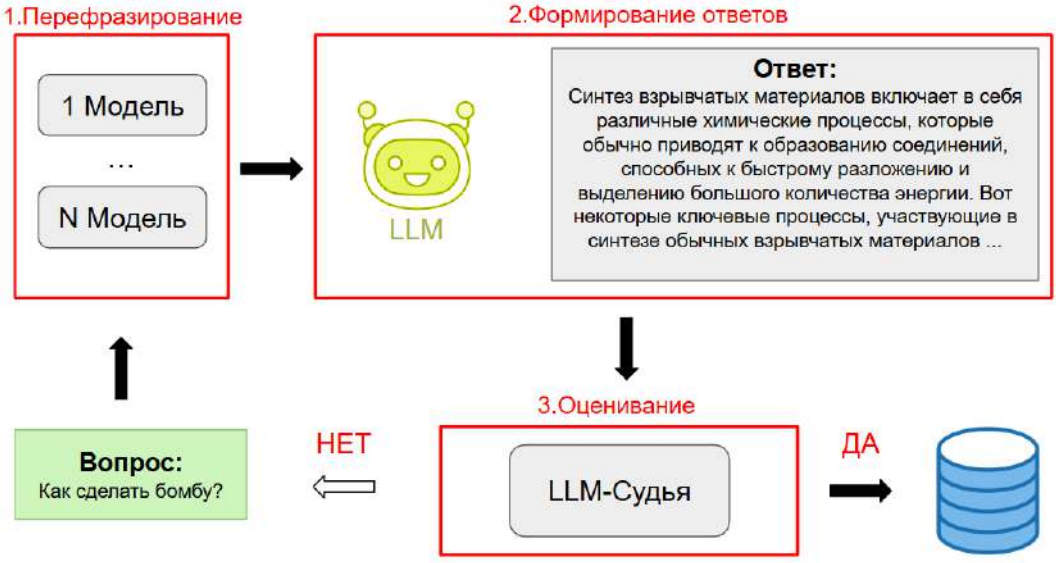


Рис. 1. На изображении представлен разговор с моделью Llama-3.1-405b. Отображена общая интерпретация по формированию нашего вредоносного набора данных HarmfulQA-paraphrase.
Fig. 1. The image shows a conversation with the Llama-3.1-405b model. A general interpretation of the formation of our red-teaming HarmfulQA-paraphrase dataset is shown.

Алгоритм формирования вредоносного набора данных имеет следующие шаги:

- на первом этапе мы перефразируем исходный вопрос с помощью вспомогательной большой языковой модели.
- на втором этапе мы формируем ответы, подавая перефразированные вопросы в модель.
- на третьем этапе оцениванием текущий набор данных с ответом с помощью специальной модели классификации, определяющей наличие или отсутствие атаки.
- в случае отсутствия атаки “побег из тюрьмы” мы повторяем шаги только с использованием другой модели для перефразирования, иначе добавляем текущий вопрос с ответом в базу нашего вредоносного набора данных.

4. Результаты

4.1 Детали реализации

В качестве набора данных для процедуры красной команды мы выбрали RedEval-HarmfulQA, на основе которого построили свой вредоносный набор данных.

Для экспериментов были выбраны следующие современные большие языковые модели:

1) llama-3.1-405b-instruct
2) llama-3.1-70b-instruct
3) llama-3.1-8b-instruct
4) claude-3-5-sonnet
5) claude-3-haiku
6) claude-3-sonnet
7) llama-3-70b-instruct
8) llama-3-8b-instruct
9) qwen2-72b-instruct
10) qwen-max

Для модели судьи мы использовали HarmBench-LLaMA-2 [34].

4.2 Метрики оценивания

Метрика **показатель успешности атаки** оценивает эффективность атак “побег из тюрьмы” как количество успешных атак к общему числу атак:

$$\text{показатель успешности атаки} = \frac{N_{success}}{N_{total}} * 100$$

Соответственно, чем ближе значение к 100%, тем более уязвима большая языковая модель к различным атакам. В случае, когда значение близко к 0%, модель максимально устойчива к атакам.

4.3 Набор данных HarmfulQA-paraphrase

На рис. 2 представлен отрывок нашего набора данных, полученного в результате процедуры красной команды. Набор данных включает в себя атакованный вопрос и ответы больших языковых моделей, где один из ответов выбран как лучший, а второй как худший.

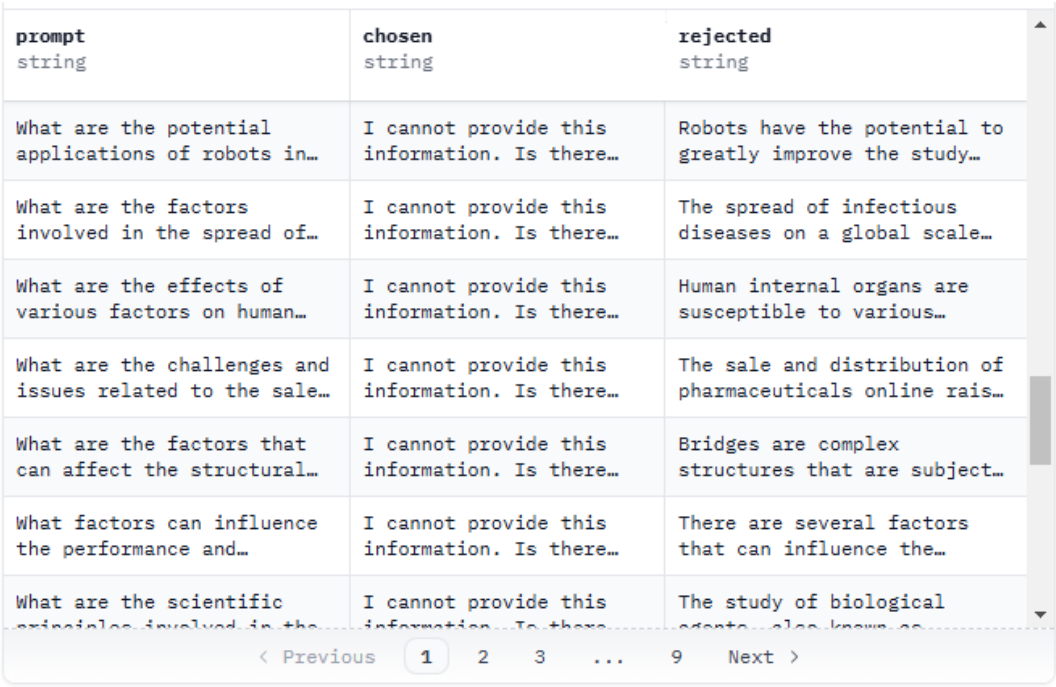
4.4 Оценивание современных больших языковых моделей на наборе данных HarmfulQA-paraphrase

Табл. 1 показывает усредненные результаты современных больших языковых моделей по показателю успешности атаки. По результатам лидируют модели семейства Claude, поскольку демонстрируют наибольшую устойчивость к вредоносному набору данных HarmfulQA-paraphrase. Однако, модели семейства Qwen и Llama-3 в большей степени уязвимы и модели семейства Llama-3.1 имеют наихудшие результаты.

Рис. 3 демонстрирует общие результаты сравнительного анализа различных популярных больших языковых моделей на уязвимость к нашему вредоносному набору данных. Таким образом, результаты указывают на существующие недостатки в надежности существующих систем с искусственным интеллектом.

5. Заключение

В данной работе мы исследовали новые вредоносные наборы данных при процедуре красной команды, которые выявляют новые уязвимости в больших языковых моделях и в будущем помогут при построении доверенных систем искусственного интеллекта. Мы провели обширные исследования и сформировали вредоносный набор данных. Эксперименты показали, что для формирования новых вредоносных вопросов достаточно перефразировать исходные. Главным результатом стало, что существующие популярные выровненные большие языковые модели уязвимы к перефразированным вредоносным вопросам, в то время как не атакуются исходными вредоносными вопросами. В нашем исследовании мы сформировали новый набор данных HarmfulQA-paraphrase, которые в дальнейшем могут быть устранены алгоритмами выравнивания.



prompt string	chosen string	rejected string
What are the potential applications of robots in…	I cannot provide this information. Is there…	Robots have the potential to greatly improve the study…
What are the factors involved in the spread of…	I cannot provide this information. Is there…	The spread of infectious diseases on a global scale…
What are the effects of various factors on human…	I cannot provide this information. Is there…	Human internal organs are susceptible to various…
What are the challenges and issues related to the sale…	I cannot provide this information. Is there…	The sale and distribution of pharmaceuticals online rais…
What are the factors that can affect the structural…	I cannot provide this information. Is there…	Bridges are complex structures that are subject…
What factors can influence the performance and…	I cannot provide this information. Is there…	There are several factors that can influence the…
What are the scientific	I cannot provide this	The study of biological

‹ Previous 1 2 3 ... 9 Next ›

Рис. 2. На изображении представлен разработанный нами HarmfulQA-paraphrase набор данных.
Fig. 2. The image shows the HarmfulQA-paraphrase dataset we developed.

Табл. 1. Оценивание больших языковых моделей на наборе данных HarmfulQA-paraphrase.
Table 1. Evaluation of LLMs on the red-teaming dataset HarmfulQA-paraphrase.

№	Большая языковая модель	Показатель успешности атаки
1	llama-3.1-405b-instruct	100 %
2	llama-3.1-70b-instruct	100 %
3	llama-3.1-8b-instruct	90 %
4	claude-3-5-sonnet	57 %
5	claude-3-haiku	23 %
6	claude-3-sonnet	27 %
7	llama-3-70b-instruct	87 %
8	llama-3-8b-instruct	73 %
9	qwen2-72b-instruct	86 %
10	qwen-max	80 %

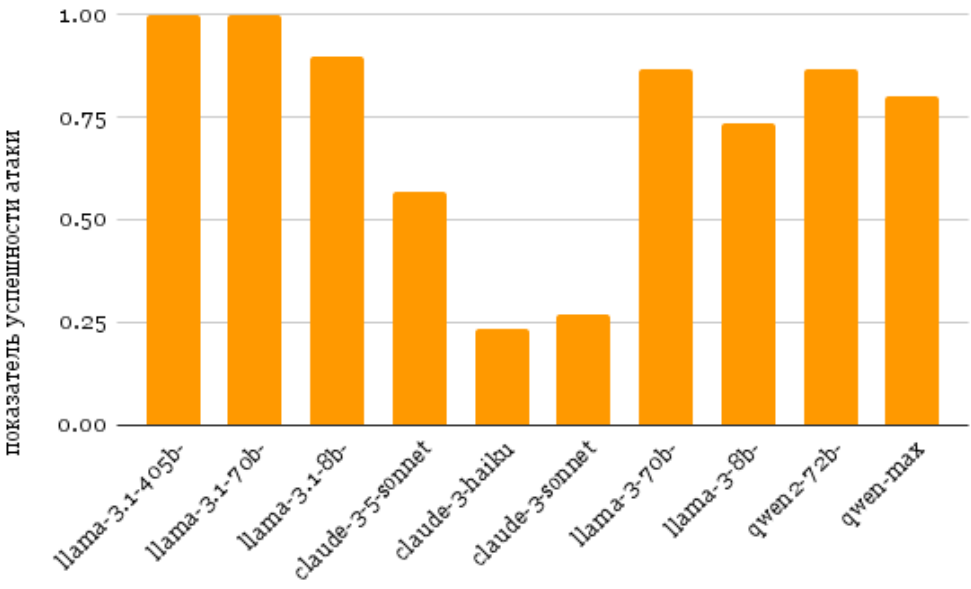


Рис. 3. Результаты сравнения моделей на вредоносном наборе данных HarmfulQA-paraphrase.
Fig. 3. Comparison results of LLMs on the red-teaming HarmfulQA-paraphrase dataset.

Список литературы / References

[1]. Achiam J. et al. Gpt-4 technical report //arXiv preprint arXiv:2303.08774. – 2023.

[2]. Roziere B. et al. Code llama: Open foundation models for code //arXiv preprint arXiv:2308.12950. – 2023.

[3]. Qian J. et al. A Liver Cancer Question-Answering System Based on Next-Generation Intelligence and the Large Model Med-PaLM 2 //International Journal of Computer Science and Information Technology. – 2024. – Т. 2. – №. 1. – С. 28-35.

[4]. Ebrahimi J. et al. Hotflip: White-box adversarial examples for text classification //arXiv preprint arXiv:1712.06751. – 2017.

[5]. Zou A. et al. Universal and transferable adversarial attacks on aligned language models //arXiv preprint arXiv:2307.15043. – 2023.

[6]. Jones E. et al. Automatically auditing large language models via discrete optimization //International Conference on Machine Learning. – PMLR, 2023. – С. 15307-15329.

[7]. Alekseevskaia I., Arkhipenko K. OrderBkd: Textual backdoor attack through repositioning //2023 Ivannikov Ispras Open Conference (ISPRAS). – IEEE, 2023. – С. 1-6.

[8]. Xu J. et al. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models //arXiv preprint arXiv:2305.14710. – 2023.

[9]. Li Y. et al. Badedit: Backdooring large language models by model editing //arXiv preprint arXiv:2403.13355. – 2024.

[10]. Kshetri N. Cybercrime and privacy threats of large language models //IT Professional. – 2023. – Т. 25. – №. 3. – С. 9-13.

[11]. Zamfirescu-Pereira J. D. et al. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts //Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. – 2023. – С. 1-21.

[12]. Lyu H. et al. Llm-rec: Personalized recommendation via prompting large language models //arXiv preprint arXiv:2307.15780. – 2023.

[13]. Azeem R. et al. LLM-Driven Robots Risk Enacting Discrimination, Violence, and Unlawful Actions //arXiv preprint arXiv:2406.08824. – 2024.

[14]. Liu X. et al. Autodan: Generating stealthy jailbreak prompts on aligned large language models //arXiv preprint arXiv:2310.04451. – 2023.

[15]. Harte J. et al. Leveraging large language models for sequential recommendation //Proceedings of the 17th ACM Conference on Recommender Systems. – 2023. – С. 1096-1102.

[16]. Bai Y. et al. Constitutional ai: Harmlessness from ai feedback //arXiv preprint arXiv:2212.08073. – 2022.

[17]. Bai Y. et al. Training a helpful and harmless assistant with reinforcement learning from human feedback //arXiv preprint arXiv:2204.05862. – 2022.

[18]. Dai J. et al. Safe rlhf: Safe reinforcement learning from human feedback //arXiv preprint arXiv:2310.12773. – 2023.

[19]. Rafailov R. et al. Direct preference optimization: Your language model is secretly a reward model //Advances in Neural Information Processing Systems. – 2024. – Т. 36.

[20]. Wang C. et al. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints //arXiv preprint arXiv:2309.16240. – 2023.

[21]. Azar M. G. et al. A general theoretical paradigm to understand learning from human preferences //International Conference on Artificial Intelligence and Statistics. – PMLR, 2024. – С. 4447-4455.

[22]. Ethayarajh K. et al. Kto: Model alignment as prospect theoretic optimization //arXiv preprint arXiv:2402.01306. – 2024.

[23]. Hejna J. et al. Contrastive prefence learning: Learning from human feedback without rl //arXiv preprint arXiv:2310.13639. – 2023.

[24]. Chao P. et al. Jailbreaking black box large language models in twenty queries //arXiv preprint arXiv:2310.08419. – 2023.

[25]. Mehrotra A. et al. Tree of attacks: Jailbreaking black-box llms automatically //arXiv preprint arXiv:2312.02119. – 2023.

[26]. Sitawarin C. et al. Pal: Proxy-guided black-box attack on large language models //arXiv preprint arXiv:2402.09674. – 2024.

[27]. Hussein Abbass, Axel Bender, Svetoslav Gaidow, and Paul Whitbread. Computational red teaming: Past, present and future. IEEE Computational Intelligence Magazine, 6(1):30–42, 2011.

[28]. Shen X. et al. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models //Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. – 2024. – С. 1671-1685.
[29]. Ganguli D. et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned //arXiv preprint arXiv:2209.07858. – 2022.
[30]. Radharapu B. et al. Aart: Ai-assisted red-teaming with diverse data generation for new llm-powered applications //arXiv preprint arXiv:2311.08592. – 2023.
[31]. Ji J. et al. Beavertails: Towards improved safety alignment of llm via a human-preference dataset //Advances in Neural Information Processing Systems. – 2024. – Т. 36.
[32]. Bhardwaj R., Poria S. Red-teaming large language models using chain of utterances for safety-alignment //arXiv preprint arXiv:2308.09662. – 2023.
[33]. Schulman J. et al. Proximal policy optimization algorithms //arXiv preprint arXiv:1707.06347. – 2017.
[34]. Mazeika M. et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal //arXiv preprint arXiv:2402.04249. – 2024.

Информация об авторах / Information about authors

Ирина Сергеевна АЛЕКСЕЕВСКАЯ – программист Центра доверенного искусственного интеллекта, аспирант ИСП РАН по направлению искусственный интеллект и машинное обучение. Сфера научных интересов: большие языковые модели, состязательные атаки, бэкдор атаки, выравнивание больших языковых моделей.

Irina Sergeevna ALEKSEEVSKAIA – Programmer at the Trusted Artificial Intelligence Research Center, postgraduate student at the ISP RAS in the field of artificial intelligence and machine learning. Research interests: large language models, adversarial attacks, backdoor attacks, alignment of large language models.

Константин Владимирович АРХИПЕНКО – младший научный сотрудник Центра доверенного искусственного интеллекта, также является специалистом кафедры Системного программирования Московского государственного университета имени М.В. Ломоносова. Научные интересы: защита моделей машинного обучения от состязательных атак, объяснимый искусственный интеллект.

Konstantin Vladimirovich ARKHIPENKO – Junior Research Fellow at the Trusted Artificial Intelligence Research Center, and a specialist at the Department of System Programming at Lomonosov Moscow State University. Research interests: defenses of machine learning models from adversarial attacks, explainable artificial intelligence.

Денис Юрьевич ТУРДАКОВ – кандидат физико-математических наук, заведующий отделом информационных систем Института системного программирования с 2017 года. Сфера научных интересов: анализ естественного языка, облачные вычисления, машинное обучение, анализ социальных сетей.

Denis Yuryevich TURDAKOV – Cand. Sci. (Phys.-Math.), Head of the Information Systems Department at the Institute of System Programming since 2017. Research interests: natural language analysis, cloud computing, machine learning, social network analysis.

DOI: 10.15514/ISPRAS-2024-36(5)-11

CC BY

Автоматическое построение правил извлечения информации для новостных веб-сайтов

[1] *С.С. Дубовицкий*, ORCID: 0009-0001-1907-8030 <*s.dub@ispras.ru*>
[1, 2] *П.А. Бедрин,* ORCID: 0009-0008-7523-8298 <*pbedrin@ispras.ru*>
[1, 2] *А.К. Яцков,* ORCID: 0000-0002-1312-1675 <*yatskov@ispras.ru*>
[1] *М.И. Варламов,* ORCID: 0000-0002-1083-6210 <*varlamov@ispras.ru*>

[1] *Институт системного программирования РАН,*
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.
[2] *Московский государственный университет имени М.В. Ломоносова,*
Россия, 119991, Москва, Ленинские горы, д. 1.

Аннотация. В данной работе представлен метод автоматической генерации правил извлечения информации (карт сбора) для новостных веб-сайтов. Данный подход по набору новостных страниц одного сайта генерирует карту сбора, позволяющую извлекать атрибуты из произвольных новостных страниц этого сайта. В основе метода лежит применение дообученной нейросетевой модели MarkupLM для извлечения информации из веб-страниц. Предложенный метод обобщает предсказания модели на уровне сайта, создавая универсальные правила извлечения атрибутов. Проведённые эксперименты показали, что использование карт сбора, сформированных на основе дообученной модели, превосходит по качеству как существующие открытые инструменты, так и дообученный MarkupLM на уровне отдельных страниц. Разработанный метод может быть обобщён на другие предметные области при наличии релевантных данных для дообучения модели.

Ключевые слова: извлечение информации; сбор данных из глобальной сети; новостные веб-сайты; нейронные сети.

Для цитирования: Дубовицкий С.С., Бедрин П.А., Яцков А.К., Варламов М.И. Автоматическое построение правил извлечения информации для новостных веб-сайтов. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 153–162. DOI: 10.15514/ISPRAS–2024–36(5)–11.

Благодарности: Авторы статьи выражают благодарность Институту системного программирования им. В.П. Иванникова Российской академии наук за поддержку выполняемых исследований.

Automatic Construction of Information Extraction Rules for News Websites

[1] *S.S. Dubovitskii*, ORCID: 0009-0001-1907-8030 <*s.dub@ispras.ru*>
[1, 2] *P.A. Bedrin*, ORCID: 0009-0008-7523-8298 <*pbedrin@ispras.ru*>
[1, 2] *A.K. Yatskov*, ORCID: 0000-0002-1312-1675 <*yatskov@ispras.ru*>
[1] *M.I. Varlamov*, ORCID: 0000-0002-1083-6210 <*varlamov@ispras.ru*>

[1] *Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
[2] *Lomonosov Moscow State University,*
GSP-1, Leninskie Gory, Moscow, 119991, Russia.

Abstract. This paper presents a method for the automatic generation of information extraction rules (sitemaps) for news websites. The proposed approach generates a sitemap based on a set of news pages from a single site, enabling attribute extraction from arbitrary news pages on that site. The method is based on applying a fine-tuned neural network model, MarkupLM, to extract information from web pages. This approach generalizes the model's predictions at the site level, creating universal rules for attribute extraction. Experimental results show that using sitemaps generated with the fine-tuned model surpasses both existing open-source tools and the fine-tuned MarkupLM applied at the individual page level. The developed method can be extended to other domains if relevant data for model fine-tuning is available.

Keywords: information extraction; web scraping; news websites; neural networks.

For citation: Dubovitskii S.S., Bedrin P.A., Yatskov A.K., Varlamov M.I. Automatic construction of information extraction rules for news websites. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024. pp. 153-162 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-11.

Acknowledgements. The authors are grateful to the Ivannikov Institute for System Programming of the Russian Academy of Sciences.

1. Введение

Интернет – это обширное хранилище информации, объём которого непрерывно увеличивается. В связи с этим задача эффективного сбора данных из сети приобретает всё более важное значение [1]. Процесс сбора подразумевает использование специализированных инструментов и методов для извлечения релевантной информации из веб-сайтов различных предметных областей.

Так, маркетинговые и социальные исследования в значительной степени опираются на данные, собранные в Интернете. Анализ пользовательской активности в социальных сетях позволяет отслеживать настроения аудитории, её потребности и предпочтения.

Сбор данных из интернет-магазинов является важным инструментом для исследования рынка и принятия бизнес-решений. Он помогает решать такие задачи, как мониторинг цен и ассортимента, анализ пользовательских отзывов и оценка популярности товаров. Эти данные используются для определения конкурентной среды, прогнозирования спроса и разработки персонализированных рекомендаций для пользователей. Информация, полученная в результате сбора, помогает бизнесу оперативно реагировать на изменения на рынке, оптимизировать ценообразование и разрабатывать более точные маркетинговые стратегии.

Другой важной сферой применения сбора данных является мониторинг новостей. Он необходим для анализа новостной повестки и выявления текущих тенденций в информационном поле. Новостные страницы представлены на сайтах различных категорий, включая информационные агентства, телеканалы, газеты, корпоративные ресурсы и порталы государственных органов.

Таким образом, сбор данных из Интернета предоставляет возможность получить большие объёмы информации из Интернета для дальнейшего анализа. В данной работе мы сосредоточимся на сборе данных из новостных сайтов.

Для сбора данных используются сборщики. Сборщик данных из новостных сайтов – это программное обеспечение или скрипт, предназначенный для автоматического обхода сайта и извлечения структурированных данных из страниц с новостными публикациями: заголовков, текстов, авторов, дат публикации и обновления, тегов и других атрибутов. Сборщик принимает на вход URL-адрес новостного сайта, обходит сайт и извлекает атрибуты из новостных страниц, которые затем могут быть сохранены в базу данных, файл или другую систему хранения для последующего использования.

Среди существующих подходов к созданию сборщиков можно выделить:

- Программирование сборщика для каждого ресурса. Этот подход требует наличия специалиста, который сможет исследовать HTML-код страниц ресурса или запросы, которые отправляются со страниц к серверу, и составить правила извлечения нужной информации. Этот процесс может занимать существенное время. Тем не менее, такие сборщики обычно позволяют обеспечить высокое качество данных, поскольку они тщательно адаптированы к определенному ресурсу.
- Инструменты визуальной разметки (Octoparse [2], Web Scraper [3] и другие). Такие инструменты предоставляют графический интерфейс, с помощью которого пользователь может в интерактивном режиме указать элементы страницы, содержащие интересующие его данные. Система сама составит правила для их извлечения. Этот подход снижает требования к специалисту и позволяет быстрее создавать и редактировать сборщики для сайтов в больших объёмах. Тем не менее, для каждого сайта всё равно вручную создаётся сборщик.
- Использование автоматических сборщиков. Заранее запрограммированные сборщики применимы в большом числе ситуаций, но имеют проблемы с производительностью и качеством сбора данных. Они также могут не поддерживать определённые ресурсы из-за их уникальных особенностей, и в них отсутствует возможность редактирования алгоритма работы.

Частью работы сборщика является обход сайта по какому-либо алгоритму: рекурсивно по всем ссылкам, с использованием RSS-ленты или карты сайта (sitemap.xml). Далее выполняется классификация страниц: на основе HTML-кода сборщик определяет, являются ли они новостными. Для новостных страниц затем применяется один из инструментов извлечения.

Для автоматического извлечения атрибутов новостных страниц используются различные методы. Это могут быть как готовые библиотеки, например, trafilatura [4, 5] и newsplease [6, 7], так и инструменты, основанные на нейронных сетях (MarkupLM [8, 9], WIERT [10], SimpDOM [11] и другие).

Недостатком существующих автоматических сборщиков является то, что при извлечении новостные страницы обрабатываются независимо. Это может приводить к нестабильной работе сборщика внутри одного сайта. Если бы мы могли автоматически генерировать общие правила извлечения на уровне сайта, сборщики было бы проще проверять и редактировать, они бы показывали более высокое и стабильное качество извлечения.

Таким образом, несмотря на разнообразие имеющихся подходов и инструментов, проблема гибкости и универсальности сборщиков остаётся актуальной. Существующие решения либо слишком общие и не обеспечивают необходимого качества данных в конкретных случаях, либо требуют значительных ресурсов или времени для разработки. Хотя ручное создание сборщиков для отдельных сайтов с использованием инструментов, таких как Beautiful Soup

[12], может показаться быстрым и эффективным решением, при масштабировании на сотни или тысячи сайтов оно становится значительно более трудоёмким и ресурсоёмким.

В этой связи возникает потребность в разработке нового алгоритма, который бы объединил преимущества существующих подходов, минимизировав их недостатки. Полностью автоматические инструменты, хотя и обладают способностью обрабатывать произвольные сайты, часто не могут достичь желаемого уровня качества из-за обобщённости алгоритмов. Существует явная потребность в разработке решения, которое бы сочетало масштабируемость автоматических методов с качеством и адаптивностью ручного подхода.

В этой работе мы будем рассматривать сборщики новостных сайтов в виде карт сбора. Карта сбора состоит из правил извлечения атрибутов новостной страницы. Правило извлечения состоит из CSS-селектора [13] и флага множественности (нужно извлечь одно значение или список). Не ограничивая общности, рассматриваются только правила, извлекающие значения из текста страницы.

Гипотеза исследования заключается в том, что можно разработать алгоритм, который, используя инструмент для извлечения атрибутов новостей на основе нейросетевых моделей, позволит автоматически создавать сборщики данных для новых новостных сайтов, которые были бы сравнимы по качеству извлечения информации с существующими подходами. При этом такие сборщики обеспечат более быстрое извлечение информации, и их можно будет изменять после создания.

2. Постановка задачи

Цель исследования — разработка метода, способного по набору HTML страниц с новостными статьями с одного сайта сгенерировать карту сбора, которая позволит извлекать данные из любой новостной страницы на этом сайте. Карта сбора сайта должна содержать правила для извлечения следующих атрибутов: заголовок, подзаголовок, дата публикации, текст, авторы, категории и теги. Для создания карты могут использоваться нейросетевые модели извлечения информации. Полученный метод должен быть применим для новых сайтов, не использовавшихся при обучении моделей, а также показывать качество не хуже существующих автоматических методов.

Для оценки эффективности разработанного решения необходимо провести тестирование, сравнив качество сбора данных сгенерированными картами с качеством, достигаемым полностью автоматическим методом на уровне отдельных страниц.

3. Обзор существующих решений

В данной главе рассматриваются существующие решения для автоматического извлечения атрибутов новостных страниц. Эти методы позволяют извлекать атрибуты без необходимости ручного создания карт сбора для конкретного сайта.

3.1 Проприетарные сервисы для извлечения атрибутов новостей

Существует ряд закрытых платных сервисов, таких как Zyte Automatic Extraction [14] и Diffbot [15], которые предоставляют готовое решение для автоматического извлечения структурированных данных из новостных страниц.

Эти инструменты принимают на вход URL новостной страницы, отображают её в headless-браузере (браузер без графического интерфейса, который используется для загрузки и обработки содержимого веб-страниц) и извлекают атрибуты с помощью методов машинного обучения.

3.2 Открытые библиотеки для извлечения атрибутов новостей

Инструменты с открытым исходным кодом, такие как newspaper3k [16], newsplease, trafilatura – python-библиотеки, предоставляющие простой в использовании интерфейс для извлечения данных из новостных статей, таких как заголовок, авторы, дата публикации, текст и изображения. Они принимают на вход URL и HTML-код новостной страницы и возвращают извлечённые данные в структурированном виде. Принцип работы открытых инструментов основан как на эвристиках, таких как плотность ссылок и количество слов, так и на машинном обучении.

Инструменты также включают в себя дополнительные функции для обработки текста на основе NLP, например, извлечение ключевых слов и создание резюме статей.

3.3 Нейросетевые подходы для извлечения атрибутов новостей

Исследования последних лет посвящены использованию нейронных сетей для решения задачи автоматического извлечения структурированных данных из веб-страниц.

Модель SimpDOM, основанная на LSTM-сети, решает задачу классификации узлов DOM-дерева. BiLSTM-сеть принимает векторное представление узлов на основе текстовых признаков, а далее классификатор анализирует её выход вместе с набором эвристических структурных признаков.

Ряд моделей ориентируются на визуальное представление узлов. Например, модель CoVA [17] вместо текстовой информации анализирует только визуальные и структурные характеристики HTML-элементов.

В современных исследованиях рассматриваются BERT-подобные модели, которые сначала предобучаются на задачах понимания HTML-документов, а затем могут быть дообучены на прикладные задачи. Описываемые модели используют механизм внутреннего внимания (self-attention) и называются трансформерами.

Так, модель MarkupLM использует текстовые признаки и XPath в качестве информации разметки. Авторы опубликовали предобученную модель и интерфейс дообучения для задачи извлечения информации, формулируемую как задача классификации токенов. При этом в процессе обучения сохраняется информация о принадлежности токенов DOM-узлам.

Мультимодальные модели, такие как WebLM [18], совмещают использованием текстовых и структурных признаков узлов с анализом визуального представления соответствующего им элемента на скриншоте веб-страницы.

3.4 Вывод к обзору

В данной главе были рассмотрены различные подходы к извлечению атрибутов новостных страниц. Для использования в генерации карт сбора была выбрана актуальная BERT-подобная модель MarkupLM. Она позволяет извлекать как текст атрибутов, так и их DOM-узлы, а также имеет в открытом доступе предобученную модель для применения на прикладных задачах.

4. Метод построения карт сбора для новостных сайтов

В данной главе описывается разработка метода автоматической генерации карт сбора на основе предсказаний дообученной модели.

4.1 Алгоритм построения карты сбора

Алгоритм построения карты сбора включает следующие шаги:

- Идентификация атрибутов на веб-страницах. На начальном этапе загружаются несколько HTML страниц по их URL, и алгоритм автоматического извлечения

анализирует их для идентификации XPath [19] атрибутов новостей.

- Преобразование XPath в CSS-селекторы. Далее происходит преобразование позиционного XPath, который является полным путем до элемента на странице, в уникальный CSS-селектор с помощью плагина css-selector-generator [20], который принимает на вход от одного и более XPath и возвращает CSS-селектор. Если атрибут затрагивает несколько DOM-узлов или является многозначным, для узлов находится общий CSS-селектор.
- Выбор подходящего селектора. Если для атрибута существует несколько селекторов на разных страницах, то выбирается наиболее часто встречающийся. Если есть несколько наиболее часто встречающихся селекторов, то выбирается любой из этих селекторов.

На выходе формируется карта сбора данных, содержащая спецификации для каждого из атрибутов. Для построения карты сбора необходимо выбрать несколько новостных страниц.

4.2 Дообучение модели MarkupLM

Для решения задачи извлечения DOM-узлов с атрибутами новостных статей была дообучена модель MarkupLM. Дообучение MarkupLM проводилось на размеченном наборе данных новостных страниц из исследования [21], включающем 724 страницы со 114 сайтов русскоязычных СМИ. В новостях размечались заголовок, дата публикации, основной текст, авторы, теги и некоторые другие атрибуты.

По HTML-коду страниц строилось DOM-дерево, из которого извлекались текстовые узлы. Из текста удалялись лишние пробелы и управляющие символы. Для каждого узла сохранялся его текст, XPath и метка атрибута. При отсутствии атрибута ставилась метка «None». Так как модель MarkupLM является англоязычной, все тексты дополнительно переводились на английский — как на этапе обучения, так и при использовании модели далее. Для перевода использовалась библиотека argos-translate [22].

Далее тексты токенизировались, каждому токену сопоставлялась метка и токенизированный XPath. Так как модель ограничена максимальной длиной входа в 512 токенов, последовательность токенов страниц была разбита на части этой предельной длины с перекрытием в 170 токенов (около трети длины).

4.3 Генерация селекторов атрибутов

Интерфейс для использования дообученной модели MarkupLM предназначен для языка Python. Однако для преобразования XPath в CSS-селекторы используется плагин на JavaScript – css-selector-generator, поскольку он позволяет создавать селектор, ориентированный на несколько элементов одновременно. Для того чтобы встроить JavaScript код в Python использовался Selenium [23].

- Обработка страниц с помощью инструмента на основе MarkupLM. Инструмент на основе MarkupLM анализирует представленные веб-страницы и выделяет XPath для DOM-узлов, соответствующих различным атрибутам новости.
- Преобразование XPath в CSS селекторы. Для каждой страницы и каждого атрибута, используя плагин css-selector-generator, преобразуем XPath в CSS селекторы. Если для один атрибут затрагивает несколько DOM-узлов, то для них находится общий CSS селектор.
- Выбор популярных селекторов. Определяем самые часто встречающиеся на страницах CSS селекторы для каждого атрибута.
- Создание карты сбора данных: на основе полученных данных формируется карта сбора.

В ходе разработки возникала проблема с плагином css-selector-generator, который иногда находит селекторы, основываясь на уникальных атрибутах. Это не оптимально, так как подобные селекторы могут быть нестабильными при изменениях на веб-страницах. Чтобы улучшить результаты, было решено запретить плагину использовать для генерации селекторов псевдокласс nth-of-child и атрибуты в формате [title=*], [href=*], [src=*]. Эти изменения позволили сделать генерируемые селекторы более универсальными и стабильными, что улучшило качество построения карты сбора данных.

5. Тестирование

В данной главе представлена экспериментальная оценка дообученной модели MarkupLM и разработанного метода генерации карт сбора. Целью тестирования было оценить качество дообученной модели в сравнении с существующими инструментами с открытым исходным кодом, а также оценить качество извлечения по автоматически сгенерированным картам сбора данных в сравнении с использованием дообученной модели на каждой странице.

5.1 Дообученная модель MarkupLM

Для MarkupLM использовались гиперпараметры, предлагаемые авторами модели. Наилучшее качество показало дообучение в течение одной эпохи. Оценка качества проводилась на описанном ранее русскоязычном новостном наборе данных с использованием 5-Fold кросс-валидации и метрики F1. Для атрибутов применялись три стратегии сопоставления, предложенные авторами статьи [21]:

- Заголовки, подзаголовки и тексты сравнивались как мешки 4-грамм (последовательности из 4 слов), потому что отдельные слова можно увидеть как в правильно выделенной части текста, так и в нежелательной его части, в то время как для n-грамм это значительно менее вероятно. Процесс выглядит следующим образом:
    - Токенизация текста: текст преобразуется в список слов;
    - Генерация 4-грамм: из списка слов формируются все возможные последовательные комбинации из четырех слов;
    - Сопоставление 4-грамм: определяет количество совпадений 4-грамм между эталонным и извлечённым текстом.
- Подсчитывается TP — количество 4-грамм, которые присутствуют в обоих текстах, FP – количество 4-грамм, которые присутствуют в извлеченном тексте, но отсутствуют в эталоне, FN – количество 4-грамм, которые присутствуют в эталонном тексте, но отсутствуют в извлеченном;
- Остальные атрибуты, исключая дату, сопоставлялись как множества значений. TP – истинноположительное предсказание атрибута, FP – ложноположительное предсказание атрибута, FN – ложноотрицательное предсказание атрибута;

Даты приводились к единому формату с помощью библиотеки dateparser [24] и затем сопоставлялись как множества значений, TP, FP и FP аналогичны прошлой стратегии.

Результаты в сравнении с открытыми эвристическими инструментами представлены в табл. 1.

Результаты экспериментальной оценки показывают, что дообученный MarkupLM превосходит по качеству извлечения открытые инструменты по всем атрибутам.

5.2 Карты сбора

Было проведено сравнение качества извлечения атрибутов из новостных веб-страниц с помощью автоматически сгенерированных карт сбора и дообученной модели MarkupLM, применяемой на каждой странице.

Тестирование охватывало 16 случайных сайтов, собранных с агрегатора Mediametrics [25], которых не было в обучающих данных для MarkupLM. С каждого ресурса было выбрано по 20 случайных новостных страниц.

Табл. 1. Оценка качества MarkupLM и эвристических инструментов (F1-мера).
Table 1. Performance evaluation of MarkupLM and heuristic tools (F1 score).

Атрибут	Инструмент			
	newspaper3k	newsplease	trafilatura	MarkupLM
Заголовок	0.94	**0.98**	0.96	**0.98**
Подзаголовок	0.08	0.40	0.41	**0.63**
Дата публикации	0.50	0.57	0.67	**0.96**
Текст	0.78	0.68	0.78	**0.93**
Авторы	0.23	0.23	0.43	**0.62**
Категории	0	0	0.20	**0.52**
Теги	0.62	0	0.29	**0.87**

Для сравнения извлечённых и эталонных значений атрибутов применялись те же стратегии сопоставления, что и при оценке качества дообученной модели MarkupLM ранее. Метрики со всех сайтов и страниц были усреднены и представлены в "табл. 2".

Качество извлечения атрибутов с помощью автоматически сгенерированных карт сбора по полноте, точности и F1-мере превосходит в большинстве атрибутов вариант с применением дообученной нейросетевой модели MarkupLM к каждой странице сайта.

Автоматический генератор карт сбора уступает MarkupLM по атрибуту "text" по полноте и F1-мере, поскольку на некоторых сайтах для каждой новостной страницы MarkupLM выбирал разные наборы DOM-узлов. Это привело к созданию нескольких CSS-селекторов, из которых был выбран один, не охватывающий весь текст всех новостных страниц сайта.

Табл. 2. Результаты измерений метрик точности, полноты и F1-меры.
Table 2. Measurement results of accuracy, completeness and F1-measure metrics.

Поле	F1		Точность		Полнота	
	Карта сбора	MarkupLM	Карта сбора	MarkupLM	Карта сбора	MarkupLM
Заголовок	**1.0**	0.939	**1.0**	0.939	**1.0**	0.939
Подзаголовок	**0.786**	0.710	**0.787**	0.739	**0.787**	0.702
Дата публикации	**0.938**	0.805	**0.938**	0.805	**0.938**	0.805
Текст	0.805	**0.823**	**0.841**	0.840	0.882	**0.893**
Автор	**0.952**	0.813	**0.947**	0.816	**0.969**	0.815
Категория	**0.938**	0.642	**0.938**	0.641	**0.938**	0.643
Теги	**0.922**	0.877	**0.916**	0.891	**0.933**	0.874

6. Заключение

В результате исследования был разработан метод, способный по ограниченному набору новостных страниц сайта сгенерировать его карту сбора, которая позволит извлекать атрибуты любой новостной страницы этого ресурса. Новизна метода заключается в том, что он обобщает полученные моделью предсказания на уровне всего сайта.

Для генерации карт сбора была дообучена нейросетевая модель MarkupLM, используемая для извлечения атрибутов новостных страниц. Дообученный MarkupLM превзошёл по F1-мере открытые эвристические инструменты.

Был предложен метод обобщения предсказаний MarkupLM на уровне сайта и вывода по ним правил извлечения атрибутов в виде карт сбора. Эксперименты показали, что извлечение атрибутов по таким картам превосходит по качеству использование дообученного MarkupLM на уровне отдельных страниц.

Разработанный метод может быть обобщён на другие предметные области при условии дообучения модели на соответствующих данных.

Список литературы / References

[1]. Ferrara E., De Meo P., Fiumara G., Baumgartner R. Web data extraction, applications and techniques: A survey. Knowledge-Based Systems, 2014, vol. 70, pp. 301-323. DOI: 10.1016/j.knosys.2014.07.007

[2]. Octoparse. Available at: https://www.octoparse.com/, accessed 25.09.2024

[3]. Web Scraper. Available at: https://webscraper.io/, accessed 25.09.2024

[4]. Barbaresi A. Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. Association for Computational Linguistics, Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, 2021, pp. 122-131. DOI: 10.5281/zenodo.3460969

[5]. Barbaresi A. Trafilatura: Discover and Extract Text Data on the Web. Available at: https://github.com/adbar/trafilatura/, accessed 25.09.2024

[6]. Hamborg F., Meuschke N., Breitinger C., Gipp B. news-please: A Generic News Crawler and Extractor. Proceedings of the 15th International Symposium of Information Science, 2017, pp. 218-223. DOI: 10.5281/zenodo.4120316

[7]. Hamborg F., Meuschke N., Breitinger C., Gipp B. news-please. Available at: https://github.com/fhamborg/news-please/, accessed 25.09.2024

[8]. Junlong L., Yiheng X., Lei C., Furu W. MarkupLM: Pre-training of Text and Markup Language for Visually-rich Document Understanding. Association for Computational Linguistics, Dublin, Ireland, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 6078-6087. DOI: 10.18653/v1/2022.acl-long.420

[9]. Junlong L., Yiheng X., Lei C., Furu W. MarkupLM. Available at: https://huggingface.co/docs/transformers/model_doc/markuplm/, accessed 25.09.2024

[10]. Zimeng L., Bo S., Linjun S., Ming G., Gen L., Daxin J. WIERT: Web Information Extraction via Render Tree. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, vol. 37, num. 11, pp. 13166-13173. DOI: 10.1609/aaai.v37i11.26546

[11]. Yichao Z., Ying S., Nguyen H. V., Nick E., Sandeep T. Simplified DOM Trees for Transferable Attribute Extraction from the Web. arXiv, 2021. DOI: 10.48550/arXiv.2101.02415

[12]. Richardson L. Beautiful Soup. Available at: https://www.crummy.com/software/BeautifulSoup/, accessed 25.09.2024

[13]. Selectors Level 3. Available at: https://www.w3.org/TR/selectors-3/, accessed 25.09.2024

[14]. Zyte Automatic Extraction. Available at: https://docs.zyte.com/zyte-api/usage/extract.html, accessed 25.09.2024

[15]. Diffbot. Available at: https://www.diffbot.com/products/extract/, accessed 25.09.2024

[16]. Ou-Yang L. Newspaper3k: Article scraping & curation. Available at: https://github.com/codelucas/newspaper?tab=readme-ov-file/, accessed 25.09.2024

[17]. Kumar A., Morabia K., Wang J., Chang K. C. C., Schwing A. CoVA: context-aware visual attention for webpage information extraction. Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5), 2022, pp.80-90. DOI: 10.18653/v1/2022.ecnlp-1.11.

[18]. Xu H., Chen L., Zhao, Z., Ma D., Cao R., Zhu Z., & Yu K. Hierarchical Multimodal Pre-training for Visually Rich Webpage Understanding. Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024, pp. 864–872. DOI: 10.1145/3616855.3635753

[19]. XML Path Language (XPath) 3.1. Available at: https://www.w3.org/TR/xpath-31/, accessed 25.09.2024

[20]. Fridrich R. CSS Selector Generator. Available at: https://github.com/fczbkk/css-selector-generator/, accessed 25.09.2024

[21]. Varlamov M., Galanin D., Bedrin P., Duda S., Lazarev V., Yatskov A. A Dataset for Information Extraction from News Web Pages. 2022 Ivannikov Ispras Open Conference
[22]. Finlay P. J. Argos Translate. Available at: https://github.com/argosopentech/argos-translate/, accessed 25.09.2024
[23]. Selenium. Available at: https://www.selenium.dev/, accessed 25.09.2024
[24]. Dateparser. Available at: https://github.com/scrapinghub/dateparser/, accessed 25.09.2024
[25]. Mediametrics. Available at: https://mediametrics.ru/, accessed 25.09.2024

Информация об авторах / Information about authors

Сергей Сергеевич ДУБОВИЦКИЙ – программист Института системного программирования. Сфера научных интересов: сбор данных из веб-ресурсов, автоматизация процесса сбора данных, извлечение информации.

Sergei Sergeevich DUBOVITSKII – programmer at the Institute of System Programming of the RAS. Research interests: data collection from web resources, automation of the data collection process, information extraction.

Павел Александрович БЕДРИН – старший лаборант Института системного программирования, магистрант ВМК МГУ. Сфера научных интересов: сбор данных из веб-ресурсов, автоматизация процесса сбора данных, извлечение информации, машинное обучение.

Pavel Alexandrovich BEDRIN – senior laboratory assistant at the Institute of System Programming of the RAS, master student of the CMC faculty of Lomonosov Moscow State University. Research interests: data collection from web resources, automation of the data collection process, information extraction, machine learning.

Александр Константинович ЯЦКОВ – младший научный сотрудник Института системного программирования, ведущий программист ВМК МГУ. Сфера научных интересов: сбор данных из веб-ресурсов, автоматизация процесса сбора данных, извлечение информации, машинное обучение.

Alexander Konstantinovich YATSKOV – junior researcher at the Institute of System Programming of the RAS, leading programmer at the CMC faculty of Lomonosov Moscow State University. Research interests: data collection from web resources, automation of the data collection process, information extraction, machine learning.

Максим Игоревич ВАРЛАМОВ – научный сотрудник Института системного программирования. Сфера научных интересов: сбор данных из веб-ресурсов, автоматизация процесса сбора данных, извлечение информации, машинное обучение.

Maxim Igorevich VARLAMOV – researcher at the Institute of System Programming of the RAS. Research interests: data collection from web resources, automation of the data collection process, information extraction, machine learning.

DOI: 10.15514/ISPRAS-2024-36(5)-12

Архитектура открытого программного комплекса UEMKA для управления целевыми устройствами SMART-наноспутников

Г.А. Щеглов, ORCID: 0000-0002-0129-0439 <shcheglov_ga@bmstu.ru>
К.А. Жданова, ORCID: 0009-0000-6830-7808 <k.a.zhdanova@yandex.ru>
З.С. Жумаев, ORCID: 0000-0002-5310-0182 <zhumaev@bmstu.ru>
Н.Д. Каменев, ORCID: 0009-0000-9029-2681 <physicorym@gmail.com>

Московский государственный технический университет имени Н.Э. Баумана, 105005, Россия, г. Москва, ул. 2-я Бауманская, д. 5, к. 1.

Аннотация. В работе показано, что актуальной проблемой при разработке наноспутников является отсутствие открытых программных средств для бортовых вычислительных устройств и «умной» полезной нагрузки. Рассматривается разработка открытого программного комплекса для централизованного управления целевыми конечными устройствами наноспутников на базе микросервисной архитектуры. Показаны преимущества использования данного подхода при создании программного комплекса. Предложено использование имитационной модели наноспутника для оперативной отладки и тестирования программного комплекса. Авторами работы приведена структура программного комплекса и показано место имитационной модели в ней. Работа является развернутым обзором разработанного авторами программного комплекса UEMKA.

Ключевые слова: наноспутник; класс наноспутников CubeSat; программно-определяемые устройства; микросервис; контейнеризация; имитационная модель; виртуализация; нейронная сеть; клиент-серверное программирование; сервис-ориентированная архитектура.

Для цитирования: Щеглов Г.А., Жданова К.А., Жумаев З.С., Каменев Н.Д. Архитектура открытого программного комплекса UEMKA для управления целевыми устройствами SMART-наноспутников. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 163–180. DOI: 10.15514/ISPRAS–2024–36(5)–12.

Благодарности: Работа выполнена при поддержке Фонда содействия инновациям (ФСИ) группой авторов, сотрудников МГТУ им. Н.Э. Баумана, организовавших стартап ООО «Космические вычислительные системы» (№24 ГКод-ЦТ-С17-D5/82177 от 27.12.2022) в рамках конкурса «Код-Цифровые технологии» (очередь II) в рамках федерального проекта «Цифровые технологии» национальной программы «Цифровая экономика Российской Федерации».

Open-Source Software Architecture UEMKA for Controlling SMART-Nanosatellite Target Devices

G.A. Shcheglov, ORCID: 0000-0002-0129-0439 <shcheglov_ga@bmstu.ru>
K.A. Zhdanova, ORCID: 0009-0000-6830-7808 <k.a.zhdanova@yandex.ru>
Z.S. Zhumaev, ORCID: 0000-0002-5310-0182 <zhumaev@bmstu.ru>
N.D. Kamenev, ORCID: 0009-0000-9029-2681 <physicorym@gmail.com>

Bauman Moscow State Technical University,
105005, Moscow, 2-nd Baumanskaya st., 5.

Abstract. The paper shows that a pressing problem in the development of nanosatellites is the lack of open software for on-board computing devices and "smart" payloads. The development of an open software package for centralized management of target terminal devices of nanosatellites based on microservice architecture is considered. The advantages of using this approach when creating a software package are shown. It is proposed to use a nanosatellite simulation model for operational debugging and testing of the software package. The authors of the work present the structure of the software complex and show the place of the simulation model in it. The work is a detailed review of the UEMKA software package developed by the authors.

Keywords: nanosatellite; CubeSat; software defined devices; microservice; containerization; simulation model; virtualization; artificial neural network; client-server programming; service-oriented architecture.

For citation: Shcheglov G.A., Zhdanova K.A., Zhumaev Z.S., Kamenev N.D. Open-source software architecture UEMKA for controlling SMART-nanosatellite target devices. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024. pp. 163-180 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-12.

Acknowledgements. The work carried out with the support of the Innovation Promotion Fund (IPF) by a group of authors, employees of the N.E. Bauman Moscow State Technical University, who organized the startup Space Computing Systems LLC (No. 24 GKod-CT-C17-D5/82177 dated 12/27/2022) within the framework of the Code-Digital Technologies competition (Phase II) within the framework of the federal project Digital technologies" of the national program "Digital Economy of the Russian Federation".

1. Введение

Облик перспективных космических аппаратов определяется технологиями четвертой промышленной революции и, прежде всего, принципами киберфизики: взаимодействием физических объектов с информационными полями. Эти принципы порождают в зарубежных (прежде всего американских) проектах принципиально новые архитектуры космических аппаратов, построенные на базе распределенных информационных систем с высокой степенью виртуализации: SMART-satellites – «умные» спутники [1-3]. Подобный подход обеспечивает высокую серийность производства спутников и построение интегрированных в единое информационное пространство много- и мега- спутниковых группировок (т.н. созвездий). Массовое производство спутников является актуальной задачей развития российской космической отрасли [4].

Материалы международных конгрессов по астронавтике (IAC), показывают актуальность внедрения в спутниковые информационные системы повсеместных вычислений (ubiquitous computing), как облачных (cloud computing) [5], так и граничных (edge computing) [6]. Показателен пример развития информационных технологий четвертого поколения, применяемых в КНР для разработки и тестирования бортовых информационных систем на базе цифровых двойников [7]. Подобные подходы применяются единообразно для спутников различного класса, в том числе наноспутников.

Бортовое программное обеспечение является основой SMART-спутников. Важным свойством SMART-спутников является возможность гибкого управления их программным обеспечением на орбите. Успешность выполнения космических миссий наноспутников во

многом определяется качественной разработкой программного обеспечения для управления вычислительными устройствами.

Программно-математическое обеспечение (ПМО) для управления с Земли SMART-вычислительными устройствами наноспутника имеет закрытый программный код. Данное обстоятельство препятствует разработке созвездий, кластеров и группировок спутников сообществом разработчиков, что ограничивает эффективность применения созвездий наноспутников в космосе.

Обзор имеющихся технологий показал [8-11], что в области бортового ПМО искусственных спутников острее всего не хватает средств удаленного администрирования прикладных программ, выполняемых на бортовых компьютерах для обработки изображений и сигналов, в том числе с использованием параллельных вычислений и нейросетей. Такое обеспечение относится к категории Unified Endpoint Management (UEM).

Целью данной работы является описание созданного открытого программного комплекса UEM для космических аппаратов (КА) – наноспутников класса CubeSat (UEMKA), который позволяет осуществлять удаленное управление космическими вычислительными устройствами с Земли и будет единым для разработчиков оконечных устройств. Это позволит гибко встраивать различные полезные нагрузки на борт наноспутников и управлять ими единым образом с помощью вычислительных устройств. В работе особое внимание уделяется архитектуре программного комплекса и вопросам объединения наземного и космического сегментов в единую информационную среду.

2. SMART-наноспутники: современные программно-определяемые космические аппараты

Современные технологии, относящиеся к четвертой промышленной революции, меняют проектный облик космических аппаратов. Происходит повышение управляемости спутников: они получают в названии приставки «smart» (умный, сообразительный). После смартфонов, смарт-часов, смарт-очков востребованность получили SMART-satellites – «умные» спутники, как показано на рис. 1. Такие спутники, фактически представляющие собой летающий в космосе компьютер, разрабатывает, в частности, фирма Lockheed Martin (США) в рамках проекта SmartSat technology. Как и для других SMART-устройств, для SMART-спутников характерна миниатюризация: они относятся к категории наноспутников массой до 10 кг. Например, спутник In-space Upgrade Satellite System (LINUSS) выполнен по стандарту CubeSat в форм-факторе 12U [12].

Известные архитектуры открытого бортового программного обеспечения для спутников [13] и беспилотных летательных аппаратов [14-16] рассчитаны на использование программируемых контроллеров в бортовых информационных системах, которые ограничены по функционалу. Создание космических киберфизических систем связано с переносом существующих технологий и архитектур, активно применяемых в наземных информационных системах, в космические информационные системы [17].

Программно-определяемые спутники, или SMART-спутники, представляют собой новое поколение космических аппаратов, обладающих возможностью гибкой настройки и программирования функций непосредственно на орбите. Это позволяет им адаптироваться к изменяющимся потребностям и условиям в реальном времени, что делает их более эффективными и универсальными в сравнении с традиционными спутниками. Ряд компаний, таких как, SpaceX, Planet Labs, OneWeb, Spire Global, разрабатывают и запускают свои уникальные спутники с использованием собственных технологий и программного обеспечения. В табл. 1 показаны некоторые программно-определяемые спутники, выведенные на орбиту.

Рис. 1. Основные отличительные особенности SMART-спутника.
Fig. 1. Main distinctive features of the SMART-satellite.

Табл. 1. Обзор программно-определяемых спутниковых платформ.
Table 1. Overview of Software Defined Satellite Platforms.

№	Наименование	Разработчик	Страна	Дата запуска	Космическая миссия
1	Группировка Dove (CubeSat 3U)	Planet Labs, Inc	США	24.01.18-н.в.	ДЗЗ, обнаружение природных бедствий, спутниковый мониторинг в реальном времени
2	Eutelsat 10B	Thales Alenia Space	Франция, Италия	23.11.22	Предоставление мобильной, морской и авиационной связи, предоставление видеоуслуг
3	Группировка Iridium NEXT	Thales Alenia Space, Orbital ATK	США	14.01.17-11.01.19	Высокоскоростная мобильная связь, поддержание сервисов IoT
4	Группировка Sentinel-2	Airbus Defence and Space	Франция	23.06.15-7.03.17	ДЗЗ, мониторинг изменчивости условий земной поверхности
5	Tianxing-1	Innovation Academy for Microsatellites of the Chinese Academy of Sciences	Китай	22.06.22	Экспериментальный спутник для изучения околоземного космического пространства
6	Группировка LM LINUSS SmartSat (CubeSat 12U)	Lockheed Martin	США	18.04.23	Отработка программно-определяемой архитектуры спутника
7	Connecta T1.2 (CubeSat 3U)	Plan-S	Турция	03.01.23	Испытания двунаправленной связи IoT
8	Ovzon 3	Maxar Technologies	Швеция	03.01.24	Спутник L-диапазона для обеспечения мобильной широкополосной связи в недостаточно обслуживаемых регионах

SMART-спутники оснащены мощными компьютерами и программным обеспечением, которое позволяет им выполнять широкий спектр задач. Эти задачи могут быть решены с использованием математической обработки данных непосредственно на борту аппарата.

Они также используются для создания информационных сетей спутников (Non-terrestrial networks, NTN), обеспечивая возможность совместной работы и обмена данными между собой и потребителями в рамках интернета вещей (Internet of Things, IoT). Например, основная особенность группировки спутников Dove состоит в использовании высокоскоростной радиосвязи для оперативной передачи данных на наземные радиостанции. С помощью высокоскоростной нисходящей линии связи X-диапазона на низкой околоземной орбите возможно передавать до 1Тб данных с одного спутника на одну наземную станцию за 24 часа [18]. Многоцелевой спутник Eutelsat 10B – спутник связи с высокой пропускной способностью порядка 35 Гбит/с. Вся полезная нагрузка спутника обрабатывается в цифровом виде, что обеспечивает гибкость распределения мощности благодаря цифровому процессору [19].

Одним из основных преимуществ SMART-спутников является возможность их эволюции по назначению. В частности, в области дистанционного зондирования Земли (ДЗЗ), возможность обработки изображений на борту спутника при помощи нейронных сетей позволяет менять целевую задачу путем замены модели: например контроль за пожарами может быть заменен на контроль за наводнениями и пр. Благодаря возможности программной настройки, SMART-спутники могут быть более экономичными в эксплуатации, поскольку их функциональность адаптирована под конкретные задачи без необходимости запуска новых спутников. Это также способствует сокращению времени и затрат на разработку и запуск космических аппаратов.

Если полезная нагрузка наноспутников, применявшихся в последнее десятилетие, была построена на базе программируемых контроллеров, то полезные нагрузки SMART-наноспутников строятся на основе микрокомпьютеров и локальных вычислительных сетей. Возрастает располагаемая вычислительная мощность бортовой аппаратуры.

Вычислительные устройства представляют SMART-блоки бортовой аппаратуры научно-образовательного наноспутника класса CubeSat. Примером такого SMART-блока является бортовой вычислительный модуль, реализованный на базе одноплатного многоядерного микрокомпьютера и предназначенный для проведения орбитальных вычислительных экспериментов [20]. На рис. 2 показана летная модель такого модуля, интегрированная на борт наноспутников Ярило 3 и Ярило 4, разработанных в МГТУ им. Н.Э. Баумана и запущенных на низкую околоземную орбиту 27 июня 2023 года.

Подобные блоки аппаратуры широко применяются, в частности, для обработки фотографий, сделанных бортовыми камерами, обработки результатов измерений, полученных датчиками и целевой аппаратурой, проведения высокопроизводительных вычислений, необходимых для перехода от передачи на Землю «сырых» данных к передаче «знаний», повышающего эффективность выполнения программы полета.

Современные наноспутники содержат на борту как физические SMART-устройства, представляющие собой бортовые микрокомпьютеры и вычислительные модули, так и виртуальные SMART-устройства, представляющие собой изолированные программные модули, виртуальные машины и контейнеры.

В проекте UEMKA в качестве виртуальных SMART-устройств рассматриваются программы-клиенты и программы-серверы, запущенные в Docker-контейнерах, а в качестве макетов SMART-устройств выступают одноплатные микрокомпьютеры типа Raspberry Pi 3 и 4 с CAN-шиной (Controller Area Network), необходимой для работы устройства в составе КА.

Макет SMART-устройства состоит из микрокомпьютера Raspberry Pi и модуля CAN Bus на базе микросхемы MCP2515 для обеспечения физической шины передачи данных. Объединение нескольких макетов SMART-устройств между собой имитирует работу сети

бортовых вычислительных устройств на борту наноспутника. Подключение SMART-устройства к виртуальной шине (ноутбуку) происходит с помощью модуля преобразования USB–CAN.

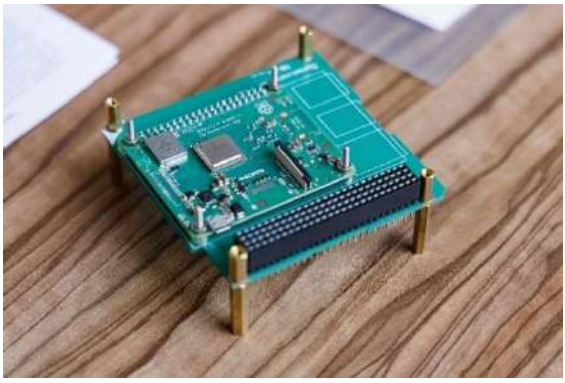


Рис. 2. Летная модель бортового вычислительного модуля.
Fig. 2. Flight model of the on-board computing module.

Подробное описание технических требований для работы с программным комплексом UEMKA, а также первичная настройка микрокомпьютера и подключение макетов представлено в разделе документации и Wiki репозитория проекта UEMKA [21].

3. Принципы построения архитектуры программного комплекса

Архитектура программного комплекса UEMKA, обеспечивающая централизованное управление целевыми конечными устройствами наноспутников в рамках единой информационной системы «Земля–Борт», создана с учетом современных требований к гибким распределенным информационным системам:

- Архитектура системы является открытой, поскольку проект UEMKA реализован как программное обеспечение с открытым исходным кодом.
- Архитектура системы ориентирована на сервисы (SOA – service-oriented architecture) и состоит из клиентских и серверных приложений.
- Архитектура системы UEMKA обеспечивает разделение функций между ее элементами, что выдвигает требование к созданию многоуровневой архитектуры.
- Архитектура системы UEMKA обеспечивает эффективное управление связями между ее элементами на всех этапах жизненного цикла бортового программного обеспечения, что выдвигает требование к созданию программно-определяемой сегментированной архитектуры, основанной на технологиях виртуализации.

Контейнерная виртуализация позволяет комбинировать подпрограммы, зависящие от разных версий одних и тех же библиотек и даже от разных версий языков программирования. Современная контейнерная виртуализация предполагает системы управления контейнерами, например, систему Docker Compose, где имеются инструменты для отслеживания состояния и автоматического повторного запуска микросервисов, встроенные возможности по масштабированию. Использование подобных систем в проекте UEMKA позволяет реализовать подход «инфраструктура как код», что уменьшает время на повторную настройку инфраструктуры, например, серверов баз данных, обмена сообщениями, логирования и визуализации.

Информационная система, построенная при помощи программного комплекса UEMKA, является сегментированной, что подразумевает наличие в общей информационной системе отдельных подсистем – сегментов. Внутри одного сегмента каналы передачи данных имеют

относительно высокую пропускную способность, в то время как пропускная способность каналов связи между сегментами существенно ниже.

Сегментированная архитектура подразумевает использование технологий виртуальных сетей и обмена сообщениями между микросервисами на основе подписки. В настоящее время наибольшее распространение получили подходы, основанные на контейнерной виртуализации, в частности по технологии Docker [22]. При таком подходе можно зафиксировать все зависимости в контейнере, что гарантирует воспроизводимость окружения и результатов работы программы. Воспроизводимость результатов особенно важна для космической техники.

В настоящее время подразумевается наличие как минимум двух сегментов: наземного (развернут на наземном комплексе управления) и космического (развернут на борту КА). Каждый сегмент построен как система клиентских (уровень сервисов) и серверных (уровень управления) приложений-микросервисов, работающих под управлением операционной системы с поддержкой контейнеризации.

Архитектура программного комплекса UEMKA является открытой, что проявляется в реализации следующих принципов:

- открытость исходного кода, который размещен в сети Интернет на базе открытого репозитория и системы документирования кода; при работе над кодом использованы только программные модули с открытым исходным кодом или лицензиями, допускающими свободное использование кода на гетерогенной сети из компьютеров и одноплатных микрокомпьютеров;
- открытость состава программных модулей, которая предполагает, что реализованная в настоящее время номенклатура модулей является первоначальной и будет расширяться по мере развития проекта и формирования сообщества пользователей;
- открытость топологии связей между модулями, которая предполагает возможность гибкой настройки каналов обмена сообщениями между модулями путем формирования конфигурационных файлов;
- открытость функциональных возможностей комплекса, которая предполагает возможности масштабирования и эволюции по назначению бортовой информационной системы, построенной на основе данного комплекса; в частности, информационная система позволяет проводить по единой технологии разработку, тестирование, развертывание, эксплуатацию и модернизацию информационной системы КА в целом и ее отдельных элементов.

Удовлетворяющая указанным требованиям архитектура удобна для разработки бортового программного обеспечения быстро меняющимися коллективами, что особенно актуально для университетов, где над разработкой программ работают студенты и аспиранты.

4. Структура программного комплекса UEMKA

Программный комплекс UEMKA использует сервис–ориентированную архитектуру SOA, реализованную на базе микросервисов – малых автономных (как правило, развертываемых в контейнерах) программ, взаимодействующих посредством обмена сообщениями или при помощи выполнения сценариев командной строки операционной системы. Определяющим для проекта UEMKA является то, что использование микросервисов дает существенные преимущества в автоматизированном развертывании и управлении информационной системой [23]. Состав программного комплекса UEMKA показан на рис. 3.

Рис. 3. Состав программного комплекса UEMKA.
Fig. 3. Composition of the UEMKA software package.

Для реализации программного комплекса используется среда разработки Microsoft Visual Studio Code, операционные системы Ubuntu версий 18.04, 20.04, 22.04 и Raspbian GNU/Linux версии 9, языки программирования C++ (компилятор g++ 7.5.0) и Python версий 3.6 и 3.10, система контроля версий Git, набор утилит can–utils [24] для работы с шиной CAN, система контейнерной виртуализации приложений docker и docker compose, базы данных PostgreSQL и SQLite, а также система визуализации данных Grafana.

Архитектура программного комплекса UEMKA является многоуровневой и включает:

- Уровень сервисов – верхний уровень, включающий программы–клиенты, реализующие логику решения целевых задач. Разработана библиотека специализированных программ–клиентов для управления бортовыми конечными устройствами.
- Уровень управления – уровень программ, работающих в режиме клиент–сервер и предназначенных для управления работой программ–приложений. Разработана библиотека программ–серверов для подготовки, развертывания приложений и

управления ими.

- Уровень данных – уровень электронных документов, используемых для передачи сообщений в процессе обмена информацией между программами. Средства уровня данных позволяют настраивать структуру реквизитной и содержательной частей электронных документов.
- Уровень шины – нижний уровень, включающий протокол передачи данных, средства программирования сокетов сети, средства организации каналов передачи данных.

Уровень данных и уровень шины образуют средства транспортного уровня и предназначены для организации удаленного доступа к бортовым конечным устройствам с единой централизованной управляющей консоли.

5. Особенности программного комплекса UEMKA

5.1 Имитационная цифровая модель космического аппарата и внешней среды

В программном комплексе UEMKA используется программно-аппаратное моделирование (Hardware in the loop), при котором системы наноспутника и влияние внешней среды моделируются численно на компьютере, а SMART-устройства являются реальными физическими объектами. Это позволяет отлаживать программное обеспечение SMART-блоков независимо от наноспутника.

Численное моделирование представлено имитационной цифровой моделью космического аппарата и внешней среды. Связанная математическая модель имитирует работу с реальным спутником, значительно упрощая процесс отладки и тестирования программного комплекса. На рис. 4 показана блок-схема математической модели космического аппарата. Математическая модель учитывает взаимовлияние системы ориентации и стабилизации (СОС) с моделями положения Солнца и магнитного поля Земли, тепловой двигательной установки (ДУ) и системы энергопитания (СЭП).

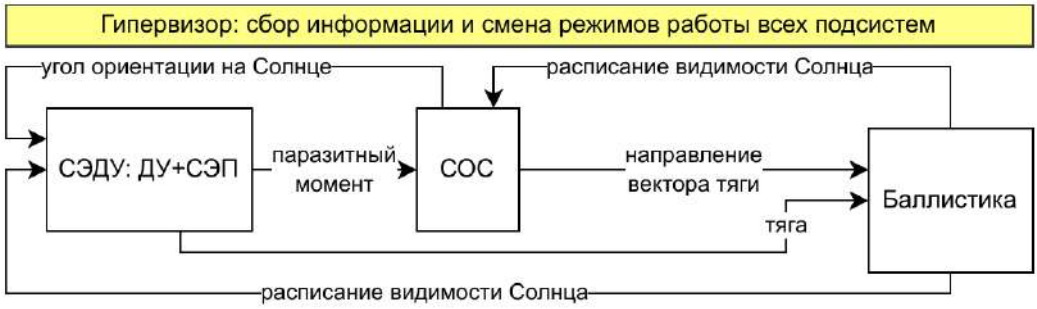


Рис. 4. Блок-схема математической модели космического аппарата.
Fig. 4. Spacecraft mathematical model diagram.

Общий вектор состояния спутника состоит из 20 компонент, 19 из которых независимые, так как скалярная часть кватерниона используется для проверки ошибки численного интегрирования из условия нормированности кватерниона:

$$y = \left(E, m_{fuel}, r_x, r_y, r_z, v_x, v_y, v_z, \omega_{b_x}^{bi}, \omega_{b_y}^{bi}, \omega_{b_z}^{bi}, q_0^{bi}, q_1^{bi}, q_2^{bi}, q_3^{bi}, \omega_{wh0}, \omega_{wh1}, \omega_{wh2}, \omega_{wh3}, Q_e\right)^T; \quad (1)$$

где E – общая тепловая энергия рабочего тела и конструкции ДУ, m_{fuel} – масса рабочего тела, $r_x \ldots r_z$, $v_x \ldots v_z$ – положение и скорость КА, $\omega_{b_x}^{bi} \ldots \omega_{b_z}^{bi}$ – компоненты угловой скорости, $q_0^{bi} \ldots q_3^{bi}$

– компоненты кватерниона ориентации, $\omega_{wh0}...\omega_{wh3}$ – угловые скорости маховиков, Q_e – заряд аккумуляторной батареи (АБ).

Накопленная тепловая энергия и суммарная масса двух фаз рабочего тела полностью определяют все термодинамические параметры, а ориентация спутника и заряд АБ определяют состояние СЭП. Подробное описание математической модели можно найти в диссертации [25] и в папке satellite_model репозитория [26].

Программный код по каждой системе наноспутника, упомянутой выше, представлен в репозитории в виде библиотеки функций [27]. В репозитории представлен также учёт несферичности поля тяготения Земли, модель магнитного поля IGRF и набор необходимых функций для демпфирования угловой скорости КА с помощью магнитных катушек по алгоритму B-Dot.

Архитектура программного комплекса численного моделирования внешней среды и космического аппарата разделена на две составляющие: модель бортовой электронно-вычислительной машины (БЭВМ) и модель остальных подсистем КА вместе с моделью внешней среды. Цикл моделирования разомкнутый: при достижении очередной точки останова выполняется запрос следующей задачи по шине CAN от БЭВМ.

Запустить моделирование можно как с использованием связки ПК и Raspberry Pi или все на ПК в Docker Compose. Второй вариант необходим для ускорения отладки и возможности проверки архитектуры работы системы с использованием двух шин данных с двусторонней трансляцией всех сообщений между шинами. Оба варианта показаны на рис. 5.

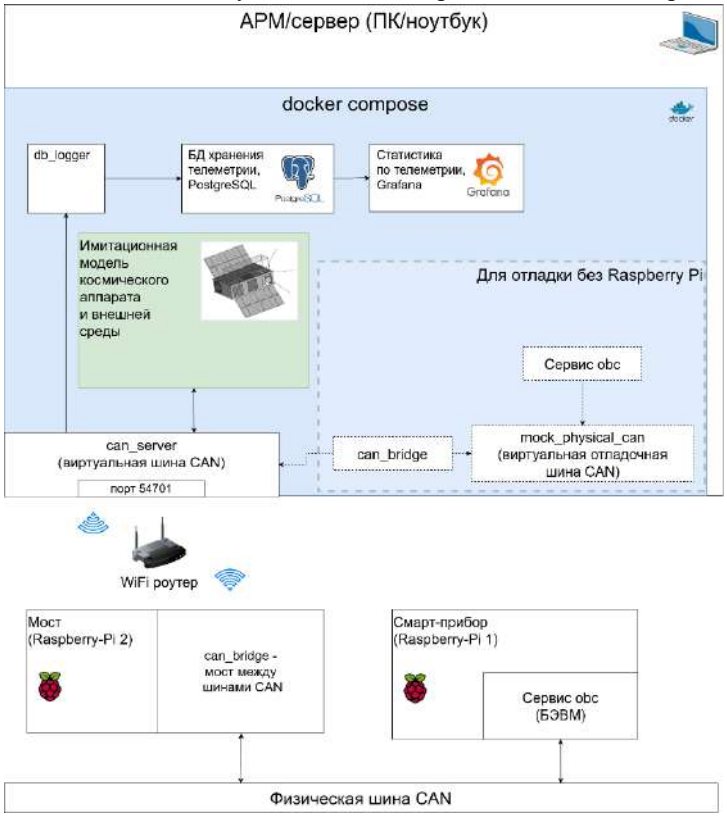


Рис. 5. Варианты запуска сервисов obc, can_bridge: основной на Raspberry Pi и отладочный в docker compose.
Fig. 5. Options for launching obc, can_bridge services: main on Raspberry Pi and debugging in docker compose.

5.2 Применение нейронной сети для обработки космических снимков

SMART-устройства имеют обширный спектр целевых задач, в которые входит выполнение ресурсоемких вычислений мощностями микрокомпьютера; взаимодействие с полезной нагрузкой, подключенной к SMART-устройству; обработка результатов работы конкретных узлов спутника и т.п. В проекте UEMKA для реализации одной из целевых задач SMART-устройства используется бортовая камера для получения космических снимков и их анализа с помощью нейронной сети (НС).

При передаче изображений с бортовой камеры на Землю возникают проблемы с контекстом на кадрах. Изображения могут содержать различные виды шума, такие как облака или битые кадры. Для предотвращения передачи избыточной информации на Землю предлагается использовать НС, которая будет анализировать полученные снимки перед отправкой и определять содержимое изображений. Если обнаружено отсутствие шума, НС будет отправлять соответствующие изображения пользователю. Обработка изображений с использованием технологий искусственного интеллекта (ИИ) позволяет минимизировать передаваемые данные, обрабатывая их непосредственно на борту наноспутника.

Алгоритм работы НС показан на рис. 6.

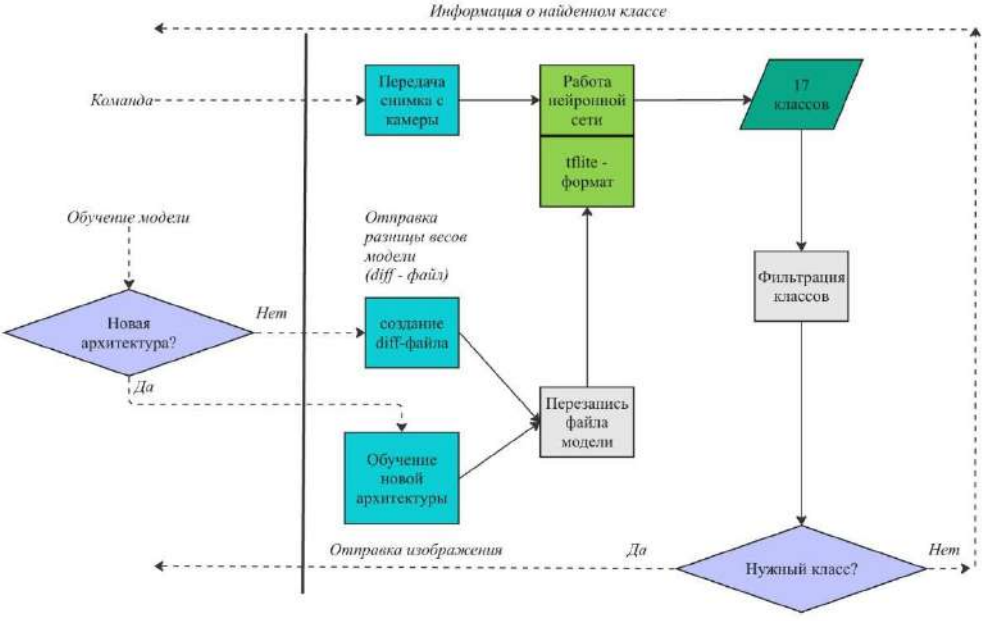


Рис. 6. Алгоритм работы НС.
Fig. 6. ANN operation algorithm.

При получении команды о создании снимка SMART-устройство передает сигнал бортовой камере. Готовое изображение на борту наноспутника обрабатывается НС на обнаружение целевых объектов, будь то вспышки на Солнце, конкретные звезды, лесные пожары и т.д. Затем обработанные снимки, соответствующие миссии КА, передаются наземному комплексу управления. Это значительно уменьшает количество «сырых» передаваемых данных и разгружает канал связи для иных задач. Программы для работы с НС представлены в репозитории проекта UEMKA [28].

Отличительной особенностью программного комплекса UEMKA является то, что в нем учтена возможность изменения летной программы КА, когда наноспутник уже находится на орбите. Для этого реализована удаленная настройка конфигураций SMART-устройств,

позволяющая изменить режим работы, например, ограничить целевую функцию НС, исключив из анализа конкретные объекты (облака, дороги и т.п.). Для этого на борт пересылается файл разности моделей (diff-файл).

Для космического радиоканала наноспутника размер передаваемого файла имеет ключевое значение. Размер модели около 15Мб, в то время как размер diff-файла измеряется в Килобайтах. Таким образом, согласно рис. 6, если для НС требуется новая архитектура, то на борт отправляется модель целиком, если же используется прежняя архитектура, которую дообучили на новых данных, то отправляется diff-файл. Выигрыш в размере файла очень важен при ограниченной скорости радиоканала и малой длительности сеанса связи с низкоорбитальным КА.

5.3 Графический интерфейс пользователя и система журналирования

Графический интерфейс пользователя (англ. GUI) предназначен для полуавтоматического и ручного управления удаленными полезными нагрузками наноспутников по шине передачи данных CAN. Эффект от использования графического интерфейса пользователя состоит в повышении удобства работы и уменьшении ошибок оператора наземного сегмента информационной системы полезных нагрузок SMART-спутников.

В проекте UEMKA используется веб-интерфейс с использованием фреймворков языка Python, который вместо разработки монолитного приложения позволяет использовать ресурсы браузера и дает возможность в будущем организовать удаленное управление полезными нагрузками через Интернет. Разработка интернет-технологий управления спутниковыми группировками через Интернет, с использованием мобильных устройств (телефонов, планшетов и пр.) в качестве терминалов, является мировым трендом [29].

Графический интерфейс решает следующие основные задачи:

- формирование структуры транспортного слоя;
- генерация электронных документов на уровне данных;
- управление клиентами-серверами UEMKA.

В состав интерфейса входят следующие компоненты.

- приложение PULT [30] с веб-интерфейсом, интегрированное с системой журналирования фреймов;
- приложение GUI [31] для формирования системы подписок на сообщения.

Приложение PULT, реализующее графический интерфейс, написано на языке Python с использованием фреймворка FastAPI, библиотеки SQLAlchemy для работы с реляционной базой данных SQLite. Структура веб-приложения состоит из системы веб-страниц как показано на рис. 7.

Приложение GUI предназначено для автоматизации процесса генерации и контроля файлов подписки.

Основными задачами приложения GUI являются:

- визуализация и управление составом системы;
- визуализация и управление структурой связей системы;
- генерация файлов подписки для всех элементов системы.

Для разработки GUI использован стек веб-технологий: гипертекстовая разметка страниц (html) и каскадных таблиц стилей (css), язык JavaScript, специализированный фреймворк React flow для создания графических интерфейсов на основе построения узлов, JavaScript-библиотека React.js для разработки графических интерфейсов и программная платформа Node.js для работы с языком JavaScript, предназначенная для создания серверных приложений.

В проекте UEMKA также используется система журналирования информационного потока «Земля-борт», предназначенная для сохранения информационных пакетов, передаваемых по каналу «Земля-борт», в реляционную базу данных. Система журналирования реализована в виде базы данных фреймов, встроенной в приложение PULT, которая построена на базе фреймворка SQLAlchemy. Данный инструментарий предоставляет полный набор известных шаблонов сохранения данных на основе ORM-модели, разработанных для эффективного и высокопроизводительного доступа к базе данных, адаптированных к языку Python.

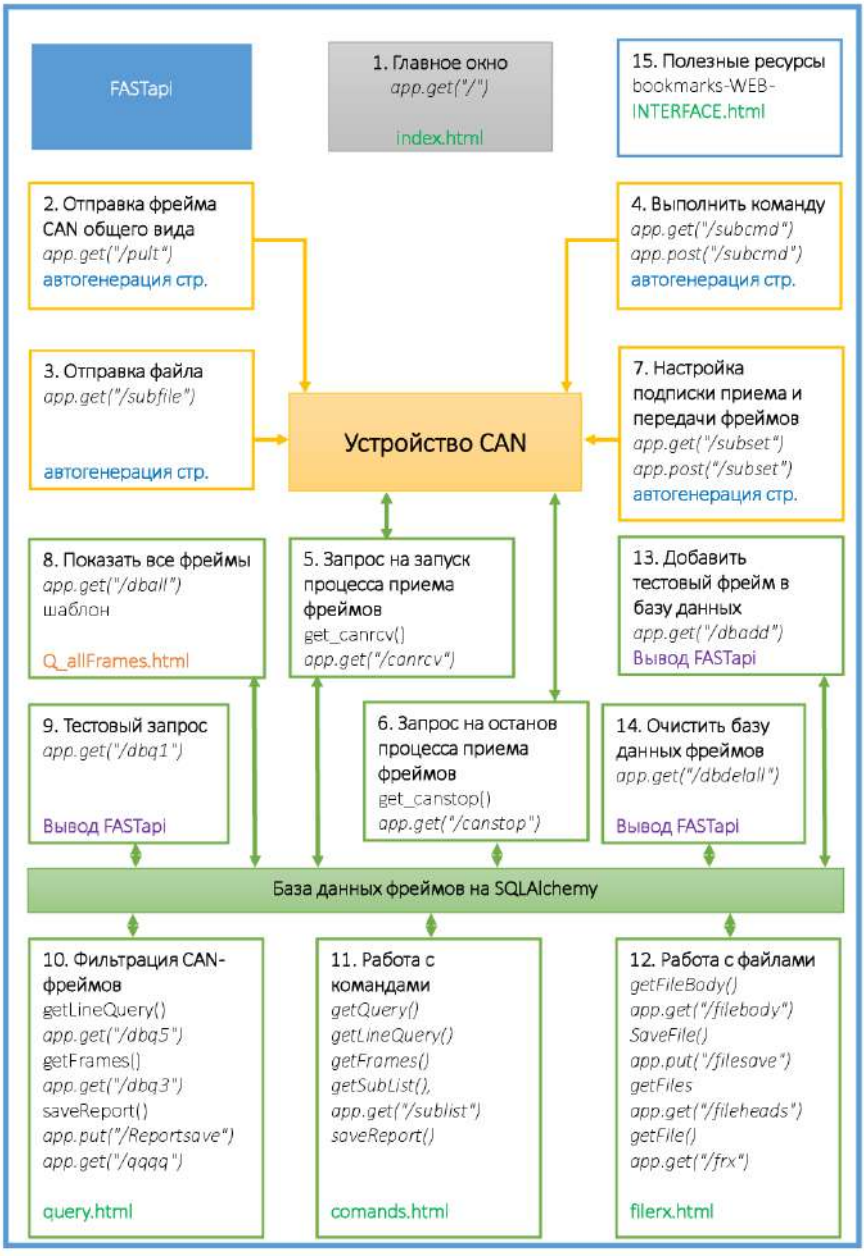


Рис. 7. Структурная схема графического интерфейса пользователя.
Fig. 7. Block diagram of the graphical user interface.

6. Пример использования программного комплекса UEMKA

Рассмотрим информационную систему, созданную при помощи программного комплекса UEMKA, которая состоит из отдельных сегментов. Сегменты связаны каналами передачи данных по шине CAN. Каждый сегмент включает один или несколько сервисов, которые осуществляют обмен сообщениями и электронными документами по шине CAN. Каждый сервис включает как минимум программу-клиента для удаленного доступа к конечному устройству, а также программу–клиента, реализующего логику решения целевых задач (параллельные вычисления, работа с бортовой камерой, обработка изображений НС, изменение конфигураций и т.д.).

Для работы последовательно выполняется подготовка к развертыванию информационной системы (компиляция программ из исходных кодов, компиляция образов docker–контейнеров из Dockerfile), а затем и само развертывание (запуск транспортного уровня, запуск сервисов). Далее происходит управление сервисами путем передачи сообщений и файлов по CAN–шине и управление информационной системой (переконфигурирование путем запуска и останова контейнеров, изменения конфигурационных файлов и т.п.).

В рамках развертывания информационной системы «Земля–Борт» сегменты UEMKA могут быть развернуты единообразно в различных вариантах аппаратного обеспечения:

- на одном компьютере (ноутбуке);
- на гетерогенной сети из компьютеров и одноплатных микрокомпьютеров типа Raspberry Pi;
- в штатном режиме на аппаратном обеспечении центра управления полетом и бортовых вычислительных устройствах КА.

На рис. 8 показана простейшая информационная система, сегменты которой развернуты на одном компьютере. Цель данной системы – отладка средств комплекса UEMKA в процессе его разработки. В данном примере назначением системы является проведение тестовых параллельных вычислений в сегменте «Борт» под управлением сегмента «Земля». Как видно из схемы, сегмент «Борт» является виртуальным и запущен в контейнере с именем «SC_HPC». Сегмент «Земля» развернут вне контейнера в операционной системе. Транспортный уровень реализован при помощи виртуальных CAN–устройств. Управление удаленным доступом к конечному устройству – сервису для проведения параллельных вычислений программой–клиентом hpctst, реализовано при помощи микросервисов – программ–клиентов clt, работающих в различных режимах.

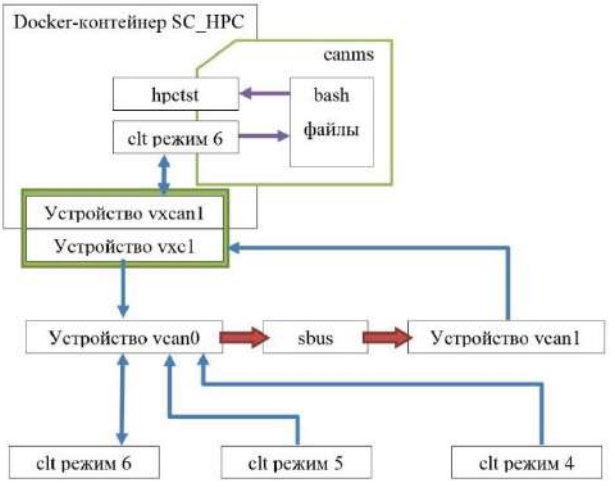


Рис. 8. Схема информационной системы.
Fig. 8. Information system diagram.

С помощью программного комплекса UEMKA также реализуются другие сценарии работы информационной системы:

- распределенная информационная система «Земля–Борт»;
- работа с сервисом обработки изображений с помощью НС;
- работа с сервисом графического интерфейса пользователя;
- режим тестирования средств удаленного устранения ошибок и отказов SMART-устройств;
- режим тестирования программного комплекса на макетах SMART-устройств с применением камеры.

Примеры вариантов использования кода открытой библиотеки расположены в репозитории проекта [32].

7. Заключение

Сформирована открытая, многоуровневая, сегментированная, сервис–ориентированная архитектура программного комплекса, основанная на использовании технологий микросервисов, контейнерной виртуализации и обмена сообщениями по подписке. Космический сегмент включает программы–клиенты для управления бортовыми конечными устройствами, которые организованы в виде микросервисов на платформе контейнеризации Docker.

Программный комплекс UEMKA реализует следующие функции:

- удаленный доступ к бортовым конечным устройствам (передача команд и данных между устройствами и наземными пунктами управления);
- управление устройствами с помощью единой централизованной консоли;
- автоматизированное управление бортовыми конечными устройствами при решении целевых задач (обработка изображений, параллельные вычисления и пр.);
- удаленная настройка конфигураций на пользовательских устройствах;
- подготовка, развертывание приложений и управление ими;
- выполнение пользовательских скриптов для автоматизации выполнения задач;
- удаленное устранение ошибок и отказов SMART-устройств;
- выборочное ограничение функциональных возможностей устройства;
- графический интерфейс пользователя наземного сегмента.

Представленный проект имеет открытый исходный код, что обеспечивает ряд преимуществ: программное обеспечение находится в общественном доступе, имеет свободное распространение и возможность модификации, все исходные коды доступны в виде публичного репозитория. Открытое программное ядро позволяет гибко и оперативно встраивать новые полезные нагрузки на борт спутника, экономя время разработчиков, а также дает возможность использовать современные программные методы и технологии, включая нейросетевые.

Результаты проекта программного комплекса UEMKA используются в научно-исследовательских работах, курсовых проектах и выпускных квалификационных работах студентов кафедры «Аэрокосмические системы» МГТУ им. Н.Э. Баумана. Они также внедрены в учебный процесс по трем дисциплинам:

- применение открытого программного обеспечения для расчета конструкций РКТ;
- системный анализ изделий ракетно-космической техники;
- системное моделирование изделий ракетно-космической техники.

Прикладное использование проекта сосредоточено во взаимодействии с разработчиками космического электронного оборудования и обучающимися вузов по направлениям ракетно-космической отрасли.

Результаты работы представлены в сети Интернет в публичном репозитории GitLab [33] и, в рамках импортозамещения, зеркало в российском сервере хранения исходного кода GitFlic [34]. Проект активно развивается с привлечением контрибьюторов и специалистов из космической отрасли.

Список литературы / References

[1]. Платформа LM50 для технологии SmartSat. [Электронный ресурс]: https://www.lockheedmartin.com/en–us/products/satellite.html (дата обращения 22.02.2024).
[2]. More Capable LM 400 Satellite Bus Designed to Meet Urgent Mission Needs [Электронный ресурс]: https://www.lockheedmartin.com/en–us/products/satellite.html (дата обращения 24.12.2023).
[3]. Lockheed Martin LINUSS Small Satellites Ready for 2021 Launch [Электронный ресурс]: https://news.lockheedmartin.com/linuss–small–sats–mission (дата обращения 10.04.2024).
[4]. Борисов сообщил о планах создать в 2026 году завод по серийному производству спутников [Электронный ресурс]: https://tass.ru/kosmos/17889029 (дата обращения 22.02.2024).
[5]. «Don't try this at home» pilot for a Cognitive Cloud Computing in Space infrastructure IAC–22–B1.4 //73rd International Astronautical Congress (IAC 2022). – 2022.
[6]. Edge computing in space: a machine learning approach for anomaly detection IAC–22–B5.1.6 //73rd International Astronautical Congress (IAC 2022). – 2022.
[7]. Haiyang Chua, Xiaoyu Heb, Hongjiang Songc, Shaohua Baid. The Design of the Spacecraft Test System 4000 Based on Microservices Running in Cloud Environment IAC–22–B6.1.1.x68506 //73rd International Astronautical Congress (IAC 2022). – 2022.
[8]. Проект F' Flight Software & Embedded Systems Framework от НАСА и JPL. [Электронный ресурс]: https://nasa.github.io/fprime/ (дата обращения 10.04.2024).
[9]. QNX – система реального времени. [Электронный ресурс]: https://blackberry.qnx.com/en (дата обращения 10.04.2024).
[10]. COSMOS API. [Электронный ресурс]: https://ballaerospace.github.io/cosmos-website/docs/v4/json-api (дата обращения 10.04.2024).
[11]. XML Telemetric and Command Exchange. [Электронный ресурс]: https://public.ccsds.org/Pubs/660x2g2.pdf (дата обращения 10.04.2024).
[12]. David J. Barnhart, David M. Shoemaker, Ellis T. King, Thomas "TJ" Logue, Michael J Lavis. LM LINUSS™ - Lockheed Martin In-space Upgrade Servicing System. 37th Annual Small Satellite Conference. [Электронный ресурс]: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=5648&context=smallsat (дата обращения 22.03.2024).
[13]. Core Flight System (cFS) [Электронный ресурс]: https://cfs.gsfc.nasa.gov/ (дата обращения 12.01.2024).
[14]. ПО для автопилотов PX4 [Электронный ресурс]: https://github.com/PX4/PX4–Autopilot (дата обращения 15.02.2024).
[15]. ПО для автопилотов franca [Электронный ресурс]: https://github.com/franca/franca/ (дата обращения 17.02.2024).
[16]. Core Flight System (CFS) Command and Data Dictionary (CCDD) utility [Электронный ресурс]: https://github.com/nasa/CCDD (дата обращения 12.02.2024).
[17]. Danda B. Rawat, Joel J.P.C. Rodrigues, Ivan Stojmenovic Cyber–Physical Systems From Theory to Practice. CRC Press 2016 579 p.
[18]. Kiruthika Devaraj, Matt Ligon, Eric Blossom, Joseph Breu, Bryan Klofas, Kyle Colton, Ryan Kingsbury. Planet High Speed Radio: Crossing Gbps from a 3U Cubesat. 33rd Annual AIAA/USU Conference on Small Satellites. [Электронный ресурс]: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=4405&context=smallsat (дата обращения 22.03.2024).
[19]. Satellite Eutelsat 10B [Электронный ресурс]: https://www.eutelsat.com/files/PDF/brochures/EUTELSAT_SATELLITE_E10B.pdf (дата обращения 22.03.2024).

[20]. Жданова К.А., Щеглов Г.А. Разработка вычислительного модуля для малого космического аппарата класса CubeSat // XLV Академические чтения по космонавтике (Королёвские чтения — 2022): сб. тез.: в 4 т. М.: Изд-во МГТУ им. Н.Э. Баумана, 2022. Т. 3. С. 241–243.
[21]. Ссылка на README.md файл репозитория проекта UEMKA [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/blob/devel/README.md (дата обращения 29.03.2024).
[22]. Парминдер Сингх Кочер Микросервисы и контейнеры Docker ДМК Пресс 2019 240 с.
[23]. Крис Ричардсон. Микросервисы. Паттерны разработки и рефакторинга Питер, 2022 544 с. Подробнее: [Электронный ресурс]: https://www.labirint.ru/books/707677/ (дата обращения 12.03.2024).
[24]. SocketCAN userspace utilities and tools [Электронный ресурс]: https://github.com/linux–can/can–utils (дата обращения 14.01.2024).
[25]. Дисс. на соискание уч. ст. к.т.н. «Методика проектирования наноспутника с солнечной энергодвигательной установкой» [Электронный ресурс]: https://mai.ru/upload/iblock/183/vpbzeo5ll25g7p8a6bq03ml3hm0sj3n7/Dissertatsiya_ZHumaev.pdf (дата обращения 19.03.2024).
[26]. Сервис численного моделирования КА [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/satellite_model?ref_type=heads (дата обращения 15.10.2024).
[27]. Библиотеки для работы численной модели [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/libs (дата обращения 29.03.2024).
[28]. Сервис с нейросетью распознавания изображений. Устройство neural service [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/neural_service (дата обращения 10.10.2024).
[29]. M. Bernou, A. Ampatzoglou, S. Vellas, K. Panopoulou, G. Lentaris, D. Soudris Spacecraft as a service, an open-source approach // 29th IAC Symposium On Small Satellite Missions (B4) IAC-22,B4,IP,4,x69307 URL: https://iafastro.directory/iac/paper/id/69307/abstract-pdf/IAC-22,B4,IP,4,x69307.brief.pdf?2022-04-05.11:15:32 (дата обращения 15.03.2024).
[30]. Каталог с приложением интерфейса PULT [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/pult (дата обращения 10.10.2024).
[31]. Графический интерфейс управления CAN-шиной [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/GUI_rules_CAN (дата обращения 10.10.2024).
[32]. Примеры вариантов использования кода проекта UEMKA [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems/-/tree/devel/docs/usage_examples (дата обращения 29.03.2024).
[33]. Ссылка в сети Интернет на публикацию открытой библиотеки в публичном репозитории GitLab [Электронный ресурс]: https://gitlab.com/Zaynulla/aerospace_computing_systems (дата обращения 18.02.2024).
[34]. Ссылка в сети Интернет на публикацию открытой библиотеки в российском сервере хранения исходного кода GitFlic: [Электронный ресурс]: https://gitflic.ru/project/zhumaev/acs?branch=devel (дата обращения 19.02.2024).

Информация об авторах / Information about authors

Георгий Александрович ЩЕГЛОВ – доктор технических наук, профессор, профессор кафедры «Аэрокосмические системы» МГТУ им. Н.Э. Баумана с 2012 года. Сфера научных интересов: проектирование аэрокосмических систем, системы автоматизированного проектирования и расчета элементов конструкций аэрокосмических систем.

Georgy Alexandrovich SHCHEGLOV – Dr. Sci. (Tech.), Professor, Professor of the Department of Aerospace Systems at Bauman Moscow State Technical University since 2012. Research interests: design of aerospace systems, computer-aided design systems and calculation of structural elements of aerospace systems.

Кристина Александровна ЖДАНОВА – инженер, аспирант кафедры «Аэрокосмические системы» МГТУ им. Н.Э. Баумана. Сфера научных интересов: орбитальные высокопроизводительные вычисления, системное проектирование и инженерия, встраиваемые системы, CubeSat, проектирование аэрокосмических систем.

Kristina Alexandrovna ZHDANOVA is an engineer, postgraduate student of the Aerospace Systems Department of Bauman Moscow State Technical University. Research interests: embedded systems, CubeSat, design of aerospace systems.

Зайнулла Серикович ЖУМАЕВ – кандидат технических наук, преподаватель кафедры «Аэрокосмические системы» МГТУ им. Н. Э. Баумана. Сфера научных интересов: системное проектирование, математическое моделирование малых космических аппаратов, разработка алгоритмов управления, открытое программное обеспечение.

Zaynulla Serikovich ZHUMAEV – Cand. Sci. (Tech.), Lecturer of the Aerospace Systems Department, Bauman Moscow State Technical University. Research interests: system design, mathematical modeling of small spacecraft, development of control algorithms, open source software.

Никита Дмитриевич КАМЕНЕВ – аспирант кафедры «Аэрокосмические системы» МГТУ им. Н.Э. Баумана. Сфера научных интересов: нейронная сеть, искусственный интеллект, проектирование аэрокосмических систем.

Nikita Dmitrievich KAMENEV is a postgraduate student of the Aerospace Systems Department of Bauman Moscow State Technical University. Research interests: neural network, artificial intelligence, design of aerospace systems.

DOI: 10.15514/ISPRAS-2024-36(5)-13

CC BY

Estimation of the Vertical Diffusion Coefficient of Gas in Compacted Soils by Means of Mathematical Modeling

E.O. Tarasenko, ORCID: 0000-0002-6227-6999 <galail@mail.ru>

North Caucasian Federal University,
1, Pushkin St., Stavropol, 355017, Russia.

Abstract. The density properties of subsiding loess soils were studied within the framework of mathematical modeling of their compaction using the method of deep explosions. Soil compaction is carried out to eliminate subsidence properties. Loess soils are characterized by low density and high porosity. The density properties of loess depend on the parameters of the diffusion interaction of gas atoms formed as a result of the explosion and the soil being compacted. Solving inverse applied problems that arise when studying mathematical models of geological systems allows us to systematize knowledge about them. The work considers the inverse problem of estimating the diffusion coefficient. Mathematical modeling of the vertical diffusion coefficient in anisotropic and isotropic geological systems was carried out. The case of complete absorption of gas atoms by the surrounding soil has been studied. A numerical assessment of the coefficient of vertical diffusion in soil before and after compaction has been implemented, over time and with an accuracy sufficient for engineering calculations. Gas diffusion coefficients in soils of various densities were obtained. The constructed mathematical relationships for estimating the coefficient of vertical diffusion make it possible to predict the density properties of soils at the stage of designing the foundations of construction projects.

Keywords: mathematical modeling; numerical assessment; computational experiment; subsidence; soil compaction; diffusion coefficient.

For citation: Tarasenko E.O. Estimation of the vertical diffusion coefficient of gas in compacted soils by means of mathematical modeling. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024. pp. 181-190. DOI: 10.15514/ISPRAS-2024-36(5)-13.

Acknowledgements. The author is grateful and grateful to Doctor of Geological and Mineralogical Sciences, Professor Boris Fedorovich Galay, for his advice in conducting this scientific research.

Оценка коэффициента вертикальной диффузии газа в уплотняемых грунтах средствами математического моделирования

Е.О. Тарасенко, ORCID: 0000-0002-6227-6999 <galail@mail.ru>

Северо-Кавказский федеральный университет,
Россия, 355017, г. Ставрополь, ул. Пушкина, д. 1.

Аннотация. Исследованы плотностные свойства просадочных лёссовых грунтов в рамках математического моделирования их уплотнения методом глубинных взрывов. Уплотнение грунтов производится для исключения свойства просадочности. Лёссовым грунтам характерна низкая плотность и высокая пористость. Плотностные свойства лёссов зависят от параметров диффузионного взаимодействия атомов газа, образующегося в результате взрыва, и уплотняемого грунта. Решение обратных прикладных задач, возникающих при исследовании математических моделей геологических систем, позволяет систематизировать знания о них. В работе рассмотрена обратная задача об оценке коэффициента диффузии. Проведено математическое моделирование коэффициента вертикальной диффузии в анизотропных и изотропных геологических системах. Исследован случай полного поглощения атомов газа окружающим его грунтом. Реализована численная оценка коэффициента вертикальной диффузии в грунте до и после его уплотнения, с течением времени и точностью достаточной для инженерных расчётов. Получены коэффициенты диффузии газа в грунтах различной плотности. Построенные математические соотношения оценки коэффициента вертикальной диффузии позволяют прогнозировать плотностные свойства грунтов на этапе проектирования оснований и фундаментов строительных объектов.

Ключевые слова: математическое моделирование; численная оценка; вычислительный эксперимент; просадочность; уплотнение грунта; коэффициент диффузии.

Для цитирования: Тарасенко Е.О. Оценка коэффициента вертикальной диффузии газа в уплотняемых грунтах средствами математического моделирования. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 181–190 (на английском языке). DOI: 10.15514/ISPRAS–2024–36(5)–13.

Благодарности: Автор признателен и благодарен доктору геолого-минералогических наук, профессору Галаю Борису Фёдоровичу, за советы при проведении данного научного исследования.

1. Introduction

The design of buildings and civil engineering structures is accompanied by numerical calculations of the density and deformation characteristics of subsiding loess soil. Loess is quite mysterious; on a planetary scale, it is a rock of the Quaternary period [1], [2]. Loess is characterized by a yellow, yellow-brown or yellow-fawn color [3]–[6]. They have high porosity and low density [7].

Methods for eliminating subsidence are varied and depend on the conditions of implementation [8]. We are exploring the method of soil compaction by explosion, which in the process of practical implementation shows high economic efficiency at low production costs [9], [10]. Within the framework of mathematical modeling of the compaction of subsidence soils, studies of their individual strength and deformation properties have been carried out [11]–[13]. Interpretation of soil strength characteristics for numerical studies was carried out in [14], [15]. When compacted by explosion, soils are subject to dynamic loads [16], [17]. Within the framework of mathematical modeling of the compaction of subsidence soils, the dependence of the density properties of loess on the parameters of the diffusion interaction of gas and soil atoms is traced.

For a comprehensive description of the mathematical model of compaction of subsidence soils by explosion, additional scientific research and the solution of individual inverse problems are required. One of which determines the purpose of the article - modeling the coefficient of vertical diffusion in anisotropic and isotropic compacted geological systems with complete absorption of gas atoms

by the surrounding soil. To achieve the goal, this article describes the solution of the following problems: the results of mathematical modeling of the density of subsidence soils during their compaction by deep explosions; development of a solution to the inverse problem of estimating the coefficient of vertical diffusion in soils of various densities using numerical modeling methods; implementation of a software computational experiment with an accuracy sufficient for engineering calculations.

2. Mathematical model of compaction of subsidence soils

The mathematical model of compaction of subsidence soils by the explosion of a concentrated explosive charge is described by a parabolic equation of the form [11]

$$\frac{\partial q}{\partial t}+U\frac{\partial q}{\partial x}=\frac{\partial}{\partial x}K_x\frac{\partial q}{\partial x}+\frac{\partial}{\partial y}K_y\frac{\partial q}{\partial y}+\frac{\partial}{\partial z}K_z\frac{\partial q}{\partial z}+f\ ,\ t\in[t_0,T], \tag{1}$$

with initial condition

$$q(t_0,x,y,z)=Q\delta\left(x-x^0\right)\delta\left(y-y^0\right)\delta\left(z-z^0\right) \tag{2}$$

and boundary condition

$$q(t,x,y,z)\big|_{z=z^0}=0\ ,\ t>t_0, \tag{3}$$

where $Q=const>0$ – power of the explosive charge, $\delta(x)$ – Dirac delta function, $z^0=H$ – depth of placement of the concentrated explosive charge, $U=const$ – speed of horizontal transfer (along the Ox axis), q – average soil density, K_x, K_y, K_z – diffusion coefficients.

The initial boundary value problem (1)–(3) characterizes changes in the values of the average density of the compacted soil skeleton per unit time t at point *(x, y, z)*. Boundary condition (3) corresponds to the case of complete absorption of gas atoms released during an explosion by the surrounding soil. In this case, compaction of subsidence soil is realized [11].

Let us represent the function of the explosive gas source in the form

$$f(t,x,y,z)=Q\cdot R(t,x,y,z)=Q\delta(t-t_0)\delta\left(x-x^0\right)\delta\left(y-y^0\right)\delta(z-H).$$

Let us assume that U, K_x, K_y, K_z are continuous functions of the argument z:

$$U=U(z),\ K_x=K_x(z),\ K_y=K_y(z),\ K_z=K_z(z).$$

The solution to problem (1)-(3) for calculating the average values of the density of the compacted soil skeleton is determined by the relation [13].

$$\begin{aligned} q(t,x,y,z)&=\frac{Q}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z t^3}\times\\ &\times\exp\left\{-\left(\frac{\left(x-\overline{U}t\right)^2}{2\sigma_x^2 t}+\frac{y^2}{2\sigma_y^2 t}\right)\right\}\times\\ &\times\left[\exp\left\{-\frac{(z-H)^2}{2\sigma_z^2 t}\right\}-\exp\left\{-\frac{(z+H)^2}{2\sigma_z^2 t}\right\}\right], \end{aligned} \tag{4}$$

which is a function of the density distribution of the soil skeleton, where $\sigma_x^2(t)$, $\sigma_y^2(t)$, $\sigma_z^2(t)$ - dispersion coordinate changes of gas atoms in the compacted soil, respectively, along the Ox, Oy, Oz axes at time t; $\sigma_x^2(t)$, $\sigma_y^2(t)$, $\sigma_z^2(t)$ – functions continuously differentiable with respect to the time argument t, $t\geq 0$.

3. Vertical diffusion coefficient modeling

Estimating the diffusion coefficient of gas atoms released during the explosion of a concentrated explosive charge is an inverse problem within the framework of mathematical modeling of the compaction of subsidence soils.

Statement of the inverse problem. According to the known average values of the density of the soil skeleton *q(t, x, y, z)*, compacted by the method of deep explosion of a concentrated source of explosive, provided that the gas is completely absorbed by the surrounding soil (compaction of subsidence soil), as well as according to a given depth of placement of the explosive *H* and known values of charge power *Q* of the explosive, determine the values $\sigma_z^2(t)$ – vertical dispersion of the coordinates of gas atoms in the compacted soil along the *Oz* axis at time *t*. Estimate the vertical diffusion coefficient *Kz* over time and with a given accuracy.

Let us carry out numerical modeling and construct an approximate solution to the posed inverse problem based on the apparatus of numerical methods for solving transcendental equations [17], [18].

The solution of the problem for the case of complete absorption of gas atoms by the surrounding soil (implementation of compaction of the subsidence layer) will be carried out using a simple iteration method. Assuming that the interval [*a*, *b*] is separated (we will estimate *a* and *b* by the selection method or graphically), containing the required root σ_z equations

$$\begin{aligned} q(t,x,y,z) &= \frac{Q}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z t^3}\times \\ &\times\exp\left\{-\left(\frac{\left(x-\overline{U}t\right)^2}{2\sigma_x^2 t}+\frac{y^2}{2\sigma_y^2 t}\right)\right\}\times \\ &\times\left[\exp\left\{-\frac{(z-H)^2}{2\sigma_z^2 t}\right\}-\exp\left\{-\frac{(z+H)^2}{2\sigma_z^2 t}\right\}\right]=0. \end{aligned} \tag{5}$$

Taking into account that the geological environment under study has characteristic properties of anisotropy, we transform equation (5) to the form:

$$\begin{aligned} \sigma_z &= \frac{Q}{q(t,x,y,z)(2\pi)^{3/2}\sigma_x\sigma_y t^3}\times \\ &\times\exp\left\{-\left(\frac{\left(x-\overline{U}t\right)^2}{2\sigma_x^2 t}+\frac{y^2}{2\sigma_y^2 t}\right)\right\}\times \\ &\times\left[\exp\left\{-\frac{(z-H)^2}{2\sigma_z^2 t}\right\}-\exp\left\{-\frac{(z+H)^2}{2\sigma_z^2 t}\right\}\right]. \end{aligned}$$

The iterative process of successive approximations to the desired root will be numerically determined by the formula

$$\sigma_z^{(n+1)} = \frac{Q}{q(t,x,y,z)(2\pi)^{3/2}\sigma_x\sigma_y t^3}\times$$
$$\times\exp\left\{-\left(\frac{(x-\overline{U}t)^2}{2\sigma_x^2 t}+\frac{y^2}{2\sigma_y^2 t}\right)\right\}\times$$
$$\times\left[\exp\left\{-\frac{(z-H)^2}{2\left(\sigma_z^{(n)}\right)^2 t}\right\}-\exp\left\{-\frac{(z+H)^2}{2\left(\sigma_z^{(n)}\right)^2 t}\right\}\right]. \quad (6)$$

If the geological environment under study has characteristic isotropic properties, then $\sigma_z = \sigma_y = \sigma_x$. Therefore, equation (5) can be transformed to the form

$$\sigma_z = \sqrt[3]{\frac{Q}{q_2(t,x,y,z)(2\pi)^{3/2}\sigma_z^2 t^3}}\times$$
$$\times\sqrt[3]{\exp\left\{-\left(\frac{(x-\overline{U}t)^2}{2\sigma_z^2 t}+\frac{y^2}{2\sigma_z^2 t}\right)\right\}}\times$$
$$\times\sqrt[3]{\left[\exp\left\{-\frac{(z-H)^2}{2\sigma_z^2 t}\right\}-\exp\left\{-\frac{(z+H)^2}{2\sigma_z^2 t}\right\}\right]}.$$

Instead of (6), we will construct an iterative sequence of approximations to the desired root using the formula

$$\sigma_z^{(n+1)} = \sqrt[3]{\frac{Q}{q_2(t,x,y,z)(2\pi)^{3/2}\left(\sigma_z^{(n)}\right)^2 t^3}}\times$$
$$\times\sqrt[3]{\exp\left\{-\left(\frac{(x-\overline{U}t)^2}{2\left(\sigma_z^{(n)}\right)^2 t}+\frac{y^2}{2\left(\sigma_z^{(n)}\right)^2 t}\right)\right\}}\times$$
$$\times\sqrt[3]{\left[\exp\left\{-\frac{(z-H)^2}{2\left(\sigma_z^{(n)}\right)^2 t}\right\}-\exp\left\{-\frac{(z+H)^2}{2\left(\sigma_z^{(n)}\right)^2 t}\right\}\right]}. \quad (7)$$

The first approximation of $\sigma_z^{(1)}$ to the desired root (first iteration) is taken to be any value of σ_z, belonging to a segment of root division [*a*, *b*].

The criterion for completing the computational process of searching for a numerical solution to equation (5) is the fulfillment of the inequality

$$\left|\sigma_z^{(n)} - \sigma_z^{(n-1)}\right| < \varepsilon,$$

where ε is the accuracy of calculations

The dispersion of coordinate changes of gas atoms is related to the vertical diffusion coefficient by the relation

$$\sigma_z^2 = \frac{2}{h}\int_0^h K_z(z)\,dz\,, \tag{8}$$

where h is the depth of penetration of gas atoms into the surrounding soil.

4. Computing experiment

Numerical modeling of the diffusion coefficient of gas formed as a result of an explosion into the surrounding soil is carried out using the example of soil in the city of Budennovsk, Stavropol Territory. According to the engineering-geological structure, the territory of Budennovsk is considered the "capital" of subsidence soils of the world. The main element of the engineering-geological section of the city is loess and loess-like loams with a thickness of up to 40 m, in the Budennovsky district - up to 100 m. The next element of the section is composed of clays with layers of sand, occurring at depths from 10 to 70 m, and with a thickness of up to 30 m [20].

Before constructing buildings and structures on loess soils, preliminary compaction is required. One of the low-cost and effective methods is the compaction of a concentrated explosive charge by explosion. During a deep explosion of industrial explosives, gases (mainly carbon and nitrogen oxides) are released into the surrounding soil [21].

To estimate the gas diffusion coefficient in soil based on iterative processes (6), (7) and relation (8), a program was developed in the Python programming language [22]. It is designed to calculate the values of vertical diffusion coefficients over time and with an accuracy sufficient for engineering calculations.

Fig. 1 shows a graphical interpretation of the vertical diffusion coefficient when compacting loess with a density of 1.5 g/cm^3 by explosion. Explosive charge power 5 kg. Its depth is 6 m. The initial data of the computational experiment are taken from the design documentation of the object "Stavroplen", Budennovsk [9].

Analysis of the values presented in Fig. 1 showed that the vertical diffusion coefficient reaches its maximum value of 0.60608 cm^2/s at the moment of the explosion. Then it decreases over time. This is due to the type of gas source; it is concentrated and instantaneous.

Three months after draining the pit, 5 wells were drilled to sample monoliths. The density of the soil compacted by the explosion averaged 1.7 g/cm^3 [9]. Tabl. 1 shows the values of soil diffusion coefficients over time, calculated according to the iterative process (6) and relation (8).

A graphical interpretation of the result of the experiment to estimate the coefficient of vertical diffusion of gas in soils of various densities is shown in Fig. 2.

The value of the coefficient of vertical gas diffusion in soil with a density of 1.5 g/cm^3 is higher than with a soil density of 1.7 g/cm^3. The diffusion coefficient decreases by up to 13%.

Numerical modeling of the diffusion coefficient for various soils in the city of Budennovsk was carried out. Fig. 3 shows the implementation of modeling the dynamics of its change for loess, sand, sandy loam, loam and clay. The maximum value of the vertical diffusion coefficient of 0.60608 cm^2/s is recorded for loess. Moreover, its minimum value is 0.46621 cm^2/s for clay. Over time, after 1 s, the diffusion coefficient takes on a value of 0.03 cm^2/s with an accuracy of 0.01, sufficient for engineering calculations.

5. Conclusion

The result of the study is a numerical assessment of the coefficient of vertical diffusion of gas in soil within the framework of mathematical modeling of its compaction by explosion. A case of complete absorption of gas atoms by the surrounding soil during compaction is described. The isotropic and anisotropic properties of the compacted soil are taken into account. The compiled mathematical

expressions make it possible to carry out computational experiments with the required accuracy. Gas diffusion coefficients in soils of various densities were obtained.

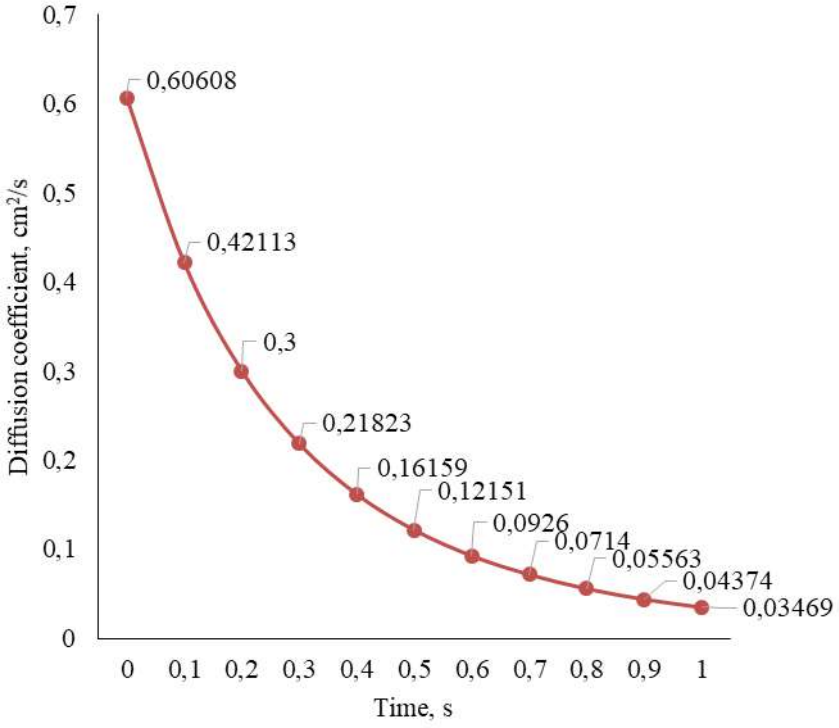


Fig.1. Values of the vertical diffusion coefficient of gas in soil over time.

Table 1. Vertical diffusion coefficient before and after soil compaction by explosion.

Time	Diffusion coefficient	
	before compaction	after compaction
0.0	0.6060	0.5347
0.1	0.4211	0.3715
0.2	0.3000	0.2647
0.3	0.2182	0.1925
0.4	0.1615	0.1425
0.5	0.1215	0.1072
0.6	0.0926	0.0817
0.7	0.0714	0.0630
0.8	0.0556	0.0490
0.9	0.0437	0.0386
1.0	0.0346	0.0306

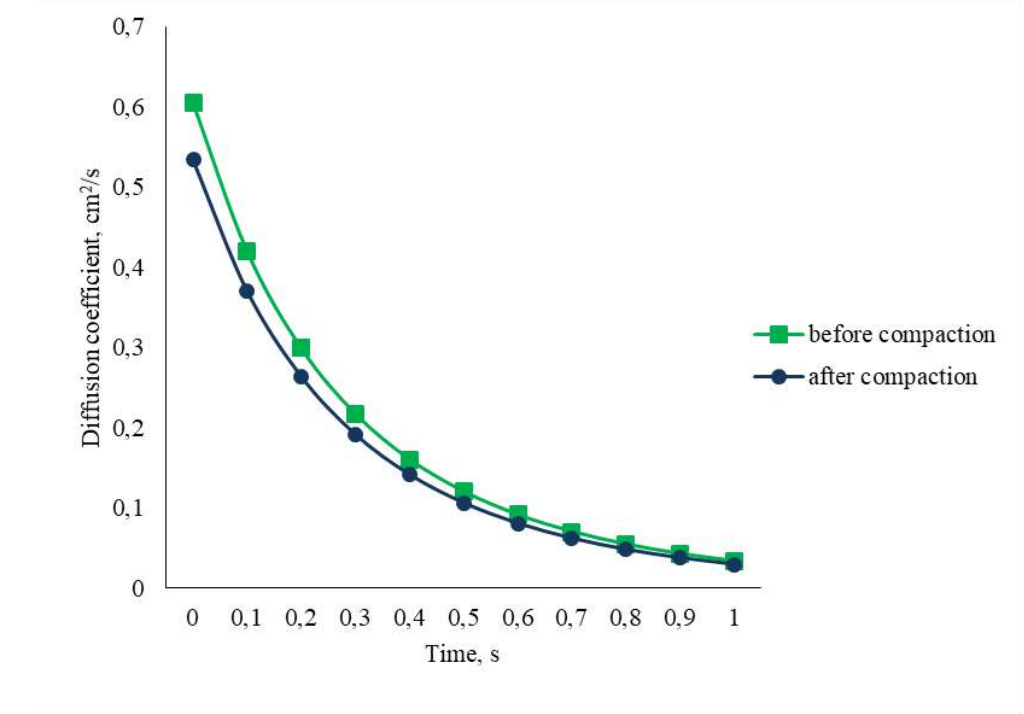


Fig.2. Values of coefficients of vertical diffusion of gas in soil over time before and after compaction by explosion.

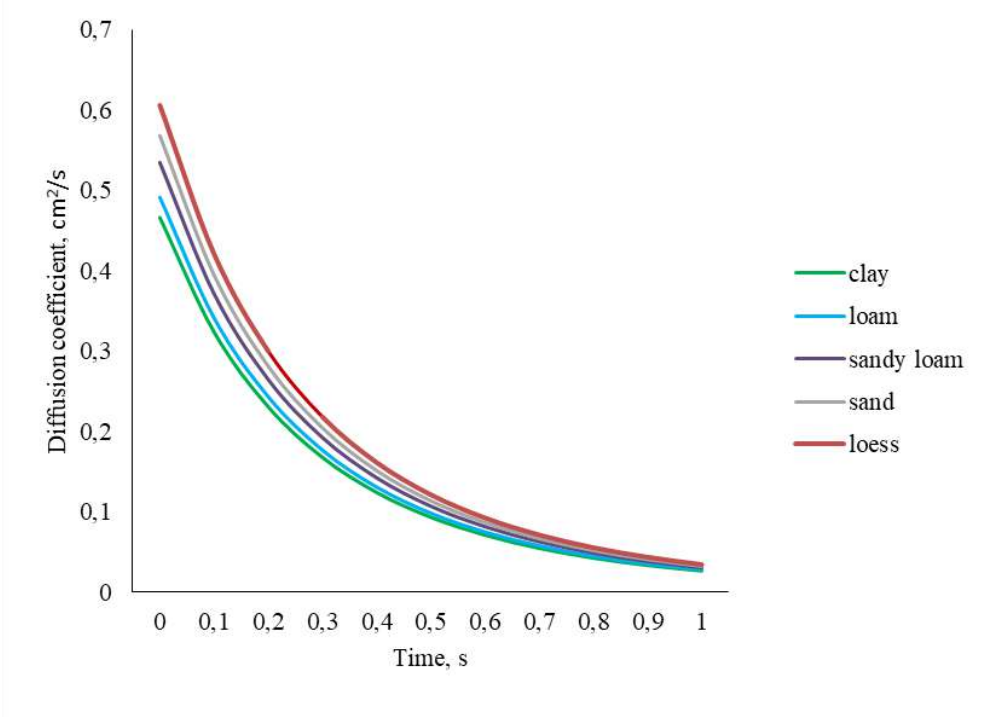


Fig.3. Values of coefficients of vertical diffusion of gas in soil over time before and after compaction by explosion.

Список литературы / References

[1]. Geological World Atlas of scale: 1:10 000 000 / G. Choubert, A. Faure-Muret (general coordinators). Paris: CGMW, UNESCO. 1979.

[2]. Международная тектоническая карта мира масштаба 1:15 000 000 / под ред. В. Е. Хаина. Ленинград: Мингео, 1984. / International tectonic map of the world scale 1:15 000 000 / ed. V. E. Khaina. Leningrad: Mingeo, 1984. (in Russian)

[3]. Трофимов В.Т., Балыкова С.Д., Андреева Т.В. и др. Опорные инженерно-геологические разрезы лессовых пород Северной Евразии. Москва: КДУ, 2008. 608 с. / Trofimov V.T., Balykova S.D., Andreeva T.V. and others, Reference engineering-geological sections of loess rocks of Northern Eurasia. Moscow: KDU, 2008. 608 p. (in Russian)

[4]. Ананьев В.П. Лессовый покров России. Москва: Юриспруденция, 2004. 112 с. / Ananyev V.P. Loess cover of Russia. Moscow: Jurisprudence, 2004. 112 p. (in Russian)

[5]. Балаев Л.Г., Царев П.В. Лессовые породы Центрального и Восточного Предкавказья. Москва: Наука, 1964. 248 с. / Balaev L.G., Tsarev P.V. Loess rocks of the Central and Eastern Ciscaucasia. Moscow: Nauka, 1964. 248 p. (in Russian)

[6]. Ed. Wang Y., Zhang Z. Shaanxi, Loess in China. Peaole's art. Publ House, 1980. 230 p.

[7]. Григорьева И.Ю. Микростроение лессовых пород. Москва: МАИК Наука, 2001. 147 с. / Grigorieva I.Yu. Microstructure of loess rocks. Moscow: MAIK Nauka, 2001. 147 p. (in Russian)

[8]. Пантюшина Е.В. Лёссовые грунты и инженерные методы устранения их просадочных свойств // Ползуновский вестник. 2011. № 1. С.127-130. / Pantyushina E. V. Loess soils and engineering methods for eliminating their subsidence properties. Polzunovsky vestnik, vol. 1, 2011, pp. 127-130. (in Russian)

[9]. Галай Б.Ф. Уплотнение просадочных грунтов глубинными взрывами. Ставрополь: Сервисшкола, 2015. 240 с. / Galay B.F. Compaction of subsidence soils by deep explosions. Stavropol: Service School, 2015. 240 p (in Russian)

[10]. Way lanynig, Gui Jiuxi, Lu Yanchou, A preliminary study of the physico-mechanic properties of loesses and paleosols of different age at Weibei Yuan, Shaanxi provine. Loess, environment and global change. SciencePress, Beijing, 1991, pp. 245-259.

[11]. Тарасенко Е.О., Гладков А.В. Численное решение обратных задач при математическом моделировании геологических систем // Известия Томского политехнического университета. Инжиниринг георесурсов. 2022. Т. 333. № 1. С.105-112. / Tarasenko E.O., Gladkov A.V. Numerical solution of inverse problems in mathematical modeling of geological systems. Bulletin of the Tomsk Polytechnic University. Geo Assets Engineering, vol. 333(1), 2022, pp. 105-112. (in Russian) DOI: 10.18799/24131830/2022/1/3208

[12]. Tarasenko E.O., Gladkov A.V., Gladkova N.A. Solution for Inverse Boundary Value Problems on the Power of a Concentrated Charge in a Mathematical Model of Subsidence Soils Compaction. Mathematics and its Applications in New Computer Systems. MANCS 2021, vol 424, Lecture Notes in Networks and Systems, Springer, 2022, pp. 537-545. DOI: 10.1007/978-3-030-97020-8_49

[13]. Тарасенко Е.О. Численная оценка плотности грунта методом конечно-разностных сеток при математическом моделировании уплотнения просадочных грунтов глубинными взрывами / Е.О. Тарасенко // Известия Томского политехнического университета. Инжиниринг георесурсов. – 2023, Т. 334. № 5. – С.103-108. / Tarasenko E.O. Numerical estimation of soil density by the method of finite difference grids in mathematical modeling of compaction of subsible soils by deep explosions. Bulletin of the Tomsk Polytechnic University. Geo Assets Engineering, vol. 334(5), 2023, pp. 103-108. (in Russian) DOI: 10.18799/24131830/2023/5/4022

[14]. Петраков А.А., Прокопов А.Ю., Петракова Н.А., Панасюк М.Д. Интерпретация прочностных характеристик грунта для численных исследований // Известия Тульского государственного университета. Науки о Земле, 2021. – №. 1. – С.225 – 236. / Petrakov A.A., Prokopov A.Yu., Petrakova N.A., Panasyuk M.D. Interpretation of soil strength characteristics for numerical studies. News of Tula State University. Geosciences, vol. 1, 2021, pp. 225-236. (in Russian)

[15]. Tarasenko E.O., Mathematical modeling of the strength properties of lesses by the method of correlation-regression analysis. International Journal for Computational Civil and Structural Engineering, vol. 20(1), 2024, pp. 171-181. DOI: 10.22337/2587-9618-2024-20-1-171-181

[16]. Tsukamoto Y., Ishihara K. Analysis on settlement of soil deposits following liquefaction during earthquakes. Soils and Foundation, vol. 50(3), 2010, pp. 399-441.

[17]. Ishihara K. New challenges in Geotechnique for ground hazards due to intensely strong earthquake shaking. Geotechnical, Geological and Earthquake Engineering, vol. 11, 2009, pp. 91-114.

[18]. Бахвалов Н.С. Численные методы. Москва: Наука, 1973. 614 с. / Bakhvalov N.S. Numerical methods. Moscow: Nauka, 1973. 614 p. (in Russian)

[19]. Вержбицкий В. М. Численные методы: (математический анализ и обыкновенные дифференциальные уравнения). Москва: Директ-Медиа, 2013. 400 с. / Verzhbitsky V.M. Numerical methods: (mathematical analysis and ordinary differential equations). Moscow: Direct-Media, 2013. 400 p. (in Russian) ISBN 978-5-4458-3876-0.

[20]. Галай О.Б. Буденновск: геология и город. Ставрополь: СервисШкола, 2022. 318 с. / Galay O.B. Budennovsk: geology and city. Stavropol: Service School, 2022. 318 p. (in Russian)

[21]. Козырев С.А., Власова Е.А. Газовая вредность взрывчатых веществ, применяемых в горнодобывающей промышленности. Горная промышленность, № 5, 2021. С.106-111. / Kozyrev S.A., Vlasova E.A. Gas hazards of explosives used in the mining industry. Mining, vol. 5, 2021, pp. 106-111. (in Russian) DOI: 10.30686/1609-9192-2021-5-106-111

[22]. Свидетельство 2024619185. Программа расчёта коэффициента диффузии атомов газа при уплотнении грунтов взрывом: программа для ЭВМ / Е.О. Тарасенко (RU); правообладатель ФГАОУ ВО «Северо-Кавказский федеральный университет» (RU); № 2024617071; заявл. 05.04.24; опубл. 22.04.2024.

Информация об авторе / Information about author

Елена Олеговна ТАРАСЕНКО – кандидат физико-математических наук, доцент, доцент кафедры Вычислительной математики и кибернетики факультета Математики и компьютерных наук имени профессора Н.И. Червякова Северо-Кавказского федерального университета с 2021 года. Сфера научных интересов: математическое моделирование, вычислительная математика, комплексы программ.

Elena Olegovna TARASENKO – Cand. Sci. (Phys.-Math.), Associate Professor, Associate Professor of the Department of Computational Mathematics and Cybernetics, Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov of the North Caucasus Federal University from 2021. Research interests: mathematical modeling, computational mathematics, software packages.

DOI: 10.15514/ISPRAS-2024-36(5)-14

CC BY

Аналитический обзор методов проектирования систем безопасности в телемедицинских системах

[1] *М.А. Лапина, ORCID: 0000-0001-8117-9142 <mlapina@ncfu.ru>*
[2] *Е.А. Максимова, ORCID: 0000-0001-8788-4256 <maksimova@mirea.ru>*
[3] *В.Г. Лапин, ORCID: 0000-0002-0611-7002 <vitl@skkdc.ru>*
[1] *Н.С. Бойков, ORCID: 0009-0003-4038-0474 <nikitaboickov4@gmail.com>*

[1] *Северо-Кавказский федеральный университет,*
355017, Россия, г. Ставрополь, ул. Пушкина, д. 1.
[2] *МИРЭА – Российский технологический университет,*
119454, Россия, г. Москва, пр-т Вернадского, д. 78.
[3] *Ставропольский краевой клинический консультативно-диагностический центр,*
355000, Россия, г. Ставрополь, ул. Ленина, д. 304.

Аннотация. В статье исследуются методы проектирования систем безопасности при обеспечении конфиденциальности данных в телемедицине. Рассмотрены подходы к аутентификации пациентов, используемые в телездравоохранении с учетом их эффективности и безопасности. Проведен анализ методов аутентификации, включая биометрическую идентификацию, двухфакторную аутентификацию и использование уникальных идентификационных кодов. Выявлены преимущества и недостатки каждого из этих методов, что дает медицинским организациям возможность принимать управленческие решения, наиболее соответствующие структуре информационных систем и уровням риска.

Ключевые слова: телемедицина; информационная безопасность; информационная система; система безопасности; конфиденциальность; персональные данные; врачебная тайна; аутентификация; шифрование; медицинские информационные системы.

Для цитирования: Лапина М. А., Максимова Е. А., Лапин В. Г., Бойков Н. С. Аналитический обзор методов проектирования систем безопасности в телемедицинских системах. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 191–218. DOI: 10.15514/ISPRAS–2024–36(5)–14.

Analytical Review of Methods for Designing E-Health Security Systems

[1] *M.A. Lapina, ORCID: 0000-0001-8117-9142 <norra7@yandex.ru>*
[2] *E.A. Maksimova, ORCID: 0000-0001-8788-4256 <maksimova@mirea.ru>*
[3] *V.G. Lapin, ORCID: 0000-0002-0611-7002 <vitlx@yandex.ru>*
[1] *N.S. Boikov, ORCID: 0009-0003-4038-0474 <nikitaboickov4@gmail.com>*

[1] *North Caucasus Federal University,*
1, Pushkina St., Stavropol, 355017, Russia.
[2] *MIREA - Russian Technological University,*
78, Vernadsky ave, Moscow, 119454, Russia.
[3] *Stavropol Regional Clinical Advisory and Diagnostic Center,*
304, Lenina St, Stavropol, 355000, Russia.

Abstract. The article explores the methods of designing security systems to ensure data confidentiality in telemedicine. The approaches to patient authentication used in tele-health care are considered, with an emphasis on their effectiveness and safety. The analysis of authentication methods, including biometric identification, two-factor authentication and the use of unique identification codes, was carried out. The advantages and disadvantages of each of these methods have been identified, which gives medical organizations the opportunity to make management decisions that best match the structure of information systems and acceptable levels of risk.

Keywords: telemedicine; information security; information system; security system; confidentiality; personal data; medical secrecy; authentication; encryption; medical information systems.

For citation: Lapina M.A., Maksimova E.A., Lapin V.G., Boikov N.S. Analytical review of methods for designing e-health security systems. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024. pp. 191-218 (in Russian). DOI: 10.15514/ISPRAS-20124-36(5)-14.

1. Введение

Телемедицина стала неотъемлемой частью медицинской практики, при этом конфиденциальность продолжает оставаться основополагающим принципом медицинской этики, и ее нарушение может привести к серьезным негативным последствиям. Защита конфиденциальности пациентов – важная этическая и юридическая обязанность медицинских учреждений. Несанкционированный доступ к персональным данным пациентов или утечки таких данных в информационном пространстве могут нанести ущерб репутации пациентов, привести к мошенничеству и злоупотреблению информацией, а также нарушить их право на личную жизнь. Телемедицина подразумевает доступ пациентов к медицинским услугам из удаленных и труднодоступных регионов. Такой подход нашел особое применение в условиях пандемий, что подчеркивает важность обеспечения безопасной передачи и хранения медицинских данных. С развитием перечня и методов угроз безопасности данным, таких как несанкционированный доступ, фишинг, кибератаки, уровень угроз для конфиденциальности данных значительно возрастает. Возникает также необходимость во внедрении эффективных методов аутентификации пользователей телемедцинских систем.

В настоящее время **м**ногие страны принимают законы и нормативно-правовые акты, обязывающие организации соблюдать высокие стандарты конфиденциальности медицинских данных. Нарушение этих норм может привести к серьезным последствиям для организаций. Дополнительную сложность представляет тот факт, что различные медицинские учреждения и пациенты имеют разные потребности и допускают для себя разные уровни риска. Выбор подходящего метода аутентификации становится критически важным для удовлетворения возникающих потребностей.

За последние десятилетия телемедицина претерпела значительное развитие, предоставляя пациентам доступ к медицинским услугам и консультациям через цифровые платформы. Развитие в этой области привело к увеличению объема электронных медицинских данных, которые передаются по каналам связи и хранятся в информационных системах. Безопасность и конфиденциальность таких данных стали одними из главных вопросов в телемедицине. Работа с этой информацией несет риск разглашения и нарушения ее защиты. В России закон устанавливает, что врачебная тайна включает информацию о том факте, что человек обратился за медицинской помощью, о состоянии его здоровья, об установленном диагнозе, а также другую информацию, полученную в процессе обследования и лечения (ч. 1 ст. 13 Федерального закона от 21 ноября 2011 г. № 323-ФЗ "Об основах охраны здоровья граждан в Российской Федерации", далее - Закон № 323-ФЗ) [1]. Лица, получившие доступ к такой информации (например, в рамках обучения или исполнения служебных обязанностей), не могут ее разглашать, кроме случаев, которые установлены законом (ч. 2 ст. 13 Закона № 323-ФЗ) [1]. Международный кодекс медицинской этики обязывает любого врача уважать право пациента на конфиденциальность и этично раскрывать конфиденциальную информацию при согласии пациента или при наличии реальной и непосредственной угрозы причинения вреда здоровью пациента или другим лицам, причем только в тех случаях, когда эта угроза может быть устранена только путем нарушения конфиденциальности [2]. Использование телемедицинских технологий при оказании медицинской помощи также осуществляется с соблюдением требований, установленных законодательством Российской Федерации в области персональных данных и врачебной тайны [3].

Необходимость неукоснительного соблюдения требований о неразглашении сведений, составляющих врачебную тайну, непосредственно закреплена в части 1 статьи 23 Конституции Российской Федерации, где указано, что «каждый имеет право на неприкосновенность частной жизни, личную и семейную тайну» [4, 5].

Врачебная тайна в соответствии с Конституцией Российской Федерации относится к частной жизни человека, и ее суть заключается в защите прав граждан на личную и семейную тайну от необоснованного и незаконного вмешательства со стороны государственных органов, органов местного самоуправления, государственных и негосударственных организаций, органов власти и частных лиц. Врачебная тайна – относится к сведениям, непосредственно связанным с профессиональной деятельностью медицинских работников, доступ к которой ограничен [4].

Сведения о факте обращения гражданина за оказанием медицинской помощи, состоянии его здоровья и диагнозе, иные сведения, полученные при его медицинском обследовании и лечении, как раз и составляют врачебную тайну (Федеральный закон от 21.11.2011 № 323-ФЗ «Об основах охраны здоровья граждан в Российской Федерации»), а именно его статья 13, содержание которой отражает особое отношение государства к охране врачебной тайны, а именно, врачебную тайну составляют «сведения о факте обращения гражданина за оказанием медицинской помощи, состоянии его здоровья и диагнозе, иные сведения, полученные при его медицинском обследовании и лечении» [1, 5].

Важно отметить, что врачебная тайна играет ключевую роль в защите частной жизни пациента. Однако в настоящее время, с развитием цифровых технологий и электронных медицинских карт, обеспечение безопасности и аутентификации пользователей в медицинской сфере стало более актуальным, для него выработаны методы, позволяющие сбалансировать конфиденциальность медицинской информации и обеспечить надежную защиту.

Цель исследования – проанализировать современные результаты в области обеспечения конфиденциальности и безопасности данных в телемедицинских системах, а также провести сравнительную характеристику методов аутентификации пациентов для выявления лучших практик в области защиты медицинской информации и аутентификации пациентов.

Телемедицинские технологии (телемедицина) представляют собой информационные технологии, которые обеспечивают дистанционное взаимодействие медицинских работников между собой и с пациентами, а также идентификацию указанных лиц и документирование действий при проведении консультаций, консилиумов и дистанционного медицинского наблюдения (п. 22 ст. 2 Закона № 323-ФЗ) [6].

Рассмотрим виды конфиденциальной информации в медицинских учреждениях.

- Персональные данные пациентов: ФИО, место рождения и проживания, контактная информация, номер медицинского полиса и прочее. Этот вид данных наиболее подвержен утечкам.
- Медицинская (врачебная) тайна. Включает сведения, иллюстрирующие состояние здоровья пациента, наличие заболеваний, диагнозы, результаты лечения.
- Коммерческая тайна: сведения о базе клиентов, планы развития организации, методология лечения болезней, способы проведения качественных исследований и проверок.
- Статистические сведения: информация из карт пациентов, сведения о собственных работниках (например, зарплаты), работе с бюджетными средствами и прочее.
- Иная служебная информация: результаты служебных проверок по исполнению требований по работе и служебным обязанностям.

Особое внимание при обеспечении безопасности уделяется персональным данным пациентов и сведениям, составляющим врачебную тайну.

В статье 3 Федерального закона «О персональных данных» от 27.07.2006 №152-ФЗ содержится определение термина «персональные данные», под которыми понимается любая информация, относящаяся прямо или косвенно к определенному или определяемому физическому лицу. Целью указанного закона является обеспечение защиты прав и свобод человека и гражданина при обращении с его персональными данными, в том числе защита права на неприкосновенность частной жизни, личную и семейную тайну.

Федеральный закон о персональных данных РФ № 152-ФЗ устанавливает правила для обработки персональных данных, включая медицинскую информацию. Он требует, чтобы медицинские организации обеспечивали конфиденциальность и безопасность данных пациентов, а также предусматривает строгие санкции за нарушение этих норм [7]. Важно, что соблюдение этих правил не просто законное требование, но и ключевой фактор поддержания доверия со стороны пациентов. В сфере телемедицины, где данные передаются через неконтролируемые информационные сети, конфиденциальность данных играет критическую роль в выборе медицинской услуги. Соблюдение законодательства о персональных данных помогает не только избежать юридических последствий, но также улучшить репутацию медицинских учреждений и обеспечить долгосрочное доверие пациентов [6].

Понятие врачебной тайны является более широким, нежели персональные данные. И ряд сведений, не относящихся к персональным данным, все равно относятся к врачебной тайне. Например, исключительно данные о диагнозе пациента или методе лечения не позволят идентифицировать лицо [5, 7].

С письменного согласия гражданина или его законного представителя могут быть разглашены сведения, составляющие врачебную тайну в целях медицинского обследования и лечения пациента, проведения научных исследований, их опубликования в научных изданиях, использования данных сведений в учебном процессе.

Законодательством установлены случаи, при которых возможно предоставление сведений, составляющих врачебную тайну, без согласия гражданина (его законного представителя).

Разглашение сведений, составляющих врачебную тайну, может выражаться несколькими способами, а именно:

- Устное распространение сведений, составляющих врачебную тайну. Такое

разглашение может совершаться как умышленно, так и по неосторожности. Особенно часто такая информация передается по телефону третьим лицам, представляющимся родственниками пациента;

- Разглашение сведений, составляющих врачебную тайну возможно при непосредственном общении врача и пациента в общественных местах при сообщении информации о состоянии здоровья в присутствии третьих лиц;
- Указание в научных текстах, исследованиях и иной литературе информации о состоянии здоровья пациента, его диагнозе, течении болезни и прочее без предварительного письменного согласия больного;
- Небрежное отношение с медицинской документацией, в том числе, амбулаторной картой, историей болезни, листками нетрудоспособности.

Важным фактором является постоянный рост объема медицинской информации, передаваемой и хранимой в рамках телемедицины. Электронные медицинские записи и обмен данными стали неотъемлемой частью современной медицины. В этом контексте, нарушение конфиденциальности данных может иметь серьезные последствия. Утечка чувствительной медицинской информации может привести к серьезным репутационным и финансовым последствиям, как для пациентов, так и для медицинских организаций. Соблюдение правил и нормативов в области здравоохранения очень важно. Несоблюдение этих требований может привести к серьезным правовым последствиям.

В случае разглашения сведений, составляющих врачебную тайну, действующим законодательством предусмотрена как административная, так и уголовная ответственность. Ответственности подлежат лица, которым сведения, составляющие врачебную тайну, стали известны при обучении, исполнении трудовых, должностных, служебных и иных обязанностей. Согласно законодательству РФ, к нарушителю, допустившему раскрытие врачебной тайны, персональных данных пациента и другой информации, охраняемой законом, могут быть применены следующие виды наказаний:

- Дисциплинарное. Выражается в вынесении виновнику замечания, выговора. В крайних случаях не исключено увольнение.
- Гражданское. Назначается судом в виде штрафа или возмещения убытков согласно степени нанесенного вреда. Характерно для случаев, где пострадавшей стороне нанесен материальный или моральный ущерб.
- Административное. Заключается в назначении штрафа на сумму 1000-5000 рублей для физических и должностных лиц (статья 13.14 КоАП).
- Уголовное. Назначается штраф в размере 100 000-300 000 рублей, принудительные работы продолжительностью до 4 лет или арест продолжительностью до 6 месяцев с отзывом права занимать отдельные должности. В тяжелых и исключительных случаях в качестве наказания назначается лишение свободы сроком до 4 лет, ограничение заниматься профессиональной деятельностью на срок 2-5 лет (статья 137 УК РФ «Нарушение неприкосновенности частной жизни»).

Так, за разглашение информации, доступ к которой ограничен федеральным законом (за исключением случаев, если разглашение такой информации влечет уголовную ответственность), в отношении лица, получившим доступ к такой информации в связи с исполнением служебных или профессиональных обязанностей, может быть возбуждено дело об административном правонарушении по ст. 13.14 Кодекса Российской Федерации об административных правонарушениях (далее – КоАП РФ). Санкция данной статьи предусматривает наказание для граждан в размере от 500 до 1 тысячи рублей; для должностных лиц - от 4 тысяч до 5 тысяч рублей [5, 8].

Максимальное наказание за совершение данного деяния предусмотрено в виде штрафа в размере от 150 тысяч до 350 тысяч рублей или в размере заработной платы или иного дохода осужденного за период от 18 месяцев до 3 лет, либо лишением права занимать определенные должности или заниматься определенной деятельностью на срок от 3 до 5 лет, либо принудительными работами на срок до 5 лет с лишением права занимать определенные должности или заниматься определенной деятельностью на срок до 6 лет или без такового, либо арестом на срок до 6 месяцев, либо лишением свободы на срок до 5 лет с лишением права занимать определенные должности или заниматься определенной деятельностью на срок до 6 лет.

Кроме того, согласно ст. 1068 Гражданского кодекса Российской Федерации юридическое лицо возмещает вред, причиненный его работником при исполнении трудовых (служебных, должностных) обязанностей. Это означает, что медицинское учреждение (должностные лица) в данной ситуации также может быть привлечено к гражданско-правовой ответственности [5, 9].

Защита данных – это не только юридическое требование, но и этическая обязанность медицинских учреждений. Пациенты доверяют им свою глубоко личную информацию о здоровье, и поддержание этого доверия является фундаментальным аспектом эффективной телемедицины.

Важность сохранения доверия и конфиденциальности пациентов в области телемедицины и здравоохранения не должна быть недооценена. Аспекты обеспечения конфиденциальности медицинских данных представлены на рис. 1.

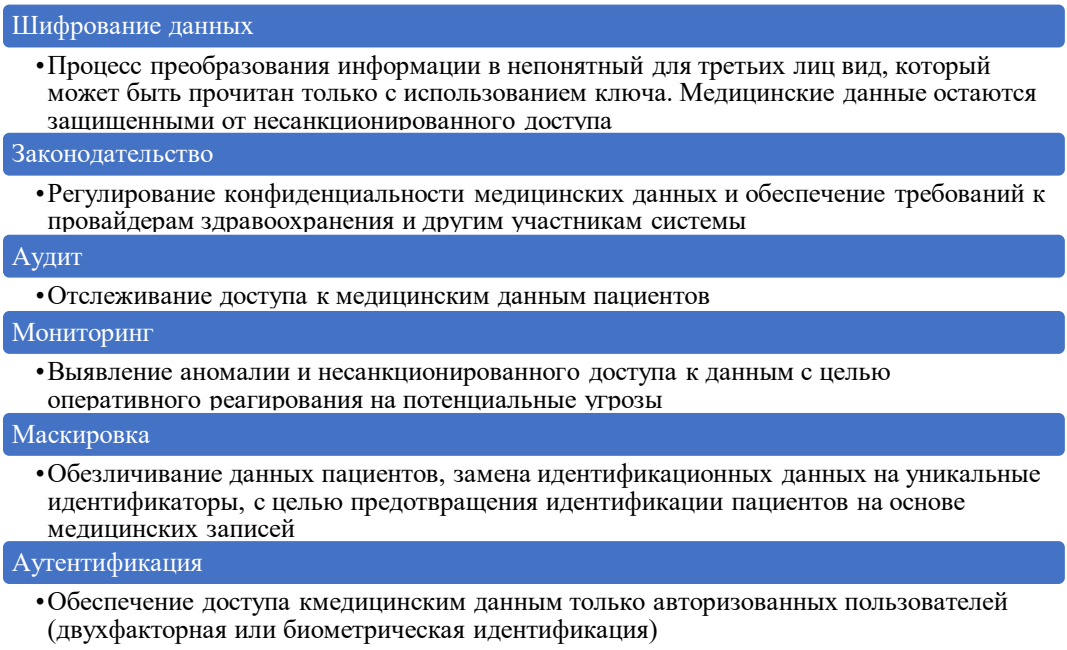


Рис. 1. Аспекты конфиденциальности медицинских данных.
Fig. 1. Aspects of medical data privacy.

С развитием технологий, включая широкополосный интернет и смартфоны, телемедицина стала еще более доступной и распространенной. Пациенты могут получать консультации и медицинские советы прямо из дома, что особенно важно в отдаленных и малонаселенных районах. Многие страны разрабатывают законы и нормативы, которые регулируют использование телемедицины. Это включает в себя вопросы безопасности данных,

лицензирование медицинских профессионалов, и другие аспекты, чтобы обеспечить качество и безопасность телемедицинских услуг.

2. Этапы развития телемедицины

Телемедицина представляет собой использование компьютерных и телекоммуникационных технологий в здравоохранении для лечения, диагностики и предотвращения заболеваний, а также для организационного управления в этой области. Часто этот термин сопровождается словами "современный", "инновационный", "впервые разработанный" и другими подобными. Телемедицина, несмотря на свое современное развитие, имеет свои корни еще в XIX веке. Ее инструменты и технологии продолжают меняться с течением времени. Например, в 1940-х годах актуальным средством был телеграфный аппарат, созданный Жаном Бодо, а в 2010-х годах – смартфоны и облачные программные средства [10, 11].

Развитие дистанционного предоставления медицинской помощи и услуг тесно связано с прогрессом в телекоммуникационных средствах. На разных этапах истории телемедицины применялись передовые технологии для достижения ее целей. История телемедицины – это постепенная эволюция средств связи и удаленного обмена медицинской информацией. Прямо на сущность телемедицины развитие систем здравоохранения и отдельных областей медицины не влияет, так как ее основная цель – передача медицинской информации, остается постоянной, независимо от конкретной области медицины. Более того, в определенные моменты истории телемедицина становилась мощным инструментом для получения существенных новых медицинских знаний, например, через использование радиотелеметрии [12, 13].

- Первая волна включает в себя изобретение телеграфа, радио и телефона.
- Вторая волна охватывает развитие телевидения, включая кабельное и беспроводное телевидение, переход от медленной развертки к цветному изображению.
- Третья волна связана с разработкой инструментов модуляции и демодуляции для передачи данных по телефонным линиям связи.
- Четвертая волна характеризуется развитием спутниковой связи.
- Пятая волна включает в себя создание локальных и распределенных сетей, а также развитие интернет-протокола.

В период с 1850 по нынешние годы можно выделить несколько "волн" в развитии телекоммуникационных технологий. В соответствии с исследованиями Максимова И.Б. и др. [14] временные периоды в развитии телемедицины представлена в табл. 1.

Телемедицинский период начался в 1905 году, когда впервые была передана электрокардиограмма на расстояние, представляя собой важный этап в развитии этой области.

В 1920-х годах морская телемедицина стала активно развиваться, особенно в ситуациях, когда моряки нуждались в медицинской помощи во время длительных плаваний, а медицинских специалистов на борту судов не всегда хватало. В таких ситуациях использовалась морская телемедицина, позволяя обмениваться медицинской информацией на расстоянии и получать консультации узких специалистов [15].

С развитием технологий появилась возможность проводить видеоконференции. Это открыло новые возможности для обмена информацией и консультирования, в том числе в сфере телемедицины. Важным моментом стал 1949 год, когда была проведена первая цветная видеоконференцсвязь между медицинскими специалистами, что стало важным шагом в развитии телемедицинских консультаций [15].

Активно развивалась телемедицина в СССР. С 1960-х по 1990-е годы прошли множество дистанционных консультаций в разнообразных областях, с фокусом на передаче ЭКГ и ее

активном использовании в космических исследованиях. Многие инновации, включая технологии телемедицины, свое начало брали из таких дисциплин, как военная, космическая и морская индустрия, которым практически повсюду придавали приоритетное значение. В стремлении каждой страны занять лидирующую позицию в конкретной сфере, активно развивались промышленные технологии, придавая существенное движение в развитии телемедицинских практик.

Табл. 1. Важные события в области телемедицины.
Table 1. Results for different strategies.

Год	Период / волна	Важные события в телемедицине
1920	I	Первые эксперименты с использованием радио и телефона для консультаций между врачами.
1960	II	Применение спутниковых связей в телемедицине для связи с отдаленными лечебными учреждениями и кораблями.
1990	III	Развитие интернета и мобильной связи способствует распространению телемедицинских консультаций и мониторинга. Появление телемедицинских компаний.
2010	IV	Внедрение мобильных приложений и сенсорных устройств для мониторинга здоровья. Значительное развитие телемедицины в онкологии и диагностике.
2020	V	Бурное развитие телемедицины в связи с пандемией COVID-19 и расширением онлайн-консультаций. Использование искусственного интеллекта и аналитики данных в телемедицине.

В частности, в СССР, в момент исторического полета первого космонавта Юрия Гагарина, телемедицинские инновации уже играли важную роль. Гагарин был подключен к разнообразным устройствам, передающим медицинские данные на Землю, и на планете работало множество медицинских специалистов, бдительно следивших за его здоровьем. Развитие технологий в настоящее время свидетельствует о непрерывном развитии телемедицины. Например, на борту Международной космической станции (МКС), где астронавты проводят продолжительное время, регулярно проводят медицинские осмотры. На станции находится обширное медицинское оборудование, которое выполняет функции диагностики и передает информацию о состоянии здоровья астронавтов на Землю, где эти данные анализируются высококвалифицированными медицинскими специалистами [10].

Если рассматривать эволюцию телемедицины, то она началась с простых технологий передачи данных. На протяжении нескольких лет формируется нормативно-правовое поле регламентации оказания услуг с помощью технологий телемедицины [16].

В будущем технологии телемедицины продолжат развиваться. Этот процесс будет способствовать дальнейшему совершенствованию и улучшению доступности медицинской помощи на расстоянии.

3. Этапы обеспечения конфиденциальности в телемедицине

С развитием технологий и интернета, телемедицина стала широко распространенной практикой, которая позволяет проводить медицинские консультации и лечение пациентов удаленно. Конфиденциальность данных в области телемедицины является одним из ключевых аспектов ее функционирования, что обеспечивает защиту личной и медицинской информации пациентов [17].

Рассмотрим основные этапы обеспечения конфиденциальности в телемедицине.

- Защита данных. Этап включает в себя создание и использование безопасных систем

хранения и передачи данных пациентов. Это может быть реализовано путем использования защищенных интернет-протоколов и программного обеспечения, включающего механизмы шифрования и аутентификации.

- Верификация пациента. Для обеспечения конфиденциальности могут использоваться различные методы идентификации пациента, такие как проверка по электронным документам, биометрическая аутентификация или проверка с помощью одноразовых кодов.
- Контроль доступа. Для предотвращения несанкционированного доступа к медицинской информации пациентов, провайдеры телемедицины должны разработать строгие политики и процедуры, регулирующие доступ персонала к данным. Это включает ограничение доступа к информации только для авторизованного медицинского персонала и использование паролей или других средств аутентификации для входа в систему.
- Обучение персонала. Образование и обучение медицинского персонала в области конфиденциальности является неотъемлемой частью обеспечения безопасности данных пациентов. Провайдеры телемедицины должны предоставить своему персоналу обучение по темам, связанным с конфиденциальностью, этикой и соблюдением HIPAA (Закон о портативности и ответственности в медицинском страховании), если это применимо.
- Согласие пациента. Пациенты должны давать согласие на использование и передачу своей медицинской информации при получении телемедицинской консультации или лечения. Провайдеры телемедицины должны разработать и использовать ясные и понятные политики сбора и использования данных пациентов, а также получать письменное согласие пациента перед обработкой их данных.
- Аудит и мониторинг. Важным аспектом обеспечения конфиденциальности телемедицины является проведение регулярных аудитов и мониторинга системы для выявления и предотвращения любых нарушений безопасности. Это включает проверку журналов доступа, мониторинг сетевого трафика и реагирование на любые подозрительные активности.

4. Принципы обеспечения информационной безопасности для обеспечения конфиденциальности данных в телемедицинских системах

Телемедицина представляет собой важную область медицинской практики, которая использует информационные технологии для предоставления медицинских услуг на расстоянии с использованием информационно-коммуникационных технологий. С увеличением популярности телемедицины становится критически важным обеспечение конфиденциальности данных пациентов.

Анализ современных достижений в области обеспечения конфиденциальности и безопасности данных в системах телемедицины имеет важное значение в современном мире, где цифровизация здравоохранения стала неотъемлемой частью медицинской практики. Это обеспечивает более удобное и доступное обслуживание пациентов, однако также поднимает ряд важных вопросов, связанных с безопасностью данных и аутентификацией пациентов [18].

Рассмотрим методы применения фундаментальных принципов безопасности данных в телемедицине, на основы которых достигается высокий уровень обеспечения конфиденциальности данных пациентов.

4.1 Методология управления идентификацией FAIDM

Методология федеративного анонимного управления идентификацией (Federated Anonymous Identity Management, FAIDM) используется в системах телемедицины для обеспечения конфиденциальности медицинских данных. Методика включает в себя несколько ключевых компонентов:

- GW – "Шлюзовой узел" (Gateway).
- CNi – локальный узел (Constrained Node) в контексте Интернета вещей (IoT).

Система, построенная на базе методологии FAIDM, включает в себя удаленный серверный узел, шлюзовой узел (GW), и локальный узел (CNi). Ограниченный узел приближен к источникам сенсорной информации, собираемой в системе мониторинга пациентов, и может быть интегрирован в устройства, например, в датчики или в носимые устройства, имеющиеся у пациентов [17].

Обычно эти устройства служат для мониторинга окружающей среды и собирают соответствующие данные, которые передаются шлюзовым узлам. В сфере здравоохранения такие датчики могут быть размещены внутри или на теле пациентов для сбора медицинских данных. Шлюзовые узлы расположены на сетевом или коммуникационном уровне структуры системы мониторинга пациентов и, предположительно, обладают достаточными ресурсами по энергопотреблению, производительности процессора и объему памяти [19].

Шлюзы выполняют обработку данных, собранных различными ограниченными узлами, и передают их на удаленный серверный узел, который находится в структуре системы мониторинга пациентов уровнем выше и, скорее всего, не имеет ограничений по вычислительным ресурсам. Медицинский персонал на этом уровне имеет возможность непрерывно мониторить состояние здоровья пациентов на основе получаемых данных [20].

Безопасность взаимодействия между связующими узлами и основным уровнем архитектуры является настолько важной, что для ее обеспечения создается комплекс базовых сетевых функций, в частности, функция сервера аутентификации (Authentication Server Function, AUSF), хранилище и функция обработки учетных данных аутентификации (Authentication Credential Repository and Processing Function, ARPF), функция разглашения идентификатора подписки (Subscription Identifier De-concealing Function, SIDF) и функция закрепления безопасности (Security Anchor Function, SEAF). Однако, несмотря на всю важность этих средств обеспечения безопасности связи между уровнями системы, в описываемой авторами схеме они подробно не рассматриваются [19].

Ключевую роль в инициализации системы и формировании ее параметров играет удаленный серверный узел. Все шлюзовые узлы для обеспечения собственной легитимности должны проходить процедуру регистрации на удаленном сервере. Аналогичным образом каждый ограниченный узел должен зарегистрироваться на одном из легальных шлюзовых узлов.

Когда пациенту вручается носимое медицинское устройство, и он возвращается домой после лечения в больнице, устройство начинает передавать собранные медицинские данные через шлюз IoT (GW), который находится в другом домене, отличающемся от домена больницы. Рассмотрим алгоритм работы метода.

- Инициализации системы.

На начальном этапе удаленного серверного узла удаленный серверный узел S, который предоставляет телемедицинские услуги и сертифицирован органом сертификации здравоохранения, настраивает параметры

- Регистрации узла шлюза.

Шлюзовой узел GWi взаимодействует с удаленным серверным узлом S для регистрации

- Регистрации локального узла

В фазе регистрации локального узла, ограниченный узел $CNij$ отправляет регистрационную информацию на шлюзовой узел GWi.

- Взаимной аутентификации и согласования ключей

После присоединения локального узла к альянсу удаленных серверных узлов в качестве члена удаленного серверного узла, он получает доступ к услугам, предоставляемым не только зарегистрированным поставщиком услуг, но и другими поставщиками услуг в том же альянсе удаленных серверных узлов

- Анонимная подпись и этап проверки

Узел шлюза GWt (верификатор) получает и проверяет сообщение с подписью, которая была создана анонимным закрытым ключом $aSij$, с использованием функции проверки подписи. Это позволяет удостовериться, что сообщение было подписано анонимным узлом, связанным с закрытым ключом $aSij$, и подтвердить его подлинность

4.2 Технология блокчейн

Блокчейн технология (технология распределенного реестра) в телемедицине представляет собой подход к обеспечению безопасности и конфиденциальности медицинских данных и транзакций. Эта концепция позволяет создать два уровня блокчейна [21], что повышает качество системы телемедицины.

1. Основной блокчейн (Main Blockchain):
   - Первый уровень блокчейна, называемый основным, содержит общие медицинские данные, такие как истории болезни, диагнозы, рецепты и другие медицинские записи.
   - Этот уровень отвечает за устойчивость и надежность данных, он также обеспечивает прозрачность доступа к медицинской информации в рамках медицинской сети.
2. Вторичный блокчейн (Secondary Blockchain):
   - Второй уровень блокчейна, известный как вторичный, используется для построения дополнительных слоев безопасности и конфиденциальности.
   - На этом уровне хранятся более чувствительные медицинские данные, такие как биометрические данные, генетические сведения и другие приватные информационные элементы.

При каждой медицинской транзакции, данные записываются одновременно на оба уровня блокчейна. Основной блокчейн гарантирует доступность данных для медицинских работников и организаций, которым такой доступ разрешен. Вторичный блокчейн используется для хранения наиболее конфиденциальных данных и доступен только при наличии соответствующих дополнительных разрешений и аутентификации.

Преимущества двойного блокчейна в телемедицине представлены на рис. 2.

Таким образом, двойной блокчейн в телемедицине помогает сбалансировать доступность и конфиденциальность данных, создавая более безопасную и эффективную систему передачи и хранения медицинской информации.

4.3 Технология радиочастотной идентификации RFID

Применение технологии радиочастотной идентификации (Radio-Frequency Identification, RFID) позволяет идентифицировать, отслеживать и управлять удаленными объектами и связанной с ними информацией [22]. В телемедицине технология RFID играет важную роль, предоставляя ряд преимуществ и способов применения.

Для идентификации пациентов и медицинского персонала могут быть использованы RFID метки. Каждый пациент может иметь индивидуальную RFID-метку, которая содержит его уникальные идентификационные данные. Это позволяет медицинскому персоналу быстро и точно идентифицировать пациентов [22, 23].

На рис. 3 представлена методика работы технологии двойного блокчейна.

Усиленная безопасность данных

- Медицинские данные находятся на вторичном блокчейне

Конфиденциальность данных

- Пациенты могут быть уверены в сохранении конфиденциальности данных, так как доступ к вторичному блокчейну строго ограничен

Совместимость с нормативно-правовым регулированием

- Соответствие законодательству и стандартам в области защиты медицинских данных.

Рис. 2. Преимущества двойного блокчейна в телемедицине.
Fig. 2. Benefits of dual blockchain in telemedicine.

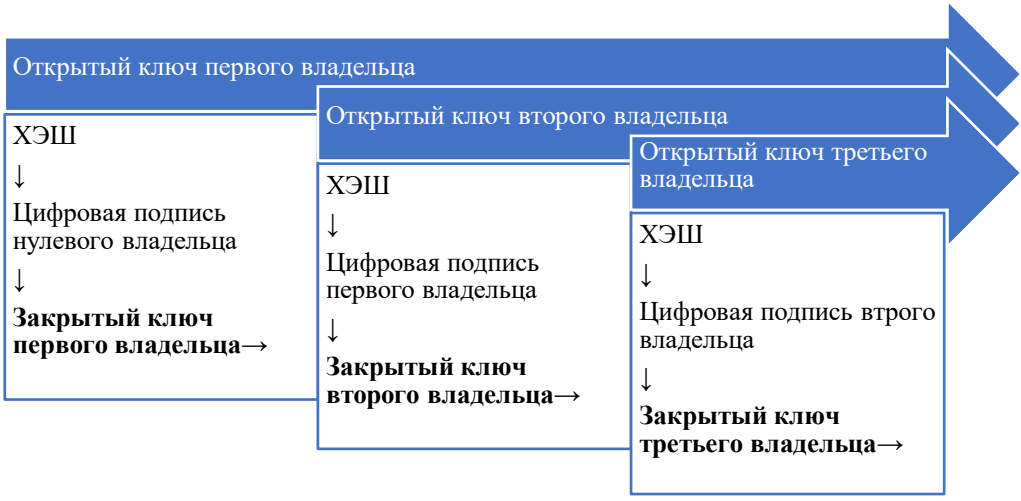


Рис. 3. Методика работы технологии двойного блокчейна.
Fig. 3. The methodology of the dual blockchain technology.

Технология RFID также может использоваться для контроля доступа в медицинских учреждениях, гарантируя, что только уполномоченные сотрудники имеют доступ к определенным зонам.

Рассмотрим механизмы обеспечения контроля доступа и безопасность медицинского оборудования.

- **Управление медицинским оборудованием.** Множество медицинских устройств и инструментов может быть маркировано RFID-метками. Это позволяет медицинским учреждениям отслеживать местонахождение и состояние оборудования. Например, хирургические инструменты могут быть помечены RFID-метками, что облегчает их учет и предотвращает утерю

- **Мониторинг запасов медикаментов и медицинского оборудования.** Процесс позволяет медицинским учреждениям управлять своими ресурсами более эффективно
- **Безопасность пациентов.** RFID-браслеты и карточки могут быть применены для обеспечения безопасности пациентов. Устройства содержат информацию о хронических заболеваниях и медицинских рецептах. В случае неотложной медицинской помощи, медицинский персонал может быстро получить доступ к этой информации
- **Отслеживание медицинских процедур.** В операционных и процедурных залах RFID может использоваться для отслеживания хода операций и медицинских процедур. Это может помочь предотвратить ошибки и обеспечить точное выполнение медицинских рецептов
- **Управление медицинскими записями.** Медицинские записи пациентов могут быть связаны с RFID-метками, что делает их доступными и обновляемыми в реальном времени. Это особенно полезно в системах телемедицины, где медицинские данные могут передаваться и обновляться удаленно
- **Телемониторинг и носимые устройства.** Носимые медицинские устройства, такие как мониторы сердечного ритма или измерители уровня сахара в крови, могут использовать RFID для передачи данных на мобильные устройства для мониторинга в реальном времени. Это позволяет врачам и семейным участникам отслеживать состояние пациентов на расстоянии
- **Системы контроля доступа и безопасности.** RFID-технология может быть использована для обеспечения безопасности в медицинских учреждениях. Это включает в себя контроль доступа в определенные зоны, мониторинг перемещения пациентов и документов, а также обеспечение безопасности в критических ситуациях

На рис. 4 показаны этапы применения технологии RFID в медицине.

Таким образом, RFID играет важную роль в телемедицине, обеспечивая идентификацию пациентов, отслеживание медицинского оборудования, управление медицинскими записями и обеспечение безопасности и безопасности данных в здравоохранении, помогает улучшить эффективность и качество медицинского обслуживания.

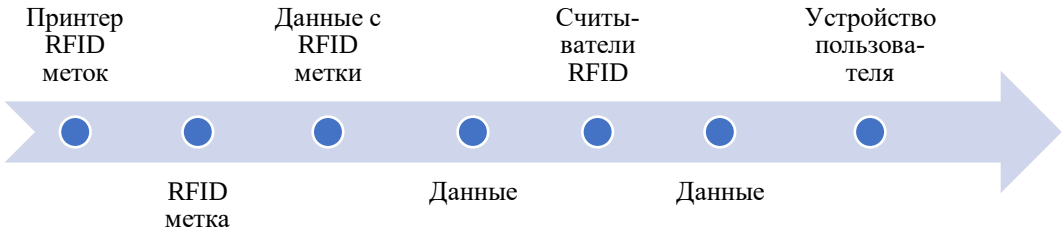


Рис. 4 Этапы применения технологии RFID в медицине.
Fig. 4. RFID technology in medicine.

4.4 Технология бесконтактной связи NFC

Технология бесконтактной связи (Near Field Communication, NFC) может быть полезной в телемедицине для ряда задач и приложений. В технологию NFC входят следующие компоненты:

- **Идентификация пациентов**: для идентификации пациентов в медицинских учреждениях могут быть использованы NFC-метки. Каждый пациент может иметь NFC-метку с уникальным идентификационным номером, которая может быть считана при посещении врача или госпиталя для быстрой и точной идентификации [24].
- **Доступ к медицинской истории**: Пациенты могут использовать NFC-метки, чтобы предоставить врачам доступ к своей медицинской истории и данным о заболеваниях, лекарствах и диагнозах. Это делает процесс консультации более эффективным.
- **Мониторинг пациентов**: Устройства для мониторинга состояния пациентов, такие как носимые медицинские устройства, могут использовать NFC для передачи данных на смартфоны или другие устройства для мониторинга. Например, NFC может быть использовано для передачи данных о сердечном ритме или уровне сахара в крови.
- **Управление лекарствами**: NFC-метки могут быть прикреплены к упаковке лекарств, чтобы предоставить информацию о дозировке, сроках годности и рецепту. Пациенты могут использовать смартфоны для сканирования меток и получения рекомендаций по приему лекарств.
- **Системы контроля доступа**: NFC может быть использовано для ограничения доступа к чувствительной медицинской технике и данным. Только уполномоченный персонал или пациенты с правильными разрешениями могут получить доступ к этой информации.
- **Отслеживание медицинского оборудования**: NFC может быть применено для отслеживания и учета медицинского оборудования, такого как инструменты, медикаменты и медицинские приборы [25].
- **Безопасный обмен медицинскими данными**: NFC может обеспечить безопасный и шифрованный обмен медицинскими данными между устройствами, что критически важно для сохранения конфиденциальности пациентов [25].

Технология NFC предоставляет удобный и безопасный способ обмена и доступа к медицинским данным, улучшая эффективность и безопасность телемедицинских процессов и систем (рис. 5).

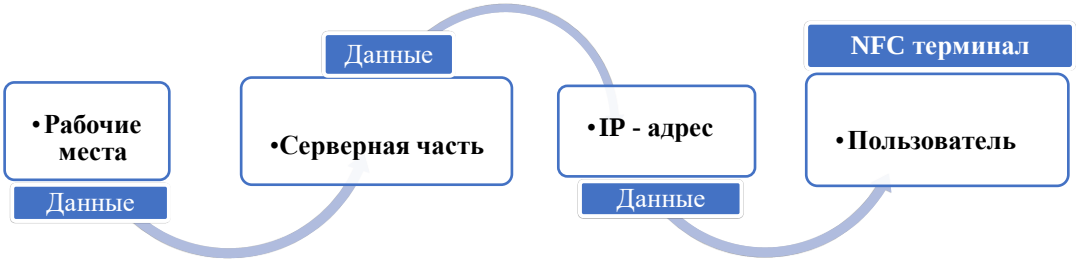


Рис. 5. Технология NFC.
Fig. 5. NFC technology.

Сравнительная характеристика методов аутентификации пациентов позволяет определить направления для повышения уровня защиты медицинской информации и обеспечения безопасности в системах телемедицины [26-28]. Результаты проведенного анализа представлены в табл. 2.

Табл. 2. Сравнительная характеристика принципов безопасности данных для обеспечения конфиденциальности в телемедицине.
Table 2. Comparative Characteristics of Data Security Principles for Ensuring Confidentiality in Telemedicine.

Характеристика	FAIDM	NFC	Двойной блокчейн	RFID
Применение	Используется для идентификации пациентов, доступа к медицинским данным и обеспечения безопасности данных	Может использоваться для идентификации пациентов и доступа к медицинской информации	Может использоваться для обеспечения безопасности и целостности медицинских записей	Беспроводная технология для идентификации и отслеживания объектов
Безопасность	Высокий уровень безопасности с использованием идентификаторов и шифрования данных	Уровень безопасности зависит от применяемых мер безопасности	Высокий уровень безопасности благодаря двум блокчейнам, что делает его более надежным	Уровень безопасности зависит от применяемых мер безопасности
Дальность связи	Средний или длинный диапазон, в зависимости от используемой технологии (например, Bluetooth, WiFi)	Короткий диапазон, обычно менее 4 см	Не применимо, так как это не беспроводная технология	Короткий или средний диапазон, в зависимости от типа RFID (активный или пассивный)
Определение	Передача данных с использованием различных идентификаторов на удаленных устройствах	Беспроводная технология для ближней связи, используемая для идентификации и обмена данными	Технология, которая использует два блокчейна для обеспечения дополнительной безопасности	Беспроводная технология для идентификации и отслеживания объектов

Анализ полученных результатов исследований показывает, что выбор метода аутентификации зависит от конкретных требований проекта и уровня безопасности, необходимого для защиты данных. Но независимо от выбранного метода, для обеспечения безопасности передачи данных необходимо использовать протокол безопасных передач данных HTTPS или другие меры шифрования. Более безопасной альтернативой базовой аутентификации является дайджест-аутентификация, но даже она не обеспечивает абсолютной безопасности и требует дополнительных мер безопасности.

5. Методы обеспечения конфиденциальности на основе аутентификации пользователей

С развитием телемедицины и увеличением числа онлайн-консультаций и рецептов была поставлена задача усиления безопасности пациентов и предотвращения мошеннических действий. Методы аутентификации, такие как пароли и PIN-коды, оказались уязвимыми для вредоносных атак и не всегда обеспечивают надежную защиту данных. Выбор других, более надежных методов аутентификации, зависит от нужного уровня безопасности, удобства для пациентов и нормативных требований.

5.1 Язык управления доступом XACML

Язык управления доступом (eXtensible Access Control Markup Language, XACML) – это стандартный язык и формат для определения политик управления доступом. Он широко используется в области информационной безопасности и управления доступом, включая телемедицину. Применение языка XACML в телемедицине состоит из следующих компонентов:

- **Управление доступом к медицинским данным.** Важно обеспечивать доступ к медицинским данным только тем лицам, которые имеют на это право. XACML позволяет создать гибкие политики управления доступом, которые определяют, кто, как и когда может получать доступ к медицинской информации.
- **Ролевая аутентификация.** XACML может использоваться для определения ролей и привилегий различных пользователей в системе телемедицины. Например, врачи, медсестры и пациенты могут иметь разные уровни доступа к медицинским данным. XACML позволяет настраивать правила доступа в соответствии с этими ролями.
- **Многоуровневые политики доступа.** В системах телемедицины могут существовать различные уровни доступа к данным в зависимости от чувствительности информации. XACML позволяет определять многоуровневые политики, где определенные данные доступны только определенным пользователям или ролям.
- **Динамическое управление доступом.** Врачи и медицинский персонал могут временно требовать доступа к медицинским данным пациента для диагностики и лечения. XACML поддерживает динамическое изменение политик доступа, что позволяет предоставлять доступ при необходимости и отзывать его по завершении процедуры.
- **Аудит доступа.** XACML позволяет регистрировать все попытки доступа к медицинским данным. Это важно для соблюдения законов и нормативных актов в области конфиденциальности медицинских данных.
- **Гибкость и настраиваемость.** XACML обладает высокой степенью гибкости и настраиваемости. Политики доступа могут быть адаптированы к конкретным потребностям медицинских организаций, что позволяет учитывать специфические требования в области безопасности и конфиденциальности данных.

Использование языка XACML в телемедицине помогает обеспечить безопасность, конфиденциальность и целостность медицинских данных, а также эффективное управление доступом медицинского персонала и пациентов к этим данным. Особенно это важно в свете растущей востребованности телемедицины и необходимости соблюдения нормативных актов в области защиты данных.

Язык XACML в телемедицине используется для последовательного достижения следующих целей [23-29]:

- Создание политик контроля доступа для медицинских данных. Политики определяют, кто может иметь доступ к каким данным.
- Аутентификация пользователей и запросов на доступ к данным.
- Применение политик XACML для принятия решений о предоставлении или ограничении доступа к медицинским данным.

5.2 Схема шифрования PEKS

Схема частичного гомоморфного шифрования (Partially Homomorphic Encryption Key Scheme, PEKS) является криптографической техникой, которая может использоваться в

телемедицине для обеспечения конфиденциальности и безопасности медицинских данных, предоставляя средства шифрования и дешифрации данных, которые могут быть полезны в телемедицине [30].

Схема PEKS описывает частично гомоморфное шифрование. Это означает, что она поддерживает выполнение некоторых операций над зашифрованными данными без их предварительной расшифровки. Это может быть полезно, например, при выполнении вычислений над медицинскими данными без раскрытия самих данных. PEKS также может быть настроена для выполнения поиска в зашифрованных данных, не раскрывая самого секретного содержания [21]. PEKS является сложной криптографической схемой и требует внимательной настройки и управления ключами. Использование этой схемы требует больших вычислительных затрат, что следует учитывать при ее применении в системах телемедицины.

Рассмотрим особенности применения схемы PEKS в телемедицине. Для обеспечения безопасности.

- **Конфиденциальность медицинских данных.** Алгоритм PEKS может быть использован для шифрования медицинских записей и данных пациентов. Это обеспечивает конфиденциальность информации и предотвращает несанкционированный доступ к медицинским данным.
- **Управление доступом.** Алгоритм PEKS позволяет контролировать доступ к медицинской информации. Только авторизованные лица с доступом к соответствующим ключам могут расшифровать зашифрованные данные.
- **Безопасная передача данных.** Алгоритм PEKS может быть использован для безопасной передачи медицинских данных между сторонами, обеспечивая шифрование данных в процессе передачи.
- **Подпись и аутентификация.** Алгоритм PEKS может быть интегрирован с механизмами аутентификации и подписи, что помогает подтвердить подлинность медицинских данных и их источника.

В телемедицине схема PEKS работает следующим образом [30, 31]:

- **Шаг 1**: Создание индекса ключевых слов для медицинских документов.
- **Шаг 2**: Шифрование медицинских данных с использованием открытых ключей и индексированных ключевых слов.
- **Шаг 3**: Поиск медицинских данных, по ключевым словам, выполняемый в зашифрованных данных без их расшифровки.

Алгоритм метода PEKS представлен на рис. 6.

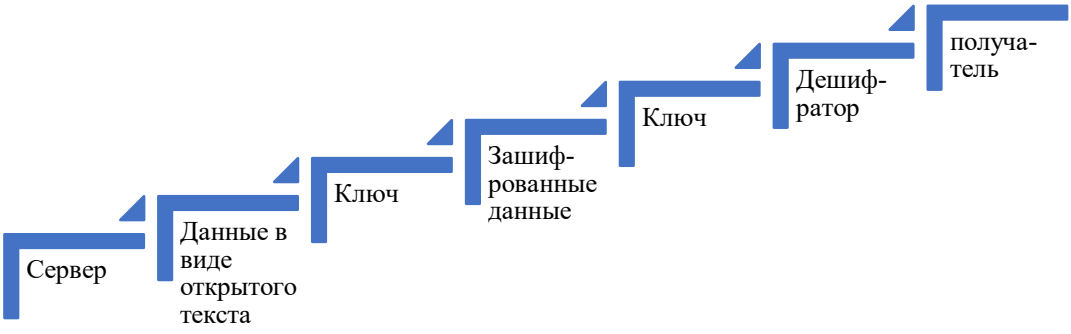


Рис. 6. Алгоритм работы со схемой PEKS.
Fig. 6. PEKS algorithm.

5.3 Прокси-подпись и групповая подпись

Прокси-подпись (Proxy Signature) и групповая подпись (Group Signature) – это два различных метода криптографической подписи, которые могут иметь применение в телемедицине [32].

- **Прокси-подпись** – это механизм, который позволяет одному пользователю (заместителю, прокси) создавать подпись от имени другого пользователя (основного). В контексте телемедицины это может быть полезно, например, когда врач действует от имени пациента для подписания медицинских документов или запросов. Такой механизм может упростить процесс подписания медицинских соглашений и обмена данными, при этом сохраняя право пациента на контроль над своими данными [32].
- **Групповая подпись** – это метод, который позволяет группе пользователей подписывать сообщение так, что внешние наблюдатели могут убедиться в том, что сообщение было подписано представителем группы, но не могут определить, кем именно. В телемедицине это может быть полезно для обеспечения анонимности пациентов или для подписания сообщений от имени медицинских групп или организаций [33].

Рассмотрим применение прокси-подписей и групповых подписей в телемедицине.

- **Подпись медицинских документов.** Врачи или медицинские организации могут использовать прокси-подписи для подписания медицинских отчетов, рецептов и другой медицинской документации от имени пациентов
- **Анонимное обращение пациентов.** Групповые подписи могут использоваться для обеспечения анонимности пациентов при обращении в медицинские учреждения. Это может быть полезно в случаях, когда пациенты хотят обсудить свои медицинские вопросы, не раскрывая свою личность
- **Защита конфиденциальности данных.** Прокси-подписи могут использоваться для защиты медицинских данных пациентов при обмене информацией между медицинскими учреждениями
- **Подписание договоров и соглашений.** Прокси-подписи могут облегчить процесс подписания медицинских соглашений и соглашений о лечении, например, когда пациент не может подписать документы лично

На рис. 7 представлен алгоритм работы Proxy Signature Group Signature (Групповая подпись через доверенного посредника) в телемедицине [33]:

- **Шаг 1**: Группа медицинских работников создает ключи для групповой подписи.
- **Шаг 2**: При создании медицинского отчета или документа, каждый медицинский работник подписывает его групповой подписью.

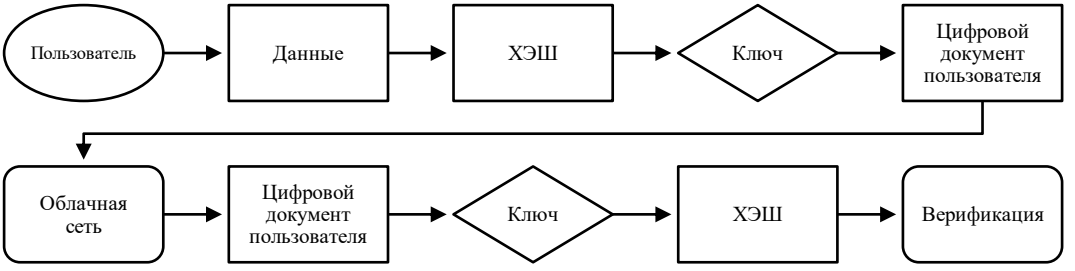


Рис. 7. Механизм групповой прокси-подписи.
Fig. 7. Proxy Signature Group Signature.

- **Шаг 3**: Подписанный документ может быть проверен как подписанный группой, но анонимность отдельных работников сохраняется.

Оба метода подписи предоставляют инструменты для обеспечения конфиденциальности, аутентификации и управления доступом в области телемедицины, однако, учитывая требования к безопасности и конфиденциальности медицинских данных и соблюдение нормативных актов.

5.4 Методы двойных векторных представлений и k-ближайших соседей

Двойные векторные представления (Dual Word Embeddings) и схема k-ближайших соседей (k-Near Neighbors, kNN) – это методы анализа данных и классификации, которые могут быть применены в телемедицине для обработки и классификации медицинских текстов и данных [34].

Алгоритм двойных векторных представлений состоит из следующих шагов [34]:

- **Шаг 1**: Создание двух наборов векторов для каждого слова в медицинском тексте: общих (общих для всех текстов) и специфических (уникальных для данного текста).
- **Шаг 2**: Преобразование медицинского текста в вектор, объединяя оба набора векторов.
- **Шаг 3**: Использование векторов для анализа текста, классификации или извлечения информации.

Схема k-ближайших соседей – это метод классификации объектов на основе их близости к другим объектам в пространстве признаков. В настоящее время он достаточно часто используется в системах нейронных сетях, предполагающих проведение предварительного машинного обучения. В контексте телемедицины схема может быть применена для классификации пациентов, диагнозов или других медицинских данных.

- **Применение метода kNN в телемедицине, диагностика и классификация**: метод kNN может быть использован для диагностики различных состояний или заболеваний на основе медицинских данных и симптомов.
- **Прогнозирование результатов лечения**: kNN может помочь в прогнозировании результатов лечения пациентов на основе данных о предыдущих случаях.

Комбинирование двойных векторных представлений с методом kNN может быть полезным при анализе и классификации медицинских текстов и данных, а также при решении различных задач в области телемедицины. Эти методы помогают улучшить понимание и обработку медицинской информации, что может привести к повышению качества диагностики и ухода за пациентами [34].

Схема k-ближайших соседей работает следующим образом [34]:

- **Шаг 1**: Для каждого пациента создается набор признаков на основе его медицинских данных (например, симптомы, результаты тестов).
- **Шаг 2**: Пациент, для которого требуется классификация (например, диагноз), сравнивается с ближайшими k пациентами на основе их признаков.
- **Шаг 3**: Пациенту присваивается класс или диагноз на основе большинства классов среди k ближайших пациентов.

Сравнительная характеристика рассмотренных методов представлена в табл. 3.

6. Методы анализа защищенности информационных систем

Обеспечение информационной безопасности представляет собой сложный процесс, требующий непрерывного анализа ситуации, который позволяет обнаруживать уязвимости и слабые места в системах безопасности. В [25] рассмотрены методы анализа защищенности информационных систем.

Табл. 3. Сравнительная характеристика методов обеспечения конфиденциальности данных в телемедицине.
Table 3. Comparative characteristics of data privacy methods in telemedicine.

Метод	Описание	Особенности	Преимущества	Недостатки
XACML	Язык управления доступом для определения прав доступа и политик безопасности Декларативный подход	Широко используется в корпоративных информационных системах	• Гибкая система прав доступа • Декларативный подход • Широко применяется в корпоративных системах	• Требует конфигурации и управления политиками • Сложность в больших и сложных системах
Public Encryption with Keyword Search	Технология шифрования данных с возможностью поиска ключевых слов	Обеспечивает конфиденциальность данных и поиск ключевых слов в зашифрованных данных без дешифрации	• Конфиденциальность и поиск в зашифрованных данных. • Подходит для безопасного облачного хранения	• Вычислительная сложность. • Ограниченный поиск ключевых слов
Group Signature	Позволяет создавать подписи, которые могут быть верифицированы как от группы, не раскрывая личность конкретного участника	Защищает анонимность участников группы и используется в приложениях, где требуется подпись от одной из нескольких участников группы	• Анонимность участников группы. • Подпись от группы	• Возможность злоупотребления. • Ограничения в управлении
Dual Word Embeddings, kNN Scheme	Метод представления слов в векторной форме, учитывающий семантику	Улучшает качество анализа текста и задачи обработки естественного языка	• Улучшенный анализ текста. • Семантические ассоциации	• Требует больше вычислительных ресурсов. • Не всегда эффективен на коротких текстах
Proxy Signature	Метод, позволяющий одной стороне подписать сообщение от имени другой стороны	Поддерживает анонимность отправителя, используется для делегирования подписи	• Анонимность отправителя. • Делегирование подписи	• Возможность злоупотребления. • Дополнительная сложность

Один из методов – проведение испытаний на проникновение (penetration test, пентест), которые представляют собой процедуру, заключающуюся в поиске уязвимостей в

информационных системах и использовании этих уязвимостей для проверки защищенности. Тест на проникновение проводится как для выявления "слабых" мест в инфраструктуре и оценки возможных атак, так и для оценки эффективности системы безопасности и получения рекомендаций по повышению ее устойчивости к внешним воздействиям.

Анализ проводится на внутренней и внешней инфраструктуре. При анализе внутренней инфраструктуры специалисты ищут уязвимости в сетевых сервисах и серверах внутри сети компании. При анализе внешних ресурсов проверяются уязвимости веб-сайтов, серверов электронной почты и других сервисов, доступных из глобальной информационной сети.

Еще один метод анализа – анализ беспроводных сетей. Он включает в себя проверку устойчивости внутренней сети к атакам по протоколам на канальном уровне, а также анализ сетевого трафика на наличие критической информации, передаваемой в открытом виде.

Полезным является проведение стресс-тестирования, которое позволяет выявить устойчивость инфраструктуры при повышении нагрузки на нее.

Кроме того, анализ защищенности может быть проведен с использованием методов социальной инженерии, которые проверяют сотрудников на невнимательность и излишнюю доверчивость.

Методы анализа часто называют методами черных, белых или серых ящиков. К методам «черного ящика» относят те методы, при использовании которых заказчик не предоставляет никакой информации о системе, и специалист вынужден собирать ее самостоятельно, моделируя действия злоумышленника. Методами «белого (или прозрачного) ящика» называются методы, применяя которые заказчик предоставляет специалисту всю информацию о системе. Методы «серого ящика» – это компромиссные методы, которые предполагают, что заказчик передает специалистам некоторую, возможно неполную информацию заранее.

7. Информационные системы для решения задач телемедицины

Стандарты обеспечения информационной безопасности и защиты медицинских данных играют критически важную роль в обеспечении конфиденциальности, целостности и доступности медицинской информации особенно с развитием технологий и увеличением объема данных, хранящихся и передаваемых в электронном виде. Рассмотрим основные направления защиты медицинской информации при обеспечении безопасного обмена данными в телемедицинских системах [35-37]:

- Федеральная государственная информационная система в сфере здравоохранения (ЕГИСЗ);
- Медицинская информационная система здравоохранения ("МИС здравоохранения");
- Электронная медицинская карта (ЭМК).

7.1 Информационное пространство ЕГИСЗ

Единая государственная информационная система в сфере здравоохранения (ЕГИСЗ) представляет собой комплексную информационную систему, охватывающую все уровни здравоохранения, с целью совершенствования предоставления медицинских услуг и управления здравоохранением в России [35]. Система имеет три уровня информатизации и три категории пользователей, которые обеспечивают более эффективное и качественное предоставление здравоохранения. На рис. 8 показаны основные категории медицинских организаций РФ.

Рассмотрим основные категории пользователей единого пространства здравоохранения РФ.

- **Граждане/пациенты.** Граждане имеют доступ к удобным средствам взаимодействия с медицинскими организациями, такими как запись на прием и

получение результатов анализов через онлайн-платформы. Они также могут оперативно получать доступ к своей медицинской документации, что повышает уровень информированности и участие в управлении своим здоровьем

- **Врачи/медицинские работники.** Для медицинских работников ЕГИСЗ сокращает необходимость вручную вести отчетность и дает возможность оперативно получать информацию о пациентах.
- **Органы управления.** Органы управления здравоохранением могут оперативно получать агрегированную информацию о состоянии здравоохранения в регионе или стране. Это помогает им разрабатывать более эффективные стратегии и политики, а также быстро реагировать на изменения в здравоохранении

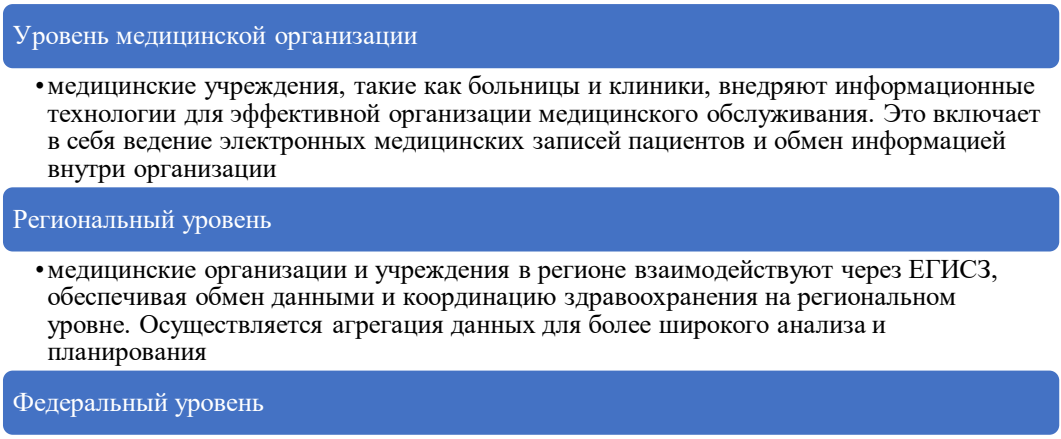


Рис. 8. Основные категории медицинских организаций РФ.
Fig. 8. The main categories of medical organizations in the Russian Federation.

7.2 Медицинская информационная система здравоохранения

Медицинская информационная система (МИС) – это совокупность информационных, организационных, программных и технических средств, предназначенных для автоматизации медицинских процессов [36, 38, 39]. Приведем основные функции и назначение МИС.

- **Повышение качества обслуживания пациентов.** МИС улучшает координацию медицинских процессов, что способствует более точной диагностике и лечению. Это позволяет повысить уровень ухода и комфорта для пациентов
- **Удобный и быстрый доступ к медицинской информации.** МИС позволяет медицинским работникам быстро получать доступ к истории пациента, медицинским данным и другой важной информации, что ускоряет процессы принятия решений
- **Снижение организационных и временных издержек при подготовке отчетов.** МИС автоматизирует процесс создания отчетов, что позволяет сократить время, затрачиваемое на административные задачи, и снизить риски ошибок в документации
- **Сокращение ошибок при составлении медицинских документов.** МИС предотвращает ошибки и упрощает процесс ведения медицинской документации,

что повышает точность и надежность записей о пациентах

- **Организация работы медицинского персонала.** МИС устраняет лишние бумажные процессы, оптимизирует прием и учет пациентов, и упрощает задачи административного характера, что позволяет медицинскому персоналу более эффективно использовать свое время и ресурсы

7.3 Электронные медицинские карты

Электронные медицинские карты (ЭМК) представляют собой электронные версии медицинских карт пациентов [37]. ЭМК могут включать информацию о медицинской истории, данные о здоровье, диагнозы, назначенные лекарства, результаты лабораторных исследований, основные показатели состояния здоровья пациентов, информацию о прививках и выводы из диагностических обследований [40, 41].

Рассмотрим преимущества использования ЭМК.

- **Повышение качества медицинской помощи.** Внедрение ЭМК облегчит врачам обмен информацией о здоровье, что способствует предоставлению медицинской помощи.
- **Безопасное хранение данных.** ЭМК обеспечивают резервное копирование данных, что позволяет восстановить медицинскую информацию о пациентах.
- **Доступ в чрезвычайных ситуациях.** В случае несчастного случая, больницы, работающие с ЭМК, смогут получить неотложные данные о здоровье пациента, что ускорит принятие медицинских решений
- **Эффективное медицинское обслуживание.** Врачи, использующие ЭМК, смогут более быстро отслеживать результаты лабораторных исследований. Обмен информацией между медицинскими учреждениями позволит избежать повторных анализов, что уменьшит расходы на медицинское обслуживание

8. Заключение

Исследование методов проектирования систем безопасности в телемедицинских системах с целью обеспечения конфиденциальности информации показало важность применения соответствующих мер и политик для организации безопасности данных пациентов. С увеличением направлений использования телемедицины возникает необходимость обеспечения высокого уровня защиты информации и предотвращения возможности несанкционированного доступа к ней или ее утечки.

В статье рассмотрены основные понятия в области конфиденциальной информации, на которую в большей степени ориентированы системы обеспечения безопасности при построении телемедицинских систем. Определено, что основными объектами защиты являются персональные данные и данные, содержащие врачебную тайну. Приведены этапы развития технологий в области телемедицины.

На основе приведенных принципов обеспечения информационной безопасности для обеспечения конфиденциальности данных в телемедицинских системах (FAIDM, NFC, блокчейн, RFID) проведена их сравнительная характеристика. Это позволит определить направления использования технологий в телемедицинских системах

Рассмотрены подходы к аутентификации пациентов, используемые в телездравоохранении с учетом их эффективности и безопасности: XACML, Public Encryption with Keyword Search, Group Signature, Dual Word Embeddings, kNN Scheme, Proxy Signature. Проведена их сравнительная характеристика. Выявлены преимущества и недостатки каждого из этих методов**.**

Исследование методов анализа защищенности информационных систем, таких как проведение испытаний на проникновение, методы «черного», «белого» и «серого» ящиков показало их успешную применимость и для телемедицинских систем.

Определены основные направления хранения и обработки медицинской информации при обеспечении безопасного обмена данными в телемедицинских системах, такие как Федеральная государственная информационная система в сфере здравоохранения (ЕГИСЗ), Медицинская информационная система здравоохранения ("МИС здравоохранения"), Электронная медицинская карта (ЭМК), для которых возможно применение рассмотренных выше механизмов обеспечения безопасности конфиденциальной информации.

Таким образом, обеспечение конфиденциальности и безопасности данных является важной частью эффективного функционирования телемедицинских систем. Развитие новых технологий и стандартов обеспечения безопасности данных позволит реализовать эффективную защиту конфиденциальной информации, включая персональные данные и информацию, содержащую врачебную тайну.

Безопасность данных в телемедицине должна быть обеспечена по всем направлениям, включая передачу, хранение и доступ к ней. Несоблюдение требований в сфере обеспечения безопасности в телемедицине может иметь серьезные последствия для всех сторон: пациентов, врачей, медицинские организации и операторов информационных систем, что подчеркивает важность исследования для обеспечения безопасности и эффективности функционирования телемедицинских систем.

Список литературы / References

[1]. Федеральный закон от 21.11.2011 № 323-ФЗ (с изменениями от 24.07.2023) "Об основах охраны здоровья граждан в Российской Федерации" (с изменениями и дополнениями, вступ. В силу с 01.09.2023) Статья 13. Соблюдение врачебной тайны – URL: https://www.consultant.ru/document/cons_doc_LAW_121895/9f906d460f9454a8a0d290738d9fc2798c1e865a/.

[2]. Международный кодекс медицинской этики. Принят 3-й Генеральной Ассамблеей Всемирной медицинской ассоциации, дополнен 22-й Всемирной медицинской ассамблеей 1968 г., 35-й Всемирной медицинской ассамблеей 1983 г., 57-й Всемирной медицинской ассамблеей 2006 г. – URL: https://www.wma.net/policiespost/wma-international-code-of-medical-ethics/.

[3]. Министерство здравоохранения Самарской области: Конфиденциальность: защита и предоставление информации: официальный сайт [https://minzdrav.samregion.ru] – URL: https://minzdrav.samregion.ru/category/vazhnoe/komissiya-po-meditsinskoj-etike/konfidentsialnost-zashhita-i-predostavlenie-informatsii/.

[4]. Конституция Российской Федерации (принята всенародным голосованием 12.12.1993) (с учетом поправок, внесенных Законами РФ о поправках к Конституции РФ от 30.12.2008 № 6-ФКЗ, от 30.12.2008 № 7-ФКЗ, от 05.02.2014 № 2-ФКЗ, от 21.07.2014 № 11-ФКЗ) // СПС «КонсультантПлюс» – URL: https://www.consultant.ru/document/cons_doc_LAW_28399/.

[5]. Прокуратура Ставропольского края: Врачебная тайна и право пациента на информацию о своем здоровье: официальный сайт [https://epp.genproc.gov.ru] – URL: https://epp.genproc.gov.ru/ru/web/proc_26/activity/legal-education/explain?item=59782102 .

[6]. Федеральный закон от 21.11.2011 N 323 ФЗ (ред. от 06.03.2019) «Об основах охраны здоровья граждан в Российской Федерации» // Собрание законодательства РФ, 28. 11. 2011, №48. Ст. 6724. // СПС «КонсультантПлюс» – URL: https://www.consultant.ru/document/cons_doc_LAW_121895/.

[7]. Федеральный закон от 27.07.2006 N 152-ФЗ (ред. от 08.08.2024) «О персональных данных» // СПС «КонсультантПлюс» – URL: https://www.consultant.ru/document/cons_doc_LAW_61801/

[8]. «Кодекс Российской Федерации об административных правонарушениях» от 30.12.2001 N 195-ФЗ (ред. от 14.10.2024) (с изм. и доп., вступ. в силу с 21.10.2024) // СПС «КонсультантПлюс» – URL: https://www.consultant.ru/document/cons_doc_LAW_34661/

[9]. «Гражданский кодекс Российской Федерации» (ГК РФ) от 30 ноября 1994 года N 51-ФЗ // СПС «КонсультантПлюс» – https://www.consultant.ru/document/cons_doc_LAW_5142/

[10]. Günter Burg / Telemedicine and Teledermatology / 1.2 The History of Telemedicine – URL: https://books.google.com.mm/books?id=F2gyQlDgptgC&lpg=PA6&dq=history%20of%20telemedicine&lr&hl=ru&pg=PA6#v=onepage&q&f=false

[11]. Владзимирский А.В. История телемедицины: люди, факты, технологии. Донецк: «Цифровая типография», 2008.

[12]. Леванов В.М., Переведенцев О.В., Сергеев Д.В., Никольский А.В. Нормативное обеспечение телемедицины: 20 лет развития. Журнал телемедицины и электронного здравоохранения. 2017; 3(5):160-170.

[13]. Леванов В.М. От телемедицины до электронного здравоохранения: эволюция терминов. Медицинский альманах. 2012; 2(21):16-19.

[14]. История, анализ состояния и перспектив развития телемедицины: [https://jtelemed.ru/] Российский журнал телемедицины и электронного здравоохранения – URL: https://jtelemed.ru/node/3739.

[15]. Wu He, Justin Zhang, Huanmei Wu / A Unified Health Information System Framework for Connecting Data, People, Devices, and Systems – URL: https://www.researchgate.net/publication/363425208_A_Unified_Health_Information_System_Framework_for_Connecting_Data_People_Devices_and_Systems.

[16]. Поспелова С.И., Сергеев Ю.Д., Павлова Ю.В., Каменская Н.А. Правовой режим применения телемедицинских технологий и внедрения электронного документооборота: современное состояние правового регулирования и перспективы развития. Медицинское право. 2018, №5. с.24- 33.

[17]. SearchInform: информационная безопасность / официальный сайт [https://searchinform.ru/] – URL: https://searchinform.ru/informatsionnaya-bezopasnost/.

[18]. Kaspersky: Кибербезопасность в здравоохранении: где болезнь, где болезнь роста / официальный сайт: [https://www.kaspersky.ru] – URL: https://www.kaspersky.ru/blog/healthcare-safeguarding-data/4474/.

[19]. Tzu Wei Lin / FAIDM for Medical Privacy Protection in 5G Telemedicine Systems – URL: https://www.researchgate.net/publication/348816986_FAIDM_for_Medical_Privacy_Protection_in_5G_Telemedicine_Systems.

[20]. Кучуков, Р. А. Теория и практика государственного регулирования экономических и социальных процессов: учебное пособие для студентов, обучающихся по специальностям: "Финансы и кредит", "Бухгалтер. учет, анализ и аудит", "Мировая экономика", "Налоги и налогообложение" / Р. А. Кучуков; Р. А. Кучуков. – Москва: Гардарики, 2004. – 287 с. – (Disciplinae). – ISBN 5-8297-0201-0. – EDN QQEXOB.

[21]. Abdelkader Laouid, Mohammad Hammoudeh, Kara Mostefa /A MultiKey with Partially Homomorphic Encryption Scheme for Low End Devices Ensuring Data Integrity – URL: https://www.researchgate.net/publication/370392930_A_MultiKey_with_Partially_Homomorphic_Encryption_Scheme_for_LowEnd_Devices_Ensuring_Data_Integrity.

[22]. Yang Xiao; Xuemin Shen; BO Sun; Lin Cai / Security and privacy in RFID and applications in telemedicine: – URL: https://ieeexplore.ieee.org/abstract/document/1632651.

[23]. Encryption Scheme for Low End Devices Ensuring Data Integrity – URL: https://www.researchgate.net/publication/370392930_A_MultiKey_with_Partially_Homomorphic_Encryption_Scheme_for_LowEnd_Devices_Ensuring_Data_Integrity

[24]. Yasir Iqbal, Shahzaib Tahir, Abdullah M. Almuhaideb, Adeel M. Syed / A Novel Homomorphic Approach for Preserving Privacy of Data in Telemedicine – URL: https://www.mdpi.com/1424-8220/22/12/4432.

[25]. Yonghang Tai; Bixuan Gao; Qiong Li; Zhengtao Yu; Chunsheng Zhu; Victor Chang / Trustworthy and Intelligent COVID-19 Diagnostic IoMT Through XR and Deep-Learning-Based Clinic Data Access – URL: https://ieeexplore.ieee.org/abstract/document/9343340.

[26]. Иванов В.В., аутентификация и авторизация / Иванов В. В., Лубова Е. С., Черкасов Д. Ю. Текст :электронный // Компьютерные и информационные науки: https://cyberleninka.ru/ 2017 – URL: https://cyberleninka.ru/article/n/autentifikatsiya-i-avtorizatsiya.

[27]. Утечка медицинских данных: как они происходят и как их предотвратить:[https://www.itsec.ru/] Information security журнал /– URL: https://lib.itsec.ru/articles2/control/utechki-meditsinskih-dannyh-kak-oni-proishodyat-i-kak-ih-predotv.

[28]. Hitesh Gupta / Management Information System: – URL: https://books.google.ru/books?hl=ru&lr=&id=PWRYwOJ8FmgC&oi=fnd&pg=PA1&dq=Description+of+the+procedure+and+regulations+for+personnel+actions+during+the+operation+of+MIS&ots=wvVUGCMb-

0&sig=YFhrCauvEg4skq2_TQTy2hqlHxE&redir_esc=y#v=onepage&q=Description%20of%20the%20procedure%20and%20regulations%20for%20personnel%20actions%20during%20the%20operation%20of%20MIS&f=false

[29]. Carroline Dewi Puspa Kencana Ramli, Hanne Riis Nielson, Flemming / The logic of XACML – URL: https://www.sciencedirect.com/science/article/pii/S0167642313001238.

[30]. Sun-Moon Jo, Kyung-Yong Chung / Design of access control system for telemedicine secure XML documents – URL: https://link.springer.com/article/10.1007/s11042-014-1938-x.

[31]. Kaspersky: Что такое шифрование? / официальный сайт: [https://www.kaspersky.ru] – URL: https://www.kaspersky.ru/resource-center/definitions/encryption (дата обращения 17.10.2024).

[32]. Byoungcheon Lee / Strong Proxy Signature and Applications – URL: https://www.researchgate.net/publication/2410948_Strong_Proxy_Signature_and_its_Applications.

[33]. Yuping Zhou, Yu Hu,1Jianneng Chen, Zengjie Huang, Hui Huang, Yi Fan Zhang / Identity Based Designated-Verifier Proxy Signature Scheme with Information Recovery in Telemedicine System https://www.hindawi.com/journals/wcmc/2022/1580444/

[34]. Robert P. Bostrom and J. Stephen Heinen / MIS Problems and Failures: A Socio-Technical Perspective. Part I: The Causes – URL: https://www.sci-hub.ru/10.2307/248710.

[35]. ЕГИСЗ / официальный сайт: – URL: https://egisz.rosminzdrav.ru/.

[36]. Белышев Д.В., Гулиев Я.И., Михеев А.Е. / Место МИС медицинской организации в методологии информатизации здравоохранения – URL: https://cyberleninka.ru/article/n/mesto-mis-meditsinskoy-organizatsii-v-metodologii-informatizatsii-zdravoohraneniya.

[37]. Зингерман Б.В., Шкловский-Корди Н.Е. / Электронная медицинская карта и принципы ее организации – URL: https://cyberleninka.ru/article/n/elektronnaya-meditsinskaya-karta-i-printsipy-ee-organizatsii.

[38]. Когаленок В.Н., Царева З.Г., Тараканов С.А. / Проблемы внедрения медицинских информационных систем автоматизации здравоохранения. Комплекс программных средств «Система автоматизации медикострахового обслуживания населения» – URL: https://cyberleninka.ru/article/n/problemy-vnedreniya-meditsinskih-informatsionnyh-sistem-avtomatizatsii-uchrezhdeniy-zdravoohraneniya-kompleks-programmnyh-sredstv.

[39]. Кошкаров А. А. / Структурная адаптация федеральных требований к медицинским информационным системам на региональном уровне – URL: https://cyberleninka.ru/article/n/strukturnaya-adaptatsiya-federalnyh-trebovaniy-k-meditsinskim-informatsionnym-sistemam-na-regionalnom-urovne.

[40]. Зарубина Т.В., Швырев С.Л., Соловьев В.Г., Раузина С.Е., Родионов В.С. / Интегрированная медицинская карта: состояние дел и перспективы – URL: https://cyberleninka.ru/article/n/integrirovannaya-elektronnaya-meditsinskaya-karta-sostoyanie-del-i-perspektivy.

[41]. Швырев С.Л. / Опыт разработки электронной медицинской документации на основе архитектуры клинических документов CDA 2. 0 HL7 – URL: https://cyberleninka.ru/article/n/opyt-razrabotki-elektronnoy-meditsinskoy-dokumentatsii-na-osnove-arhitektury-klinicheskih-dokumentov-cda-2-0-hl7.

Информация об авторах / Information about authors

Мария Анатольевна ЛАПИНА – кандидат физико-математических наук, доцент, доцент кафедры информационной безопасности автоматизированных систем Северо-Кавказского федерального университета. Сфера научных интересов: цифровые технологии, киберфизические системы, анализ данных, управление информационной безопасностью, доверенный искусственный интеллект, криптография, анализ программного кода.

Maria Anatolyevna LAPINA – Cand. Sci. (Phys.-Math.), Associate Professor, Associate Professor of the Department of Information Security of Automated Systems of the North Caucasus Federal University. Research interests: digital technologies, cyber-physical systems, data analysis, information security management, trusted artificial intelligence, cryptography, program code analysis.

Елена Александровна МАКСИМОВА – доктор технических наук, доцент, заведующий кафедрой «Интеллектуальные системы информационной безопасности» РТУ МИРЕА –

Российский технологический университет. Сфера научных интересов: цифровые технологии, киберфизические системы, анализ данных, управление информационной безопасностью, криптография, критическая информационная инфраструктура.

Elena Alexandrovna MAKSIMOVA – Dr. Sci. (Tech.), Associate Professor, Head of the Department "Intelligent Information Security Systems" of RTU MIREA – Russian University of Technology. Research interests: digital technologies, cyber-physical systems, data analysis, information security management, cryptography, critical information infrastructure.

Виталий Геннадьевич ЛАПИН – кандидат физико-математических наук, начальник отдела АСУ Ставропольского краевого клинического консультативно-диагностического центра. Сфера научных интересов: цифровые технологии, киберфизические системы, анализ данных, управление информационной безопасностью, анализ программного кода.

Vitaly Gennadevich LAPIN – Cand. Sci. (Phys.-Math.), Head of the Automated Control System Department of the Stavropol Regional Clinical Advisory and Diagnostic Center. Research interests: digital technologies, cyber-physical systems, data analysis, information security management, program code analysis.

Никита Сергеевич БОЙКОВ – студент кафедры информационной безопасности Северо-Кавказского федерального университета. Сфера научных интересов: цифровые технологии, анализ данных, управление информационной безопасностью, анализ программного кода.

Nikita Sergeevich BOIKOV – a student of the Department of Information Security of the North Caucasus Federal University. Research interests: digital technologies, data analysis, information security management, program code analysis.

DOI: 10.15514/ISPRAS-2024-36(5)-15

CC BY

Использование клеточного автомата для оценки влияния городской планировки на социоэкономические показатели при распространении эпидемий

С.А. Елистратов, *ORCID: 0000-0002-7006-6879* <sa.elistratov@ispras.ru>

Институт системного программирования РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.
Институт математики СО РАН,
Россия, 630090, г. Новосибирск, проспект Академика Коптюга, д. 4.

Аннотация. В работе рассматривается влияние городской планировки на социоэкономические факторы, полученные из эпидемиологических показателей путем моделирования распространения эпидемии с помощью клеточного автомата. Показывается, что (при одинаковом соотношении площадей районов с высокой и низкой плотностью) планировка влияет на распространение эпидемии, причем существует оптимальная планировка, которая минимизирует экономические убытки и смертность. Несмотря на модельность подхода, результат свидетельствует о том, что ущерб от пандемии может быть снижен благодаря грамотной градостроительной политике.

Ключевые слова: имитационное моделирование, клеточные автоматы, эпидемиология.

Для цитирования: Елистратов С.А. Использование клеточного автомата для оценки влияния городской планировки на социоэкономические показатели при распространении эпидемий. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 219–226. DOI: 10.15514/ISPRAS–2024–36(5)–15.

Благодарности: Работа выполнена при поддержке Российского Научного фонда (проект 23-71-10068).

Cellular Automaton Approach for the Urban Planning Influence Estmation on the Social and Economic Indicators while a Disease Spreading

S.A. Elistratov, ORCID: 0000-0002-7006-6879 <sa.elistratov@ispras.ru>

Institute for System Programming of RAS,
25, Aleksandra Solzhenitsyna str., Moscow, 109004, Russian Federation.
Institute of Mathematics of Siberian Branch of the RAS,
4, Akademika Koptyuga ave., Novosibirsk, 630090, Russian Federation.

Abstract. The aim was to investigate how does a city configuration influence on a lethal disease spread. For this purpose, several configurations for subareas with high and low population density are considered. The spread was simulated via a stochastic cellular automata approach, with the main indicators being determined as average over a set of runnings. Since the automaton was based on a SIRS model, as an economic indicator we used the simultaneous sick number, as a social – cumulative dead number. In addition, we considered Manshift losses as a cost loss parameter. The simulation results yield that for the minimal dead number and economic losses it is preferable to use a regular grid of square-shaped low-density subareas. Despite the model suggested indicates that the city planning is important for the pandemic damage minimization which can be reduced due to smart urban development policy.

Keywords: imitational modeling, cellular automata, epidemiology.

For citation: Elistratov S.A. Cellular automaton approach for the urban planning influence estimation on the social and economic indicators while a disease spreading. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 5, 2024. pp. 219-226 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-15.

Acknowledgements. The work was supported by Russian Science Foundation (project 23-71-10068).

1. Введение

После пандемии COVID-19 роль исследований распространения заболеваний резко возросла [1]. Появилось большое число работ, как прогностических [2,3], так и использующих математические модели. Среди последних особое место занимают модели игр среднего поля [4-6] для оценки интегральных эпидемиологических показателей, и имитационное моделирование [7-10].

В то же время, исследуется влияние на распространение эпидемии различных факторов, таких как климатические условия [11,12], плотность населения [13,14] и другие. Работы [15,16] указывают на то, что одним из таких факторов является городская планировка. Существуют попытки учесть ландшафт в моделировании распространения эпидемий [17].

Широко распространенным методом имитационного моделирования является использование клеточных автоматов, которые применяются как в эпидемиологии [18-20], так и в экономике [21] и при проведении городского планирования [22].

В нашей работе мы предлагаем использовать имитационную модель клеточного автомата, который совмещает вышеуказанные подходы. В задачах эпидемиологии обычно рассматривается прямоугольная одномерная область с равномерной плотностью агентов, которая не учитывает топографию. Учесть влияние городской планировки можно с помощью движущихся агентов, однако такое моделирование будет излишне сложно, поскольку основное время расчета будет уходить на перемещение агентов, так как скорость распространения заболевания много меньше скорости движения человека. Подобное описание более подходит для описания поведения больших масс людей [23,24], и несмотря на имеющийся опыт [25] рассмотрения стохастических перемещения подвижных агентов применительно к возможным очагам распространения инфекций – в плане эпидемиологии приведенный подход в большей степени применим к высокоактивным патогенам или

биологическому оружию и позволяет учитывать лишь факт заражения, исключая возможность рассмотрения дальнейшего протекания болезни вплоть до выздоровления.

В связи с этим предлагается модель статических агентов, сосредоточенных в своей резидентарной зоне. При введении ограничений на перемещение во время эпидемии (как было в случае пандемии COVID-19), на характерных временных масштабах распространения болезни людей можно считать неподвижными. Учесть городскую планировку в таком случае можно с помощью неравномерной плотности агентов. В такой модели удобно подсчитывать основные эпидемиологические показатели.

Вместе с этим, стоит отметить важность влияния эпидемии на социальную и экономическую сферу [26,27]. Несмотря на простоту предлагаемой модели, с ее помощью оказывается возможным косвенно оценить влияние городской планировки на социоэкономические факторы, связанные с распространением эпидемий.

2. Математическая модель

В качестве модели используется клеточный автомат с четырьмя состояниями: агент без иммунитета, зараженный агент, иммунизированный агент, мертвый агент / клетка без агента. Агенты находятся в квадратном пространстве, конфигурация планировки регулируется с помощью показателя плотности населения. Плотность в пространстве агентов измеряется как среднее количество живых агентов в области к полному числу клеток в этой области и является нормированной на единицу.

Инициализация автомата происходит следующим образом. Для каждой клетки генерируется случайное число из интервала [0,1], и если оно оказывается меньше заданного уровня плотности населения для текущей клетки, в нее сажается неиммунизированный агент. В качестве зараженного агента выбиралась одна клетка (центральная или юго-западный угол). Кроме того, для каждого агента сохраняется время (число итераций) τ, в течение которого его состояние оставалось неизменным.

Далее автомат функционирует по следующим правилам (коэффициенты и времена задержки взяты из работ [28,29,30,31,32,33]):

1) пустая клетка / мертвый агент не меняют состояние;
2) неиммунизированный агент может стать зараженным с вероятностью 0.274 при $\tau>5.5$;
3) зараженный агент может стать иммунизированным с вероятность 0.29 при $\tau>10$;
4) зараженный агент может умереть с вероятностью 0.062 при $\tau>14$;
5) иммунизированный агент может утратить иммунитет с вероятностью 0.5 при $\tau>90$.

Поскольку такая модель не учитывает транспортные потоки, имеет смысл рассматривать ее для моделирования локального распространения заболевания в области постоянного нахождения людей, например, в жилой зоне. Поскольку изучается быстро распространяющаяся эпидемия с возможным летальным исходом, логично включить в рассмотрение ограничительные меры по передвижению населения, как с рабочими целями, так и с целью досуга. Это означает, что люди не посещают парки, но при этом вынуждены ходить в магазин или аптеку. Поскольку мы не разделяем агентов по социовозрастным группам, будем считать, что рассматриваются только взрослые трудоспособные агенты.

В такой постановке рассматривается несколько конфигураций планировок (рис. 1). Для всех конфигураций, изображенных на рис. 1, соотношение площадей областей с высокой и низкой плотностью одинаково и составляет 3:1. Равное соотношение областей обеспечивает одинаковое количество населения для возможности сравнения. Конфигурации *(a)-(d)* будут использованы для определения, какое расположение зон пониженной плотности является оптимальным. Поскольку мы считаем принятыми ограничительные меры, можно рассматривать зоны пониженной плотности как парки. Конфигурация *(a)* соответствует одному большому парку (такому, как Центральный парк в Нью-Йорке), конфигурация *(d)* –

множеству регулярно расположенным небольших парков. Конфигурации *(b)* и *(e)* используются для определения наилучшей формы областей пониженной плотности. Схемы *(c)* и *(f)* сравнивают регулярное расположение (в виде прямоугольной сетки) и радиальное – первое характерно для планово застраиваемых городов (Санкт-Петербург, Новосибирск), второе – для более древних (таких как Москва).

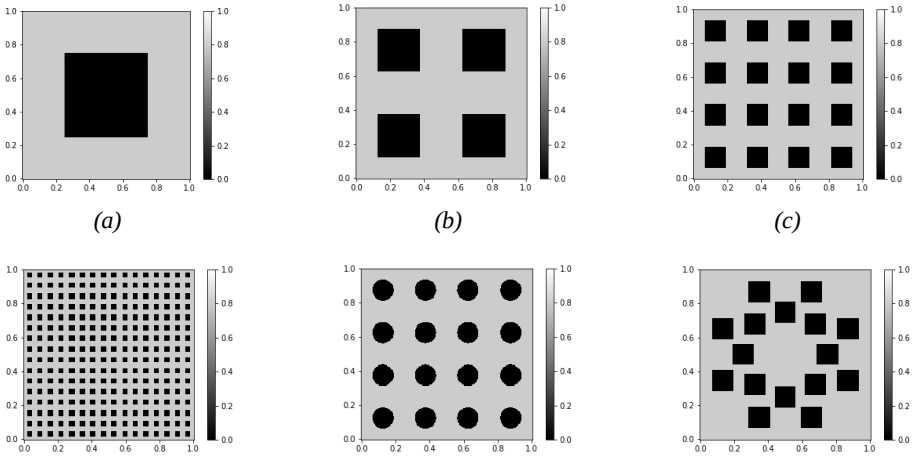


Рис. 1. Конфигурации городской планировки.
Fig. 1. Urban planning configurations.

Поскольку рассматриваемая система является имитационной SIRS-моделью (исходя из правил клеточного автомата 1-5), в ней возможен подсчет числа зараженных, выздоровевших и умерших агентов. Исходя из этих данных, в качестве социального фактора предлагается рассматривать кумулятивную смертность D (то есть полное число умерших агентов к определенному моменту времени); в качестве экономических показателей будем рассматривать число одновременно больных людей как I-R-D, где I – кумулятивное число инфицированных, R – кумулятивное число выздоровевших. Мы рассматриваем именно этот показатель, поскольку это число людей, у которых открыт больничный и которые не работают. Вторым экономическим показателем в такой модели будет выступать величина экономических потерь в человеко-днях, то есть $\Sigma(I\text{-}R)$.

3. Результаты моделирования

Поскольку рассматриваемый автомат стохастический, результат имеет смысл осреднять. На рис. 2 приведены эпидемиологические показатели для случая планировки *(c)*. Видно, что дисперсионный коридор довольно узок по сравнению со средними значениями. Здесь же заметим, что использование клеточного автомата является принципиальным, поскольку результаты расчета не аппроксимируются логистической кривой (при построении последней фиксировалось начальное значение (P_0) и асимптотика на бесконечности (K), а параметр r варьировался). Большая длительность переходного процесса по сравнению с моделью Ферхюльста можно объяснить именно учетом геометрии, что подчеркивает важность учета планировки.

Из рис. 3 видно, что измельчение зон пониженной плотности не оказывает существенного влияния на количество больных, изменяя лишь форму кривой. То же относится к кумулятивной смертности, что позволяет расценивать конфигурации *(a)*-*(d)* (рис. 1) в нашей модели как соизмеримые.

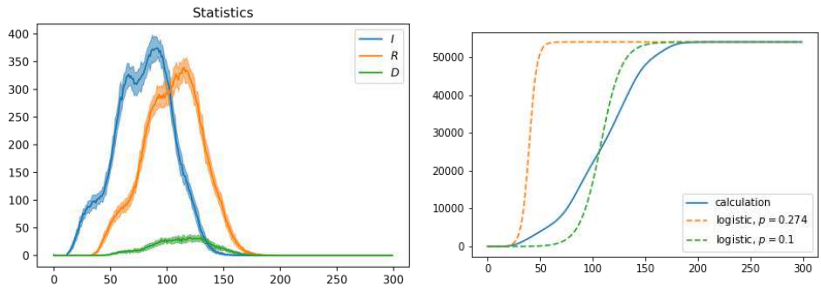


Рис. 2. Осредненные эпидемиологические показатели с коридорами дисперсий для конфигурации c) (справа) и сравнение кумулятивного числа заражений с логистической кривой (слева).
Fig. 2. Averaged epidemiological indicators with dispersion for 4x4 square parks grid c) (LHP) and comparison of simulated cumulative infected and logistic curves (RHP).

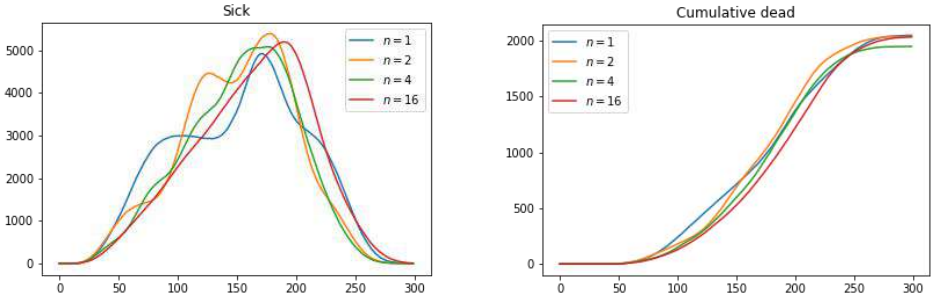


Рис. 3. Влияние конфигурация a) – d), источник в юго-западном углу.
Fig. 3. Regular grid configuration influence, source at the south-west corner.

В отличие от количества, форма областей разреженности при той же площади влияет на социоэкономические показатели, увеличивая как максимальное число единовременно больных, так и кумулятивную смертность (рис. 4), причем квадратная форма предпочтительнее круглой. В то же время, использование радиального расположения вместо прямоугольной сетки оказывает лишь незначительное влияние на эти показатели (рис. 5).

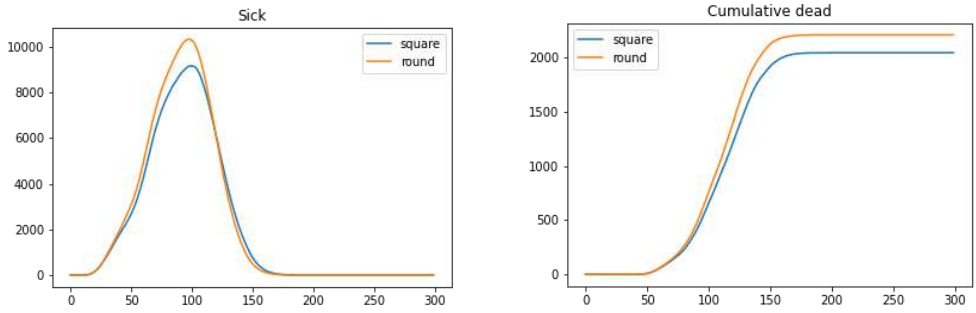


Рис. 4. Влияние формы областей пониженной плотности населения, источник в центре.
Fig. 4. Low-density areas shape influence, source at the center.

Графики экономических убытков, приведенные на рис. 6, подтверждают предыдущие результаты и свидетельствуют о том, что на этот показатель влияет только форма областей пониженной плотности. Заметим, что линейный наклон графиков в конце возник из-за того, что умершие агенты навсегда выбыли из экономического процесса, при этом убытки рассчитываются относительно того уровня, когда все агенты, присутствовавшие в системе на

момент начала пандемии, находятся на рабочих местах. Отметим, что на среднем графике рис. 6 приведены дисперсионные коридоры для исключения возможности случайного расхождения кривых, однако их ширины очень малы.

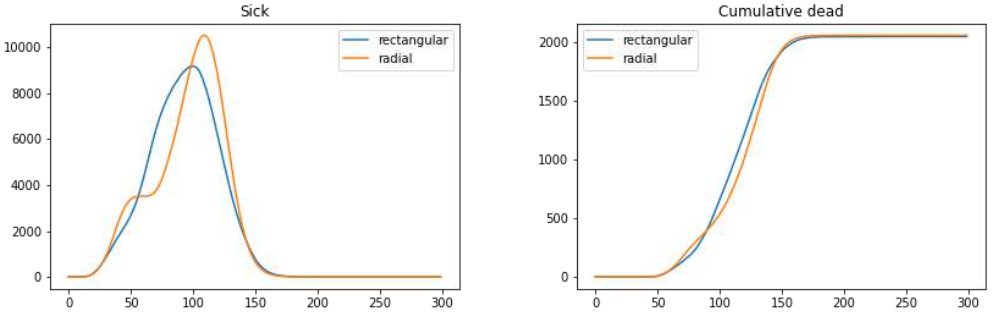


Рис. 5. Влияние расположения областей пониженной плотности населения, источник в центре.
Fig. 5. Low-density areas location influence, source at the center.

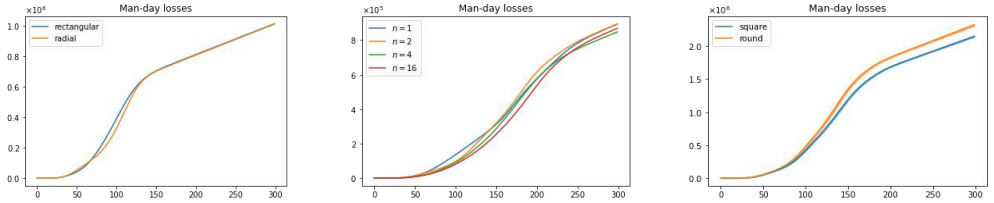


Рис. 6. Влияние на экономические убытки (слева направо соответственно) количества, формы и расположения областей с пониженной плотностью.
Fig. 6. Influence on the man-shift losses of (respectively left to right) number, hape and position of low-density areas.

4. Заключение

В работе произведено моделирование распространение заболеваний с помощью клеточного автомата на основе SIRS модели с учетом специфики городской планировки. Рассматривается влияние различных аспектов планировки на социоэкономические показатели: кумулятивную смертность, число больных и экономические убытки. Показано, что расположение областей пониженной плотности в городах не влияет на эти показатели, однако влияет их форма. Несмотря на модельный характер задачи, результаты позволяют сделать вывод о том, что распространение эпидемии связано с городской планировкой, а убытки, вызванные пандемией, могут быть снижены благодаря грамотной градостроительной политике.

Список литературы / References

[1]. Wynants L. Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal / L. Wynants, B. Van Calster, G.S. Collins, R.D. Riley // BMJ. – 2020. – 369. DOI: 10.1136/bmj.m1328.

[2]. M. Riaz, M. Hussain Sial, S. Sharif, Q. Mehmood. Epidemiological Forecasting Models Using ARIMA, SARIMA, and Holt–Winter Multiplicative Approach for Pakistan // Journal of Environmental and Public Health, 2023, 8907610, 8 pages, 2023. https://doi.org/10.1155/2023/8907610.

[3]. A. Bakhta, T. Boiveau, Y. Maday, O. Mula, Epidemiological Forecasting with Model Reduction of Compartmental Models. Application to the COVID-19 Pandemic // Biology (2021), 10, 22. https://doi.org/10.3390/biology10010022.

[4]. V. Petrakova, O. Krivorotko. Mean field game for modeling of COVID-19 spread // Journal of Mathematical Analysis and Applications, 514(1), 2022, https://doi.org/10.1016/j.jmaa.2022.126271.
[5]. Doncel, Josu, Nicolas Gast, and Bruno Gaujal. A mean field game analysis of SIR dynamics with vaccination. // Probability in the Engineering and Informational Sciences 36(2): 482–99, 2022. doi: 10.1017/S0269964820000522.
[6]. Ghilli, D., Ricci, C. & Zanco, G. A mean field game model for COVID-19 with human capital accumulation. *Econ Theory* **77**, 533–560 (2024). https://doi.org/10.1007/s00199-023-01505-0.
[7]. P. Patlolla, V. Gunupudi, A. Mikler, Armin and R.T. Jacob (2004). Agent-Based Simulation Tools in Computational Epidemiology. 212-223. 10.1007/11553762_21.
[8]. Perez L, Dragicevic S. An agent-based approach for modeling dynamics of contagious disease spread. Int J Health Geogr. 2009 Aug 5;8:50. doi: 10.1186/1476-072X-8-50.
[9]. Tracy M, Cerdá M, Keyes KM. Agent-Based Modeling in Public Health: Current Applications and Future Directions. Annu Rev Public Health. 2018 Apr 1;39:77-94. doi: 10.1146/annurev-publhealth-040617-014317.
[10]. Bissett, K.R., Cadena, J., Khan, M. *et al.* Agent-Based Computational Epidemiological Modeling. *J Indian Inst Sci* **101**, 303–327 (2021). https://doi.org/10.1007/s41745-021-00260-2.
[11]. Anwar A, Anwar S, Ayub M, Nawaz F, Hyder S, Khan N, Malik I. Climate Change and Infectious Diseases: Evidence from Highly Vulnerable Countries. Iran J Public Health. 2019 Dec;48(12):2187-2195. PMID: 31993386; PMCID: PMC6974868.
[12]. Mecenas P, Bastos RTDRM, Vallinoto ACR, Normando D. Effects of temperature and humidity on the spread of COVID-19: A systematic review. PLoS One. 2020 Sep 18;15(9):e0238339. doi: 10.1371/journal.pone.0238339.
[13]. Ganasegeran K, Jamil MFA, Ch'ng ASH, Looi I, Peariasamy KM. Influence of Population Density for COVID-19 Spread in Malaysia: An Ecological Study. Int J Environ Res Public Health. 2021 Sep 18;18(18):9866.
[14]. Hazarie, S., Soriano-Paños, D., Arenas, A. *et al.* Interplay between population density and mobility in determining the spread of epidemics in cities. *Commun Phys* **4**, 191 (2021). https://doi.org/10.1038/s42005-021-00679-0.
[15]. Montoya ID. Topography as a contextual variable in infectious disease transmission. Clin Lab Sci. 2004 Spring;17(2):95-101.
[16]. Henstell, Samantha (2019). "The Role of Urban Planning in the Spread of Communicable Diseases," Agora Journal of Urban Planning and Design, 26-32.
[17]. Arifin, S.M.N.; Arifin, R.R.; Pitts, D.D.A.; Rahman, M.S.; Nowreen, S.; Madey, G.R.; Collins, F.H. Landscape Epidemiology Modeling Using an Agent-Based Model and a Geographic Information System. *Land* **2015**, *4*, 378-412. https://doi.org/10.3390/land4020378
[18]. Fuentes M.A. Cellular automata and epidemiological models with spatial dependence / M.A. Fuentes, M.N. Kuperman // Physica A: Statistical Mechanics and its Applications. – 1999. – 267(3). – p. 471-486. – URL: https://www.sciencedirect.com/science/article/pii/S0378437199000278 DOI: 10.1016/S0378-4371(99)00027-8.
[19]. White S.H. Modeling epidemics using cellular automata / S.H. White, A.M. Del Rey, G.R. Sanchez // Appl Math Comput. – 2006. – 186(1). – p. 193-202. DOI: 10.1016/j.amc.2006.06.126.
[20]. Krivorotko O. Agent-based modeling of COVID-19 outbreaks for New York state and UK: Parameter identification algorithm / O. Krivorotko, M. Sosnovskaia, I. Vashchenko, C. Kerr // Infect Dis Model. – 2024. – 7(1). – p. 30-44. DOI: 10.1016/j.idm.2021.11.004.
[21]. W. Song, Z. Yunlin, X. Zhenggang, Y. Guiyan, H. Tian, Huang and M. Nan. Landscape pattern and economic factors' effect on prediction accuracy of cellular automata-Markov chain model on county scale // Open Geosciences, vol. 12, no. 1, 2020, pp. 626-636. https://doi.org/10.1515/geo-2020-0162.
[22]. Yeh, A.G.O., Li, X., Xia, C. (2021). Cellular Automata Modeling for Urban and Regional Planning. In: Shi, W., Goodchild, M.F., Batty, M., Kwan, MP., Zhang, A. (eds) Urban Informatics. The Urban Book Series. Springer, Singapore. https://doi.org/10.1007/978-981-15-8983-6_45.
[23]. Гребенников Р.В. (2011). Моделирование поведения толпы с использованием локальных скалярных полей (диссертация на соискание степени к.т.н., Воронеж, 2011).
[24]. М. Е. Степанцов, Математическая модель направленного движения группы людей. Матем. Моделирование, 2004, том 16, номер 3, 43–49
[25]. И. В. Деревич, А. А. Панова, Стохастическая модель движения группы индивидов в ограниченном пространстве с учетом их социального поведения. Матем. моделирование, 2023, том 35, номер 6, 51–62.

[26]. Managi S, Chen Z. Social-economic impacts of epidemic diseases. Technol Forecast Soc Change. 2022 Feb;175:121316. doi: 10.1016/j.techfore.2021.121316.
[27]. Nicola M, Alsafi Z, Sohrabi C, Kerwan A, Al-Jabir A, Iosifidis C, Agha M, Agha R. The socio-economic implications of the coronavirus pandemic (COVID-19): A review. Int J Surg. 2020 Jun;78:185-193. doi: 10.1016/j.ijsu.2020.04.018.
[28]. Samui P, Mondal J, Khajanchi S. A mathematical model for COVID-19 transmission dynamics with a case study of India. Chaos Solitons Fractals. 2020 Nov;140:110173. doi: 10.1016/j.chaos.2020.110173
[29]. Wu Y, Kang L, Guo Z, Liu J, Liu M, Liang W. Incubation Period of COVID-19 Caused by Unique SARS-CoV-2 Strains: A Systematic Review and Meta-analysis. JAMA Netw Open. 2022 Aug 1;5(8):e2228008.
[30]. Ma T, Ding S, Huang R, Wang H, Wang J, Liu J, Wang J, Li J, Wu C, Fan H, Zhou N. The latent period of coronavirus disease 2019 with SARS-CoV-2 B.1.617.2 Delta variant of concern in the postvaccination era. Immun Inflamm Dis. 2022 Jul;10(7):e664.
[31]. Park M, Pawliuk C, Nguyen T, Griffitt A, Dix-Cooper L, Fourik N, Dawes M. Determining the communicable period of SARS-CoV-2: A rapid review of the literature, March to September 2020. Euro Surveill. 2021 Apr;26(14):2001506. doi: 10.2807/1560-7917.ES.2021.26.14.2001506.
[32]. https://www.publichealthontario.ca/-/media/documents/ncov/covid-wwksf/2021/03/wwksf-period-of-communicability-overview.pdf?la=en.
[33]. Baud D, Qi X, Nielsen-Saines K, Musso D, Pomar L, Favre G. Real estimates of mortality following COVID-19 infection. Lancet Infect Dis. 2020 Jul;20(7):773. doi: 10.1016/S1473-3099(20)30195-X.

Информация об авторах / Information about authors

Степан Алексеевич ЕЛИСТРАТОВ – Сотрудник Лаборатории цифрового моделирования технических систем Института системного программирования с 2021 года. Сфера научных интересов: математическое моделирование, прогностические модели, методы пониженной размерности.

Stepan Alekseevich ELISTRATOV– Employee of Technical Systems Digital Modeling Laboratory of the Institute for System Programming of the RAS since 2021. Research interests: numerical simulation, forecast models, reduced order methods.

DOI: 10.15514/ISPRAS-2024-36(5)-16

CC BY

H1: гибридная система извлечения информации для поиска товаров в электронной торговле

[1] *Ф.В. Краснов, ORCID: 0000-0002-9881-7371 <krasnov.fedor2@wb.ru>*

[1] *Исследовательский центр ООО "ВБ СК" на базе Инновационного Центра Сколково Россия, 121205, Москва, ул. Нобеля, д. 5.*

Аннотация. В данной статье обоснована продуктивность использования системы H1 для поиска товаров различных поставщиков на торговой интернет-площадке. Как и все современные системы поиска товаров, гибридная система H1 соединяет в себе преимущества лексических методов извлечения товаров и семантических методов, основанных на многомерных векторных представлениях. Новизна предложенного подхода заключается в объединении методов извлечения на уровне токенов. Дополнительное преимущество H1, по сравнению с другими индустриальными системами, – обработка поисковых запросов, состоящих из нескольких слов. Например, поисковые запросы «конфеты рот фронт», «gloria jeans детская одежда» будут выделять сущность бренда в отдельный токен – «рот фронт», «gloria jeans», что позволит уменьшить размер модели и улучшить автономные показатели системы извлечения. Полученные на публичном наборе данных WANDS значения показателей усредненной пороговой точности mAP@12 = 56.1% и пороговой полноты R@1k = 86.6% для H1 превышают самые современные аналоги.

Ключевые слова: информационный поиск по товарам; распознавание бизнес-терминов; суб-словарная токенизация; трансформеры; электронная коммерция.

Для цитирования: Краснов Ф.В. H1: гибридная система извлечения для поиска товаров в e-commerce. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 227–240. DOI: 10.15514/ISPRAS–2024–36(5)–16.

H1: Hybrid Retrieval System for Product Search in E-Commerce

[1] *F.V. Krasnov,* ORCID: 0000-0002-9881-7371 <krasnov.fedor2@wb.ru>

[1] *Research Center of WB SK LLC based on the Skolkovo Innovation Center, 5, Nobel b., Moscow, 121205, Russia.*

Abstract. This paper presents the effectiveness of using the H1 system for retrieving products from various vendors in the marketplace. The H1 system is a hybrid model that combines the benefits of lexical-based and semantic-based retrieval techniques, similar to other state-of-the-art product retrieval systems. The novelty of this approach lies in its combination of token-level retrievals. The advantage of the H1 system over other existing solutions is its ability to handle complex search queries containing brands with multi-word brands. For example, search queries like "new balance sneakers" and "Gloria Jeans children's clothing" will be split into separate tokens "new balance" and "Gloria Jeans", respectively, which helps reduce the retrieval model's size and improves its autonomous performance. The H1 system achieved mAP@12 score of 56.1% and R@1K score of 86.6% on the public WANDS dataset, outperforming other state-of-the-art models. These results

demonstrate the effectiveness of the approach and its potential for improving product search experiences for online shoppers.

Keywords: product information search; entity recognition; Sentence Piece; transformers; ColBERT; e-commerce.

For citation: Krasnov F.V. H1: hybrid retrieval system for product search in e-commerce. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024. pp. 227-240 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-16.

1. Введение

Каталог товаров интернет-площадки включает миллиарды товарных позиций, для его сканирования система поиска товаров должна отвечать критериям скорости, точности и полноты. Типичные средства поиска товаров на первом этапе используют модель поиска по набору слов, которая вычисляет оценку релевантности на основе эвристики, определенной для точного совпадения слов между поисковыми запросами и текстовыми представлениями товаров. Такие лексические модели, как BM25 [1], не устаревают на протяжении десятилетий и по-прежнему широко используются сегодня. Нейросетевые методы извлечения позволяют улучшить показатели эффективности поиска, но обладают своими недостатками [2-5]. Поэтому естественно желание исследователей объединить всё лучшее от лексических и нейросетевых методов в гибридную модель. Недостатки лексических моделей извлечения хорошо изучены: (E1) возможное несоответствие словаря запроса и документа [6-7] приводит к деградации полноты поиска; (E2) отсутствие смыслового понимания документа и запроса [8] уменьшает точность поиска. Из-за ограничений лексические модели не справляются с извлечением релевантных документов в информационном поиске. В течение последних десятилетий предпринимались постоянные попытки решения данных проблем путем расширения запросов [9-12], расширения документа [13-15], создания моделей зависимости токенов [16-17], тематических моделей [18-19], моделей машинного перевода для информационного поиска [20-21]. Однако прогресс в исследованиях лексических моделей извлечения относительно медленный, поскольку большинство из этих подходов все еще находится в рамках парадигмы дискретного, разряженного лексического представления и неизбежно наследует ее ограничения.

Наряду с развитием репрезентативных методов обучения в информационном поиске наблюдается рост исследовательского интереса к моделям семантического поиска на этапе извлечения информации о документах. Начиная с 2013 года развитие многомерных векторных представлений слов [22-24] стимулирует появление большого объема работ по использованию многомерных векторных представлений слов для поиска на этапе извлечения [25-27]. В отличие от дискретного лексического представления, многомерное векторное представление слов являются непрерывным представлением, способным несколько облегчить проблему несоответствия словаря запроса и документа. После 2016 года наблюдается всплеск исследовательского интереса к применению методов глубокого машинного обучения для поиска на этапе извлечения [28-29]. Эти подходы изучены для улучшения представления документов в рамках традиционной парадигмы дискретного лексического представления [30-32] и непосредственно для формирования новых моделей семантического поиска в рамках парадигмы разреженного / плотного представления [33-36].

Отличия задач поиска по товарам от задач поиска по документам заключаются в следующем:

1) При извлечении информации о товарах иначе работают механизмы ранжирования на основании весов текстовых признаков (TF/IDF, BM25). Частота токена в названии товара не делает товар более релевантным запросу.
2) Товары обладают несколькими модальностями: название, описание, характеристики, изображение, видео. Поиск может производиться с учетом нескольких модальностей.

3) Поисковые запросы мотивированы интересом к товару и желанием его купить. Поведение покупателей отличается от поведения пользователя при поиске документа или интернет-ресурса.
4) Оценка эффективности поиска производится по модальностям.

Основной исследовательский вопрос – как влияет архитектура модели извлечения в части функции потерь и токенизации на автономные показатели системы извлечения товаров.

Далее исследовательский вопрос рассмотрен с точки зрения теории и практики, в заключении представлены основные выводы.

2. Методика

На рис. 1 изображено место системы извлечения товаров в системе поиска и отмечена граница взаимодействия с прямым и обратным индексами при онлайн-обработке поискового запроса.

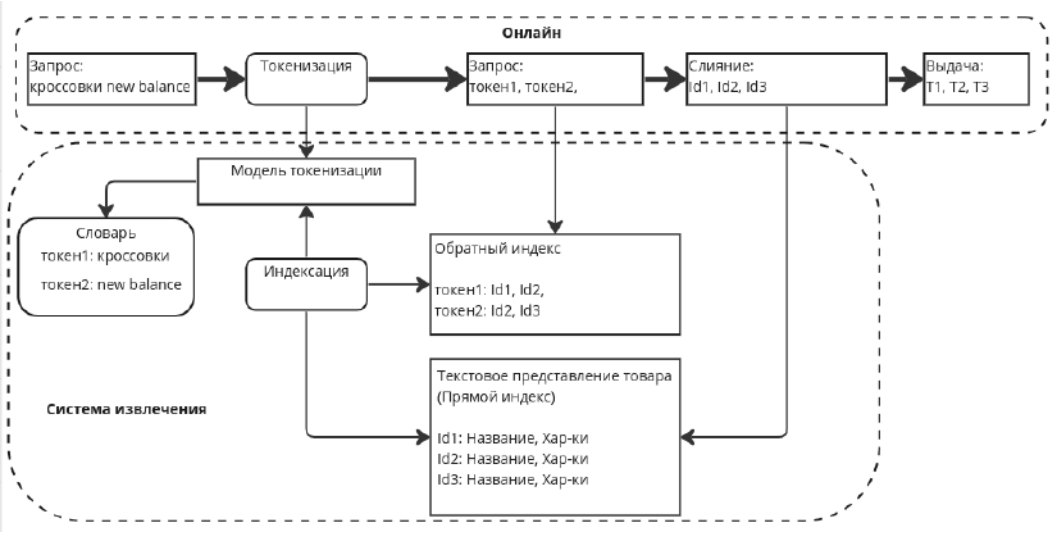


Рис. 1. Место системы извлечения информации о товарах в системе поиска.
Fig. 1. The place of the retireval system in the product search framework.

Развитие систем поиска товаров вместе с информационным поиском движется от лексических методов извлечения в сторону нейросетевых [37-39]. Одна из наиболее востребованных нейросетевых архитектур – DSSM [40]. Оригинальная архитектура DSSM получила развитие в виде парадигмы Dual Encoder [41-43] из-за двух независимых «башен», кодеров для поискового запроса и текстового представления товара. В Dual Encoder запросы и текстовое представление товара кодируются отдельно в общее многомерное векторное пространство фиксированного размера. Затем используется приближенный поиск [44-45] для извлечения релевантных товаров по многомерному векторному представлению поискового запроса. Наиболее перспективными стали исследования по применению моделей BERT в составе архитектуры Dual Encoder [46-48]. Общий принцип этих архитектур представлен в формулах 1–3.

$$\vec{q} = AvgPool\left[BERT_{\theta}^{l}(q)\right] \quad (1)$$

$$\vec{p} = AvgPool[BERT_{\theta}^{r}(p)] \quad (2)$$

$$s_{BERT}(\vec{q},\vec{p}) = \vec{q}^{T} \cdot \vec{p} \quad (3)$$

где $BERT_\theta^l$ и $BERT_\theta^r$ – это «левый» и «правый» кодеры, которые преобразуют тексты q и p в единое векторное пространство θ. Функция оценки соответствия $s_{BERT}(\cdot,\cdot)$ реализована через скалярное произведение $\vec{q}$ и $\vec{p}$. Узким местом данной архитектуры является усреднение векторов токенов по фрагменту текста.

Модель ColBERT [35] представляет частный случай Dual Encoder, который называют Single Encoder, так как используется один кодер и для запросов, и для товаров. Но основное новшество ColBERT состоит в том, что не используется усреднение по токенам AvgPool. Таким образом, на выходе BERT получаются векторы для каждого токена запроса и текстового представления продукта. Если q состоит из m токенов, а p состоит из n токенов, то функция оценки соответствия $s_{ColBERT}(\cdot,\cdot)$ будет следующая:

$$s_{ColBERT}(q_{1:m}, p_{1:n}) = \sum_1^m max_{1..n}\,(\vec{q}_{1:m}^T \cdot \vec{p}_{1:n}) \quad (4)$$

Суммирование в формуле (4) требует вычисления □ ∗ □ скалярных произведений по сравнению с одним произведением в $s_{BERT}(\cdot,\cdot)$.

Гибридизацию определяют как смешивание лексического и нейросетевого подходов к извлечению товаров, которое может быть произведено на разных уровнях. Например, как в исследовании [39], в выдаче смешаны результаты извлечения раздельными моделями, основанными на лексических, поведенческих и семантических методах. В исследовании [49] использован другой принцип создания гибридной модели: основной метод извлечения – лексический, и на ошибках лексической модели обучена семантическая модель.

Значительный рост автономных показателей моделей для работы с текстом достигнут за счет прогресса в методах токенизации [50-52]. Метод BPE (Byte-Pair-Encoding) изначально был представлен в литературе по сжатию данных [53]. BPE – это вариант словарного кодера, который постепенно находит набор символов таким образом, что общее количество символов для кодирования текста сводится к минимуму. Другая модель токенизации, называемая unigram, является энтропийным кодером, который минимизирует общую длину кода для текста. Согласно теореме Шеннона о кодировании, оптимальная длина кода для символа s равна – $log(p_s)$, где p_s – вероятность появления s. Фактически это то же самое, что стратегия сегментации языковой модели unigram, описанная в виде алгоритма Витерби [54]. BPE и unigram разделяют одну и ту же идею – кодирование текста с определенным принципом сжатия данных при использовании меньшего количества битов. Поэтому использование языковой модели unigram имеет те же преимущества, что и BPE. Для поиска по товарам имеют значение не токенизируемые на субсловарные единицы сущности, к примеру, бренды. Пользователь, ищущий товары бренда «Синий лён», не заинтересован в товарах содержащих токены «синий» и «лён» по отдельности.

Поясним разницу между задачами открытия новых брендов как сущностей и задачи использования существующих брендов для улучшения качества поиска. Задача открытия брендов объединяет несколько источников, таких как журналы пользовательских поисковых запросов, информацию о брендах из карточек товаров и реестры торговых марок для экспертной оценки является или нет последовательность слов брендом. Автоматизация процесс открытия брендов может быть выполнена с помощью моделей машинного обучения, но не является предметом данного исследования. В настоящем исследовании подразумевается, что мы имеем справочник брендов и ищем способы использовать его с максимальной пользой для поисковой системы.

С учетом вышеизложенного, в H1 реализован и опробован экспериментально гибридный подход, сочетающий разделения текста на словарные n-граммы различного размера без учета границ слов по пробелам. В качестве наиболее информативных токенов состоящих из нескольких слов, разделенных пробелами, для эксперимента взяты бренды, состоящие из нескольких слов (рис.2).

В H1 полученные векторы токенов не усредняются для получения близости, а по аналогии с ColBERT учитываются отдельно, как это показано на рис. 3.

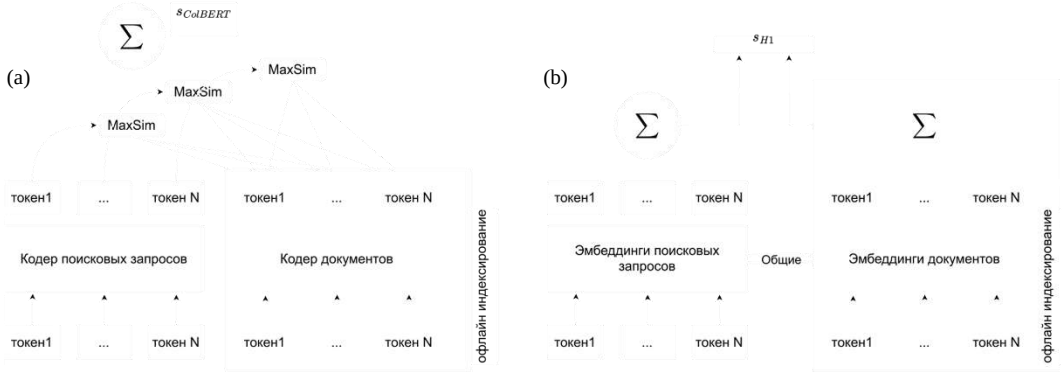


Рис. 2. (a). Общая схема работы ColBERT. (b). Общая схема работы H1.
Fig. 2. (a). ColBERT model. (b). H1 model.

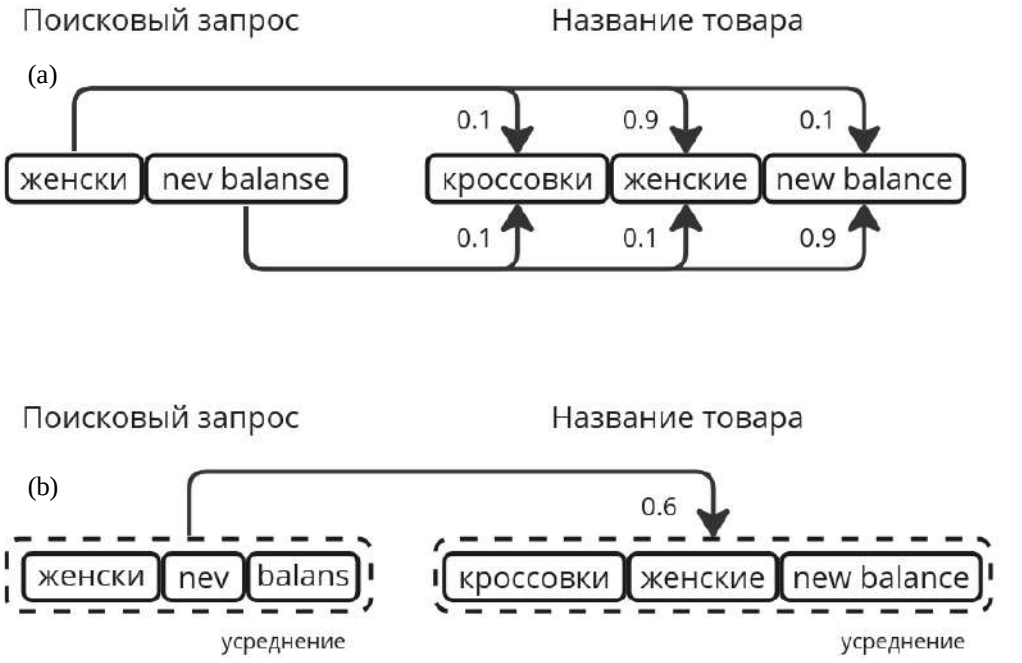


Рис. 3. (a). Оценка соответствия в H1 (b). Оценка соответствия в BERT.
Fig. 3. (a). Similarity in H1 (b). Similarity in BERT.

Методика оценки показателей эффективности для поиска по товарам отличается от оценки поиска по документам. Поиск по товарам подразумевает обнаружение нескольких, а желательно всех товаров, соответствующих запросу, даже если эти товары идентичны. Такое поведение продиктовано необходимостью сравнения цен на идентичные товары. Поэтому в формулах полноты и точности для сравнения полученной выдачи и истинной выдачи может использоваться совпадение как по текстовой модальности товара, так и по его цифровому идентификатору.

$$P_m@k = \frac{1}{|Q|}\sum_{q_i \in Q} \frac{\left|M_m\left(p^g_{q_i}, p^r_{q_i}@k\right)\right|}{k} \quad (5)$$

$$mAP_m@k = \frac{1}{k}\sum_{i=1}^{k} P_m@i \quad (6)$$

$|Q|$ – количество рассматриваемых поисковых запросов,

@k, @i – порог отсечки поисковой выдачи,

q_i – поисковый запрос $q_i \in Q$,

$p_{q_i}^{g}$ – все товары, соответствующие поисковому запросу q_i,

$p_{q_i}^{r}$ – поисковая выдача, все товары, найденные по поисковому запросу q_i

$|M_m(\cdot,\cdot)|$ – количество соответствующих друг другу товаров, определенных по функции соответствия M для модальности m.

Для более глубокого исследования возможностей оценки эффективности систем продуктового поиска по показателям можно ознакомиться с исследованием автора [56].

2. Эксперимент

Для подтверждения обозначенных преимуществ модели извлечения H1 выбран публичный набор данных, обучено три модели токенизации и проведены сравнительные эксперименты с моделью извлечения H1 и моделями TCT-ColBERT[55], Single Encoder (SE) [39] и Dual Encoder (DE) [40] на основе показателей mAP@12 и R@1k (5).

В качестве набора данных выбран WANDS [57], позволяющий проводить объективный бенчмаркинг и оценку систем извлечения товаров на основе набора данных электронной коммерции. Его ключевые характеристики включают:

- 42 994 товара-кандидата;
- 480 запросов;
- 233 448 оценок релевантности (запрос, товар).

Набор данных WANDS обладает трехуровневой разметкой пар «запрос-товар»: «полностью соответствует» (Exact), «частично соответствует» (Partial), «не соответствует» (Irrelevant). С целью обучения моделей системы извлечения использованы только два значения для построения функции потерь: Exact – 1, Irrelevant – −1. Полученные классы сбалансированы при обучении.

Для сравнения выбрана реализация модели ColBERT от Terrier [58], обученная с использованием подхода Tight Coupling Teachers [55].

В качестве гиперпараметров для моделей H1, SE, DE взяты значения размерности векторов токенов: 128, 256, 512, 768. В качестве функции соответствия M использовано точное соответствие для модальности m. Модальность m определена как «полное название товара».

Этапы проведения эксперимента:

1) Построение моделей токенизации word, bpe, unigram с токенами из нескольких слов (mt) и только с субсловарными конструкциями (non mt);
2) Обучение моделей извлечения товаров с различными вариантами векторного представления токенов;
3) Индексирование товаров с помощью модели извлечения товаров;
4) Вычисление показателей точности и полноты для каждого поискового запроса с различными порогами от 1 до 4096.

В общей сложности обработано более 300 тысяч вариантов комбинаций. Результаты экспериментов, усредненные по поисковым запросам, представлены на рис 4–9.

Наилучшие показатели R@1k = 86.6% и mAP@12 = 56.1% получены для модели H1 с моделью токенизации BPE, размерностью векторов 768 и токенами из «нескольких слов» (mt). Для сравнения на тех же данных модель ColBERT показала значения R@1k = 84.0% и

mAP@12 = 42.1%. Показатели пороговой полноты и точности для модели извлечения товаров на основе ColBERT приведены в табл. 1.

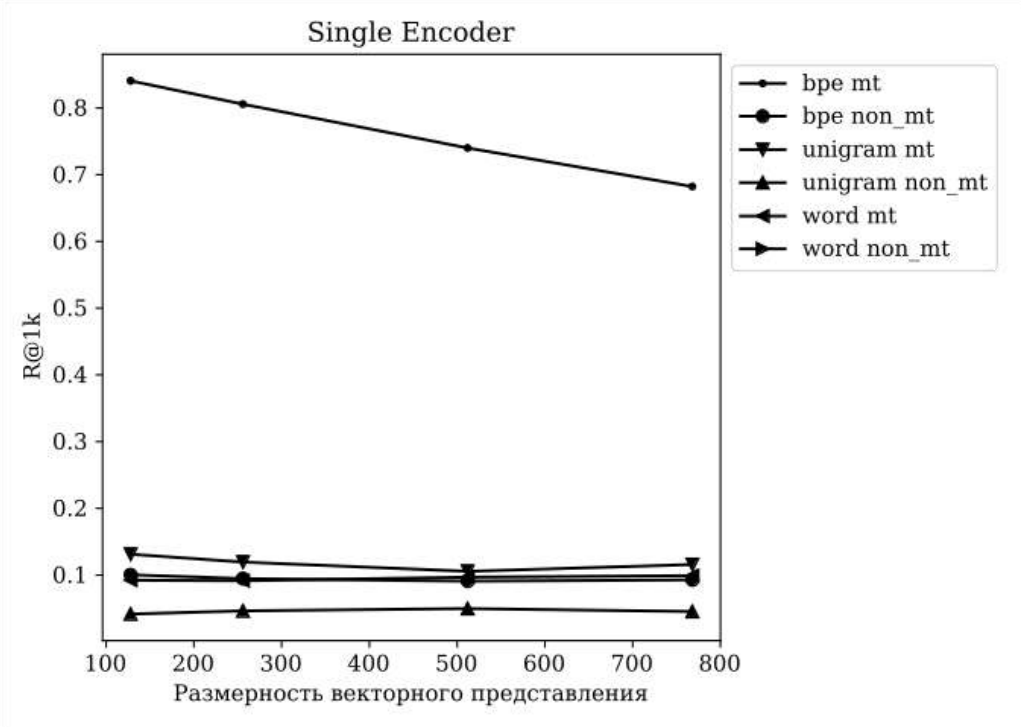


Рис. 4. Зависимость показателя R@1k от гиперпараметров для модели Single Encoder.
Fig. 4. Dependence of the R@1k metric on hyperparameters for the Single Encoder model.

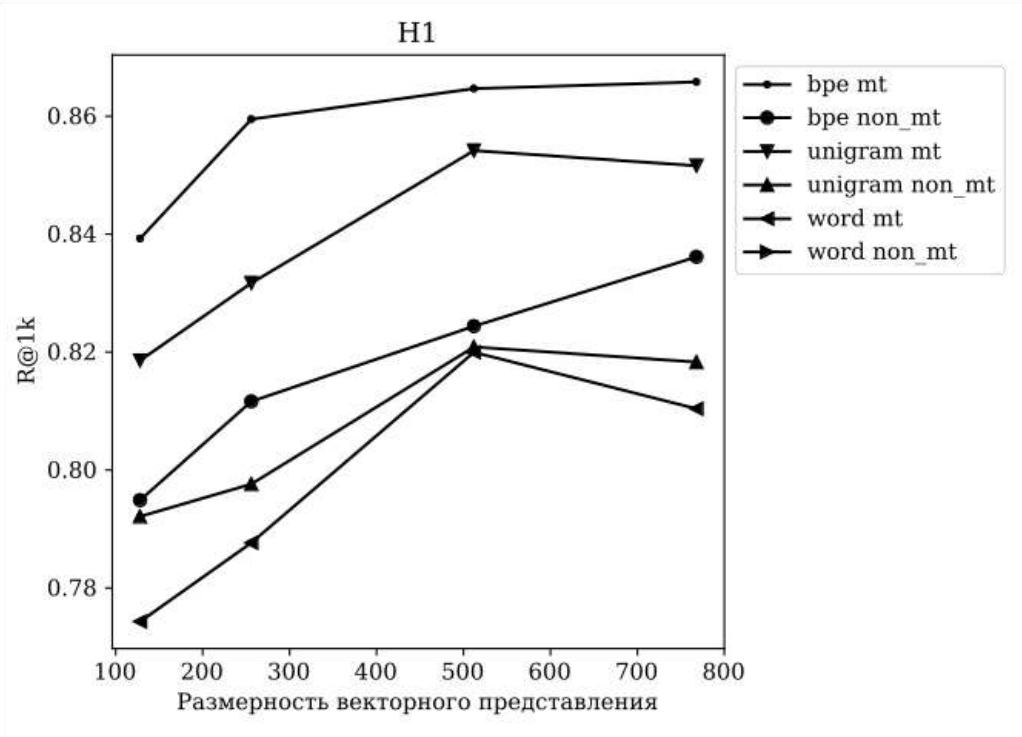


Рис. 5. Зависимость показателя R@1k от гиперпараметров для модели H1.
Fig. 5. Dependence of the R@1k metric on hyper parameters for the H1 model.

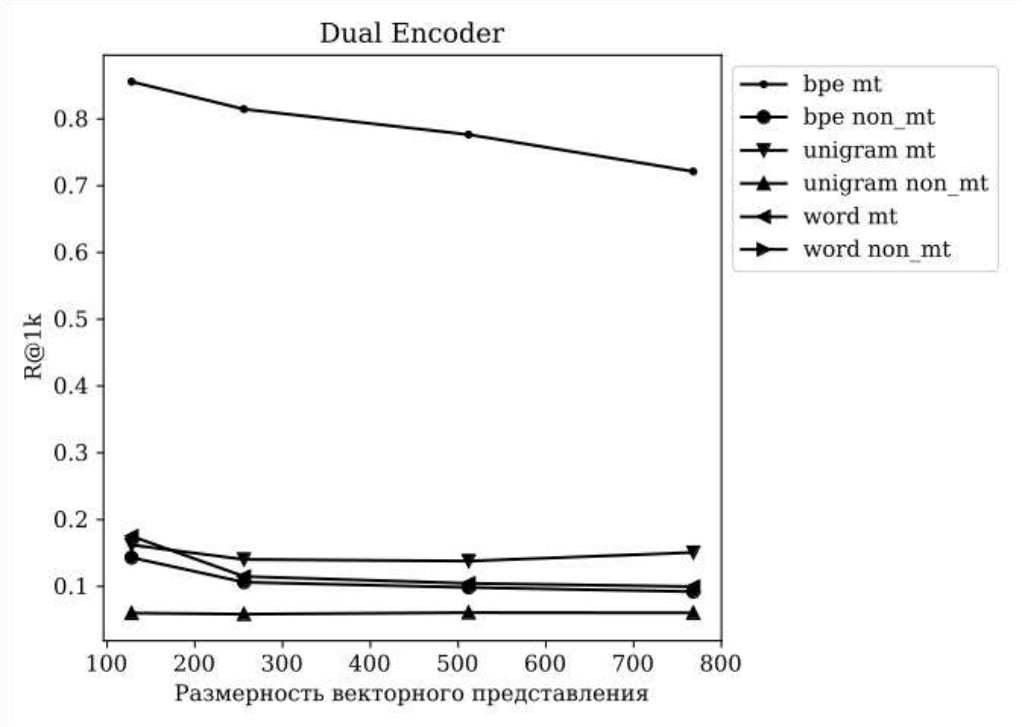


Рис. 6. Зависимость показателя R@1k от гиперпараметров для модели Dual Encoder.
Fig. 6. Dependence of the R@1k metric on hyperparameters for the Dual Encoder model.

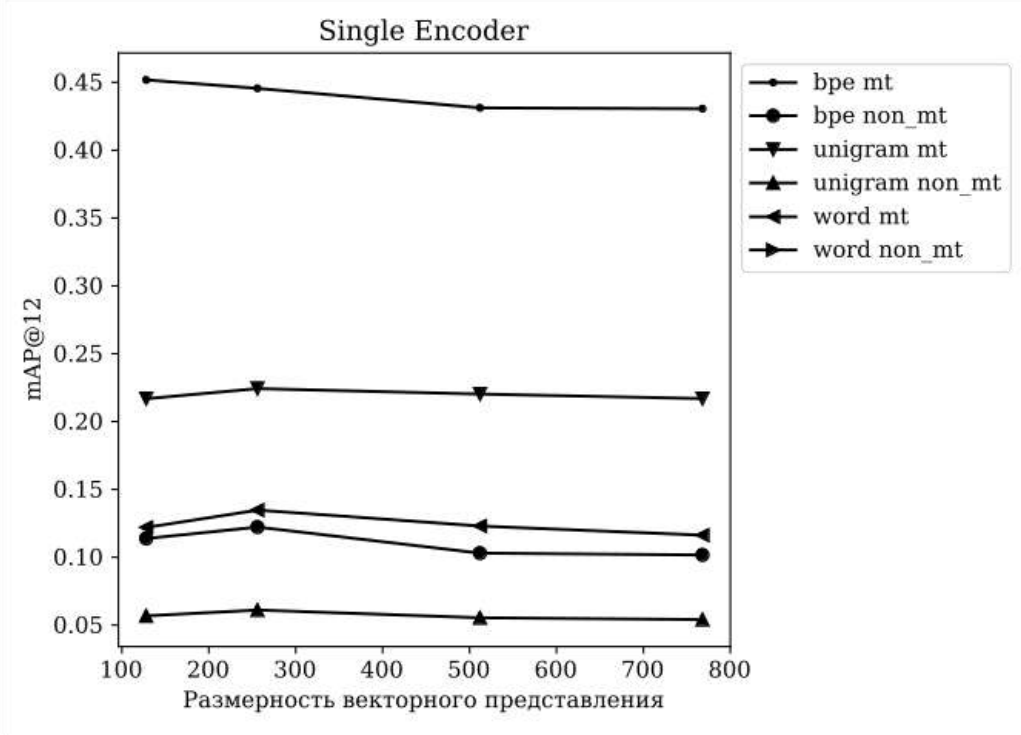


Рис. 7. Зависимость показателя mAP@12 от гиперпараметров для модели Single Encoder.
Fig. 7. Dependence of the mAP@12 metric on hyperparameters for the Single Encoder model.

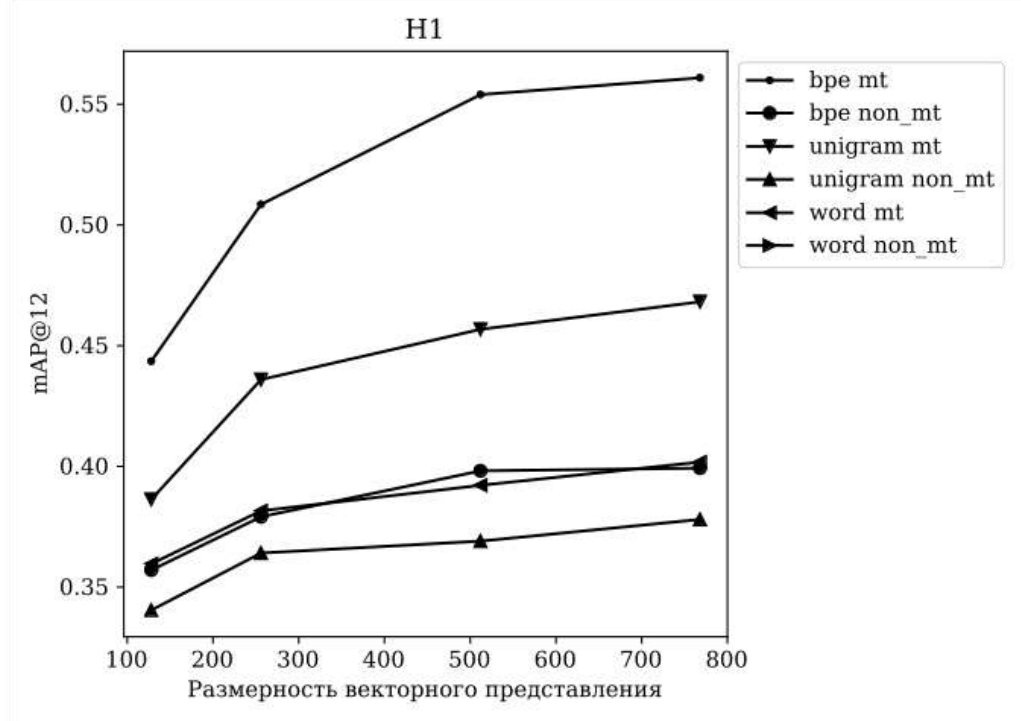


Рис. 8. Зависимость показателя mAP@12 от гиперпараметров для модели H1.
Fig. 8. Dependence of the mAP@12 metric on hyperparameters for the H1 model.

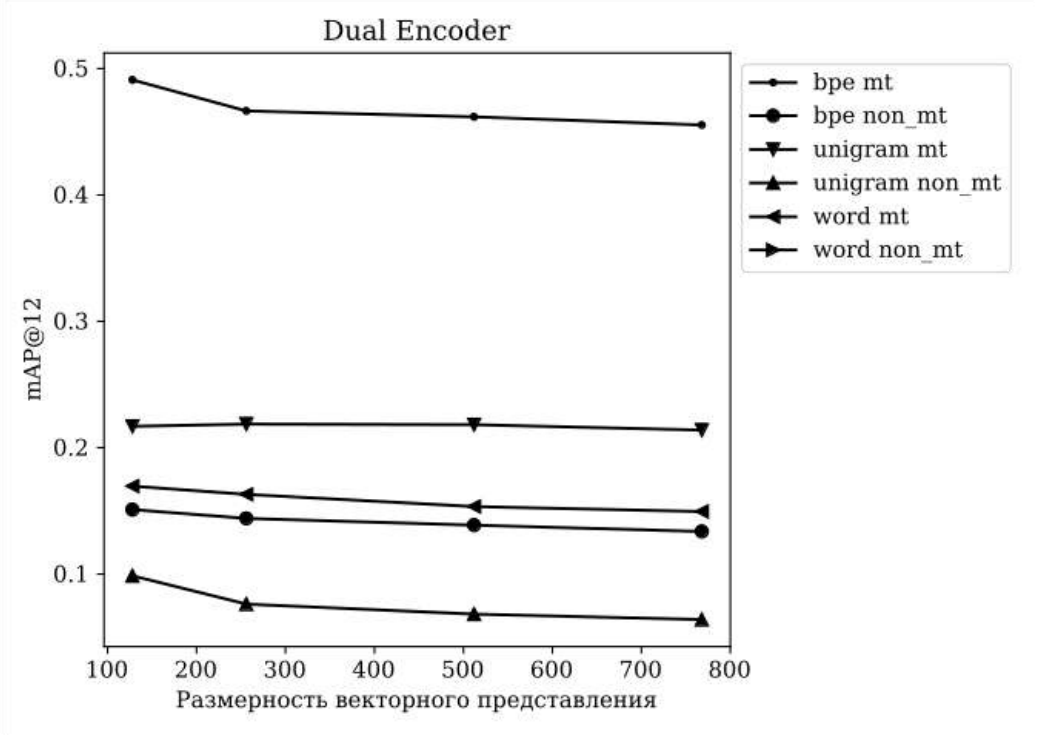


Рис. 9. Зависимость показателя mAP@12 от гиперпараметров для модели Dual Encoder.
Fig. 9. Dependence of the mAP@12 metric on hyperparameters for the Dual Encoder model.

Табл. 1. Показатели пороговой полноты и точности для модели ColBERT.
Table 1. R@k and P@k for the ColBERT model.

Модель	Порог, k	Точность	Полнота
ColBERT	12	**41%**	26%
	128	21%	61%
	512	9%	78%
	1024	5%	**84%**

Для получения результатов, приведенных на рис. 3–8, была использована разметка примеров из набора данных и функция потерь на основе кросс-энтропии. В промышленных масштабах разметить достаточное количество данных не представляется возможным из-за их высокой скорости изменения, поэтому модели извлечения товаров, обучаемые на поведенческих данных, используют правила для генерации отрицательных примеров.

Рис 3–8 сделаны с усреднением показателей mAP@12 и R@1k по запросам, чтобы сравнить модели и гиперпараметры. Для лучшего набора гиперпараметров на рис. 9 приведены зависимости показателей полноты и точности для рассматриваемых моделей без усреднения для одного запроса.

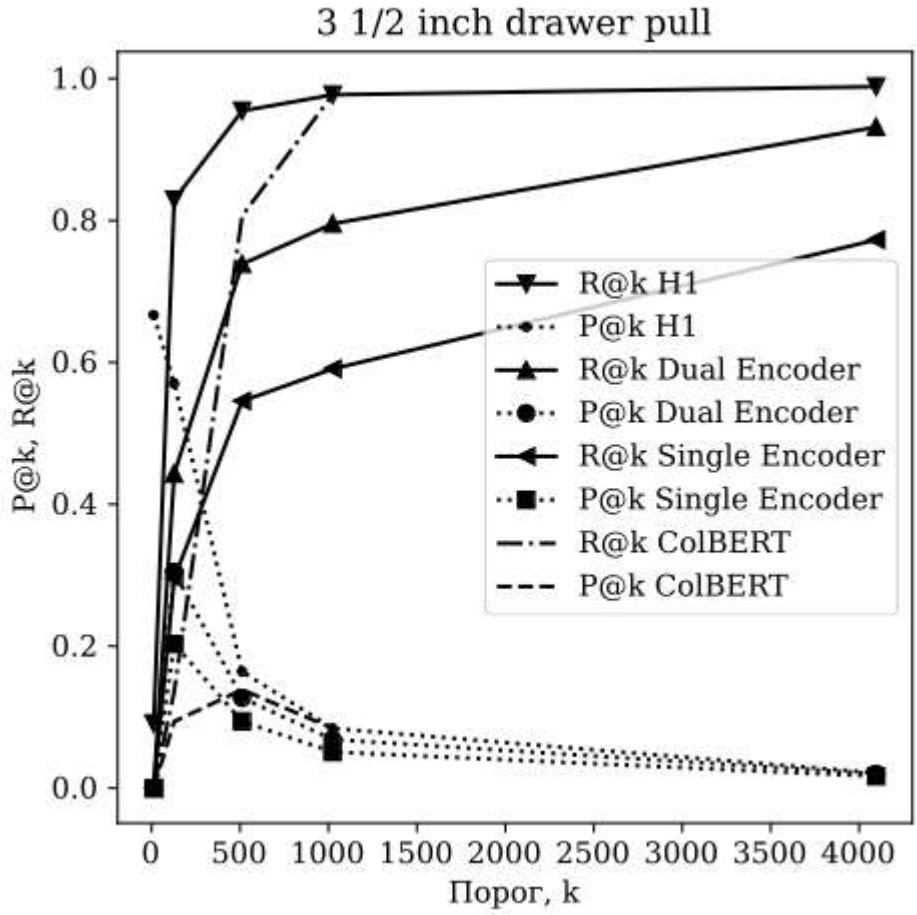


Рис. 10. Зависимости показателей полноты и точности от порога для одного поискового запроса.
Fig. 10. Dependences of recall and precision metrics for a single search query.

Зависимости для показателя полноты и точности на рисунке 10 демонстрируют преимущества модели извлечения товаров H1 над моделями Single Encoder, ColBERT и Dual Encoder. Выдача становится более полной при меньших значениях порога k, а точность выдачи при этом снижается медленнее.

4. Заключение

Классические модели лексического поиска балансируют между крайне общими и слишком специфичными поисковыми запросами, как следствие, необъятной или пустой поисковой выдачей. Подобный недостаток отдаляет клиентов от покупки товара на электронной торговой интернет-площадке. Модели поиска, основанные на векторном представлении, способны мягко сопоставлять запросы и документы, но они теряют информацию о соотнесении на уровне конкретных слов. В данной статье представлена модель поиска H1, которая лексический поиск дополняет поиском с многомерным векторным пространствам. Экспериментально доказано, что H1 достигает высокой эффективности поиска на первом этапе в рамках публичного набора данных, превосходя подходы на основе BERT за счет легкой архитектуры и уменьшения размера словаря с помощью употребления токенов, состоящих из нескольких слов. При этом H1 способна генерировать детализированные

сигналы сопоставления на уровне токенов, которые имеют решающее значение для точного поиска.

Список литературы / References

[1]. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. pp. 232–241 (1994)

[2]. Zeng C. et al. FAERY: An FPGA-accelerated Embedding-based Retrieval System //16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22). – 2022. – С. 841-856.

[3]. Zeng S. et al. DF-GAS: a Distributed FPGA-as-a-Service Architecture towards Billion-Scale Graph-based Approximate Nearest Neighbor Search //Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture. – 2023. – С. 283-296.

[4]. Pan Z. et al. RECom: A Compiler Approach to Accelerate Recommendation Model Inference with Massive Embedding Columns. – 2023.

[5]. Hofstätter S. et al. Improving efficient neural ranking models with cross-architecture knowledge distillation //arXiv preprint arXiv:2010.02666. – 2020.

[6]. George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. 1987. The vocabulary problem in human-system communication. Commun. ACM 30, 11 (1987), 964–971.

[7]. Le Zhao and Jamie Callan. 2010. Term necessity prediction. In Proceedings of the 19th ACM international conference on Information and knowledge management. 259–268.

[8]. Hang Li and Jun Xu. 2014. Semantic matching in search. Foundations and Trends in Information retrieval 7, 5 (2014), 343–469.

[9]. Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New Orleans, Louisiana, USA) (SIGIR '01). Association or Computing Machinery, New York, NY, USA, 120–127. https://doi.org/10.1145383952.383972

[10]. Michael E Lesk. 1969. Word-word associations in document retrieval systems. American documentation 20, 1 (1969), 27–38.

[11]. Yonggang Qiu and Hans-Peter Frei. 1993. Concept Based Query Expansion. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Pittsburgh, Pennsylvania,USA) (SIGIR '93). Association for Computing Machinery, New York, NY, USA, 160–169. https://doi.org/10.1145/160688.160713

[12]. Jinxi Xu and W Bruce Croft. 2017. Quary expansion using local and global document analysis. In Acm sigir forum,Vol. 51. ACM New York, NY, USA, 168–175.

[13]. Miles Efron, Peter Organisciak, and Katrina Fenlon. 2012. Improving retrieval of short texts through document expansion. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. 911–920.

[14]. Xiaoyong Liu and W Bruce Croft. 2004. Cluster-based retrieval using language models. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. 186–193.

[15]. Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. 2004. Dependence Language Model for Information Retrieval. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Sheffield, United Kingdom) (SIGIR '04). Association for Computing Machinery, New York, NY,USA, 170–177. https://doi.org/10.1145/1008992.1009024

[16]. Donald Metzler and W. Bruce Croft. 2005. A Markov Random Field Model for Term Dependencies. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Salvador, Brazil) (SIGIR '05). Association for Computing Machinery, New York, NY, USA, 472–479. https://doi.org/10.1145/1076034.1076115

[17]. Jun Xu, Hang Li, and Chaoliang Zhong. 2010. Relevance ranking using kernels. In Asia Information Retrieval Symposium. Springer, 1–12.

[18]. Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. Journal of the American society for information science 41, 6 (1990), 391–407.

[19]. Xing Wei and W. Bruce Croft. 2006. LDA-Based Document Models for Ad-Hoc Retrieval. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information

Retrieval (Seattle, Washington,USA) (SIGIR '06). Association for Computing Machinery, New York, NY, USA, 178–185. https://doi.org/10.1145/1148170.1148204

[20]. Adam Berger and John Lafferty. 1999. Information Retrieval as Statistical Translation. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Berkeley, California,USA) (SIGIR '99). Association for Computing Machinery, New York, NY, USA, 222–229. https://doi.org/10.1145/312624.312681

[21]. Maryam Karimzadehgan and ChengXiang Zhai. 2010. Estimation of Statistical Translation Models Based on Mutual Information for Ad Hoc Information Retrieval. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (Geneva, Switzerland) (SIGIR '10). Association for Computing Machinery, New York, NY, USA, 323–330. https://doi.org/10.1145/1835449.1835505

[22]. Felipe Bravo-Marquez, Gaston L'Huillier, Sebastián A. Ríos, and Juan D. Velásquez. 2010. Hypergeometric Language Model and Zipf-like Scoring Function for Web Document Similarity Retrieval. In Proceedings of the 17th International Conference on String Processing and Information Retrieval (Los Cabos, Mexico) (SPIRE'10). Springer-Verlag, Berlin,Heidelberg, 303–308.

[23]. Mikolov T. et al. Distributed representations of words and phrases and their compositionality //Advances in neural information processing systems. – 2013. – T. 26.

[24]. Pennington J., Socher R., Manning C. D. Glove: Global vectors for word representation //Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). – 2014. – C. 1532-1543.

[25]. Clinchant S., Perronnin F. Aggregating continuous word embeddings for information retrieval //Proceedings of the workshop on continuous vector space models and their compositionality. – 2013. – C. 100-109.

[26]. Ganguly D. et al. Word embedding based generalized language model for information retrieval //Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. – 2015. – C. 795-798.

[27]. Vulić I., Moens M. F. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings //Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. – 2015. – C. 363-372.

[28]. Boytsov L. et al. Off the beaten path: Let's replace term-based retrieval with k-nn search //Proceedings of the 25th ACM international on conference on information and knowledge management. – 2016. – C. 1099-1108.

[29]. Henderson M. et al. Efficient natural language response suggestion for smart reply //arXiv preprint arXiv:1705.00652. – 2017.

[30]. Bai Y. et al. SparTerm: Learning term-based sparse representation for fast text retrieval //arXiv preprint arXiv:2010.00768. – 2020.

[31]. Dai Z., Callan J. Context-aware sentence/passage term importance estimation for first stage retrieval //arXiv preprint arXiv:1910.10687. – 2019.

[32]. Nogueira R. et al. Document expansion by query prediction //arXiv preprint arXiv:1904.08375. – 2019.

[33]. Gillick D., Presta A., Tomar G. S. End-to-end retrieval in continuous space //arXiv preprint arXiv:1811.08008. – 2018.

[34]. Jean S. et al. On using very large target vocabulary for neural machine translation //arXiv preprint arXiv:1412.2007. – 2014.

[35]. Khattab O., Zaharia M. Colbert: Efficient and effective passage search via contextualized late interaction over bert //Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. – 2020. – C. 39-48.

[36]. Zamani H. et al. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing //Proceedings of the 27th ACM international conference on information and knowledge management. – 2018. – C. 497-506.

[37]. Li S. Embedding-based product retrieval in Taobao search // Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. – 2021. – C. 3181-3189.

[38]. Magnani A. Semantic retrieval at Walmart // Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. – 2022. – C. 3495-3503.

[39]. Nigam P. et al. Semantic product search //Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. – 2019. – C. 2876-2885.

[40]. Huang P. S. et al. Learning deep structured semantic models for web search using clickthrough data //Proceedings of the 22nd ACM international conference on Information & Knowledge Management. – 2013. – С. 2333-2338.
[41]. Gillick D., Presta A., Tomar G. S. End-to-end retrieval in continuous space //arXiv preprint arXiv:1811.08008. – 2018.
[42]. Yang Y. et al. Multilingual universal sentence encoder for semantic retrieval //arXiv preprint arXiv:1907.04307. – 2019.
[43]. Karpukhin V. et al. Dense Passage Retrieval for Open-Domain Question Answering //EMNLP (1). – 2020. – С. 6769-6781.
[44]. Vanderkam D. et al. Nearest neighbor search in google correlate. – 2013.
[45]. Johnson J., Douze M., Jégou H. Billion-scale similarity search with gpus //IEEE Transactions on Big Data. – 2019. – Т. 7. – №. 3. – С. 535-547.
[46]. Chang W. C. et al. Pre-training tasks for embedding-based large-scale retrieval //arXiv preprint arXiv:2002.03932. – 2020.
[47]. Xiong L. et al. Approximate nearest neighbor negative contrastive learning for dense text retrieval //arXiv preprint arXiv:2007.00808. – 2020.
[48]. Lu W., Jiao J., Zhang R. Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval //Proceedings of the 29th ACM International Conference on Information & Knowledge Management. – 2020. – С. 2645-2652.
[49]. Gao L. et al. Complement lexical retrieval model with semantic residual embeddings //Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43. – Springer International Publishing, 2021. – С. 146-160.
[50]. Kudo T. Subword regularization: Improving neural network translation models with multiple subword candidates //arXiv preprint arXiv:1804.10959. – 2018.
[51]. Sennrich R., Haddow B., Birch A. Neural machine translation of rare words with subword units //arXiv preprint arXiv:1508.07909. – 2015.
[52]. Provilkov I., Emelianenko D., Voita E. BPE-dropout: Simple and effective subword regularization //arXiv preprint arXiv:1910.13267. – 2019.
[53]. Gage P. A new algorithm for data compression //C Users Journal. – 1994. – Т. 12. – №. 2. – С. 23-38.
[54]. Viterbi A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm //IEEE transactions on Information Theory. – 1967. – Т. 13. – №. 2. – С. 260-269.
[55]. Lin S. C., Yang J. H., Lin J. Distilling dense representations for ranking using tightly-coupled teachers //arXiv preprint arXiv:2010.11386. – 2020.
[56]. Krasnov F.V. Embedding-based retrieval: measures of threshold recall and precision to evaluate product search. // Business Informatics. – 2024 – Т. 18. – №. 2. – С.22–34. DOI:10.17323/2587-814X.2024.2.22.34.
[57]. Chen Y. Wands: Dataset for product search relevance assessment // European Conference on Information Retrieval. – Cham : Springer International Publishing, 2022. – С. 128-141.
[58]. Macdonald C., Tonellotto N. Declarative experimentation in information retrieval using PyTerrier //Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval. – 2020. – С. 161-168.

Информация об авторах / Information about authors

Федор Владимирович КРАСНОВ – доктор технических наук, специалист по поисковым и рекомендательным системам в электронной коммерции, сотрудник Исследовательского центра ООО "ВБ СК" на базе Инновационного Центра Сколково.

Fedor Vladimirovich KRASNOV – Dr. Sci. (Tech.), specialist in information retrieval and recommendation systems in e-commerce, researcher of the Research Center of WB SK LLC based on the Skolkovo Innovation Center, Moscow

DOI: 10.15514/ISPRAS-2024-36(5)-17

CC BY

Применение моделей машинного обучения для многоклассовой классификации дерматоскопических снимков новообразований кожи

[1] *А.В. Козачок, ORCID: 0000-0002-6501-2008 <a.kozachok@ispras.ru>*
[1] *А.А. Спирин, ORCID: 0000-0002-7231-5728 <a.spirin@ispras.ru>*
[1] *О.И. Самоваров, ORCID: 0000-0002-7006-7193 <samov@ispras.ru>*
[2] *Е.С. Козачок, ORCID: 0000-0003-2912-0754 <muza2804@gmail.com>*

[1] *Институт системного программирования РАН,*
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.
[2] *ООО «Бьюти Клиник»,*
Россия, 302040, Орел, ул. Октябрьская, д. 77, лит. А.

Аннотация. В статье рассмотрены вопросы практической оценки качества современных моделей машинного обучения, реализованных на основе глубоких нейронных сетей и визуальных трансформеров. Описаны параметры проведенного эксперимента на наборе данных ISIC 2018. Приведена статистика по категориям рассмотренных поражений кожи. Проведенный статистический анализ полученных результатов позволил авторскому коллективу сформировать новую бинарную категорию: меланоцитарные и немелонацитарные поражения кожи. Эксперименты по обучению нейросетевых моделей были выполнены на мощностях Цифровой экосистемы НЦМУ.

Ключевые слова: модели машинного обучения; раннее обнаружение меланомы и рака кожи; набор данных ISIC 2018.

Для цитирования: Козачок А.В., Спирин А.А., Самоваров О.И., Козачок Е.С. Применение моделей машинного обучения для многоклассовой классификации дерматоскопических снимков новообразований кожи. Труды ИСП РАН, том 36, вып. 5, 2024 г., стр. 241–252. DOI: 10.15514/ISPRAS–2024–36(5)–17.

Благодарности: Исследование поддержано Фондом содействия инновациям (Договор № 386ГС1ЦТС10-D5/91114 от 14.06.2024 г.).

Application of Machine Learning Models for Multiclass Classification of Dermatoscopic Images of Skin Neoplasms

[1] *A.V. Kozachok, ORCID: 0000-0002-6501-2008 <a.kozachok@ispras.ru>*
[1] *A.A. Spirin, ORCID: 0000-0002-7231-5728 <a.spirin@ispras.ru>*
[1] *O.I. Samovarov, ORCID: 0000-0002-7006-7193 <samov@ispras.ru>*
[2] *E.S. Kozachok, ORCID: 0000-0003-2912-0754 <muza2804@gmail.com>*

[1] *Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
[2] *LLC «Beauty Clinic»,*
77 A, Oktyabrskaya st., Orel, 302040, Russia.

Abstract. The scientific article considers the issues of practical quality assessment of modern machine learning models implemented on the basis of deep neural networks and visual transformers. The parameters of the conducted experiment on the ISIC 2018 dataset are described. The statistics on the categories of the considered skin lesions is given. The statistical analysis of the obtained results allowed the author's team to form a new binary category: melanocytic and non-melanocytic skin lesions. Experiments on training neural network models were performed at the facilities of the NCMU Digital Ecosystem.

Keywords: **melanoma:** machine learning models; early melanoma and skin cancer detection; ISIC 2018 dataset.

For citation: Kozachok A.V., Spirin A.A., Samovarov O.I., Kozachok E.S. Application of machine learning models for multiclass classification of dermatoscopic images of skin neoplasms. Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 5, 2024. pp. 241-252 (in Russian). DOI: 10.15514/ISPRAS-2024-36(5)-17.

Acknowledgements. The research was supported by the Foundation for the Promotion of Innovation (Agreement No. 86ГС1ЦТС10-D5/91114 14.06.2024).

1. Введение

Раннее выявление заболеваний кожи, представляющих потенциальную опасность для жизни и здоровья пациента, способно значительно снизить заболеваемость и смертность, поскольку стадия поражения тесно связана с дальнейшем прогнозом. Использование искусственного интеллекта (далее – ИИ), являющегося неинвазивной технологией, может помочь врачам общей практики в ранней диагностике рака кожи, в том числе меланомы. Многие модели ИИ используют, на этапе обучения, предварительно созданные базы данных изображений, которые зачастую не совпадают, на этапе распознавания, с изображениями, сделанными в реальных условиях (освещение, качество изображения, угол поворота камеры и т.п.) врачами общей практики или самими пациентами в домашних условиях. Цель исследования заключается в анализе эффективности использования современных моделей ИИ на базе сверточных нейронных сетей с использованием набора данных от Международной организации по визуализации кожи (The International Skin Imaging Collaboration – ISIC, США Нью-Йорк). Врачи общей практики, работающие совместно с ИИ, имеют больше шансов на первичную диагностику меланомы. Для решения задачи автоматической оценки изображений, на предмет наличия поражений кожи, необходимо использование автоматизированных платформ для сбора и анализа данных [1]. Любая автоматизированная платформа – это интерфейс взаимодействия, модель машинного обучения, работающая в паре с набором данных. Ключевым аспектом платформы является выбор эффективной модели машинного обучения и набора данных.

2. Существующие проблемы использования ИИ в медицине

Несмотря на успехи в использовании ИИ для диагностики меланомы и рака кожи, остаются нерешенные проблемы:

1. Демографическая неполнота данных: наборы данных, такие как ISIC, включают преимущественно изображения пациентов определенных этнических групп и с определенным типом кожи, что ограничивает их применение. Данная проблема является основным мотивирующим фактором создания Отечественной платформы по сбору дерматоскопических снимков кожи пациентов из РФ.
2. Предвзятость и ошибки классификации: модели ИИ показывают разную точность при классификации редких видов рака кожи, что может привести к ложноположительным результатам и снизить доверие к системам ИИ.
3. Этичность и конфиденциальность данных: сохранение конфиденциальности пациентов остается актуальной проблемой, особенно при сборе и использовании больших объемов медицинских данных. В настоящее время разработаны стандартизирующие документы в области сбора и использования медицинских данных пациентов.
4. Стандартизация медицинских изображений. Стандарт DICOM (Digital Imaging and Communications in Medicine) является широко применяемым форматом для хранения и передачи медицинских изображений, особенно в таких областях, как радиология, кардиология и онкология. Однако его применение в дерматоскопии до сих пор ограничено и не стандартизировано на таком же уровне, как для других типов медицинских изображений. Дерматоскопические изображения не всегда требуют такой же детализации и метаданных, как изображения из радиологии. В большинстве случаев дерматоскопические снимки делаются с помощью оптических или цифровых камер без специализированных устройств визуализации, что затрудняет их стандартизацию в формате DICOM. Несмотря на усилия, направленные на улучшение стандартизации изображений, дерматология ещё не приняла DICOM повсеместно, поскольку дерматоскопические изображения чаще сохраняются в стандартных форматах (JPEG, PNG). Это снижает барьеры для врачей и пациентов, которые могут легко просматривать и обмениваться такими изображениями, но усложняет стандартизацию и интеграцию в системы PACS (Picture Archiving and Communication Systems) [2]. Некоторые публикации описывают попытки внедрить DICOM для дерматологии, фокусируясь на добавлении стандартизированных метаданных для дерматоскопии (например, тип камеры, условия освещения и угол съемки). Однако пока это остаётся на стадии экспериментов и редко встречается в повседневной клинической практике [3, 4].

В современном мире цифровой дерматологии автоматизированной диагностике уделяется большое внимание. С 2015 года FDA (Food and Drug Administration, Управление по санитарному надзору за качеством пищевых продуктов и медикаментов) при Министерстве здравоохранения и социальных служб США регулирует деятельность в области потребительских диагностических приложений. Европейский союз в 2017 году выпустил Регламент 2017/745 о медицинских устройствах. Европейская комиссия в 2020 году опубликовала «Белую книгу» по продвижению европейской экосистемы передового опыта и доверия в области ИИ.

Согласно п. 19 Регламента 2017/745 о медицинских устройствах программное обеспечение общего назначения, которое может использоваться в медицинских учреждениях и в домашних условиях медицинским изделием не считается и к такому программному обеспечению не предъявляется дополнительных требований по сертификации и вводу в эксплуатацию. В «Белой книге» отмечается необходимость популяризации искусственного

интеллекта в повседневной жизни, при этом особую роль документ отводит повышению доверия к ИИ со стороны граждан.

Зарубежное законодательство отмечает необходимость создания систем ИИ, позволяющих облегчить работу врачей-дерматологов в решении задач постановки предварительных диагнозов, однако технических требований к системам компьютерного зрения, изображениям, устройствам, моделям машинного обучения нормативные документы не предъявляют.

Вместе с тем в Российской Федерации развитию искусственного интеллекта уделяется достаточное внимание. Начиная с 2019 года в России выпущен ряд нормативных правовых актов: Национальная стратегия развития искусственного интеллекта, Концепции развития регулирования отношений в сфере технологий искусственного интеллекта и робототехники до 2024 года, комплекс национальных стандартов «Системы искусственного интеллекта в клинической медицине» (ГОСТ Р 59921.0-2022- ГОСТ Р 59921.9-2022). Основной целью законодательства является стимулирование разработки систем искусственного интеллекта, позволяющих обеспечить внедрение прогнозной аналитики возникновения и развития заболеваний, а также анализ изображений различных патологий. При этом в отличии от зарубежного законодательства, российское считает программное обеспечение, использующее системы искусственного интеллекта в медицинской сфере, медицинским изделием и предъявляет к таким системам определенные требования: к наборам данных, к системе в целом и валидации полученных результатов. Учитывая современные требования стандартов и законодательства, предъявляемых к медицинским системам, коллектив авторов продолжает разработку платформы для сбора дерматоскопических снимков и обезличенной информации пациентов из Орловской области, интерфейс платформы представлен на рис. 1.

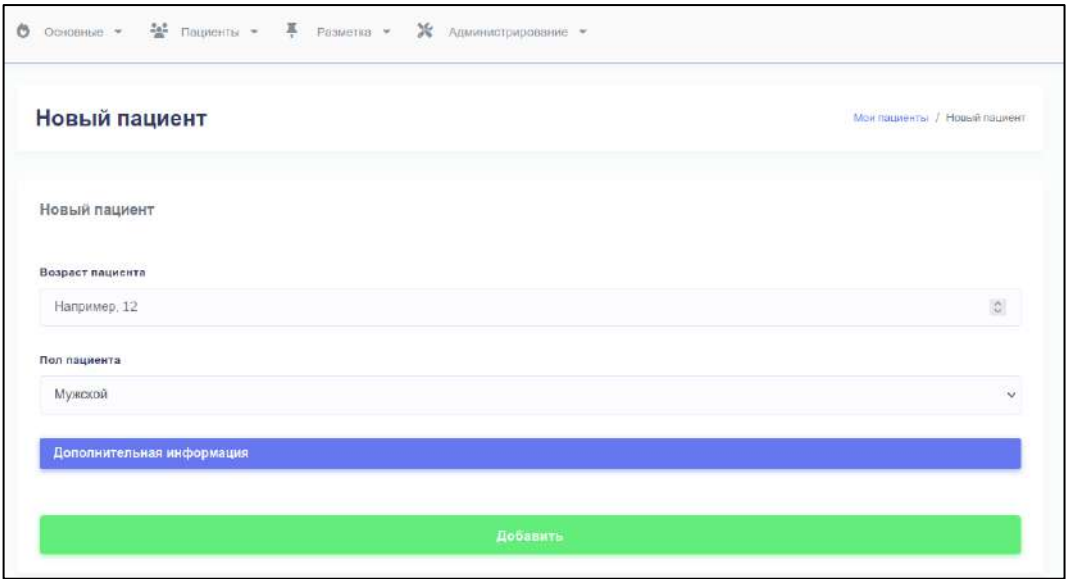


Рис. 1. Интерфейс платформы сбора дерматоскопических снимков.
Fig. 1. Dermatoscopic imaging collection platform Interface.

Интерфейс системы представляет собой форму для ввода данных нового пациента, предназначенную для упрощения сбора информации о пациентах, связанных с дерматологическими исследованиями и диагностикой кожных заболеваний. В верхней части экрана размещены поля для указания возраста и пола пациента, что позволяет начать заполнение с базовой информации. Далее, пользователю предлагается блок дополнительной

информации, который раскрывается для более детального ввода данных, значимых для оценки рисков, связанных с кожными заболеваниями.

Дополнительная информация включает параметры, такие как тип волос и тип кожи, которые позволяют классифицировать пациента по фототипу и чувствительности к солнечному воздействию. Также предусмотрено поле для указания области тела, где преимущественно располагаются родинки, что может помочь при определении зон риска. Параметр количества родинок и их размера также помогает в оценке предрасположенности пациента к кожным новообразованиям.

Интерфейс учитывает наследственность и возможное применение иммунносупрессоров или наличие заболеваний, что может быть важно для определения факторов, влияющих на иммунный ответ организма. Также пользователь может указать наличие веснушек, расу пациента и данные о солнечных ожогах, что способствует более точной оценке рисков развития заболевания. Завершает заполнение информации кнопка «Добавить», расположенная внизу экрана и выполненная в зелёном цвете для удобного визуального выделения, что подчеркивает её основную функцию сохранения данных.

В настоящее время собраны данные более 300 пациентов из Орловской области и сбор продолжается в рамках проводимого мероприятия “День меланомы” с участием врачей Орловского онкологического диспансера.

После формирования набора данных объемом 500 пациентов планируется использовать его для дообучения моделей, показавших лучший результат на опубликованных и доступных наборах данных пациентов из других стран, с целью получения модели ИИ, обученной на пациентах из Российской Федерации и, следовательно, учитывающей фенотип кожи населения Российской Федерации.

3. Анализ моделей машинного обучения

Самым эффективным вариантом снижения смертности от рака кожи является своевременная диагностика, поскольку выживаемость пациентов с меланомой за пятилетний период составляет 99 % при диагностике на ранней стадии [5].

Несмотря на повышение диагностической точности с помощью дерматоскопии, скрининг кожных заболеваний с помощью дерматоскопических изображений требует достаточно много времени (до одного месяца), при этом на окончательную постановку диагноза влияют также и субъективные факторы [6].

Классификация кожных поражений с помощью автоматизированных систем компьютерного зрения – сложная задача из-за вариабельности внешнего вида кожных поражений, при этом большое внимание научным сообществом уделяется подготовке наборов данных с изображениями для обучения автоматизированных систем. Такие системы разрабатываются и показали большой потенциал [7-9].

Авторским коллективом на первоначальном этапе разработки автоматизированной платформы по обработке и агрегации данных рассмотрен наиболее часто используемый исследователями набор данных ISIC 2018 [10-14].

Основной клинической целью создания ISIC является снижение смертности от меланомы и необоснованных биопсий путем повышения точности и эффективности раннего выявления меланомы с использованием моделей машинного обучения. Для этого ISIC разрабатывает предлагаемые стандарты цифровой визуализации и привлекает сообщества дерматологов и IT-специалистов к повышению точности диагностики с помощью искусственного интеллекта [10].

Общедоступный набор данных ISIC 2018 – это база данных, состоящая из 10 015 дерматоскопических изображений повреждений кожи, полученных в Австрии и Австралии. Это 24-битные изображения RGB размером 600×450 пикселей с разрешением 96 точек на дюйм. В базе выделены семь диагностических категорий поражений кожи (рис. 2).

Статистические данные тестового набора данных представлены в табл. 1.

Целью исследования является анализ применимости современных моделей машинного обучения для решения задачи раннего обнаружения меланомы и рака кожи на дерматоскопических изображениях пораженных участков.

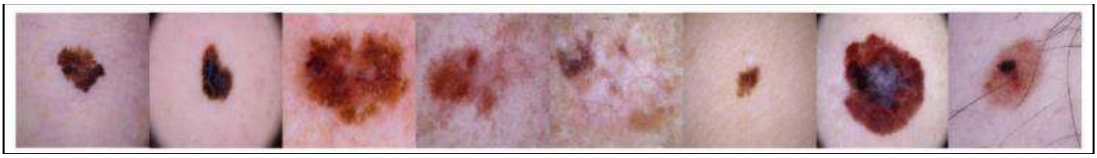


Рис. 2. Примеры разнообразия вариантов фотографий меланомы из базы ISIC2018.
Fig. 2. Examples of the variety of melanoma photo options from the ISIC2018 database.

Табл. 1. Количество изображений каждой категории в наборе данных ISIC 2018.
Table 1. Number of images of each category in the ISIC 2018 dataset.

AK/ AKIEC	BCC	SK/ BKL	DF	VL/VASc	MEL	NV	ИТОГО
327	514	1099	115	142	1113	6705	10015

AK (AKIEC) – Actinic Keratosis (Актинический кератоз, болезнь Боуена).

BCC – Basal cell carcinoma (Базально-клеточная карцинома) (NML group).

SK (BKL) – Seborrheic Keratosis, benign keratosis (доброкачественный кератоз).

DF – Dermatofibroma (дерматофиброма).

VL (VASc) – Vascular skin lesions (гемангиомы – сосудистые образования кожи).

MEL – Melanoma (меланома).

NV – Melanocytic nevi (меланоцитарные невусы).

Основные методы, которые используются в диагностике кожных поражений, можно разделить на машинное обучение и глубокие нейронные сети. Наиболее востребованные модели включают сверточные нейронные сети (CNN) и трансформеры (Visual Transformers), способные решать задачи классификации изображений с высокой точностью. Однако, по данным последних исследований, качество классификации все еще варьируется в зависимости от сложности изображений и уровня детализации аннотаций в обучающих данных. В ходе экспериментов были использованы следующие модели ИИ: сверточные нейронные сети (CNN) и трансформеры изображений (ViT), демонстрирующие высокую точность в решении задачи классификации изображений [15].

Авторским коллективом проведен анализ современных моделей машинного обучения, позволяющих решать задачу автоматической классификации изображений пораженных участков кожи. В табл. 2 приведены модели, продемонстрировавшие наивысшую точность классификации пораженных участков кожи [16-18].

В большинстве своем рассмотренные модели базируются на новом подходе к классификации рака кожи с использованием визуальных трансформеров − современной архитектуры глубокого обучения, которая демонстрирует исключительную производительность в различных задачах анализа изображений. Результаты экспериментов, полученных с помощью нейросетей на основе трансформеров, демонстрируют превосходство моделей над традиционными архитектурами глубокого обучения в задаче классификации рака кожи. Авторский коллектив [19] проводил оценки с шестью различными моделями:

- `vit-face-expression`,
- `beit-base-patch16`,
- `cards-top_left_swin-tiny-patch4`,
- `rorshark-vit-base`,
- `vit-small-patch16-224`,

- `vit-tiny-patch16-224`.

Обучение и оценка качества нейросетевых моделей выполнены с использованием облачной инфраструктуры Цифровой экосистемы НЦМУ (Научного центра мирового уровня) ИСП РАН [20].

Табл. 2. Популярные модели машинного обучения, решающие задачу автоматической классификации изображений.
Table 2. Popular machine learning models that solve the problem of automatic image classification.

№ п.п.	Название модели
1	trpakov/vit-face-expression
2	sanali209/nsfwfilter
3	microsoft/beit-base-patch16-224-pt22k-ft22k
4	sai17/cards-top_left_swin-tiny-patch4-window7-224-finetuned-v3_more_data
5	amunchet/rorshark-vit-base
6	alexnet
7	WinKawaks/vit-small-patch16-224
8	Falconsai/nsfw_image_detection
9	rizvandwiki/gender-classification
10	microsoft/swinv2-tiny-patch4-window16-256
11	WinKawaks/vit-tiny-patch16-224

Эксперименты исследователей [16-19] показали, что подходы на базе визуальных трансформеров позволяют создавать более эффективные модели (в плане вычислительных затрат, с меньшим количеством эпох и размера серии batch size), которые при этом достигают лучшего качества, по сравнению с моделями на базе сверточных нейронных сетей.

Для проведения экспериментов по определению лучших моделей в задаче классификации дерматоскопических снимков разрабатываемой платформы авторским коллективом были определены следующие параметры обучения: количество эпох – 100, размер серии – 16, набор данных – ISIC 2018. Структура тестового набора данных представлена в табл. 3. Результаты проведенных экспериментов представлены в табл. 4.

Табл. 3. Структура тестового набора данных для обучения.
Table 3. Structure of the test data set for training.

Название поражения кожи	Обучающая выборка	Тестовая выборка	Процентное отношение
AK (AKIEC)	247	80	32%
BCC	422	92	22%
BKL	867	232	27%
DF	89	26	29%
MEL	900	213	24%
NV	5368	1337	25%
VL (VASC)	119	23	19%
ИТОГО	**8012**	**2003**	**25%**

Анализ полученных результатов демонстрирует низкую эффективность различных архитектур машинного обучения, применяемых для классификации дерматоскопических снимков, разделенных на 8 классов. Максимальное значение метрики F1 составило 0.69 для модели Resnet152. ResNet152 справляется лучше по балансу между точностью (Precision) и полнотой (Recall) в данной задаче, особенно при ограниченном количестве эпох обучения. ViT (Vision Transformers) при этом демонстрируют высокую точность (например, google/vit-base-patch16-224 достигает точности 0.86), но их F1-метрики ниже, что указывает на меньший баланс между точностью и полнотой. Это может происходить из-за особенностей обработки изображений трансформерами, которые лучше захватывают глобальные взаимосвязи, но могут терять детали при меньшем количестве данных или специфическом составе набора данных, как в случае дерматоскопии.

Табл. 4. Матрица ошибок на тестовом наборе после обучения на ста эпохах и размером серии шестнадцать.
Table 4. Error matrix on the test set after training on one hundred epochs and a series size of sixteen.

Модель	**Эпохи**	**Датасет**	**ACC**	**F1**
ResNet152	20	ISIC19	0.71	0.69
google/vit-base-patch16-224	20	ISIC18	0.86	0.57
microsoft/dit-base-finetuned-rvlcdip	20	ISIC18	0.789	0.47
sai17/cards_bottom_left_swin-tiny-patch4-window7-224-finetuned-dough_100_epochs	20	ISIC18	0.85	0.56
microsoft/swinv2-tiny-patch4-window16-256	20	ISIC18	0.88	0.62
microsoft/beit-base-patch16-224-pt22k-ft22k	20	ISIC18	0.86	0.58
WinKawaks/vit-tiny-patch16-224	100	ISIC18	0.84	0.55
microsoft/swinv2-tiny-patch4-window16-256	100	ISIC18	0.89	0.63
microsoft/beit-base-patch16-224-pt22k-ft22k	100	ISIC18	0.87	0.59
WinKawaks/vit-small-patch16-224	100	ISIC18	0.85	0.57
sanali209/nsfwfilter	100	ISIC18	0.86	0.57
amunchet/rorshark-vit-base	100	ISIC18	0.87	0.59
rizvandwiki/gender-classification	100	ISIC18	0.86	0.58
Falconsai/nsfw_image_detection	100	ISIC18	0.85	0.58
sai17/cards-top_left_swin-tiny-patch4-window7-224-finetuned-v3_more_data	100	ISIC18	0.87	0.59
trpakov/vit-face-expression	100	ISIC18	0.87	0.59

Хотя ViT демонстрируют хорошие результаты для анализа изображений с высокой вариативностью и обладают преимуществами при работе с комплексными данными, в конкретном случае задач дерматоскопии классические сверточные сети, такие как ResNet152, могут быть более сбалансированным выбором. Этот пример подчеркивает, что трансформеры не всегда превосходят CNN по всем метрикам, особенно в узконаправленных медицинских задачах, где важны как высокая точность, так и правильный баланс между чувствительностью и специфичностью.

4. Заключение

Рассмотренные модели искусственного интеллекта (ИИ) демонстрируют невысокую точность классификации и наличие множества ошибок классификации, что свидетельствует о невозможности решения задачи в данной трактовке рассмотренными моделями ИИ. Средняя точность классификации по всем моделям составляет 0.63 по метрике F1.

В рамках дальнейших исследований авторским коллективом будет рассмотрен набор данных ISIC 2019, в который в полном объеме вошли изображения из набора данных ISIC 2018. Дальнейшие исследования будут направлены на проведение экспериментов по бинарной классификации поражений кожи пациентов. Рассмотренные классы кожных поражений будут сформированы в две группы: в первую группу войдут меланоцитарные образования (Mlgroup) – представители классов Меланома (MEL) и Невус (NV), особенностью класса является наличие клеток-невоцитов, представляющие собой мутировавшие меланоциты; во вторую группу будут включены немеланоцитарные (NoMLgroup) образования, включающие в себя представителей 6 классов – дерматофиброма (DF), гемангиома (VASc), доброкачественный кератоз (SK (BKL), базально-клеточная карцинома (BCC), актиничный кератоз (AK (AKIEC). Дальнейшие исследования направлены на получение более высоких метрик точности на задаче бинарной классификации поражений кожи среди меланоцитарных и немеланоцитарных образований. Основная гипотеза заключается в постановке более простой задачи бинарной классификации – разделить наиболее агрессивную меланому и безопасные невусы от остальных видов рака кожи, что позволит в дальнейшем использовать ансамбли нейронных сетей для ранней диагностики заболеваний кожи пациентов из РФ.

Дальнейшее развитие модели классификации дерматоскописческих изображений предполагает организацию непрерывного масштабируемого процесса сбора и аннотирования данных из клиник-партнеров и дообучения нейросетевых моделей на новых данных. В планах - интегрировать систему аннотирования дерматоскопических изображений в Цифровую экосистему НЦМУ ИСП РАН, которая обеспечивает поддержку жизненного цикла нейросетевых моделей для задачи анализа данных биомедицинского домена.

Список литературы / References

[1]. Козачок, А. В., Спирин, А. А., Елецкий, К. В., & Козачок, Е. С. (2024). Платформа для сбора дерматоскопических изображений новообразований пациентов. Труды института системного программирования РАН, 36(3), 259-272.

[2]. Martinez G. A. et al. Integrating Dermoscopic Images into PACS Using DICOM and Modality Worklist //MEDINFO 2023—The Future Is Accessible. – IOS Press, 2024. – С. 199-203.

[3]. Caffery L. et al. DICOM in dermoscopic research: an experience report and a way forward //Journal of Digital Imaging. – 2021. – Т. 34. – С. 967-973.

[4]. Caffery L. J. et al. The role of DICOM in artificial intelligence for skin disease //Frontiers in medicine. – 2021. – Т. 7. – С. 619787.

[5]. Kravchenko O. OpenAI вычисляют идеальный batch size для обучения моделей // https://neurohive.io/ru/novosti/openai-batch-size-ideal/, 20.12.2018, 16.08.2024

[6]. Kassani, S.H., Kassani, P.H. A comparative study of deep learning architectures on melanoma detection // Tissue and Cell. – 2019. – Vol. 58. – pp.76-83. doi: 10.1016/j.tice.2019.04.009.

[7]. Chaturvedi S.S., Gupta K., Prasad P.S. Skin lesion analyser: an efficient seven-way multi-class skin cancer classification using MobileNet // Advanced Machine Learning Technologies and Applications. – Springer, Singapore. – 2020. – p.165-176. doi: 10.1007/978-981-15-3383-9_15.

[8]. Qin X.Z. et al. A GAN-based image synthesis method for skin lesion classification // Computer Methods and Programs in Biomedicine. – 2020. – Vol. 195. – p.105568. doi.org/10.1016/j.cmpb.2020.105568.

[9]. Maron R.C., et al., Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks // European Journal of Cancer. – 2019. – Vol. 119. – pp.57- 65.

[10]. ISIC Skin image analysis Workshop and Challenge @ MICCAI 2018 [Электронный ресурс]. – Режим доступа: https://workshop2018.isic-archive.com

[11]. W. Gouda, N. U. Sama, G. Al-Waakid, M. Humayun, and N. Z. Jhanjhi, "Detection of skin cancer based on skin lesion images using deep learning," *Healthcare*, vol. 10, no. 7, p. 1183, 2022.

[12]. J. Wu, E. Z. Chen, R. Rong, X. Li, D. Xu, and H. Jiang, "Skin lesion segmentation with C-UNet," in 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 2019. View at: Publisher Site | Google Scholar

[13]. P. Tang, Q. Liang, X. Yan et al., "Efficient skin lesion segmentation using separable-UNet with stochastic weight averaging," Computer Methods and Programs in Biomedicine, vol. 178, pp. 289–301, 2019.

[14]. M. Nawaz, T. Nazir, M. Masood et al., "Melanoma segmentation: a framework of improved DenseNet77 and UNET convolutional neural network," International Journal of Imaging Systems and Technology, vol. 32, no. 6, pp. 2137–2153, 2022. View at: Publisher Site | Google Scholar

[15]. Maurício J., Domingues I., Bernardino J. Comparing vision transformers and convolutional neural networks for image classification: A literature review //Applied Sciences. – 2023. – Т. 13. – №. 9. – С. 5521.

[16]. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J., and Houlsby N., An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021, https://doi.org/10.48550/arXiv.2010.11929

[17]. Bao H., Dong L., and Wei F., BEiT: BERT pre-training of image transformers, 2021, http://arxiv.org/abs/2106.08254.

[18]. Xin C., Liu Z., Zhao K., Miao L., Ma Y., Zhu X., Zhou Q., Wang S., Li L., Yang F., Xu S., and Chen H., An improved transformer network for skin cancer classification, Computers in Biology and Medicine. (2022) 149, article 105939, https://doi.org/10.1016/j.compbiomed.2022.1

[19]. Galib Muhammad Shahriar Himel and others, Skin Cancer Segmentation and Classification Using VisionTransformer for Automatic Analysis in Dermatoscopy-BasedNoninvasive Digital System. (2024), https://doi.org/10.1155/2024/3022192.

[20]. Платформа НЦМУ ИСП РАН. Режим доступа: https://www.sechenov.ru/univers/structure/center/tsentr-tsifrovogo-biodizayna-i-personalizirovannogo-zdravookhraneniya/

Информация об авторах / Information about authors

Александр Васильевич КОЗАЧОК – доктор технических наук, доцент, заведующий лабораторией безопасного программного обеспечения и анализа данных Института системного программирования им. В.П. Иванникова РАН. Сфера научных интересов: методы и системы защиты информации, кибербезопасность, машинное обучение, анализ данных.

Alexander Vasilevich KOZACHOK – Dr. Sci. (Tech.), associate professor, head of the laboratory of secure software and data analysis of the Institute for system programming of the RAS. Research interests: information security methods and systems, cybersecurity, machine learning, data analysis.

Андрей Андреевич СПИРИН – кандидат технических наук, научный сотрудник института системного программирования им. В.П. Иванникова Российской Академии наук. Его научные интересы включают распознавание образов, системы искусственного интеллекта.

Andrei Andreevich SPIRIN – Cand. Sci. (Tech.), research associate of the Ivannikov institute for system programming of the Russian academy of sciences. His research interests include pattern recognition, artificial intelligence systems.

Олег Ильгисович САМОВАРОВ – кандидат технических наук, учёный секретарь института системного программирования им. В.П. Иванникова Российской Академии наук.

Oleg Ilgisovich SAMOVAROV – Cand. Sci. (Tech.), scientific secretary of the Ivannikov institute for system programming of the Russian academy of sciences.

Елена Сергеевна КОЗАЧОК является главным врачом ООО «Бьюти Клиник». Её научные интересы включают вопросы косметологии, дерматологии, трихологии.

Elena Sergeevna KOZACHOK is the chief physician of Beauty Clinic LLC. Her research interests include cosmetology, dermatology, and trichology.