

www.aurora-group.eu

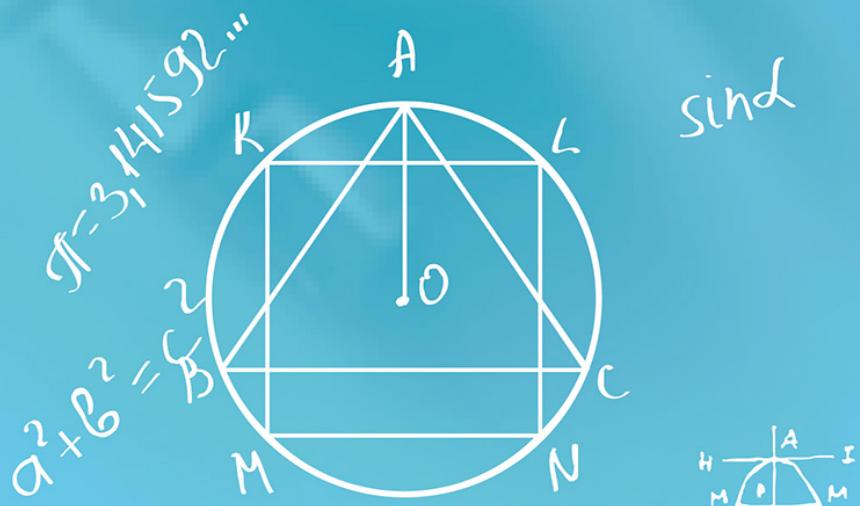
www.nbpublish.com

ПРОГРАММНЫЕ СИСТЕМЫ

и

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ

научный журнал



AURORA Group s.r.o.

nota bene

Выходные данные

Номер подписан в печать: 03-07-2025

Учредитель: Даниленко Василий Иванович, w.danilenko@nbpublish.com

Издатель: ООО <НБ-Медиа>

Главный редактор: Морозов Михаил Николаевич, кандидат технических наук,
mikhail.n.morozov@gmail.com

ISSN: 2454-0714

Контактная информация:

Выпускающий редактор - Зубкова Светлана Вадимовна

E-mail: info@nbpublish.com

тел.+7 (966) 020-34-36

Почтовый адрес редакции: 115114, г. Москва, Павелецкая набережная, дом 6А, офис 211.

Библиотека журнала по адресу: http://www.nbpublish.com/library_tariffs.php

Publisher's imprint

Number of signed prints: 03-07-2025

Founder: Danilenko Vasiliy Ivanovich, w.danilenko@nbpublish.com

Publisher: NB-Media ltd

Main editor: Morozov Mikhail Nikolaevich, kandidat tekhnicheskikh nauk,
mikhail.n.morozov@gmail.com

ISSN: 2454-0714

Contact:

Managing Editor - Zubkova Svetlana Vadimovna

E-mail: info@nbpublish.com

тел.+7 (966) 020-34-36

Address of the editorial board : 115114, Moscow, Paveletskaya nab., 6A, office 211 .

Library Journal at : http://en.nbpublish.com/library_tariffs.php

Редакционный совет

Гельман Виктор Яковлевич – доктор технических наук, профессор, профессор кафедры медицинской информатики и физики ФГБОУ ВО «Северо-Западный государственный медицинский университет им. И.И.Мечникова», 191015, Россия, г. Санкт-Петербург, ул. Кирочная, д.41, gelm@sg2104.spb.edu

Поляков Виктор Павлович – доктор педагогических наук, профессор, Главный научный сотрудник лаборатории психолого-педагогического и учебно-методического обеспечения развития информатизации образования Центра информатизации образования Федерального государственного бюджетного научного учреждения «Институт управления образованием Российской академии образования», 105062, г. Москва, ул. Макаренко, д. 5/16, стр. 1Б, polvikpal@mail.ru

Гармаев Баир Заятуевич – кандидат физико-математических наук, научный сотрудник, Институт физического материаловедения Сибирского Отделения РАН, 670000, Россия, республика Бурятия, г. Улан-Удэ, ул. Сахьяновой, 6, каб. 313

Клименко Анна Борисовна – кандидат технических наук, научный сотрудник Научно-исследовательского института многопроцессорных вычислительных систем имени академика А.В. Каляева Южного федерального университета (НИИ МВС ЮФУ), 347935, Россия, Ростовская область, г. Таганрог, ул. 8 Переулок, 15

Лютикова Лариса Адольфовна – кандидат физико-математических наук, заведующая отделом Нейроинформатики и машинного обучения, Институт прикладной математики и автоматизации Кабардино-Балкарского научного центра РАН – филиал Кабардино-Балкарского научного центра РАН (ИПМА КБНЦ РАН), 360000, Россия, Республика Кабардино-Балкария, г. Нальчик, ул. Шортанова, 89а

Мустафаев Арслан Гасанович – доктор технических наук, профессор, Государственное автономное образовательное учреждение высшего образования "Дагестанский государственный университет народного хозяйства", кафедра «Информационные технологии и информационная безопасность», 367015, Россия, Республика Дагестан, г. Махачкала, ул. Атаева, 5, каб. 4.5

Шестаков Александр Валентинович – кандидат технических наук, доцент Южный Федеральный университет, кафедра вычислительной техники, 347902, Россия, Ростовская область, г. Таганрог, ул. Свободы, 24/2

Сидоркина Ирина Геннадьевна - доктор технических наук, профессор, декан факультета Информатики и вычислительной техники Поволжского государственного технологического университета, Йошкар-Ола, Россия E-mail: dekan_fivt@mail.ru

Екатерина Прасолова-Forland - PhD, Норвежский университет науки и технологии (NTNU), Тронхейм, Норвегия E-mail: Ekaterina.Prasolova-Forland@idi.ntnu.no

Голенков Владимир Васильевич - доктор технических наук, профессор, заведующий кафедрой Интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, г. Минск, Республика Беларусь E-mail: golen@bsuir.by

Домошницкий Александр Исаакович - кандидат физико-математических наук, декан естественно-научного факультета Университетского центра в г.Ариэль, Израиль, Самария E-mail: adom@ariel.ac.il Department of Mathematics and Computer Sciences, The Ariel

University Center of Samaria, 44837 Ariel, ISRAEL

Коробейников Анатолий Григорьевич - доктор технических наук, профессор «Институт земного магнетизма, ионосфера и распространения радиоволн РАН (ИЗМИРАН)», Санкт-Петербургский филиал E-mail: korobeynikov_a_g@mail.ru

Заболеева-Зотова Алла Викторовна, доктор технических наук, профессор Волгоградского технического университета, Волгоград, Россия E-mail: zabzot@gmail.com

Бенкевич Леонид Владимирович - кандидат физических наук и инженерной физики, научный сотрудник Массачусетского Технологического Института (MIT), обсерватория Хэйстек, Бостон, США E-mail: lbenkey@gmail.com

Морозов Михаил Николаевич - кандидат технических наук, профессор, руководитель лаборатории мультимедиа, заведующий кафедрой Информатики и системного программирования Поволжского государственного технологического университета, Йошкар-Ола, Россия E-mail: mikhail.n.morozov@gmail.com

Олзоева Сэсэг Ивановна - доктор технических наук, профессор, Восточно-Сибирский государственный университет технологий и управления (г. Улан-Удэ) E-mail: sseseg@yandex.ru

Курейчик Владимир Викторович - доктор технических наук, профессор, заведующий кафедрой Систем автоматизации проектирования Технологического института «Южного федерального университета» в г.Таганрог, Россия E-mail: vkur@tsure.ru

Филатова Наталья Николаевна - доктор технических наук, профессор, Тверской государственный технический университет, Тверь, Россия E-mail: nfilatova99@mail.ru

Песошин Валерий Андреевич - член-корреспондент Академии наук Республики Татарстан, заслуженный деятель науки Республики Татарстан и Российской Федерации, доктор технических наук, профессор, заслуженный деятель науки и техники Республики Татарстан. Заведующий кафедрой Компьютерных систем Казанского национальный исследовательский университет им. А.Н. Туполева, Казань, Россия E-mail: pesoshin@evm.kstu-kai.ru

Краснов Сергей Викторович - доктор технических наук, профессор, проректор по научно-исследовательской работе, заведующий кафедрой Информатика и системы управления Волжского университета им. Татищева, Тольятти, Россия E-mail: krasnovlt@mail.ru

Горохов Алексей Витальевич - доктор технических наук, профессор кафедры Прикладной математики и информационных технологий Поволжского государственного технологического университета, Йошкар-Ола, Россия E-mail: agv64@mail.ru

Галанина Наталья Андреевна - доктор технических наук, профессор, Чувашский государственный университет им. И.Н.Ульянова, Чебоксары, Россия E-mail: galaninacheb@mail.ru

Сюзев Владимир Васильевич - доктор технических наук, профессор, заведующий кафедрой Компьютерные системы и сети Московского государственного технического университета им. Н. Э. Баумана, Москва, Россия E-mail: v.suzev@bmstu.ru

Леухин Анатолий Николаевич - доктор физико-математических наук, профессор, заведующий кафедрой Информационной безопасности Поволжского государственного технологического университета, Йошкар-Ола, Россия E-mail: code@volgatech.net

Гвинианидзе Темур Николаевич - Доктор технических наук, профессор,
Государственный университет им. Ак. Церетели Грузия, г. Кутаиси, пр. Тамар-мепе 59.
П.и 4600. temuri1951@mail.ru

Council of Editors

Gelman Viktor Yakovlevich – Doctor of Technical Sciences, Professor, Professor of the Department of Medical Informatics and Physics of the I.I.Mechnikov Northwestern State Medical University, 41 Kirochnaya Str., St. Petersburg, 191015, Russia, gelm@sg2104.spb.edu

Polyakov Viktor Pavlovich – Doctor of Pedagogical Sciences, Professor, Chief Researcher of the Laboratory of Psychological, Pedagogical and Educational methodological support for the development of Informatization of Education of the Center for Informatization of Education of the Federal State Budgetary Scientific Institution "Institute of Education Management of the Russian Academy of Education", 105062, Moscow, Makarenko str., 5/16, p. 1B, polvikpal@mail.ru

Garmaev Bair Zayatuevich – Candidate of Physical and Mathematical Sciences, Researcher, Institute of Physical Materials Science of the Siberian Branch of the Russian Academy of Sciences, 670000, Russia, Republic of Buryatia, Ulan-Ude, Sakhyanova str., 6, room 313

Klimenko Anna Borisovna – Candidate of Technical Sciences, Researcher at the Research Institute of Multiprocessor Computing Systems named after Academician A.V. Kalyaev of the Southern Federal University (Research Institute of the Ministry of Internal Affairs of the Southern Federal University), 347935, Russia, Rostov region, Taganrog, ul. 8 Lane, 15

Lyutikova Larisa Adolfovna – Candidate of Physical and Mathematical Sciences, Head of the Department of Neuroinformatics and Machine Learning, Institute of Applied Mathematics and Automation of the Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences - branch of the Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences (IPMA KBSC RAS), 360000, Russia, Republic of Kabardino-Balkaria, Nalchik, 89a Shortanova str.

Mustafayev Arslan Hasanovich – Doctor of Technical Sciences, Professor, State Autonomous Educational Institution of Higher Education "Dagestan State University of National Economy", Department of "Information Technologies and Information Security", 367015, Russia, Republic of Dagestan, Makhachkala, Ataeva str., 5, office 4.5

Alexander V. Shestakov – Candidate of Technical Sciences, Associate Professor, Southern Federal University, Department of Computer Engineering, 24/2 Svobody str., Taganrog, Rostov Region, 347902, Russia

Sidorkina Irina Gennadievna - Doctor of Technical Sciences, Professor, Dean of the Faculty of Computer Science and Computer Engineering of the Volga State Technological University, Yoshkar-Ola, Russia E-mail: dekan_fivt@mail.ru

Ekaterina Prasolova-Forland - PhD, Norwegian University of Science and Technology (NTNU), Trondheim, Norway E-mail: Ekaterina.Prasolova-Forland@idi.ntnu.no

Golenkov Vladimir Vasilyevich - Doctor of Technical Sciences, Professor, Head of the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus E-mail: golen@bsuir.by

Domoshnitsky Alexander Isaakovich - Candidate of Physical and Mathematical Sciences, Dean of the Faculty of Natural Sciences of the University Center in Ariel, Israel, Samaria E-mail: adom@ariel.ac.il Department of Mathematics and Computer Sciences, The Ariel University Center of Samaria, 44837 Ariel, ISRAEL

Korobeynikov Anatoly Grigorievich - Doctor of Technical Sciences, Professor, Institute of

Terrestrial Magnetism, Ionosphere and Radio Wave Propagation of the Russian Academy of Sciences (IZMIRAN), St. Petersburg Branch E-mail: korobeynikov_a_g@mail.ru

Zaboleeva-Zotova Alla Viktorovna, Doctor of Technical Sciences, Professor of Volgograd Technical University, Volgograd, Russia E-mail: zabzot@gmail.com

Leonid V. Benkevich - Candidate of Physical Sciences and Engineering Physics, Researcher at the Massachusetts Institute of Technology (MIT), Haystack Observatory, Boston, USA E-mail: ibenkev@gmail.com

Mikhail N. Morozov - Candidate of Technical Sciences, Professor, Head of the Multimedia Laboratory, Head of the Department of Computer Science and System Programming of the Volga State Technological University, Yoshkar-Ola, Russia E-mail: mikhail.n.morozov@gmail.com

Olzoeva Seseg Ivanovna - Doctor of Technical Sciences, Professor, East Siberian State University of Technology and Management (Ulan-Ude) E-mail: sseseg@yandex.ru

Kureychik Vladimir Viktorovich - Doctor of Technical Sciences, Professor, Head of the Department of Design Automation Systems of the Technological Institute of the Southern Federal University in Taganrog, Russia E-mail: vkur@tsure.ru

Natalia Filatova - Doctor of Technical Sciences, Professor, Tver State Technical University, Tver, Russia E-mail: nfilatova99@mail.ru

Pesoshin Valery Andreevich - Corresponding member of the Academy of Sciences of the Republic of Tatarstan, Honored Scientist of the Republic of Tatarstan and the Russian Federation, Doctor of Technical Sciences, Professor, Honored Worker of Science and Technology of the Republic of Tatarstan. Head of the Department of Computer Systems of Kazan National Research University named after A.N. Tupolev, Kazan, Russia E-mail: pesoshin@evm.kstu-kai.ru

Krasnov Sergey Viktorovich - Doctor of Technical Sciences, Professor, Vice-Rector for Research, Head of the Department of Computer Science and Control Systems of the Volga State University. Tatishcheva, Togliatti, Russia E-mail: krasnovtlt@mail.ru

Gorokhov Alexey Vitalievich - Doctor of Technical Sciences, Professor of the Department of Applied Mathematics and Information Technologies of the Volga State Technological University, Yoshkar-Ola, Russia E-mail: agv64@mail.ru

Galanina Natalia Andreevna - Doctor of Technical Sciences, Professor, I.N.Ulyanov Chuvash State University, Cheboksary, Russia E-mail: galaninacheb@mail.ru

Vladimir V. Syuzev - Doctor of Technical Sciences, Professor, Head of the Department of Computer Systems and Networks of the Bauman Moscow State Technical University, Moscow, Russia E-mail: v.suzev@bmstu.ru

Leukhin Anatoly Nikolaevich - Doctor of Physical and Mathematical Sciences, Professor, Head of the Department of Information Security of the Volga State Technological University, Yoshkar-Ola, Russia E-mail: code@volgattech.net

Gvinianidze Temur Nikolaevich - Doctor of Technical Sciences, Professor, Ak. Tsereteli State University Georgia, Kutaisi, 59 Tamar-mepe Ave., and 4600. temuri1951@mail.ru

Требования к статьям

Журнал является научным. Направляемые в издательство статьи должны соответствовать тематике журнала (с его рубрикатором можно ознакомиться на сайте издательства), а также требованиям, предъявляемым к научным публикациям.

Рекомендуемый объем от 12000 знаков.

Структура статьи должна соответствовать жанру научно-исследовательской работы. В ее содержании должны обязательно присутствовать и иметь четкие смысловые разграничения такие разделы, как: предмет исследования, методы исследования, апелляция к оппонентам, выводы и научная новизна.

Не приветствуется, когда исследователь, трактуя в статье те или иные научные термины, вступает в заочную дискуссию с авторами учебников, учебных пособий или словарей, которые в узких рамках подобных изданий не могут широко излагать свое научное воззрение и заранее оказываются в проигрышном положении. Будет лучше, если для научной полемики Вы обратитесь к текстам монографий или докторских диссертаций работ оппонентов.

Не превращайте научную статью в публицистическую: не наполняйте ее цитатами из газет и популярных журналов, ссылками на высказывания по телевидению.

Ссылки на научные источники из Интернета допустимы и должны быть соответствующим образом оформлены.

Редакция отвергает материалы, напоминающие реферат. Автору нужно не только продемонстрировать хорошее знание обсуждаемого вопроса, работ ученых, исследовавших его прежде, но и привнести своей публикацией определенную научную новизну.

Не принимаются к публикации избранные части из докторских диссертаций, книг, монографий, поскольку стиль изложения подобных материалов не соответствует журнальному жанру, а также не принимаются материалы, публиковавшиеся ранее в других изданиях.

В случае отправки статьи одновременно в разные издания автор обязан известить об этом редакцию. Если он не сделал этого заблаговременно, рискует репутацией: в дальнейшем его материалы не будут приниматься к рассмотрению.

Уличенные в плагиате попадают в «черный список» издательства и не могут рассчитывать на публикацию. Информация о подобных фактах передается в другие издательства, в ВАК и по месту работы, учебы автора.

Статьи представляются в электронном виде только через сайт издательства <http://www.enotabene.ru> кнопка "Авторская зона".

Статьи без полной информации об авторе (соавторах) не принимаются к рассмотрению, поэтому автор при регистрации в авторской зоне должен ввести полную и корректную информацию о себе, а при добавлении статьи - о всех своих соавторах.

Не набирайте название статьи прописными (заглавными) буквами, например: «ИСТОРИЯ КУЛЬТУРЫ...» — неправильно, «История культуры...» — правильно.

При добавлении статьи необходимо прикрепить библиографию (минимум 10–15 источников, чем больше, тем лучше).

При добавлении списка использованной литературы, пожалуйста, придерживайтесь следующих стандартов:

- [ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления.](#)
- [ГОСТ 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления](#)

В каждой ссылке должен быть указан только один диапазон страниц. В теле статьи ссылка на источник из списка литературы должна быть указана в квадратных скобках, например, [1]. Может быть указана ссылка на источник со страницей, например, [1, с. 57], на группу источников, например, [1, 3], [5-7]. Если идет ссылка на один и тот же источник, то в теле статьи нумерация ссылок должна выглядеть так: [1, с. 35]; [2]; [3]; [1, с. 75-78]; [4]....

А в библиографии они должны отображаться так:

[1]
[2]
[3]
[4]....

Постраничные ссылки и сноски запрещены. Если вы используете сноски, не содержащую ссылку на источник, например, разъяснение термина, включите сноски в текст статьи.

После процедуры регистрации необходимо прикрепить аннотацию на русском языке, которая должна состоять из трех разделов: Предмет исследования; Метод, методология исследования; Новизна исследования, выводы.

Прикрепить 10 ключевых слов.

Прикрепить саму статью.

Требования к оформлению текста:

- Кавычки даются углками (« ») и только кавычки в кавычках — лапками (“ ”).
- Тире между датамидается короткое (Ctrl и минус) и без отбивок.
- Тире во всех остальных случаяхдается длинное (Ctrl, Alt и минус).
- Даты в скобках даются без г.: (1932–1933).
- Даты в тексте даются так: 1920 г., 1920-е гг., 1540–1550-е гг.
- Недопустимо: 60-е гг., двадцатые годы двадцатого столетия, двадцатые годы XX столетия, 20-е годы ХХ столетия.
- Века, король такой-то и т.п. даются римскими цифрами: XIX в., Генрих IV.
- Инициалы и сокращения даются с пробелом: т. е., т. д., М. Н. Иванов. Неправильно: М.Н. Иванов, М.Н. Иванов.

ВСЕ СТАТЬИ ПУБЛИКУЮТСЯ В АВТОРСКОЙ РЕДАКЦИИ.

По вопросам публикации и финансовым вопросам обращайтесь к администратору Зубковой Светлане Вадимовне
E-mail: info@nbpublish.com
или по телефону +7 (966) 020-34-36

Подробные требования к написанию аннотаций:

Аннотация в периодическом издании является источником информации о содержании статьи и изложенных в ней результатах исследований.

Аннотация выполняет следующие функции: дает возможность установить основное

содержание документа, определить его релевантность и решить, следует ли обращаться к полному тексту документа; используется в информационных, в том числе автоматизированных, системах для поиска документов и информации.

Аннотация к статье должна быть:

- информативной (не содержать общих слов);
- оригинальной;
- содержательной (отражать основное содержание статьи и результаты исследований);
- структурированной (следовать логике описания результатов в статье);

Аннотация включает следующие аспекты содержания статьи:

- предмет, цель работы;
- метод или методологию проведения работы;
- результаты работы;
- область применения результатов; новизна;
- выводы.

Результаты работы описывают предельно точно и информативно. Приводятся основные теоретические и экспериментальные результаты, фактические данные, обнаруженные взаимосвязи и закономерности. При этом отдается предпочтение новым результатам и данным долгосрочного значения, важным открытиям, выводам, которые опровергают существующие теории, а также данным, которые, по мнению автора, имеют практическое значение.

Выводы могут сопровождаться рекомендациями, оценками, предложениями, гипотезами, описанными в статье.

Сведения, содержащиеся в заглавии статьи, не должны повторяться в тексте аннотации. Следует избегать лишних вводных фраз (например, «автор статьи рассматривает...», «в статье рассматривается...»).

Исторические справки, если они не составляют основное содержание документа, описание ранее опубликованных работ и общеизвестные положения в аннотации не приводятся.

В тексте аннотации следует употреблять синтаксические конструкции, свойственные языку научных и технических документов, избегать сложных грамматических конструкций.

Гонорары за статьи в научных журналах не начисляются.

Цитирование или воспроизведение текста, созданного ChatGPT, в вашей статье

Если вы использовали ChatGPT или другие инструменты искусственного интеллекта в своем исследовании, опишите, как вы использовали этот инструмент, в разделе «Метод» или в аналогичном разделе вашей статьи. Для обзоров литературы или других видов эссе, ответов или рефератов вы можете описать, как вы использовали этот инструмент, во введении. В своем тексте предоставьте prompt - командный вопрос, который вы использовали, а затем любую часть соответствующего текста, который был создан в ответ.

К сожалению, результаты «чата» ChatGPT не могут быть получены другими читателями, и хотя невосстановимые данные или цитаты в статьях APA Style обычно цитируются как личные сообщения, текст, сгенерированный ChatGPT, не является сообщением от человека.

Таким образом, цитирование текста ChatGPT из сеанса чата больше похоже на совместное использование результатов алгоритма; таким образом, сделайте ссылку на автора алгоритма записи в списке литературы и приведите соответствующую цитату в тексте.

Пример:

На вопрос «Является ли деление правого полушария левого полушария реальным или метафорой?» текст, сгенерированный ChatGPT, показал, что, хотя два полушария мозга в некоторой степени специализированы, «обозначение, что люди могут быть охарактеризованы как «левополушарные» или «правополушарные», считается чрезмерным упрощением и популярным мифом» (OpenAI, 2023).

Ссылка в списке литературы

OpenAI. (2023). ChatGPT (версия от 14 марта) [большая языковая модель].
<https://chat.openai.com/chat>

Вы также можете поместить полный текст длинных ответов от ChatGPT в приложение к своей статье или в дополнительные онлайн-материалы, чтобы читатели имели доступ к точному тексту, который был сгенерирован. Особенno важно задокументировать созданный текст, потому что ChatGPT будет генерировать уникальный ответ в каждом сеансе чата, даже если будет предоставлен один и тот же командный вопрос. Если вы создаете приложения или дополнительные материалы, помните, что каждое из них должно быть упомянуто по крайней мере один раз в тексте вашей статьи в стиле APA.

Пример:

При получении дополнительной подсказки «Какое представление является более точным?» в тексте, сгенерированном ChatGPT, указано, что «разные области мозга работают вместе, чтобы поддерживать различные когнитивные процессы» и «функциональная специализация разных областей может меняться в зависимости от опыта и факторов окружающей среды» (OpenAI, 2023; см. Приложение А для полной расшифровки). .

Ссылка в списке литературы

OpenAI. (2023). ChatGPT (версия от 14 марта) [большая языковая модель].
<https://chat.openai.com/chat> Создание ссылки на ChatGPT или другие модели и программное обеспечение ИИ

Приведенные выше цитаты и ссылки в тексте адаптированы из шаблона ссылок на программное обеспечение в разделе 10.10 Руководства по публикациям (Американская психологическая ассоциация, 2020 г., глава 10). Хотя здесь мы фокусируемся на ChatGPT, поскольку эти рекомендации основаны на шаблоне программного обеспечения, их можно адаптировать для учета использования других больших языковых моделей (например, Bard), алгоритмов и аналогичного программного обеспечения.

Ссылки и цитаты в тексте для ChatGPT форматируются следующим образом:

OpenAI. (2023). ChatGPT (версия от 14 марта) [большая языковая модель].
<https://chat.openai.com/chat>

Цитата в скобках: (OpenAI, 2023)

Описательная цитата: OpenAI (2023)

Давайте разберем эту ссылку и посмотрим на четыре элемента (автор, дата, название и

источник):

Автор: Автор модели OpenAI.

Дата: Дата — это год версии, которую вы использовали. Следуя шаблону из Раздела 10.10, вам нужно указать только год, а не точную дату. Номер версии предоставляет конкретную информацию о дате, которая может понадобиться читателю.

Заголовок. Название модели — «ChatGPT», поэтому оно служит заголовком и выделено курсивом в ссылке, как показано в шаблоне. Хотя OpenAI маркирует уникальные итерации (например, ChatGPT-3, ChatGPT-4), они используют «ChatGPT» в качестве общего названия модели, а обновления обозначаются номерами версий.

Номер версии указан после названия в круглых скобках. Формат номера версии в справочниках ChatGPT включает дату, поскольку именно так OpenAI маркирует версии. Различные большие языковые модели или программное обеспечение могут использовать различную нумерацию версий; используйте номер версии в формате, предоставленном автором или издателем, который может представлять собой систему нумерации (например, Версия 2.0) или другие методы.

Текст в квадратных скобках используется в ссылках для дополнительных описаний, когда они необходимы, чтобы помочь читателю понять, что цитируется. Ссылки на ряд общих источников, таких как журнальные статьи и книги, не включают описания в квадратных скобках, но часто включают в себя вещи, не входящие в типичную рецензируемую систему. В случае ссылки на ChatGPT укажите дескриптор «Большая языковая модель» в квадратных скобках. OpenAI описывает ChatGPT-4 как «большую мультимодальную модель», поэтому вместо этого может быть предоставлено это описание, если вы используете ChatGPT-4. Для более поздних версий и программного обеспечения или моделей других компаний могут потребоваться другие описания в зависимости от того, как издатели описывают модель. Цель текста в квадратных скобках — кратко описать тип модели вашему читателю.

Источник: если имя издателя и имя автора совпадают, не повторяйте имя издателя в исходном элементе ссылки и переходите непосредственно к URL-адресу. Это относится к ChatGPT. URL-адрес ChatGPT: <https://chat.openai.com/chat>. Для других моделей или продуктов, для которых вы можете создать ссылку, используйте URL-адрес, который ведет как можно более напрямую к источнику (т. е. к странице, на которой вы можете получить доступ к модели, а не к домашней странице издателя).

Другие вопросы о цитировании ChatGPT

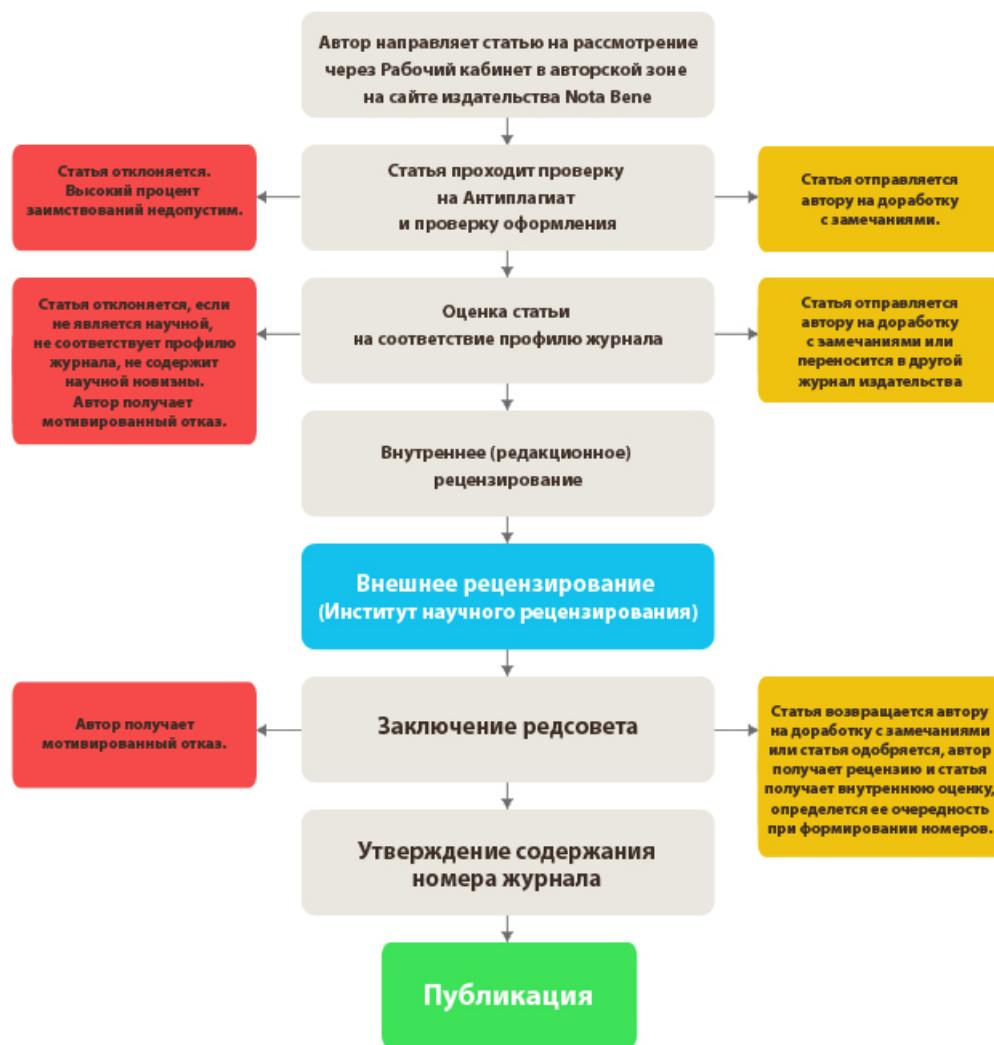
Вы могли заметить, с какой уверенностью ChatGPT описал идеи латерализации мозга и то, как работает мозг, не ссылаясь ни на какие источники. Я попросил список источников, подтверждающих эти утверждения, и ChatGPT предоставил пять ссылок, четыре из которых мне удалось найти в Интернете. Пятая, похоже, не настоящая статья; идентификатор цифрового объекта, указанный для этой ссылки, принадлежит другой статье, и мне не удалось найти ни одной статьи с указанием авторов, даты, названия и сведений об источнике, предоставленных ChatGPT. Авторам, использующим ChatGPT или аналогичные инструменты искусственного интеллекта для исследований, следует подумать о том, чтобы сделать эту проверку первоисточников стандартным процессом. Если источники являются реальными, точными и актуальными, может быть лучше прочитать эти первоисточники, чтобы извлечь уроки из этого исследования, и перефразировать или процитировать эти статьи, если применимо, чем использовать их интерпретацию модели.

Материалы журналов включены:

- в систему Российского индекса научного цитирования;
- отображаются в крупнейшей международной базе данных периодических изданий Ulrich's Periodicals Directory, что гарантирует значительное увеличение цитируемости;
- Всем статьям присваивается уникальный идентификационный номер Международного регистрационного агентства DOI Registration Agency. Мы формируем и присваиваем всем статьям и книгам, в печатном, либо электронном виде, оригинальный цифровой код. Префикс и суффикс, будучи прописанными вместе, образуют определяемый, цитируемый и индексируемый в поисковых системах, цифровой идентификатор объекта — digital object identifier (DOI).

[Отправить статью в редакцию](#)

Этапы рассмотрения научной статьи в издательстве NOTA BENE.



Содержание

Шаброва А.С., Князев М.А., Колесников А.В. Динамическое распределение слотов RACH для минимизации коллизий в сетях NB-IoT на основе алгоритмов обучения с подкреплением	1
Карпович В.Д., Государев И.Б. Исследование производительности WebAssembly в среде исполнения Node.js	12
Бондаренко О.С. Анализ методов обновления DOM в современных веб-фреймворках: Virtual DOM и Incremental DOM	35
Крохин А.С., Гусев М.М. Анализ влияния обfuscации входных данных на эффективность языковых моделей в обнаружении инъекции подсказок	44
Ратушняк Е.А. Исследование производительности современных клиентских веб-фреймворков	63
Чикалева Ю.С. Анализ гранулярности микросервисов: эффективность архитектурных подходов	79
Рогов Д.В., Аллатов А.Н. Интеллектуальная инфраструктура автоматизированного управления и интероперабельности микросервисов в облачных средах	94
Дагаев Д.В. Ареальные типы данных в инструментальном подходе к программированию	115
Макаров И.С., Райков А.В., Казанцев А.А., Нехаев М.В., Романов М.А. Применение нейросетей для анализа больших данных в реальном времени	132
Петровский А.А., Рысин М.Л. Реализация поведения Drag&Drop в Android-приложении на основе API обработки жестов	148
Душкин Р.В., Климов В.В. Имитационное моделирование функционального двойника системы управления микроклиматом интеллектуального здания	165
Ковалев С.В., Смирнова Т.Н., Зверев Р.Е., Раков И.В. Разработка проекта PLAY VISION AI для просмотра спортивных матчей с помощью искусственного интеллекта	175
Козырева Н.И., Мухтулов М.О., Ершов С.А., Новосельцева С.В., Ахмадуллин Д.А. Современные методы предотвращения DDoS-атак и защиты веб-серверов	190
Родионов Д.Г., Сергеев Д.А., Конников Е.А., Пашинина П.А. Анализ пространственно-временных закономерностей движения на аэрофотоснимках с использованием оптического потока	204
Родионов Д.Г., Сергеев Д.А., Конников Е.А., Попова С.Д. Метод анализа аэрофотоснимков с БПЛА на основе SSIM и MSE для оценки надежности технических систем	217
Англоязычные метаданные	231

Contents

Shabrova A.S., Knyazev M.A., Kolesnikov A.V. Dynamic RACH-Slot Allocation for Collision Minimization in NB-IoT Networks Based on Reinforcement Learning Algorithms	1
Karpovich V.D., Gosudarev I.B. WebAssembly performance in the Node.js environment	12
Bondarenko O.S. Analysis of DOM update methods in modern web frameworks: Virtual DOM and Incremental DOM	35
Krohin A.S., Gusev M.M. Analysis of the impact of prompt obfuscation on the effectiveness of language models in detecting prompt injections	44
Ratushniak E.A. Research on performance in modern client-side web-frameworks.	63
Chikaleva Y.S. Analysis of Microservices Granularity: Effectiveness of Architectural Approaches	79
Rogov D.V., Alpatov A.N. Intellectual infrastructure for automated control and interoperability of microservices in cloud environments	94
Dagaev D.V. Areal data types in instrumental approach to programming	115
Makarov I.S., Raikov A.V., Kazantsev A.A., Nekhaev M.V., Romanov M.A. The use of neural networks for real-time big data analysis	132
Petrovsky A.A., Rysin M.L. Implementation of Drag&Drop behavior in an Android application based on the gesture processing API.	148
Dushkin R.V., Klimov V.V. Simulation modeling of the functional twin of the microclimate control system of an intelligent building.	165
Kovalev S.V., Smirnova T.N., Zverev R.E., Rakov I.V. Development of the PLAY VISION AI project for watching sports matches using artificial intelligence	175
Kozyreva N.I., Muhtulov M.O., Ershov S.A., Novoseltseva S.V., Akhmadullin D.A. Modern methods of preventing DDoS attacks and protecting web servers	190
Rodionov D., Sergeev D.A., Konnikov E.A., Pashinina P.A. Analysis of Spatiotemporal Motion Patterns in Aerial Images Using Optical Flow	204
Rodionov D., Sergeev D.A., Konnikov E.A., Popova S.D. Method of UAV Aerial Image Analysis Based on SSIM and MSE for Assessing the Reliability of Technical Systems	217
Metadata in english	231

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Шаброва А.С., Князев М.А., Колесников А.В. Динамическое распределение слотов RACH для минимизации коллизий в сетях NB-IoT на основе алгоритмов обучения с подкреплением // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.73848 EDN: TPWJOS URL: https://nbpublish.com/library_read_article.php?id=73848

Динамическое распределение слотов RACH для минимизации коллизий в сетях NB-IoT на основе алгоритмов обучения с подкреплением**Шаброва Анна Сергеевна**

ORCID: 0009-0009-1675-1558

студент; кафедра Информационная безопасность; МГТУ им. Н.Э. Баумана

105005, Россия, г. Москва, ул. 2-я Бауманская, 5, стр. 4

✉ shabrova.anna.2410@list.ru**Князев Максим Андреевич**

ORCID: 0009-0007-3931-7442

аспирант; кафедра информационной безопасности ; ФГБОУ ВО «МИРЭА – Российский технологический университет», Институт Искусственного Интеллекта

119454, Россия, г. Москва, пр-т Вернадского, 78

✉ maxiknyaz@mail.ru**Колесников Александр Владимирович**

ORCID: 0009-0008-9669-804X

к.т.н.; кафедра Информационная безопасность; МГТУ им. Н.Э. Баумана

105005, Россия, г. Москва, Басманный р-н, ул. 2-я Бауманская, д. 5 стр. 4

✉ avkolesnikov@bmstu.ru[Статья из рубрики "Телекоммуникационные системы и компьютерные сети"](#)**DOI:**

10.7256/2454-0714.2025.2.73848

EDN:

TPWJOS

Дата направления статьи в редакцию:

27-03-2025

Дата публикации:

10-04-2025

Аннотация: Предметом исследования является управление доступом к каналам случайного доступа (Random Access Channel, RACH) в сетях узкополосного Интернета вещей (Narrowband Internet of Things, NB-IoT), испытывающих перегрузки при высокой плотности размещения устройств. Объектом исследования выступают процедуры динамического распределения слотов RACH с использованием методов машинного обучения. Основное внимание уделяется применению алгоритмов обучения с подкреплением (Reinforcement Learning, RL), в частности методов Q-learning и Deep Q-Network (DQN). Авторы детально рассматривают проблему перегрузки каналов и связанных с ней коллизий, приводящих к задержкам передачи данных и повышенному энергопотреблению устройств. Анализируется недостаточная эффективность традиционных статических методов управления слотами и обосновывается необходимость внедрения динамического подхода, способного адаптироваться к изменяющимся условиям сети. В исследовании использовались методы машинного обучения, включающие Q-learning и DQN, а также симуляционное моделирование в среде NS-3 с интеграцией RL-агента для динамического перераспределения RACH-слотов. Научная новизна исследования заключается в разработке и интеграции специализированного RL-агента, позволяющего осуществлять адаптивное распределение слотов RACH на основе текущего состояния сети NB-IoT. Основными выводами проведённого исследования являются подтверждённая высокая эффективность предложенного динамического подхода, обеспечившего снижение числа коллизий на 74%, увеличение числа успешных подключений на 16% и улучшение энергоэффективности устройств на 15% по сравнению с традиционными статическими методами. Перспективность дальнейших исследований состоит в масштабировании предложенного подхода на сети с большим числом устройств, изучении мультиагентных RL-подходов, экспериментальной проверке и интеграции разработанного метода с реальными сетями NB-IoT, а также в разработке гибридных моделей управления, сочетающих алгоритмы обучения с подкреплением с другими методами машинного обучения.

Ключевые слова:

NB-IoT, Reinforcement Learning, Q-learning, DQN, коллизии, NS-3, обучение с подкреплением, RACH, IoT, Интернет вещей

ВВЕДЕНИЕ

Современные тенденции цифровой трансформации привели к интенсивному росту количества устройств Интернета вещей (Internet of Things, IoT), которые активно используются в различных областях – от бытовых устройств до промышленных и инфраструктурных решений. Одной из наиболее востребованных технологий для обеспечения коммуникации таких устройств является стандарт Narrowband Internet of Things (NB-IoT), отличающийся низкими скоростями передачи данных и высокой энергоэффективностью. Однако увеличение плотности размещения устройств усложнило работу процедур случайного доступа (Random Access Channel, RACH), предназначенных для установления первичной связи устройств с базовыми станциями [\[4\]](#).

Традиционные статические методы распределения временных слотов, используемых в процедурах RACH, перестают отвечать современным требованиям по причине недостаточной адаптивности к динамическим изменениям нагрузки и непредсказуемости поведения большого количества устройств [5]. Следствием этого является увеличение числа коллизий, рост задержек при передаче данных и повышенное энергопотребление устройств, что в совокупности негативно отражается на общей производительности и надежности функционирования сети [3].

В связи с этим возникает необходимость в разработке и внедрении новых адаптивных механизмов, способных оперативно реагировать на изменяющиеся условия сетевой нагрузки и эффективно минимизировать вероятность возникновения коллизий [6]. Одним из перспективных направлений решения данной задачи является применение алгоритмов обучения с подкреплением (Reinforcement Learning, RL), которые позволяют системам принимать оптимальные решения на основе накопленного опыта взаимодействия с окружающей средой [8].

Цель настоящего исследования заключается в разработке и экспериментальном подтверждении эффективности применения RL-алгоритмов (Q-learning и DQN) для динамического управления процедурой распределения временных слотов RACH в сетях NB-IoT. В работе будет проведён сравнительный анализ результатов практического применения предложенного подхода с традиционным статическим методом по эксплуатационным метрикам числа коллизий, успешности подключения устройств, задержки передачи данных и показателям энергопотребления.

Результаты исследования могут быть использованы для оптимизации функционирования реальных NB-IoT сетей, повышения их производительности и энергоэффективности, а также служить основой для дальнейших исследований в области адаптивного и интеллектуального управления сетями IoT.

ТЕКУЩЕЕ СОСТОЯНИЕ ТЕХНОЛОГИИ

Narrowband Internet of Things (NB-IoT) представляет собой специализированный стандарт, разработанный консорциумом 3rd Generation Partnership Project (3GPP), предназначенный для передачи небольших объёмов данных с высокой энергоэффективностью и низкими требованиями к полосе пропускания [9]. Технология позволяет обслуживать большое количество устройств, даже в условиях затрудненного покрытия, таких как подвальные помещения и удалённые сельские территории, что увеличивает актуальность решения проблемы коллизий. Проблема минимизации коллизий в сетях NB-IoT является актуальной и активно изучается многими исследователями ввиду растущего числа устройств, подключаемых к сетям IoT [11].

В мировой практике для минимизации коллизий применяются как алгоритмические, так и технологические методы. Одним из традиционных подходов является модификация алгоритмов случайного доступа путём интеграции механизмов динамического управления интервалами повторных попыток и перераспределения временных слотов.

Однако в последние годы особое внимание уделяется применению методов машинного обучения, в частности обучения с подкреплением (Reinforcement Learning, RL), для решения задачи перегрузки и минимизации коллизий. RL является методом машинного обучения, где агент обучается взаимодействовать с окружающей средой, принимая оптимальные решения на основе накопленного опыта с целью максимизации получаемой

награды [8]. Популярными алгоритмами в этом направлении являются Q-learning и Deep Q-Network (DQN), которые позволяют эффективно адаптировать стратегии управления ресурсами сети в зависимости от текущего состояния и динамики нагрузки. Исследование «Transmission Control in NB-IoT With Model-Based Reinforcement Learning» подтверждает эффективность применения RL для адаптивного управления слотами доступа, существенно снижая вероятность возникновения коллизий [7]. В работе «Analysis of the Effect of the Reliability of the NB-IoT Network on the Intelligent System» представлен детальный анализ и конкретные рекомендации по снижению задержек и коллизий с использованием подобных подходов [2].

Также перспективным направлением считается мультиагентный подход, основанный на распределённом управлении сетевыми ресурсами, где множество агентов совместно принимают решения, адаптируясь к изменениям нагрузки и неопределённости среды.

Несмотря на многообразие предложенных методов, актуальной остаётся задача создания универсальных, масштабируемых и высокоэффективных решений, пригодных для практического применения в реальных условиях эксплуатации сетей NB-IoT. В контексте обозначенной задачи подход, основанный на RL-алгоритмах, представленный в настоящем исследовании, является перспективным направлением для эффективного и адаптивного управления сетевыми ресурсами и минимизации числа коллизий.

Задачи и инструменты

Для достижения поставленной цели исследования были сформулированы и решены следующие задачи:

- проведён анализ текущего состояния технологии NB-IoT и выявлены основные проблемы, связанные с процедурами случайного доступа;
- спроектирована модель сети NB-IoT, состоящая из трёх различных типов устройств, отражающих специфику их применения;
- выполнено моделирование работы сети и поведения устройств с применением дискретно-событийного симулятора NS-3;
- разработан RL-агент для динамического управления распределением временных слотов доступа;
- проведена серия экспериментов с использованием RL-агента и без его применения (с традиционным статическим подходом);
- зафиксированы и обработаны результаты экспериментальных исследований;
- выполнен сравнительный анализ результатов по числу коллизий, статусу подключений, задержке передачи данных и энергопотреблению устройств;
- сформулированы выводы о преимуществах и эффективности предложенного подхода на основе проведённого анализа.

В исследовании использовался дискретно-событийный сетевой симулятор NS-3 – широко используемая платформа с открытым исходным кодом, написанная на языке C++ с поддержкой интеграции модулей на Python. NS-3 позволил реализовать детализированную модель NB-IoT сети, отражающую реалистичные условия её функционирования, включая различные типы устройств и паттерны генерации трафика.

Благодаря высокой точности моделирования была осуществлена серия симуляций, что позволило собрать и проанализировать данные.

РАЗРАБОТКА МОДЕЛИ СЕТИ NB-IOT И ИНТЕГРАЦИЯ RL-АГЕНТА

Для разработки детализированной модели сети NB-IoT в среде NS-3 было выполнено несколько последовательных этапов моделирования и конфигурации. На начальном этапе был сформирован набор узлов сети: создано 50 независимых узлов, соответствующих устройствам NB-IoT, и отдельный узел, выполняющий функции базовой станции. Узлы были объединены в единую сетевую топологию типа «точка-точка» (Point-to-Point), при которой каждое устройство напрямую взаимодействует с базовой станцией через отдельный канал связи [11].

Для дополнительного контроля корректности реализации модели и наглядной демонстрации процессов обмена данными был использован интегрированный в NS-3 инструмент визуализации NetAnim. Данный инструмент позволил в режиме реального времени визуально представить сформированную топологию сети, расположение устройств и базовой станции, а также пронаблюдать динамические процессы взаимодействия узлов и обмена пакетами данных между ними (см. рисунок 1).



Рисунок 1 – Визуализация топологии сети

Конфигурация используемых каналов связи между узлами была задана с высокой пропускной способностью в 1 Гбит/с и минимальной задержкой передачи данных в 1 мс. Такие параметры были выбраны для того, чтобы свести к минимуму влияние канальных характеристик на результаты симуляции и обеспечить возможность сфокусироваться на исследовании эффективности работы механизма RACH. После завершения конфигурации каналов связи на всех узлах был установлен стек интернет-протоколов IPv4, обеспечивающий корректную передачу пакетов данных по сети.

Для реализации механизма случайного доступа RACH был использован специально разработанный компонент DynamicSlotManager. Данный компонент моделирует реальное поведение устройств NB-IoT в процессе доступа к слотам. В рамках базовой модели были размещены пять временных RACH-слотов с длительностью 10 мс. Устройства осуществляли попытки доступа к слотам и в случае возникновения коллизии переходили в режим ожидания с последующей повторной передачей пакета после случайного временного интервала задержки (backoff).

Для более точного моделирования поведения устройств NB-IoT было разработано специализированное сетевое приложение NbIoTDeviceApp, реализованное в виде

отдельного класса в симуляторе NS-3. Данное приложение позволяет имитировать три различных типа устройств, отличающихся режимами и регулярностью отправки данных:

- периодические устройства (PERIODIC), отправляющие данные с фиксированными интервалами (каждые 60 секунд);
- спорадические устройства (SPORADIC), отправляющие пакеты данных случайным образом и с непредсказуемой частотой, характерной для различных аварийных датчиков и устройств событийного типа;
- устройства с низким приоритетом (LOW_PRIORITY), характеризующиеся нерегулярной отправкой данных и допускающие значительные задержки при передаче информации.

Распределение узлов по указанным типам было задано следующим образом: 30% устройств периодического типа, 40% спорадического и 30% устройств с низким приоритетом. Размер пакетов данных, отправляемых устройствами, составлял 50 байт. Энергетические затраты каждого устройства на отправку одного пакета данных были смоделированы в размере 0.2 Дж, при этом изначальный запас энергии устройств был установлен равным 10 Дж.

Для интеграции алгоритма обучения с подкреплением (RL) в симуляционную модель был разработан специальный внешний Python-модуль rl_agent.py, реализующий логику RL с использованием методов Q-learning и Deep Q-Network (DQN). Взаимодействие между симулятором NS-3 и внешним Python-агентом обеспечивалось через специально созданный интеграционный слой на стороне NS-3 с использованием Python C API. Этот слой осуществлял вызов функций RL-агента [10], передачу агенту данных о текущем состоянии сети (число коллизий, количество успешных и неуспешных попыток подключения, текущее распределение слотов) и получение нового оптимизированного распределения слотов доступа.

На этапе инициализации агент устанавливал начальные параметры обучения (коэффициенты обучения, дисконта, вероятность случайного выбора действий) и формировал начальное равномерное распределение слотов доступа для каждого устройства.

Каждые 30 секунд агент получал от симулятора NS-3 актуальные метрики состояния сети и на их основе оценивал текущее состояние системы.

Используя стратегию ϵ -жадности, агент выбирал действие, направленное на оптимизацию текущей ситуации (уменьшение коллизий, повышение успешности подключений и энергоэффективности).

После выбора действия RL-агент рассчитывал новое оптимальное распределение слотов доступа, передавая полученные конфигурации обратно в симулятор NS-3, где они непосредственно применялись в работе устройств сети.

После применения новых конфигураций RL-агент фиксировал изменения состояния сети, рассчитывал полученную награду и корректировал значения Q-функции, готовясь к следующему циклу анализа.

Экспериментальные исследования проводились на интервале в 3600 секунд (один час), в течение которого каждые 10 секунд фиксировались метрики состояния сети для последующего анализа и сравнительной оценки эффективности предложенного подхода с использованием RL-агента по сравнению с традиционным статическим методом

управления слотами.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТОВ

Для оценки эффективности интегрированного RL-алгоритма были проведены два комплекса экспериментов: первый – с традиционным статическим подходом к распределению слотов RACH, второй – с применением адаптивного алгоритма обучения с подкреплением (RL). Время проведения каждого эксперимента составляло 3600 секунд, с идентичными исходными условиями и параметрами сети.

Анализ числа и доли коллизий показал существенное преимущество применения RL-алгоритма. При статическом подходе количество коллизий достигло 50 за весь период симуляции, что соответствует доле 2.84% от общего числа попыток доступа. В случае применения RL-алгоритма число коллизий значительно сократилось – до 13, снизив долю коллизий до 0.65%. Таким образом, адаптивный подход позволил уменьшить количество коллизий на 74%, а долю коллизий – на 77%.

Средняя задержка при обоих подходах удерживалась на низком уровне и составила примерно 0.01 секунды. Несмотря на отсутствие заметной разницы в средней задержке, снижение числа коллизий при использовании RL-алгоритма способствует более стабильной работе сети, уменьшая необходимость повторных попыток доступа, что в долгосрочной перспективе приводит к общему повышению эффективности передачи данных.

Энергопотребление устройств является важным показателем для сетей NB-IoT. В ходе эксперимента с традиционным статическим распределением слотов средний остаточный заряд устройств составил 3.7 Дж, в то время как при использовании адаптивного RL-подхода – 3.12 Дж. Это свидетельствует об улучшении энергоэффективности на 15%, вызванном сокращением числа повторных передач данных вследствие снижения коллизий.

Количество успешно переданных пакетов данных за время проведения экспериментов также улучшилось. При использовании статического подхода было зафиксировано 1710 успешных подключений. В случае использования RL-алгоритма это значение выросло до 1990 успешных подключений, что соответствует приросту успешных подключений на 16%. Данный результат подтверждает высокую эффективность адаптивного подхода с точки зрения повышения успешности передачи данных.

Сводная таблица сравнительного анализа результатов экспериментов представлена ниже:

Таблица 1. Результаты моделирования без RL-агента и с его использованием

Метрика	Статическое распределение	RL-алгоритм	Изменение
Число коллизий	50	13	↓74%
Доля коллизий (%)	2.84	0.65	↓77%
Средняя задержка (сек.)	0.01	0.01	стабильно
Среднее энергопотребление (Дж)	3.7	3.12	↓15%
Число успешных	1710	1990	↑16%

подключений	17.10	1000	1100
-------------	-------	------	------

Полученные результаты демонстрируют преимущества предложенного адаптивного подхода на основе алгоритмов обучения с подкреплением перед традиционными методами распределения слотов RACH в сетях NB-IoT. Адаптивность RL-алгоритма к текущей сетевой нагрузке позволяет значительно улучшить основные показатели производительности сети, подтвердив возможность практического применения данного подхода в реальных условиях.

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ И ИХ ОБСУЖДЕНИЕ

Результаты проведенного исследования подтверждают высокую эффективность использования алгоритмов обучения с подкреплением для динамического управления процедурой доступа к временным слотам RACH в сетях NB-IoT. По сравнению с традиционным статическим подходом, применение RL-агента позволило существенно улучшить ключевые эксплуатационные показатели. Количество коллизий было сокращено на 74%, при этом доля коллизий в общем количестве попыток подключения снизилась на 77%. Также удалось увеличить число успешных подключений на 16% и улучшить энергоэффективность устройств на 15%.

Полученные данные подтверждают целесообразность внедрения адаптивных методов управления сетевыми ресурсами на основе RL-алгоритмов в реальные NB-IoT-сети, так как это позволяет существенно повысить общую производительность и надежность функционирования сети.

В рамках дальнейших исследований представляется целесообразным масштабировать предложенный подход с целью изучения поведения алгоритмов обучения с подкреплением при увеличении количества устройств и усложнении сценариев генерации трафика. Это позволит более точно оценить производительность разработанных решений в условиях, максимально приближенных к реальным эксплуатационным условиям.

Дополнительным перспективным направлением является исследование мультиагентных подходов, при которых одновременно несколько RL-агентов управляют различными аспектами сетевого взаимодействия. Предполагается, что это позволит обеспечить дополнительный прирост эффективности в управлении ресурсами сети и снизить вероятность коллизий за счет координации агентов.

Важной задачей будущих исследований является экспериментальная проверка возможности интеграции разработанных RL-алгоритмов с реальным сетевым оборудованием и тестирование в действующих сетях. Это позволит выявить потенциальные ограничения и сложности, возникающие при переходе от симуляции к практическому применению.

Авторы убеждены, что результаты настоящей работы могут служить основой для дальнейших исследований и способствовать совершенствованию технологий интеллектуального управления сетями Интернета вещей.

ВКЛАД АВТОРОВ / AUTHORS' CONTRIBUTION

Авторы внесли равный вклад в написание статьи / The authors contributed equally to the writing of the article.

БИБЛИОГРАФИЯ

1. Liu Y., Deng Y., Jiang N. [и др.]. Analysis of Random Access in NB-IoT Networks With Three Coverage Enhancement Groups: A Stochastic Geometry Approach // IEEE Transactions on Wireless Communications. – 2021. – Т. 20, № 1. – С. 549-563. DOI: 10.1109/twc.2020.3026331. EDN: WLVLVU.
2. Jia G., Zhu Y., Li Y., Zhu Z. Analysis of the Effect of the Reliability of the NB-IoT Network on the Intelligent System // Special Section on Innovation and Application of Internet of Things and Emerging Technologies in Smart Sensing. – 2019. – № 7. – С. 112809-112820.
3. Sahithya R., Pouria Z., Mohieddine E. S., Majid N. Evaluation, Modeling and Optimization of Coverage Enhancement Methods of NB-IoT // Electrical Engineering Department. – 2019. – № 1. – С. 1-17.
4. Chougrani H., Kisseleff S., Martins W. A., Chatzinotas S. NB-IoT Random Access for Nonterrestrial Networks: Preamble Detection and Uplink Synchronization // IEEE Internet of Things Journal. – 2022. – Т. 9, № 16. – С. 14913-14927. DOI: 10.1109/jiot.2021.3123376. EDN: RVWLFC.
5. Agiwal M., Kumar M. M., Jin H. Power Efficient Random Access for Massive NB-IoT Connectivity // Sensors. – 2019. – № 19. – С. 1-24.
6. Jiang N., Deng Y., Nallanathan A. Deep Reinforcement Learning for Real-Time Optimization in NB-IoT Networks // School of Electronic Engineering and Computer Science. – 2018. – № 1. – С. 1-31.
7. Alcaraz J., Losilla F., Gonzalez-Castaño F.-J. Transmission Control in NB-IoT With Model-Based Reinforcement Learning // IEEE Access. – 2023. – № 11. – С. 57991-58005. DOI: 10.1109/access.2023.3284990. EDN: KTFAKS.
8. Anbazhagan S., Mugelan R.K. Next-Gen Resource Optimization in NB-IoT Networks: Harnessing Soft Actor-Critic Reinforcement Learning // Computer Networks. – 2024. – № 252. – С. 110670-110684. DOI: 10.1016/j.comnet.2024.110670. EDN: XHDOCK.
9. Шорин О.А., Асланян В.А. Подходы к интеграции технологии NB-IoT с сетью 5G // Экономика и качество систем связи. – 2024. – № 3. – С. 56-62. EDN: CXRPPW.
10. Намиот Д.Е., Ильюшин Е.А. Архитектура LLM агентов // International Journal of Open Information Technologies. – 2025. – Т. 13, № 1. – С. 2307-8162.
11. Исаева О.С. Построение цифрового профиля устройств Интернета вещей // Информационные и математические технологии в науке и управлении. – 2023. – Т. 30, № 2. – С. 36-44. DOI: 10.25729/ESI.2023.30.2.004. EDN: EFPGJP.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Динамическое распределение слотов RACH для минимизации коллизий в сетях NB-IoT на основе алгоритмов обучения с подкреплением» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена вопросу создания универсальных, масштабируемых и высокоэффективных решений, пригодных для практического применения в реальных условиях эксплуатации сетей NB-IoT.

В качестве цели авторы указывают разработку и экспериментальное подтверждение эффективности применения RL-алгоритмов (Q-learning и DQN) для динамического управления процедурой распределения временных слотов RACH в сетях NB-IoT.

Для достижения поставленной цели авторы сформулировали следующие задачи:

- провести анализ текущего состояния технологии NB-IoT и выявлены основные проблемы, связанные с процедурами случайного доступа;

- спроектировать модель сети NB-IoT, состоящая из трёх различных типов устройств, отражающих специфику их применения;
- выполнить моделирование работы сети и поведения устройств с применением дискретно-событийного симулятора NS-3;
- разработать RL-агент для динамического управления распределением временных слотов доступа;
- провести серия экспериментов с использованием RL-агента и без его применения (с традиционным статическим подходом);
- зафиксировать и обработаны результаты экспериментальных исследований;
- выполнить сравнительный анализ результатов по числу коллизий, статусу подключений, задержке передачи данных и энергопотреблению устройств;
- сформулировать выводы о преимуществах и эффективности предложенного подхода на основе проведённого анализа.

В списке литературы представлены зарубежные источники по теме исследования. Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Практическая значимость статьи четко обоснована, авторы указывают, что результаты исследования могут быть использованы для оптимизации функционирования реальных NB-IoT сетей, повышения их производительности и энергоэффективности, а также служить основой для дальнейших исследований в области адаптивного и интеллектуального управления сетями IoT.

Статья достаточно структурирована - в наличии введение, результаты исследований и их обсуждение, внутреннее членение основной части (текущее состояние технологии, разработка модели сети Nb-Iot и интеграция RI-агента, сравнительный анализ результатов экспериментов).

Авторами при проведении исследования использовался дискретно-событийный сетевой симулятор NS-3 – широко используемая платформа с открытым исходным кодом, написанная на языке C++ с поддержкой интеграции модулей на Python.

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна. Отсутствует четкое выделение предмета, объекта исследования.

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет, объект. Также будет целесообразным добавить о перспективах дальнейшего исследования, расширить список литературы, указав работы отечественных авторов.

Статья «Динамическое распределение слотов RACH для минимизации коллизий в сетях NB-IoT на основе алгоритмов обучения с подкреплением» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией рецензируемого научного журнала.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена актуальной проблеме минимизации коллизий в сетях Narrowband Internet of Things (NB-IoT) через динамическое распределение слотов Random Access Channel (RACH) с использованием алгоритмов обучения с подкреплением (Reinforcement Learning, RL). Авторы предлагают инновационный подход на основе Q-learning и Deep Q-Network (DQN), направленный на повышение эффективности работы сети.

Исследование выполнено на высоком научном уровне с применением современных

методов моделирования. Авторы использовали дискретно-событийный симулятор NS-3 для создания детализированной модели сети NB-IoT, включающей три типа устройств с различными паттернами генерации трафика. Интеграция RL-агента, реализованного на Python, позволила динамически управлять распределением слотов RACH. Эксперименты проводились в контролируемых условиях с фиксацией ключевых метрик: числа коллизий, задержки передачи данных, энергопотребления и успешности подключений. Работа отвечает на вызовы, связанные с ростом числа устройств IoT и перегрузкой сетей. Проблема коллизий в NB-IoT, усугубляемая статическими методами распределения ресурсов, требует адаптивных решений. Применение RL-алгоритмов, как демонстрируют авторы, является перспективным направлением для повышения производительности и энергоэффективности сетей.

Основной вклад статьи заключается в:

1. Разработке RL-агента для динамического управления слотами RACH, что позволило сократить число коллизий на 74%.
2. Экспериментальном подтверждении преимуществ RL-подхода перед статическими методами, включая улучшение энергоэффективности на 15% и увеличение успешных подключений на 16%.
3. Предложении масштабируемой модели, пригодной для дальнейших исследований, включая мультиагентные системы и интеграцию с реальным оборудованием.

Статья отличается четкой логической структурой, последовательным изложением и глубиной проработки темы. Введение обосновывает актуальность исследования, раздел «Текущее состояние технологии» содержит обзор литературы, а методическая часть детализирована и воспроизводима. Результаты представлены наглядно, включая сравнительную таблицу, что облегчает восприятие данных. Стиль изложения соответствует академическим стандартам, терминология использована корректно.

Авторы убедительно демонстрируют эффективность RL-алгоритмов для управления ресурсами NB-IoT. Полученные результаты имеют значительную практическую ценность и могут быть применены для оптимизации реальных сетей. Статья вносит весомый вклад в область интеллектуального управления IoT и открывает направления для дальнейших исследований, таких как мультиагентные системы и тестирование в промышленных условиях.

Материал будет востребован исследователями в области IoT, телекоммуникаций и машинного обучения, а также инженерами, занимающимися развертыванием и оптимизацией NB-IoT сетей. Четкие выводы и практические рекомендации делают статью полезной для широкого круга специалистов.

Статья соответствует высоким академическим стандартам, обладает значительной научной новизной и практической ценностью. Уверен, что работа вызовет интерес у научного сообщества и послужит основой для дальнейших исследований в данной области.

Рекомендация: принять к публикации без доработок.

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Карпович В.Д., Государев И.Б. Исследование производительности WebAssembly в среде исполнения Node.js // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.74049 EDN: QOVQLO URL: https://nbpublish.com/library_read_article.php?id=74049

Исследование производительности WebAssembly в среде исполнения Node.js

**Карпович Владимир Дмитриевич**

ORCID: 0009-0009-5397-3731

студент; факультет Программной Инженерии и Компьютерной Техники; Национальный исследовательский университет ИТМО

109369, Россия, г. Москва, р-н Марьино, ул. Перерва, д. 26 к. 1, кв. 94

 mywinter4@yandex.ru**Государев Илья Борисович**

ORCID: 0000-0003-4236-5991

кандидат педагогических наук

доцент; Мегафакультет компьютерных технологий и управления; Национальный исследовательский университет ИТМО

190000, Россия, г. Санкт-Петербург, Кронверкский проспект, 49

 goss@itmo.ru

[Статья из рубрики "Математическое и программное обеспечение новых информационных технологий"](#)

DOI:

10.7256/2454-0714.2025.2.74049

EDN:

QOVQLO

Дата направления статьи в редакцию:

10-04-2025

Дата публикации:

18-04-2025

Аннотация: Современные среды исполнения, такие как браузеры, Node.js и пр. предоставляют разработчикам инструменты, позволяющие выходить за рамки

традиционного JavaScript. Объектом данного исследования выступает современный подход к созданию веб-приложений, в которых возможно выполнение и совместное использование компонентов, написанных на разных языках программирования, в результате применения WebAssembly. Предметом исследования является тестирование и анализ результатов тестов, направленных на измерение производительности JavaScript и WebAssembly-модулей в среде выполнения Node.js, с акцентом на сравнение эффективности выполнения вычислительных задач, взаимодействия с памятью, обработки данных и межъязыкового взаимодействия. Автор подробно рассматривает такие аспекты темы, как интеграция WebAssembly в приложения, оценка его преимуществ при решении ресурсоёмких задач, таких как обработка изображений, объективность, репрезентативность и воспроизводимость тестирования. В работе используется прикладной, экспериментальный подход. Было произведено сравнение скорости выполнения операций при использовании чистого JavaScript и WebAssembly-модулей. Для оценки эффективности были использованы данные о времени ответа на запрос, о потреблении ресурсов системы приложением. Научная новизна данной работы заключается в разработке и теоретическом обосновании подходов к тестированию веб-приложений с использованием технологии WebAssembly. В отличие от большинства существующих исследований, сосредоточенных на производительности и безопасности WebAssembly в браузерной среде, в данной работе внимание акцентировано на автоматизированном тестировании WebAssembly-модулей вне браузера, что до настоящего времени оставалось слабо проработанным направлением. Предложен методологический подход к организации тестирования WebAssembly-модулей в среде выполнения Node.js, включая принципы структурирования тестов, интеграции с JavaScript-компонентами и анализ результатов выполнения. Такой подход позволяет учитывать специфику серверного окружения, где WebAssembly всё активнее применяется — в частности, при разработке высоконагруженных вычислительных модулей, кросс-языковой логики и безопасного изолированного исполнения. Научная новизна также заключается в выведении критериев, позволяющих оценивать пригодность тех или иных компонентов приложения для переноса в WebAssembly с точки зрения тестируемости, что даёт разработчикам дополнительный инструмент принятия архитектурных решений. Предложенные идеи подтверждены экспериментальной частью, включающей примеры реализации тестирования сценариев взаимодействия между WebAssembly и JavaScript.

Ключевые слова:

WebAssembly, JavaScript, оптимизация производительности, вычислительно интенсивные задачи, обработка изображений, интеграция технологий, серверные вычисления, экспериментальный анализ, алгоритмическая оптимизация, компиляция и интерпретация

ВВЕДЕНИЕ

В современных условиях развития веб-технологий требования к производительности веб-приложений постоянно растут. Увеличение объёма обрабатываемых данных и сложности выполняемых алгоритмов делает актуальным вопрос поиска эффективных решений в области оптимизации кода. JavaScript, как один из основных языков программирования для разработки веб-приложений, предоставляет разработчикам мощные инструменты и возможности; однако, своей интерпретируемой природой он ограничен в производительности по сравнению с компилируемыми языками.

С появлением WebAssembly возникли новые перспективы для повышения производительности веб-приложений. WebAssembly представляет собой низкоуровневый бинарный формат, который позволяет запускать код, написанный на языках программирования, таких как C, C++, Rust и других, непосредственно в среде выполнения JavaScript. Эта технология создает возможность для разработчиков писать производительный код, использующий преимущества компиляции, что позволяет избежать недостатков JavaScript в задачах, требующих высокой вычислительной мощности.

Актуальность изучения WebAssembly обусловлена его растущей популярностью и внедрением в широкий спектр приложений, от игр до обработки видео и графики. Тем не менее, эта технология еще недостаточно исследована, и образуется пробел в литературе, касающейся тестирования и производительности WebAssembly по сравнению с традиционными подходами. Цель данной работы заключается в анализе возможности интеграции WebAssembly в существующие приложения на JavaScript, а также в оценке его влияния на производительность.

ЭТАПЫ

В ходе проведения тестирования производительности JavaScript и WebAssembly в среде Node.js были выделены следующие этапы:

- выбор тестового приложения: было определено, что для оценки производительности подходящей задачей будет обработка изображений, так как это вычислительно интенсивная операция, требующая относительно больших ресурсов. В частности, была выбрана задача наложения фильтра путём применения алгоритма свёртки;
- разработка тестового приложения: был реализован простой веб-сервер, включающий функционал обработчика изображений как на JavaScript, так и на WebAssembly. Для реализации функции обработки изображений на WebAssembly использован язык C, который был скомпилирован в Wasm-файл с помощью инструмента Emscripten;
- настройка тестового окружения: для создания стабильной тестовой среды и минимизации влияния внешних факторов был использован инструмент контейнеризации Docker; в контейнерах были подняты инструменты для снятия, хранения и отображения метрик, веб-сервер с тестируемым функционалом, ПО для нагружочного тестирования;
- проведение тестов: в вышеупомянутом тестовом окружении были запущены тесты, использующие различные изображения, и последовательно отправляющие запросы к серверу; во время тестирования собирались метрики о работе приложения и состоянии системы;
- анализ результатов: после завершения тестирования собранные данные были проанализированы, на основании результатов анализа были сделаны выводы о том, в каких случаях использование WebAssembly может быть уместным.

ТЕСТОВОЕ ПРИЛОЖЕНИЕ

Результаты исследования не смогут быть объективными и отражающими действительность, если эксперименты будут проводиться над приложением, не подходящим для моделируемой ситуации. В рамках данного исследования предполагается, что WebAssembly будет использован в качестве дополнения к JavaScript как к основному языку программирования; WebAssembly будет использован точечно там, где использование JavaScript нецелесообразно, нерационально или неэффективно.

Таким образом, предполагается, что данный инструмент сможет расширить спектр ситуаций, в которых целесообразно использовать Node.js в качестве основной платформы для разработки [1].

Из этого следует, что приложение, над которым проводятся эксперименты, должно содержать в себе подлежащий оборачиванию в Wasm-формат участок. В случае, когда функционал может быть эффективно реализован нативными средствами Node.js, будем считать, что использование сторонних инструментов не оправдано.

В качестве основного функционала, который можно подвергнуть оптимизации с помощью WebAssembly, была выбрана обработка изображений. Наложение сложного фильтра – высоконагруженная операция, связанная с большим объёмом данных, регулярно применяемая в реальных приложениях; такая предметная область позволит достаточно объективно изучить влияние WebAssembly на производительность и оптимизацию приложения [2].

Архитектурно, приложение представляет собой HTTP-сервер с публичным API, позволяющим загружать, скачивать и обрабатывать изображения. Технологический стек включает в себя платформу Node.js, библиотеку для роутинга express и TypeScript.

Маршруты для обработки изображения представлены в трёх версиях: с алгоритмом, реализованным на JS и реализованным на C и скомпилированным в wasm-код в однопоточном и многопоточном режимах. Модуль WebAssembly инициируется единожды и затем вызывает экспортированную функцию обработки.

Исходный код приложенияложен в репозиторий на GitHub и доступен по ссылке <https://github.com/Winter4/image-processor-backend> [3].

АЛГОРИТМ ОБРАБОТКИ ИЗОБРАЖЕНИЯ

Обработка изображения производится с помощью алгоритма на основании свёртки. Свёртка – это математическая операция, основанная на концепции ядра (также называют фильтром или маской). Ядро – это квадратная матрица с обязательно нечётным размером, содержащая числа, которые определяют веса пикселей [4].

Свёртка выполняется следующим образом: центр ядра накладывается на текущий пиксель. Каждый элемент ядра умножается на свой пиксель, и результаты суммируются. Полученная сумма заменяет текущий (центральный) пиксель. После этого ядро сдвигается на следующий пиксель, и процесс повторяется. В итоге формируется новое изображение, где каждый пиксель является результатом свёртки. Все вычисления должны производиться с исходным изображением, не включающим уже «свёрнутые» пиксели.

Формула свёртки для изображения с одним каналом на пиксель выглядит следующим образом:

$$I'(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k I(x + i, y + j) * M(i, j),$$

где x, y – координаты текущего пикселя; I'(x, y) – значение пикселя результирующего изображения в координатах (x, y); i, j – величина смещения по осям x и y соответственно, относительно координат текущего пикселя; k - половина стороны матрицы, округлённая вниз до целого; I(x + i, y + j) – значение пикселя исходного изображения в координатах (x + i, y + j); M(i, j) – значение элемента ядра в

координатах ($k + i, k + j$).

При обработке краёв изображения, где ядро выходит за границы, используются различные методы: заполнение нулями, отражение соответствующих пикселей на другой стороне ядра, повторение ближайшего пикселя или циклическое повторение, при котором края «замыкаются» друг на друге так, будто изображение циклически повторяется.

Если пиксель представляет собой совокупность чисел с одинаковым диапазоном значений, например, 3 цветовых канала в системе RGB, свёртка будет применяться для каждого отдельного канала. То есть, для вычисления нового значения R-канала текущего пикселя необходимо произвести вычисления с R-каналами окрестных пикселей, далее – с G- и B-каналами соответственно. Если пиксель представляет собой совокупность чисел с разными диапазонами значений, например, RGBL, где L – яркость от 0 до 9, то есть два варианта: нормализация значений, произведение вычислений и последующая денормализация, или применение отдельного ядра для вычисления нового значения яркости. Применение единого ядра к ненормализованным значениям с большой вероятностью приведёт к отклонениям от ожидаемого результата, однако теоретически допустимо.

Свёртка является универсальным инструментом, лежащим в основе обработки изображений и даже более сложных систем, таких как свёрточные нейронные сети, используемые в задачах распознавания объектов и других областях анализа данных.

ТЕСТОВОЕ ОКРУЖЕНИЕ

Тестовое окружение состоит из сервера приложения, ПО для снятия метрик, ПО для хранения и анализа метрик, ПО для визуализации метрик и ПО для нагружочного тестирования.

В качестве ПО для снятия метрик был использован Cadvisor – инструмент с открытым исходным кодом, разработанный Google, для мониторинга и анализа производительности контейнеров. Cadvisor автоматически обнаруживает контейнеры, работающие на хосте, собирает статистику о их производительности и предоставляет данные в реальном времени. Хоть в основном инструмент рассчитан на Docker, его можно использовать со всеми платформами контейнеризации, поддерживающими cgroups [\[5\]](#).

Для хранилища метрик был выбран Prometheus. Это популярная система мониторинга и оповещения с открытым исходным кодом, разработанная для сбора метрик из различных источников, их хранения и анализа. Изначально был создан в SoundCloud, и в процессе стал фактическим стандартом в индустрии. Prometheus работает по принципу pull-модели, то есть сам опрашивает целевые источники данных через HTTP; формат метрик также стал стандартом. В качестве языка запросов Prometheus использует PromQL. Метрики хранятся в собственной временной базе данных, где данные организованы по временным рядам с метками в формате "ключ-значение". Для оповещений Prometheus интегрируется с другим ПО, к примеру Alertmanager, что позволяет настраивать гибкие правила уведомлений и отправлять их через Slack, email и пр [\[6\]](#).

За визуализацию метрик отвечает Grafana - платформа с открытым исходным кодом для визуализации данных и работы с ними. Она позволяет подключаться к различным источникам данных, таким как Prometheus, InfluxDB, Elasticsearch, MySQL, PostgreSQL и многим другим, чтобы создавать настраиваемые панели (дашборды) с визуализацией. Grafana поддерживает широкий набор графических компонентов, включая графики,

гистограммы, тепловые карты, таблицы, что делает её универсальным инструментом для анализа данных. Есть возможность задавать гибкие фильтры, применять шаблоны и использовать переменные для упрощения работы с дашбордами в сложных системах. Поддерживаются настройки оповещений: можно задать пороговые значения для метрик и получать. Grafana широко применяется в DevOps для мониторинга инфраструктуры, серверов, приложений и баз данных, а также в бизнесе для построения аналитических отчетов. Благодаря своей модульности, масштабируемости и интеграции с различными системами Grafana является ключевым инструментом для визуализации данных в условиях современной разработки [\[7\]](#).

Нагрузочное тестирование производится с помощью k6 — это инструмент с открытым исходным кодом для нагрузочного и производительного тестирования, разработанный создателями Grafana. k6 написан на Go и предоставляет удобный синтаксис для настройки на JavaScript. Благодаря интеграции с CI/CD системами, такими как Jenkins, GitLab CI и GitHub Actions, k6 легко можно включить в процессы автоматизации. Он также поддерживает remote write - экспорт данных в реальном времени. k6 отличается гибкостью, простотой использования, высокой производительностью и возможностью работы с большими объемами запросов [\[8\]](#).

Для минимального влияния сторонних факторов на результаты тестирования, необходимо изолировать все элементы. Для этого отлично подходит технология контейнеризации. Для реализации был выбран контейнерный движок Docker – из-за своей популярности и распространённости (фактически является стандартом индустрии).

Для контейнеризации приложения-обработчика необходимо написать скрипт сборки образа – Dockerfile:

```

Dockerfile

FROM node:20-alpine

WORKDIR /app

COPY . .

RUN npm ci
RUN npm run tsc

ENV NODE_ENV=production
CMD [ "node", "./compiled/source/api.js" ]

```

Рисунок 1 – Dockerfile для сборки образа приложения

Код Dockerfile в текстовом виде представлен в приложении А. Для консистентной развёртки инфраструктуры и тестового окружения был написан docker-compose файл [\[9\]](#). Полный код файла представлен в приложении Б. Для повышения объективности исследования ограничим контейнер с приложением 1 ядром центрального процессора и

2 ГБ оперативной памяти.

ТЕСТИРОВАНИЕ

Проведём тестирование приложения в условиях вышеописанного тестового окружения. Так как узким местом приложения является функционал обработки изображений, и целью исследования является оценка производительности именно этого функционала, тестирование будет проводиться путём последовательной отправки запросов в одном потоке, один за одним. Это позволит избежать накладных расходов, связанных с обработкой параллельно-идущих запросов, произвести максимально точную оценку производительности, а также отследить сбои в стабильности работы системы, такие как утечка памяти [\[10\]](#).

Тестирование будет проводиться в двух режимах: с использованием маленького и большого изображения. Маленькое изображение – файл формата .jpg, объёмом 314 КБ, разрешением 1680x1050 пикселей; суммарное количество пикселей 1 764 000. Большое изображение – файл формата .jpg, объёмом 10.3 МБ, разрешением 11384 x 4221 пикселей; суммарное количество пикселей – 48 051 864. Тест с использованием маленького изображения будет представлять собой последовательную отправку 5000 запросов; с использованием большого – последовательную отправку 150 запросов. Часть файлов для тестирования была взята с ресурса examplefile.com [\[11\]](#).

Стандартное отклонение фильтра Гаусса (сигма) возьмём равное 7, размер ядра 7x7 [\[12\]](#).

Однопоточный режим

Результаты теста с маленьким изображением и JS-реализацией представлены на рисунке 2:

```

✓ status is 200

checks.....: 100.00% 5000 out of 5000
data_received.....: 646 MB 300 kB/s
data_sent.....: 1.6 GB 748 kB/s
http_req_blocked.....: avg=4.27µs min=1.84µs med=3.47µs max=366.66µs p(90)=5.19µs p(95)=6.42µs
http_req_connecting.....: avg=47ns min=0s med=0s max=123.65µs p(90)=0s p(95)=0s
http_req_duration.....: avg=430.62ms min=47.61ms med=430.29ms max=499.94ms p(90)=444.26ms p(95)=454.16ms
{ expected_response=true }.....: avg=430.62ms min=47.61ms med=430.29ms max=499.94ms p(90)=444.26ms p(95)=454.16ms
http_req_failed.....: 0.00% 0 out of 5000
http_req_receiving.....: avg=163.8µs min=72.5µs med=151.76µs max=717.29µs p(90)=210.84µs p(95)=242.23µs
http_req_sending.....: avg=345.42µs min=222.18µs med=329.81µs max=1.63ms p(90)=421.59µs p(95)=468.53µs
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=430.11ms min=47.12ms med=429.77ms max=499.46ms p(90)=443.78ms p(95)=453.68ms
http_reqs.....: 5000 2.328642/s
iteration_duration.....: avg=430.97ms min=397.67ms med=430.54ms max=500.19ms p(90)=444.59ms p(95)=454.42ms
iterations.....: 5000 2.320178/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h35m55.0s), 0/1 VUs, 5000 complete and 0 interrupted iterations
sequential_requests / [ 100% ] 1 VUs 0h35m55.0s/1h0m0s 5000/5000 shared iters

```

Рисунок 2 - результаты теста с маленьким изображением на JS-реализации

Показатели системы в процессе теста с маленьким изображением и JS-реализацией представлены на рисунке 3:



Рисунок 3 – показатели системы во время теста с маленьким изображением на JS-реализации

Как можно видеть, 5000 запросов были обработаны за 35 минут 55 секунд; среднее время ответа – 431 мс, это соответствует среднему показателю RPS 2.32. Средняя загрузка оперативной памяти при этом составила 40 МБ. Использование ЦПУ упиралось в лимит всё время теста.

Результаты теста с маленьким изображением и Wasm-реализацией представлены на рисунке 4:

```

k6
k6 ✓ status is 200
k6
k6 checks.....: 100.00% 5000 out of 5000
k6 data_received.....: 646 MB 223 kB/s
k6 data_sent.....: 1.6 GB 555 kB/s
k6 http_req_blocked.....: avg=4.18µs min=2.02µs med=3.4µs max=406.33µs p(90)=4.93µs p(95)=5.94µs
k6 http_req_connecting.....: avg=45ns min=0s med=0s max=126.11µs p(90)=0s p(95)=0s
k6 http_req_duration.....: avg=580.6ms min=35.6ms med=581.35ms max=637.37ms p(90)=594.79ms p(95)=600.5ms
k6 http_req_failed.....: 0.00% 0 out of 5001
k6 http_req_receiving.....: avg=173.97µs min=67.87µs med=155.42µs max=26.08ms p(90)=212.91µs p(95)=244.1µs
k6 http_req_sending.....: avg=344.82µs min=223.7µs med=333.67µs max=1.16ms p(90)=422.26µs p(95)=458.88µs
k6 http_req_tls_handshaking.....: avg=8s min=0s med=0s max=0s p(90)=0s p(95)=0s
k6 http_req_waiting.....: avg=580.08ms min=35.12ms med=580.83ms max=636.9ms p(90)=594.27ms p(95)=600.05ms
k6 http_reqs.....: 5001 1.721491/s
k6 iteration_duration.....: avg=580.98ms min=542.34ms med=581.6ms max=637.57ms p(90)=595.11ms p(95)=600.71ms
k6 iterations.....: 5000 1.721147/s
k6 vus.....: 1 min=1 max=1
k6 vus_max.....: 1 min=1 max=1
k6
k6
k6 running (0h48m25.0s), 0/1 VUs, 5000 complete and 0 interrupted iterations
k6 sequential_requests ✓ [ 100% ] 1 VUs 0h48m25.0s/1h0m0s 5000/5000 shared iters

```

Рисунок 4 – результаты теста с маленьким изображением на Wasm-реализации

Показатели системы в процессе теста с маленьким изображением и Wasm-реализацией представлены на рисунке 5:



Рисунок 5 – результаты системы во время теста с маленьким изображением на Wasm-

реализации

5000 запросов были обработаны за 48 минут 25 секунд; среднее время ответа – 581 мс, это соответствует среднему показателю RPS 1.72. Средняя загрузка оперативной памяти при этом составила 43.7 МБ. Использование ЦПУ упиралось в лимит всё время теста.

Результаты теста с большим изображением и JS-реализацией представлены на рисунке 6:

```

✓ status is 200
checks.....: 100.00% 150 out of 150
data_received.....: 370 MB 234 kB/s
data_sent.....: 1.6 GB 1.0 MB/s
http_req_blocked.....: avg=9.97μs min=2.47μs med=3.65μs max=564.41μs p(90)=5.43μs p(95)=10.05μs
http_req_connecting.....: avg=2.17μs min=0s med=0s max=171.73μs p(90)=0s p(95)=0s
http_req_duration.....: avg=10.47s min=366.62ms med=10.54s max=10.71s p(90)=10.63s p(95)=10.65s
{ expected_response:true }.....: avg=10.47s min=366.62ms med=10.54s max=10.71s p(90)=10.63s p(95)=10.65s
http_req_failed.....: 0.00% 0 out of 151
http_req_receiving.....: avg=1.84ms min=67.5μs med=1.41ms max=5.54ms p(90)=3.18ms p(95)=4.85ms
http_req_sending.....: avg=15.41ms min=10.92ms med=14.55ms max=41.77ms p(90)=20.02ms p(95)=22.19ms
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=10.45s min=345.84ms med=10.52s max=10.69s p(90)=10.61s p(95)=10.64s
http_reqs.....: 151 0.095468/s
iteration_duration.....: avg=10.54s min=10.21s med=10.54s max=10.71s p(90)=10.63s p(95)=10.65s
iterations.....: 150 0.094836/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h26m21.7s), 0/1 VUs, 150 complete and 0 interrupted iterations
sequential_requests ✓ [ 100% ] 1 VUs 0h26m21.3s/1h0m0s 150/150 shared iters

```

Рисунок 6 – результаты теста с большим изображением на JS-реализации

Показатели системы в процессе теста с большим изображением и JS-реализацией представлены на рисунке 7:



Рисунок 7 – показатели системы во время теста с большим изображением на JS-реализации

150 запросов были обработаны за 26 минут 21 секунду; среднее время ответа – 10.47 с, это соответствует среднему показателю RPS 0.09. Средняя загрузка оперативной памяти при этом составила 409.4 МБ. Использование ЦПУ упиралось в лимит всё время теста.

Результаты теста с большим изображением и Wasm-реализацией представлены на рисунке 8:

```

✓ status is 200

checks.....: 100.00% 150 out of 150
data_received.....: 372 MB 169 kB/s
data_sent.....: 1.6 GB 742 kB/s
http_req_blocked.....: avg=10.03µs min=2.19µs med=3.61µs max=528.11µs p(90)=6.27µs p(95)=17.08µs
http_req_connecting.....: avg=1.83µs min=0s med=0s max=182.81µs p(90)=0s p(95)=0s
http_req_duration.....: avg=14.56s min=346.89ms med=14.64s max=15.14s p(90)=14.78s p(95)=14.81s
{ expected_response:true }.....: avg=14.56s min=346.89ms med=14.64s max=15.14s p(90)=14.78s p(95)=14.81s
http_req_failed.....: 0.00% 0 out of 151
http_req_receiving.....: avg=1.95ms min=63.96µs med=1.67ms max=12.79ms p(90)=3.18ms p(95)=3.77ms
http_req_sending.....: avg=13.93ms min=10.42ms med=11.99ms max=38.2ms p(90)=19.08ms p(95)=25.46ms
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=14.55s min=328.03ms med=14.62s Max=15.12s p(90)=14.77s p(95)=14.78s
http_reqs.....: 151 0.06863/s
iteration_duration.....: avg=14.66s min=14.49s med=14.64s max=15.15s p(90)=14.78s p(95)=14.81s
iterations.....: 150 0.068179/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h36m40.s), 0/1 VUs, 150 complete and 0 interrupted iterations
sequential_requests ✓ [ 100% ] 1 VUs 0h36m39.7s/1h0m0s 150/150 shared iters

```

Рисунок 8 – результаты теста с большим изображением на Wasm-реализации

Показатели системы в процессе теста с большим изображением и Wasm-реализацией представлены на рисунке 9:



Рисунок 9 – показатели системы во время теста с большим изображением на Wasm-реализации

150 запросов были обработаны за 36 минут 39 секунд; среднее время ответа – 14.56 с, это соответствует среднему показателю RPS 0.06. Средняя загрузка оперативной памяти при этом составила 539.3 МБ. Использование ЦПУ упиралось в лимит всё время теста.

Многопоточный режим

В JavaScript отсутствует нативная многопоточность. Пакет `cluster` из стандартного набора поставки Node.js может реплицировать приложение, однако каждая реплика в отдельности всё ещё будет однопоточной, и таким образом удельная производительность останется неизменной. В Emscripten (компилятор WebAssembly), в свою очередь, нативная поддержка многопоточности появилась в 2019 году в версии 1.38.27. Emscc использует Web Workers API [\[13\]](#) и SharedArrayBuffer [\[14\]](#) для её реализации. При этом, многопоточность инкапсулирована на уровне .wasm-файла и настроек компилятора. То есть, при репликации такого приложения, на каждой из реплик будет использована многопоточность, что потенциально повышает удельную производительность.

Для исследования эффективности применения многопоточности в emcc исходный код на .c был доработан для использования Worker API, а в настройки компилятора были добавлены соответствующие флаги. Модуль был скомпилирован с использованием пула из 4x потоков. При этом, реализация алгоритма свёртки как таковая не изменилась – ни объём вычислений, ни их сложность. Исходный код этой версии приложения доступен в том же репозитории.

Тесты будут проведены с разными ограничениями по ядрам для контейнера с приложением: 1 ядро, 2 ядра, 4 ядра. Ограничения по оперативной памяти остаются неизменными – 2 ГБ.

Результаты теста с маленьким изображением и ограничением в 1 ядро ЦПУ представлены на рисунке 10:

```

✓ status is 200

checks.....: 100.00% 5000 out of 5000
data_received.....: 646 MB 214 kB/s
data_sent.....: 1.6 GB 535 kB/s
http_req_blocked.....: avg=4.73µs min=2.03µs med=3.6µs max=1.34ms p(90)=5.46µs p(95)=6.55µs
http_req_connecting.....: avg=43ns min=0s med=0s max=123.32µs p(90)=0s p(95)=0s
http_req_duration.....: avg=602.42ms min=36.21ms med=599.96ms max=967.49ms p(90)=609.64ms p(95)=661.66ms
(expected_response=true )....: avg=602.42ms min=36.21ms med=599.96ms max=967.49ms p(90)=609.64ms p(95)=661.66ms
http_req_failed.....: 0.00% 0 out of 5000
http_req_receiving.....: avg=190us min=78.12µs med=164.15µs max=55.08ms p(90)=244.04µs p(95)=290.15µs
http_req_sending.....: avg=382.72µs min=213.05µs med=371.12µs max=1.14ms p(90)=523.11µs p(95)=583.34µs
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=601.85ms min=35.74ms med=599.41ms max=966.84ms p(90)=609.02ms p(95)=661.21ms
http_reqs.....: 5001 1.659123/s
iteration_duration.....: avg=602.82ms min=517.64ms med=600.23ms max=967.9ms p(90)=609.93ms p(95)=662.18ms
iterations.....: 5000 1.658791/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h50m14.2s), 0/1 VUs, 5000 complete and 0 interrupted iterations
sequential_requests ✓ [ 100% ] 1 VUs 0h50m14.2s/1h0m0s 5000/5000 shared iters
Tested with CodeBench

```

Рисунок 10 – результаты теста с маленьким изображением и ограничением в 1 ядро процессора

Показатели системы во время теста с маленьким изображением и ограничением в 1 ядро ЦПУ представлены на рисунке 11:



Рисунок 11 – показатели системы во время теста с маленьким изображением и ограничением в 1 ядро процессора

5000 запросов были обработаны за 50 минут 14 секунд; среднее время ответа – 602 мс, это соответствует среднему показателю RPS 1.65. Средняя загрузка оперативной памяти при этом составила 67.2 МБ. Использование ЦПУ упиралось в лимит всё время теста.

Результаты теста с большим изображением и ограничением в 1 ядро ЦПУ представлены на рисунке 12:

```

✓ status is 200

checks.....: 100.00% 150 out of 150
data_received.....: 372 MB 165 kB/s
data_sent.....: 1.6 GB 723 kB/s
http_req_blocked.....: avg=8.51µs min=2.27µs med=3.54µs max=443.02µs p(90)=5.15µs p(95)=7.35µs
http_req_connecting.....: avg=2.26µs min=0s med=0s max=198.59µs p(90)=0s p(95)=0s
http_req_duration.....: avg=14.94s min=340.58ms med=15.02s max=16.73s p(90)=15.07s p(95)=15.11s
{ expected_response:true }.....: avg=14.94s min=340.58ms med=15.02s max=16.73s p(90)=15.07s p(95)=15.11s
http_req_failed.....: 0.00% 0 out of 151
http_req_receiving.....: avg=2.07ms min=64.39µs med=1.59ms max=6.69ms p(90)=4.14ms p(95)=4.82ms
http_req_sending.....: avg=15.23ms min=9.15ms med=14.37ms max=30.93ms p(90)=20.61ms p(95)=23.47ms
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=14.92s min=314.2ms med=15.01s max=16.7s p(90)=15.06s p(95)=15.09s
http_reqs.....: 150 0.066888/s
iteration_duration.....: avg=15.04s min=14.88s med=15.03s max=16.74s p(90)=15.08s p(95)=15.11s
iterations.....: 150 0.066445/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h37m37.5s), 0/1 VUs, 150 complete and 0 interrupted iterations
sequential_requests ✓ [ 100% ] 1 VUs 0h37m37.5s/1h0m0s 150/150 shared iters
exited with code 0

```

Рисунок 12 – результаты теста с большим изображением и ограничением в 1 ядро процессора

Показатели системы во время теста с большим изображением и ограничением в 1 ядро ЦПУ представлены на рисунке 13:



Рисунок 13 – показатели системы во время теста с большим изображением и ограничением в 1 ядро процессора

150 запросов были обработаны за 37 минут 37 секунд; среднее время ответа – 14.94 с, это соответствует среднему показателю RPS 0.06. Средняя загрузка оперативной памяти при этом составила 547.2 МБ. Использование ЦПУ упиралось в лимит всё время теста.

Результаты теста с маленьким изображением и ограничением в 2 ядра ЦПУ представлены на рисунке 14:

```

✓ status is 200

checks.....: 100.00% 5000 out of 5000
data_received.....: 646 MB 432 kB/s
data_sent.....: 1.6 GB 1.1 MB/s
http_req_blocked.....: avg=4.52µs min=2.07µs med=3.51µs max=438.83µs p(90)=5.34µs p(95)=6.72µs
http_req_connecting.....: avg=41ns min=0s med=0s max=118.06µs p(90)=0s p(95)=0s
http_req_duration.....: avg=298.8ms min=44.91ms med=299.02ms max=395.81ms p(90)=385.55ms p(95)=387.99ms
{ expected_response:true }.....: avg=298.8ms min=44.91ms med=299.02ms max=395.81ms p(90)=385.55ms p(95)=387.99ms
http_req_failed.....: 0.00% 0 out of 5001
http_req_receiving.....: avg=133.68µs min=66.34µs med=118.69µs max=923.44µs p(90)=181.29µs p(95)=222.15µs
http_req_sending.....: avg=343.87µs min=195.11µs med=337.1µs max=1.5ms p(90)=463.89µs p(95)=518.23µs
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=298.32ms min=44.41ms med=298.56ms max=395.39ms p(90)=305ms p(95)=307.51ms
http_reqs.....: 5000 3.343267/s
iteration_duration.....: avg=299.14ms min=238.49ms med=299.31ms max=396.01ms p(90)=305.85ms p(95)=308.27ms
iterations.....: 5000 3.342599/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h24m55.8s), 0/1 VUs, 5000 complete and 0 interrupted iterations
sequential_requests ✓ [ 100% ] 1 VUs 0h24m55.8s/1h0m0s 5000/5000 shared iters
exited with code 0

```

Рисунок 14 – результаты теста с маленьким изображением и ограничением в 2 ядра процессора

Показатели системы во время теста с маленьким изображением и ограничением в 2 ядра ЦПУ представлены на рисунке 15:



Рисунок 15 – показатели системы во время теста с маленьким изображением и ограничением в 2 ядра процессора

5000 запросов были обработаны за 24 минуты 55 секунд; среднее время ответа – 299 мс, это соответствует среднему показателю RPS 3.34. Средняя загрузка оперативной памяти при этом составила 67.1 МБ. Использование ЦПУ упиралось в лимит всё время теста.

Результаты теста с большим изображением и ограничением в 2 ядра ЦПУ представлены на рисунке 16:

```
✓ status is 200
checks.....: 100.00% 150 out of 150
data_received.....: 372 MB 321 kB/s
data_sent.....: 1.6 GB 1.4 MB/s.
http_req_blocked.....: avg=10.83µs min=2.14µs med=3.6µs max=614.6µs p(90)=7.43µs p(95)=17.29µs
http_req_connecting.....: avg=2.01µs min=0s med=0s max=194.14µs p(90)=0s p(95)=0s
http_req_duration.....: avg=7.65s min=341.41ms med=7.69s max=8.88s p(90)=7.77s p(95)=7.78s
{ expected_response:true }.....: avg=7.65s min=341.41ms med=7.69s max=8.88s p(90)=7.77s p(95)=7.78s
http_req_failed.....: 0.00% 0 out of 151
http_req_receiving.....: avg=1.94ms min=70.79µs med=1.48ms max=7.18ms p(90)=3.62ms p(95)=4.92ms
http_req_sendng.....: avg=12.98ms min=10.34ms med=12.02ms max=24.75ms p(90)=16.75ms p(95)=17.57ms
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=7.64s min=321.79ms med=7.67s max=8.86s p(90)=7.75s p(95)=7.76s
http_reqs.....: 151 0.130489/s
iteration_duration.....: avg=7.71s min=7.59s med=7.69s max=8.88s p(90)=7.77s p(95)=7.78s
Iterations.....: 150 0.129625/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h19m17.2s), 8/1 VUs, 150 complete and 0 interrupted iterations
sequential_requests ✓ [ 100% ] 1 VUs 0h19m16.8s/1h0m0s 150/150 shared iters
ended with code 0
```

Рисунок 16 – результаты теста с большим изображением и ограничением в 2 ядра процессора

Показатели системы во время теста с большим изображением и ограничением в 2 ядра ЦПУ представлены на рисунке 17:



Рисунок 17 – показатели системы во время теста с большим изображением и

ограничением в 2 ядра процессора

150 запросов были обработаны за 19 минут 17 секунд; среднее время ответа – 7.65 с, это соответствует среднему показателю RPS 0.13. Средняя загрузка оперативной памяти при этом составила 541.6 МБ. Использование ЦПУ упиралось в лимит всё время теста.

Результаты теста с маленьким изображением и ограничением в 4 ядра ЦПУ представлены на рисунке 18:

```

✓ status ls 200
checks.....: 100.00% 5000 out of 5000
data_received.....: 646 MB 724 kB/s
data_sent.....: 1.6 GB 1.8 MB/s
http_req_blocked.....: avg=4.2μs min=1.7μs med=3.25μs max=481.21μs p(90)=5.01μs p(95)=6.51μs
http_req_connecting.....: avg=52ns min=0s med=0s max=137.72μs p(90)=0s p(95)=0s
http_req_duration.....: avg=178.32ms min=45.6ms med=177.25ms max=460.53ms p(90)=185.13ms p(95)=187.47ms
[ expected_response:true ].....: avg=178.32ms min=45.6ms med=177.25ms max=460.53ms p(90)=185.13ms p(95)=187.47ms
http_req_failed.....: 0.00% 0 out of 5001
http_req_receiving.....: avg=153.82μs min=68.57μs med=121.61μs max=1.59ms p(90)=245.23μs p(95)=335.32μs
http_req_sending.....: avg=222.42μs min=142.13μs med=197.01μs max=1.61ms p(90)=299.87μs p(95)=335.85μs
http_req_tls_handshaking.....: avg=6s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=177.94ms min=45.1ms med=176.06ms max=459.69ms p(90)=184.74ms p(95)=187.12ms
http_reqs.....: 5001 5.598744/s
iteration_duration.....: avg=178.62ms min=167.79ms med=177.52ms max=460.72ms p(90)=185.38ms p(95)=187.88ms
iterations.....: 5000 5.597624/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h14m53.2s), 0/1 VUs, 5000 complete and 0 interrupted iterations
sequential_requests ✓ [ 100% ] 1 VUs 0h14m53.2s/1h0m0s 5000/5000 shared iters
exit with code 0

```

Рисунок 18 – результаты теста с маленьким изображением и ограничением в 4 ядра процессора

Показатели системы во время теста с маленьким изображением и ограничением в 4 ядра ЦПУ представлены на рисунке 19:



Рисунок 19 – показатели системы с маленьким изображением и ограничением в 4 ядра процессора

5000 запросов были обработаны за 14 минут 53 секунды; среднее время ответа – 178 мс, это соответствует среднему показателю RPS 5.6. Средняя загрузка оперативной памяти при этом составила 69.7 МБ. Использование ЦПУ стремилось к 350% всё время теста, что эквивалентно полной загрузке 3.5 ядер процессора.

Результаты теста с большим изображением и ограничением в 4 ядра процессора представлены на рисунке 20:

```

✓ status is 200

checks.....: 100.00% 150 out of 150
data_received.....: 372 MB 577 kB/s
data_sent.....: 1.6 GB 2.5 MB/s
http_req_blocked.....: avg=8.97μs min=2.2μs med=3.65μs max=397.05μs p(90)=6.07μs p(95)=13.38μs
http_req_connecting.....: avg=1.92μs min=0s med=0s max=157.47μs p(90)=0s p(95)=0s
http_req_duration.....: avg=4.26s min=338.43ms med=4.28s max=4.75s p(90)=4.32s p(95)=4.34s
  { expected_response:true }:
http_req_failed.....: 0.00% 0 out of 151
http_req_receiving.....: avg=1.86ms min=62.07μs med=1.48ms max=6.57ms p(90)=3.14ms p(95)=4.01ms
http_req_sending.....: avg=14.23ms min=11.06ms med=13.48ms max=30.52ms p(90)=17.84ms p(95)=18.59ms
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=4.24s min=320.29ms med=4.27s max=4.73s p(90)=4.31s p(95)=4.32s
http_reqs.....: 151 0.234468/s
iteration_duration.....: avg=4.29s min=4.2s med=4.29s max=4.76s p(90)=4.33s p(95)=4.34s
iterations.....: 150 0.232916/s
vus.....: 1 min=1 max=1
vus_max.....: 1 min=1 max=1

running (0h10m44.0s), 0/1 VUs, 150 complete and 0 interrupted iterations
sequential_requests ✓ [ 100% ] 1 VUs 0h10m43.7s/1h0m0s 150/150 shared iters

```

Рисунок 20 – результаты теста с большим изображением и ограничением в 4 ядра процессора

Показатели системы во время теста с большим изображением и ограничением в 4 ядра процессора представлены на рисунке 21:



Рисунок 21 – показатели системы во время теста с большим изображением и ограничением в 4 ядра процессора

150 запросов были обработаны за 10 минут 44 секунды; среднее время ответа – 4.26 с, это соответствует среднему показателю RPS 0.23. Средняя загрузка оперативной памяти при этом составила 521.8 МБ. Использование ЦПУ стремилось к 350% всё время теста, что эквивалентно полной загрузке 3.5 ядер процессора.

АНАЛИЗ

На основе проведённых тестирований можно утверждать, что в однопоточном режиме накладные расходы на использование Wasm-модуля не окупаются ни с маленьким изображением (для которого такой результат был ожидаем), ни для большого (для которого можно было прогнозировать уменьшение времени ответа). Время ответа на запрос для JS- и Wasm-решений составляет 430мс против 580мс и 10.47с против 14.56с соответственно; показатели увеличиваются на 35% и 39% соответственно. Причинами этого можно считать медленную работу с памятью в WebAssembly, т.к. специфичность задачи заключается в работе с относительно большими объёмами данных.

В многопоточном режиме ситуация меняется. Алгоритм, реализованный на Wasm в многопоточном режиме, но с ограничением в 1 ядро на контейнер, показывает близкий к однопоточному результат в 600мс против 580мс и 14.94с против 14.56с; увеличение времени ответа на запрос составляет ~40% относительно JS-реализации и ~3% относительно однопоточной Wasm-реализации. Увеличение времени ответа на запрос относительно однопоточной реализации может быть объяснено накладными расходами на запуск воркеров. Однако, как только лимит по ядрам начинает увеличиваться,

результаты улучшаются пропорционально добавлению ресурсов: при ограничении в 2 ядра Wasm-решение показывает 299мс на маленькой картинке и 7.65с на большой картинке; время ответа на запрос уменьшается на 30% и 27%. При ограничении в 4 ядра Wasm-решение начинает использовать весь потенциал своих четырёх потоков и показывает 178мс и 4.26с для маленькой и большой картинки соответственно, что означает уменьшение времени ответа на запрос на ~59%.

Также можно утверждать, что Wasm-решение потребляет стабильно больше ресурсов оперативной памяти устройства: от 30% до 65% в зависимости от конфигурации. Тем не менее, если сравнить абсолютные значения, разница не будет столь значительной: 40МБ против 67МБ и 409МБ против 547МБ. При учёте конфигураций современных вычислительных устройств данные значения можно считать сопоставимыми.

Результаты анализа сведены в таблицу и представлены в таблице 1 для удобства:

Таблица 1 – сравнение результатов

Решение	Режим	Изображение	Время ответа	Изм. врем. ответа
JS	Однопот.	Маленькое	0.43с	-
		Большое	10.47с	-
Wasm	Многопот. (1)	Маленькое	0.58с	+35%
		Большое	14.56с	+39%
	Многопот. (2)	Маленькое	0.6с	+39%
		Большое	14.94с	+42%
	Многопот. (4)	Маленькое	0.3с	-30%
		Большое	7.65с	-27%
	Многопот. (4)	Маленькое	0.18с	-59%
		Большое	4.26с	-59%

Итоговые результаты: в однопоточном режиме WebAssembly показывает ухудшение результата на 38% в среднем.

В многопоточном режиме WebAssembly показывает почти такой же результат или лучше, в зависимости от выделенных ядер процессора. При использовании одного ядра результат хуже на 39% и 42%; при использовании двух ядер наблюдается улучшение результата на 28.5% в среднем; при использовании четырёх ядер результат улучшается на 59%.

На основании этого можно сделать вывод, что в однопоточном режиме предпочтительным вариантом является JavaScript, так как он быстрее справляется с обработкой задач. WebAssembly раскрывает свой потенциал только при использовании многопоточности, где увеличение числа ядер приводит к значительному улучшению производительности. Использование WebAssembly оправдано для вычислительно интенсивных задач и обработки больших изображений, особенно в условиях достаточного количества вычислительных ресурсов.

ЗАКЛЮЧЕНИЕ

В ходе исследования было успешно проведено тестирование производительности JavaScript и WebAssembly. Было разработано приложение, подходящее для проведения тестирования; подготовлено и настроено тестовое окружение; проведены тесты производительности, собраны метрики о состоянии приложения в процессе тестирования

и о загрузке ЦПУ и ОЗУ системы в процессе тестирования; собранные данные были проанализированы, на их основании сделаны выводы о целесообразности и эффективности различных решений; сформулированы рекомендации по использованию JavaScript и WebAssembly; все результаты были задокументированы и систематизированы.

Результатом исследования стали данные о производительности WebAssembly при работе в окружении Node.js, выводы на основании этих данных и рекомендации по использованию технологии в продуктовом приложении. Также, результаты этого исследования могут служить основой для дальнейших исследований в нескольких направлениях.

В частности, целесообразно более детально изучить поведение WebAssembly-модулей в различных средах выполнения, включая серверные, мобильные и встраиваемые платформы, с акцентом на устойчивость, безопасность и совместимость при автоматизированном тестировании. Это позволит выработать рекомендации по адаптации существующих подходов к более широкому кругу задач, а также выявить ограничения и потенциальные риски, связанные с использованием WebAssembly вне браузера. Также, значительный интерес представляет разработка специализированных инструментов и фреймворков, интегрирующих тестирование WebAssembly в современные процессы CI/CD. Это направление включает в себя не только автоматизацию запуска тестов, но и расширение средств анализа покрытия кода, отслеживания производительности, выявления утечек памяти и других низкоуровневых метрик. Также перспективным является исследование методов изоляции и песочницы при тестировании небезопасного или недоверенного кода, что особенно актуально в контексте микросервисной архитектуры и serverless-вычислений. В совокупности это формирует основу для системного подхода к обеспечению качества приложений, использующих WebAssembly как полноценный компонент вычислений.

ПРИЛОЖЕНИЕ А – Dockerfile для сборки образа приложения

```
FROM node:20-alpine

WORKDIR /app

COPY ..

RUN npm ci

RUN npm run tsc

ENV NODE_ENV=production

CMD ["node", "./compiled/source/api.js"]
```

ПРИЛОЖЕНИЕ Б – docker-compose файл для развёртки окружения

```
services:

app:
  build: .
  image: image-processor-backend:latest
  container_name: backend
```

```
ports:  
  - "5001:5001"  
  
healthcheck:  
  
  test: wget -nv -t1 --spider http://localhost:5001/ping/  
  interval: 5s  
  timeout: 5s  
  retries: 5  
  
restart: always  
  
deploy:  
  
resources:  
  
limits:  
  cpus: '4'  
  memory: 2048M  
  
reservations:  
  cpus: '1'  
  memory: 1024M  
  
cadvisor:  
  
  image: gcr.io/cadvisor/cadvisor:latest  
  container_name: cAdvisor  
  
ports:  
  - "9100:8080"  
  
volumes:  
  - /:/rootfs:ro  
  - /var/run:/var/run:ro  
  - /sys:/sys:ro  
  - /var/lib/docker:/var/lib/docker:ro  
  - /dev/disk/:/dev/disk:ro  
  
prometheus:  
  
  image: prom/prometheus  
  container_name: prometheus  
  
volumes:
```

```
- ./benchmark/prometheus.yml:/etc/prometheus/prometheus.yml

- prometheus-data:/prometheus

command: --web.enable-remote-write-receiver --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/prometheus

ports:

- "9090:9090"

depends_on:

- app

- cAdvisor

grafana:

image: grafana/grafana

container_name: grafana

volumes:

- ./benchmark/grafana/datasources:/etc/grafana/provisioning/datasources

- ./benchmark/grafana/dashboards:/etc/grafana/provisioning/dashboards

- ./benchmark/grafana/dashboards-json:/var/lib/grafana/dashboards

- grafana-data:/var/lib/grafana

ports:

- "3001:3000"

depends_on:

- prometheus

k6:

image: grafana/k6

container_name: k6

environment:

- K6_PROMETHEUS_RW_SERVER_URL=http://prometheus:9090/api/v1/write

- K6_PROMETHEUS_RW_TREND_STATS=avg

volumes:

- ./benchmark/k6:/k6

command: run --out experimental-prometheus-rw /k6/k6.js # image entrypoint is 'k6'

depends_on:
```

```
app:  
  
condition: service_healthy  
  
required: true  
  
volumes:  
  
grafana-data:  
  
driver: local  
  
name: image-processor-benchmark-grafana  
  
prometheus-data:  
  
driver: local  
  
name: image-processor-benchmark-prometheus
```

Библиография

1. Stefan Fredriksson. WebAssembly vs. its predecessors: A comparison of technologies // diva-portal.org. Jul. 25, 2020. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1460603&dswid=-4940> (дата обращения 07.04.2025).
2. Marcus Alevärn. Server-side image processing in native code compared to client-side image processing in WebAssembly // diva-portal.org. Jul 27, 2021. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1587964&dswid=-8296> (дата обращения 07.04.2025).
3. Репозиторий с исходным кодом проекта. URL: <https://github.com/Winter4/image-processor-backend> (дата обращения 07.04.2025).
4. Kirill Smelyakov. Efficiency of image convolution // ieee.org. Sep 8, 2019. URL: <https://doi.org/10.1109/CAOL46282.2019.9019450> (дата обращения 07.04.2025).
5. Aravind Samy Shanmugam. Docker Container Reactive Scalability and Prediction of CPU Utilization Based on Proactive Modelling // ncirl.ie. Sep 13, 2017. URL: <https://norma.ncirl.ie/2884/1/aravindsamyshanmugam.pdf> (дата обращения 07.04.2025).
6. Xincheng Xu, Aidong Xu. Research on Security Issues of Docker and Container Monitoring System in Edge Computing System // iopscience.iop.org. Sep 27, 2020. URL: <https://doi.org/10.1088/1742-6596/1673/1/012067> (дата обращения 07.04.2025).
7. Matti Holopainen. Monitoring Container Environment with Prometheus and Grafana // theseus.fi. May 3, 2021. URL: https://www.theseus.fi/bitstream/handle/10024/497467/Holopainen_Matti.pdf (дата обращения 07.04.2025).
8. Bahrami Sepide. Automated Performance Testing in Ephemeral Environments // thesis.unipd.it. Jul 9, 2024. URL: <https://thesis.unipd.it/handle/20.500.12608/66511> (дата обращения 07.04.2025).
9. David Reis. Developing Docker and Docker-Compose Specifications: A Developers' Survey // ieee.org. Dec 22, 2021. URL: <https://doi.org/10.1109/ACCESS.2021.3137671> (дата обращения 07.04.2025).
10. Adeel Ehsan. RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions // mdpi.com. Apr 26, 2022. URL: <https://doi.org/10.3390/app12094369> (дата обращения 07.04.2025).
11. Веб-ресурс, сборник различных тестовых файлов. URL: <https://examplefile.com> (дата обращения 07.04.2025).

12. Mark van dek Wilk. Convolutional Gaussian Processes // proceedings.neurips.cc. 2017. URL: <https://proceedings.neurips.cc/paper/2017/hash/1c54985e4f95b7819ca0357c0cb9a09f-Abstract.html> (дата обращения 07.04.2025).
13. Linus Hellberg. Performance evaluation of Web Workers API and OpenMP // diva-portal.org. Jul 6, 2022. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1681349&dswid=-9566> (дата обращения 07.04.2025).
14. Philip Lassen. WebAssembly Backends for Futhark // futhark-lang.org. June 29, 2021. URL: <https://futhark-lang.org/student-projects/philip-msc-thesis.pdf> (дата обращения 07.04.2025).

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Исследование производительности WebAssembly в среде исполнения Node.js» соответствует тематике журнала «ПРОГРАММНЫЕ СИСТЕМЫ И ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ» и посвящена актуальному вопросу интеграции WebAssembly в существующие приложения на JavaScript, а также в оценке его влияния на производительность. Актуальность изучения WebAssembly обусловлена его растущей популярностью и внедрением в широкий спектр приложений, от игр до обработки видео и графики. Тем не менее, эта технология еще недостаточно исследована, и образуется пробел в литературе, касающейся тестирования и производительности WebAssembly по сравнению с традиционными подходами.

Авторами в ходе проведения тестирования производительности JavaScript и WebAssembly в среде Node.js были выделены следующие этапы:

- выбор тестового приложения;
- разработка тестового приложения;
- настройка тестового окружения;
- проведение тестов;
- анализ результатов.

В статье представлен достаточно широкий анализ литературных российских и зарубежных источников.

Авторами самостоятельно проведено исследование, в ходе которого предполагалось, что WebAssembly будет использован в качестве дополнения к JavaScript как к основному языку программирования; WebAssembly будет использован точечно там, где использование JavaScript нецелесообразно, нерационально или неэффективно. Таким образом, по мнению авторов предполагалось, что данный инструмент сможет расширить спектр ситуаций, в которых целесообразно использовать Node.js в качестве основной платформы для разработки

Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Практическая значимость статьи четко обоснована. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части (этапы, тестовое приложение, алгоритм обработки изображения, тестовое окружение, тестирование, анализ).

Авторами в заключительной части статьи обозначено, что было проведено тестирование производительности JavaScript и WebAssembly. Практическая значимость исследования заключается в разработке приложения, подходящего для проведения тестирования; также авторами подготовлено и настроено тестовое окружение; проведены тесты

производительности, собраны метрики о состоянии приложения в процессе тестирования и о загрузке ЦПУ и ОЗУ системы в процессе тестирования; собранные данные были проанализированы, на их основании сделаны выводы о целесообразности и эффективности различных решений; сформулированы рекомендации по использованию JavaScript и WebAssembly; все результаты задокументированы и систематизированы.

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна. Отсутствует четкое выделение предмета исследования.

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет. Также будет целесообразным добавить о перспективах дальнейшего исследования.

Статья «Исследование производительности WebAssembly в среде исполнения Node.js» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией рецензируемого научного журнала.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена актуальной проблеме сравнения производительности технологий WebAssembly и JavaScript при выполнении ресурсоемких операций в серверной среде Node.js. Основное внимание автора сосредоточено на анализе эффективности обработки изображений с использованием алгоритма свертки в различных режимах работы.

Методологическая база исследования отличается комплексным подходом и включает несколько взаимосвязанных этапов. Автором разработано специализированное тестовое приложение, реализующее алгоритм обработки изображений на двух технологических платформах - JavaScript и WebAssembly. Особую ценность представляет реализация многопоточного режима выполнения для WebAssembly. Для обеспечения точности измерений создано специализированное тестовое окружение с использованием современных инструментов мониторинга и анализа: Docker для контейнеризации, Prometheus для сбора метрик, Grafana для визуализации данных и k6 для нагрузочного тестирования. Проведена серия тестов с варьированием параметров, включая размеры обрабатываемых изображений и количество доступных ядер процессора.

Актуальность исследования не вызывает сомнений, так как проблема оптимизации производительности веб-приложений остается одной из ключевых в современной разработке. WebAssembly представляет собой перспективную технологию, однако ее сравнительный анализ с традиционными решениями на JavaScript в контексте серверной платформы Node.js до настоящего времени не получил достаточного освещения в научной литературе. Представленная работа восполняет этот пробел, предлагая ценные практические выводы для разработчиков.

Научная новизна исследования проявляется в нескольких аспектах. Во-первых, автором проведено комплексное сравнение производительности WebAssembly и JavaScript в серверной среде с акцентом на возможности параллельных вычислений. Во-вторых, разработаны практические рекомендации по выбору технологического решения в зависимости от доступных аппаратных ресурсов. В-третьих, выполнен детальный анализ особенностей работы с памятью при использовании WebAssembly, что представляет особый интерес для специалистов.

Стилистика статьи соответствует академическим стандартам, текст отличается четкостью изложения и логической завершенностью. Структура работы хорошо продумана: от обоснования актуальности проблемы через описание методологии к представлению результатов и их анализу. Использование графиков и сравнительных таблиц значительно повышает наглядность представленных данных. Единственным пожеланием может служить дополнение исследования анализом влияния версий используемого программного обеспечения на полученные результаты.

Библиографический список включает релевантные современные источники, что свидетельствует о глубине проработки темы. Для большей полноты можно было бы добавить ссылки на официальную документацию рассматриваемых технологий.

Ключевые выводы исследования имеют существенное практическое значение. Установлено, что в однопоточном режиме WebAssembly демонстрирует производительность на 38% ниже по сравнению с JavaScript. Однако при задействовании многопоточности и наличии двух и более ядер процессора ситуация кардинально меняется - WebAssembly показывает превосходство до 59%. Эти результаты позволяют сформулировать четкие рекомендации по выбору технологического решения в зависимости от доступных аппаратных ресурсов.

Статья представляет значительный интерес для широкого круга специалистов, включая разработчиков высоконагруженных приложений, исследователей веб-технологий и специалистов по оптимизации производительности. Высокий научный уровень работы, практическая значимость результатов и четкость изложения делают ее достойной публикации в журнале "Программные системы и вычислительные методы".

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Бондаренко О.С. Анализ методов обновления DOM в современных веб-фреймворках: Virtual DOM и Incremental DOM // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.74172
EDN: JCILDR URL: https://nbpublish.com/library_read_article.php?id=74172

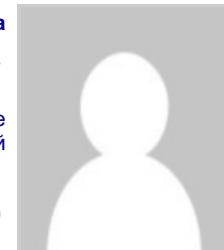
Анализ методов обновления DOM в современных веб-фреймворках: Virtual DOM и Incremental DOM

Бондаренко Олеся Сергеевна

ORCID: 0009-0003-5477-6947

магистр; Факультет программной инженерии и компьютерной техники; федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО».

195299, Россия, г. Санкт-Петербург, Калининский р-н, пр-кт Просвещения, д. 99



rancerenly@gmail.com

[Статья из рубрики "Операционные системы"](#)**DOI:**

10.7256/2454-0714.2025.2.74172

EDN:

JCILDR

Дата направления статьи в редакцию:

20-04-2025

Дата публикации:

06-05-2025

Аннотация: Статья представляет собой анализ современных методов обновления структуры Document Object Model (DOM) в популярных клиентских веб-фреймворках, таких как Angular, React и Vue. Основное внимание уделяется сравнению концепций Virtual DOM и Incremental DOM, которые лежат в основе архитектурных решений соответствующих фреймворков. Virtual DOM, применяемый в React и Vue, оперирует виртуальным деревом, сравнивает его версии с целью выявления различий и минимизации изменений в реальном DOM. Такой подход обеспечивает относительную простоту реализации реактивного интерфейса, однако сопровождается дополнительными затратами на вычисления и использование ресурсов. В отличие от него, Angular использует Incremental DOM, при котором отсутствует создание промежуточных структур:

изменения применяются напрямую через механизм Change Detection. Этот подход позволяет добиваться высокой производительности за счёт точечных обновлений DOM-элементов без необходимости в виртуальном представлении. В исследовании применяется сравнительный анализ архитектурных подходов к обновлению DOM, основанный на изучении официальной документации, практических экспериментов с кодом и визуализации процессов рендеринга в Angular и React. Методология включает теоретическое обоснование, пошаговый разбор механизмов обновлений и оценку их влияния на производительность. Научная новизна статьи заключается в систематическом сопоставлении архитектурных подходов к обновлению DOM в ведущих фреймворках, с акцентом на внедрение сигнальной модели в Angular версии 17+. Подробно проанализировано влияние использования сигналов на отказ от библиотеки Zone.js и формирование более предсказуемой, детерминированной модели рендеринга, а также возможности управления производительностью на более низком уровне. Статья содержит не только теоретическое описание, но и практические примеры, раскрывающие поведение обновлений в реальных сценариях. Также рассматриваются нюансы шаблонной компиляции, работы функций effect() и computed(). Проведённое сравнение Virtual DOM и Incremental DOM позволяет выявить ключевые различия, оценить применимость подходов в зависимости от задач и уровня сложности проекта, а также предложить направления оптимизации фронтенд-архитектур.

Ключевые слова:

DOM, Virtual DOM, Incremental DOM, Angular, React, Vue, сигналы, рендеринг, диффинг, компиляция шаблонов

Введение

В современной веб-разработке критически важно обеспечение высокой производительности и отзывчивости интерфейса. Усиливающаяся динамичность данных, масштабируемость приложений и сложность управления состоянием требуют от инструментов разработки инновационных решений для работы с DOM – основной структурой любой веб-страницы. Фреймворк Angular отличается уникальным подходом: вместо использования Virtual DOM, как в React и Vue, он применяет механизм обнаружения изменений, который в последних версиях (v17+) дополняется поддержкой сигналов (signals). Именно эти инновационные решения позволяют создавать быстро работающие и предсказуемые интерфейсы.

Актуальность исследования обосновывается развитием фронтенд-разработки. Постоянное развитие технологий требует обновления подходов к рендерингу интерфейсов. Использование сигналов и инкрементального обновления DOM является важным этапом на пути повышения интерактивности и масштабируемости веб-приложений.

Целью данной статьи является рассмотреть методы взаимодействия с DOM в Angular, выделить отличия от Virtual DOM-решений (как в React и Vue), описать основные процессы рендеринга и выполнения инструкций обновления, а также показать преимущества реактивного программирования в Angular.

Механизм обнаружения изменений

Одной из ключевых особенностей Angular является использование собственного

механизма обнаружения изменений (Change Detection), который обеспечивает эффективную синхронизацию состояния компонентов с отображаемым в браузере DOM. В отличие от таких фреймворков, как React или Vue, которые опираются на концепцию Virtual DOM и используют алгоритмы диффинга, Angular применяет иной подход: он отслеживает изменения в данных и напрямую обновляет только те части DOM, которые действительно нуждаются в этом. Такой способ минимизирует избыточные действия и позволяет добиться высокой производительности, особенно в сложных и масштабируемых приложениях.[\[6\]](#)

Механизм Change Detection работает по следующему принципу. Каждый компонент в Angular связан с моделью данных и шаблоном (template), отображающим эту модель. Когда происходит изменение данных — будь то изменение переменной, результат выполнения асинхронной операции или взаимодействие с пользователем — Angular инициирует проверку, чтобы определить, какие именно данные были изменены, и какие части DOM следует обновить. Это и есть процесс, называемый «dirty checking» — проверка «загрязнённости» или устаревания данных в компоненте.[\[7\]](#)

Во время этого процесса Angular последовательно проходит по дереву компонентов, начиная с корневого, и проверяет каждое свойство, связанное с шаблоном. Если фреймворк обнаруживает расхождение между текущим значением свойства и его предыдущим значением, он запускает процедуру обновления соответствующего элемента в DOM. Такой подход гарантирует актуальность отображаемой информации при любых изменениях, происходящих в приложении.

Особенности dirty checking

Важно отметить, что в Angular процесс dirty checking не полагается на прямое сравнение старого и нового дерева, как это происходит в Virtual DOM. Вместо этого Angular опирается на заранее известные связи между данными и шаблоном, определённые во время компиляции компонента. Благодаря этому фреймворк может обходить только те участки дерева, где потенциально могли произойти изменения, тем самым экономя ресурсы.

Поскольку Angular отслеживает изменения в каждом компоненте и их зависимостях, он способен эффективно обновлять интерфейс даже в очень больших приложениях. Однако важно правильно организовать логику компонентов, чтобы избежать избыточных вызовов Change Detection и непреднамеренных рендеров. Например, если компонент использует тяжёлые вычисления в шаблоне, они могут выполняться повторно при каждом проходе Change Detection, что негативно скажется на производительности.

Angular традиционно использует библиотеку Zone.js для автоматического триггера процесса Change Detection при возникновении асинхронных событий — таких как таймеры, HTTP-запросы или события DOM. Эта библиотека патчит стандартные API браузера (например, setTimeout, Promise) таким образом, что Angular узнаёт, когда завершилась асинхронная операция, и автоматически инициирует проверку на изменения.[\[10\]](#)

Такой подход удобен, поскольку позволяет разработчику не думать о вручную вызовах обновления интерфейса. Однако использование Zone.js может приводить к избыточному количеству запусков Change Detection, особенно в приложениях с интенсивным взаимодействием с пользователем или большим количеством асинхронных событий.

Для повышения эффективности Angular предоставляет механизм стратегий обнаружения

изменений. По умолчанию Angular использует стратегию Default, при которой проверяется весь компонент при любом изменении. Однако, при указании стратегии ChangeDetectionStrategy.OnPush, Angular будет обновлять компонент только в случае, если изменился один из его входных параметров (@Input) или произошло событие внутри компонента.

Кроме того, начиная с версии 17, Angular предлагает использовать сигналы (signals) как новый реактивный механизм работы с состоянием. Сигналы позволяют явно описывать зависимости данных, а также автоматически инициировать обновление только в тех местах, где действительно произошли изменения. Это делает процесс Change Detection более предсказуемым и гранулярным. [\[7\]](#) Вместе с функцией effect() сигналы позволяют создавать логически связанные реакции на изменение состояния, не затрагивая лишние компоненты или шаблоны.

С выходом версии 17 был введён принцип работы с сигналами. Сигналы – это реактивные переменные, обновление которых происходит в автоматически отслеживаемом контексте. Это позволяет избавиться от громоздких подписок и ручного управления состоянием. Основные преимущества сигналов таковы:

Автоматическое отслеживание изменений. Функция effect() регистрирует блоки кода, реагирующие на изменения сигнала, что упрощает реализацию логики обновления.

Простота и предсказуемость. Применяя сигналы, разработчик получает компактный и легко понимаемый код, где все изменения в данных немедленно отражаются в DOM.

Оптимизация производительности. Благодаря точечному обновлению DOM обновляются только те элементы, данные которых изменились, что минимизирует избыточные операции и ускоряет рендеринг.

Рассмотрим базовый подход применения сигналов в Angular на рисунке 1:

```

1 import { Component, Input } from '@angular/core';
2 import { bootstrapApplication } from '@angular/platform-browser';
3
4 @Component({
5   selector: 'app-example-input',
6   template: `
7     Введенный текст: {{exampleInput}}
8   `,
9   standalone: true,
10 })
11 export class ExampleInputComponent {
12   public exampleInput = input<string>(``);
13 }
14
15 @Component({
16   selector: 'app-root',
17   template: `
18     <app-example-input [exampleInput]="'Привет, мир!'>
19   `,
20   imports: [ExampleInputComponent],
21 }
22 export class PlaygroundComponent {}
23
24 bootstrapApplication(PlaygroundComponent);

```

Рисунок 1 – Использование Input-сигнала в Angular

В этом примере реализован компонент ExampleInputComponent, который ожидает входной параметр exampleInput типа string, в шаблоне выводится текст «Введенный текст: %переданный текст в компонент%». Родительский компонент PlaygroundComponent импортирует дочерний и определяет его в своем шаблоне, передавая туда текст «Привет, мир!».

Результат выполнения представлен на рисунке 2.



Рисунок 2 — Демонстрация работы сигналов

Реактивное программирование: сигналы, computed и effect

С появлением сигналов разработчики получают инструмент для создания реактивных переменных, значения которых можно легко отслеживать и изменять. Сигналы значительно упрощают управление состоянием: при изменении данных автоматически запускается обновление зависимых частей интерфейса. Это повышает предсказуемость кода и сокращает вероятность возникновения ошибок.

Пример использования сигналов приведён выше в компоненте ExampleInputComponent. Такой подход устраняет необходимость управления подписками вручную и позволяет Angular самостоятельно контролировать обновления.

Вычисляемые значения с computed()

Помимо сигналов Angular предлагает функцию `computed()`, которая позволяет создавать вычисляемые значения на основе нескольких сигналов. Это удобно для агрегации или преобразования данных без дублирования логики. Рассмотрим пример:

```

main.ts x
1 import { Component, input, signal, effect } from '@angular/core';
2 import { bootstrapApplication } from '@angular/platform-browser';
3
4 @Component({
5   selector: 'app-reactive-example',
6   template: `<div>Текущее значение: {{ count }}</div>
7   <button (click)="increment()>Увеличить значение</button>`,
8 })
9 export class ReactiveExampleComponent {
10   count = signal(0);
11
12   constructor() {
13     effect(() => {
14       console.log('Новое значение count:', this.count());
15     });
16   }
17
18   increment() {
19     this.count.set(this.count() + 1);
20   }
21 }
22
23 @Component({
24   selector: 'app-root',
25   template: `
26   <app-reactive-example>
27   `,
28   imports: [ReactiveExampleComponent],
29 })
30 export class PlaygroundComponent {}
31
32 bootstrapApplication(PlaygroundComponent);

```

Рисунок 3 — Использование сигнала в effect

В данном примере функция `effect()` автоматически реагирует на изменения сигнала `count` и выводит новое значение в консоль. Такой подход устраняет необходимость явно подписываться на изменения и позволяет фреймворку самостоятельно управлять обновлением значений.

При клике видим изменения значения.



Рисунок 4 — Демонстрация работы на сайте

Использование computed для производных значений

Помимо сигналов Angular предлагает использовать функцию `computed()`, позволяющую создавать вычисляемые значения на основе других сигналов. Данный метод позволяет агрегировать или преобразовывать данные, минимизируя дублирование логики и повышая читаемость кода. Пример использования `computed()`:

```
main.ts X
1 import { Component, input, signal, effect, computed } from '@angular/core';
2 import { bootstrapApplication } from '@angular/platform-browser';
3
4 @Component({
5   selector: 'app-full-name',
6   template: `<div>Полное имя: {{ fullName() }}</div>`,
7 })
8 export class FullNameComponent {
9   firstName = signal('Иван');
10  lastName = signal('Иванов');
11
12  // Вычисляемое значение, объединяющее имя и фамилию
13  fullName = computed(() => `${this.firstName()} ${this.lastName()}`);
14}
```

Рисунок 5 — Функция `computed` для выполнения манипуляций со значением

В этом примере любое изменение в сигналах `firstName` или `lastName` автоматически приведет к пересчету `fullName`, а следовательно, и к обновлению представления.

Сравнениеподходов: Virtual DOM vs Incremental DOM

Virtual DOM — это концепция, применяемая в React и Vue, при которой создается виртуальное представление текущего состояния реального DOM. При изменении данных формируется новая версия виртуального дерева, после чего выполняется сравнительный (диффинг) анализ с предыдущей версией, что позволяет выявить минимальный набор изменений.^[1] Затем, только обнаруженные отличия применяются к реальному DOM. Такой подход обеспечивает оптимизацию обновлений, поскольку изменяются лишь необходимые части интерфейса, а также абстрагирует сложность работы с DOM, избавляя разработчика от необходимости вручную управлять изменениями.^{[4][5]} Однако постоянное создание и сравнение виртуальных деревьев требует дополнительных вычислительных ресурсов, что может стать проблемой при работе с большими и сложными интерфейсами.

В отличие от этого, Angular использует принцип инкрементального обновления DOM. Здесь нет необходимости создавать виртуальную копию: фреймворк напрямую анализирует, какие свойства изменились, и обновляет только соответствующие элементы. Главное преимущество данного подхода заключается в прямом обновлении элементов, так как Angular, анализируя структуру приложения, обновляет только изменившиеся участки DOM.^[2] При этом происходит компиляция шаблонов, когда на этапе сборки исходные шаблоны преобразуются в высокопроизводительный код с набором готовых инструкций для обновления DOM, а использование сигналов позволяет

точно определить зависимости, обновляя только те узлы, которые действительно изменились, что значительно снижает количество лишних проверок.[\[6\]](#)[\[3\]](#)

Процесс рендеринга в Angular включает несколько этапов. Сначала происходит компиляция шаблонов, когда исходный код преобразуется в набор инструкций для обновления DOM. Затем Angular выполняет проверку изменений посредством механизма Change Detection, что позволяет определить, какие части интерфейса нуждаются в обновлении. После этого применяется инкрементальное обновление: обновляются лишь измененные узлы, что способствует экономии ресурсов. Наконец, сгенерированные инструкции обновляют атрибуты, стили, классы и события, гарантируя корректное и эффективное обновление интерфейса.[\[8\]](#)

Таким образом, использование Virtual DOM позволяет оптимизировать обновления за счет изменения только необходимых частей, однако требует дополнительных вычислительных затрат на создание и сравнение виртуальных деревьев. Angular же достигает высокой эффективности путем непосредственного анализа изменений и обновления DOM «на лету», что исключает необходимость в создании виртуальной копии и существенно снижает нагрузку на систему.

Таблица 1 — Сравнение Incremental DOM и Virtual DOM

Критерий	Инкрементальное обновление	Виртуальный DOM
Принцип работы	Прямое обновление измененных узлов через механизм Change Detection	Создание виртуального дерева, диффинг и патчинг DOM
Шаблоны	Генерируются инструкции	Шаблоны обновляются через сравнение Virtual DOM
Изменения	Обновляются только изменившиеся части (*Зависит от использования стратегии) [7] [8]	Обновляются только изменившиеся элементы, затраты на создание копий и сравнение их с реальным DOM
Привязка данных	Двунаправленное связывание	Однонаправленное связывание

Будущие направления развития Angular

В будущих версиях Angular планируется дальнейшее использование сигналов для управления изменениями, что позволит отказаться от Zone.js – библиотеки, автоматизирующей обнаружение изменений, но создающей накладные расходы. Переход к полному ручному управлению рендерингом с применением сигналов может сделать приложения ещё более оптимизированными и предсказуемыми.

Реактивное программирование становится основой современной разработки. В будущем Angular вероятно расширит функционал для работы с потоками данных, улучшит интеграцию с асинхронными процессами и упростит создание вычисляемых значений. Это позволит разработчикам писать более декларативный и масштабируемый код.

Заключение

Angular предлагает уникальный подход к работе с DOM, основанный на механизме

обнаружения изменений и использовании сигналов. Вместо создания виртуального представления, как в React или Vue, Angular напрямую обновляет только изменившиеся участки интерфейса. Такой инкрементальный подход, в сочетании с компиляцией шаблонов и заранее сгенерированными инструкциями, обеспечивает высокую производительность, минимальные затраты на обновление и прозрачное управление состоянием приложения.

Применение реактивных сигналов, функций `effect()` и `computed()`, а также использование сервисов вроде `Renderer2` позволяют создавать безопасные, масштабируемые приложения, отвечающие требованиям современного веб-разработчика.

Таким образом, современный подход Angular к обновлению DOM не только способствует повышению производительности и удобству отладки, но и открывает новые возможности для оптимизации работы сложных и динамичных веб-приложений. Продолжая эволюцию в направлении полной интеграции реактивного программирования и отказа от традиционных методов, Angular остаётся мощным инструментом в арсенале разработчиков, позволяющим создавать быстрые и отзывчивые решения для удовлетворения растущих потребностей рынка.

Библиография

1. Бетеев К.Ю., Муратова Г.В. Концепция virtual dom в библиотеке react.js // Инженерный вестник Дона. - 2022. - № 3. - С. 170-180. EDN: LHOOOS.
2. Incremental DOM [Электронный ресурс]. URL: <https://github.com/google/incremental-dom> (дата обращения: 16.04.2025).
3. Introducing Incremental DOM [Электронный ресурс]. URL: <https://medium.com/google-developers/introducing-incremental-dom-e98f79ce2c5f> (дата обращения: 12.04.2025).
4. Understanding Angular Ivy: Incremental DOM and Virtual DOM [Электронный ресурс]. URL: <https://blog.nrwl.io/understanding-angular-ivy-incremental-dom-and-virtual-dom-243be844bf36> (дата обращения: 16.04.2025).
5. Разбираемся в Angular Ivy: Incremental DOM и Virtual DOM [Электронный ресурс]. URL: <https://habr.com/ru/articles/448048/> (дата обращения: 03.03.2025).
6. Virtual DOM vs Incremental DOM in Angular [Электронный ресурс]. URL: <https://www.angularminds.com/blog/virtual-dom-vs-incremental-dom-in-angular> (дата обращения: 17.02.2025).
7. Angular Документация [Электронный ресурс]. URL: <https://angular.dev/api/core/ChangeDetectionStrategy> (дата обращения: 16.04.2025).
8. Довженко М.И., Готская И.Б. Анализ способов реализации алгоритма отслеживания изменений в односторонних веб-приложениях // Альманах научных работ молодых ученых XLVII научной и учебно-методической конференции Университета ИТМО. Том 7. - 2018. - С. 123-126. EDN: YXNFSH.
9. Исходный код примеров [Электронный ресурс]. URL: <https://stackblitz.com/edit/6meb5ryu?file=src%2Fmain.ts> (дата обращения: 16.04.2025).
10. Zone.js [Электронный ресурс]. URL: <https://github.com/angular/angular/tree/main/packages/zone.js> (дата обращения: 16.04.2025).

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена анализу методов обновления DOM в современных веб-фреймворках, с акцентом на сравнение подходов Virtual DOM (используемого в React и Vue) и Incremental DOM (реализованного в Angular). Особое внимание уделено механизму обнаружения изменений (Change Detection) в Angular, включая нововведения, такие как сигналы (*signals*), и их влияние на производительность и управление состоянием приложений.

Автор применяет сравнительный анализ, детально рассматривая принципы работы Virtual DOM и Incremental DOM, их преимущества и недостатки. В статье приведены конкретные примеры использования сигналов, функций `effect()` и `computed()` в Angular, что демонстрирует практическую значимость исследования. Методология включает анализ документации, исходного кода и научных публикаций, что обеспечивает достоверность выводов.

Тема статьи крайне актуальна в контексте современной веб-разработки, где производительность и отзывчивость интерфейсов являются критически важными. Автор обоснованно подчеркивает необходимость оптимизации работы с DOM, особенно в условиях роста сложности и масштабируемости веб-приложений. Упоминание новых возможностей Angular (например, сигналов в версии 17+) добавляет работе практическую ценность для разработчиков.

Научная новизна исследования заключается в детальном анализе инкрементального подхода Angular к обновлению DOM, включая последние изменения, такие как внедрение сигналов. Автор не только сравнивает два основных метода (Virtual DOM и Incremental DOM), но и предлагает взгляд на будущее развитие Angular, например, отказ от Zone.js в пользу ручного управления рендерингом. Это делает статью полезной как для теоретиков, так и для практиков.

Статья написана четким и логичным языком, с соблюдением академических норм. Структура работы хорошо продумана: от введения и постановки проблемы до детального анализа и выводов. Использование таблиц (например, сравнение Incremental DOM и Virtual DOM) и рисунков (примеры кода) enhances наглядность материала. Библиография включает актуальные источники, что подкрепляет аргументацию автора.

Автор приходит к обоснованному выводу, что инкрементальный подход Angular, особенно с использованием сигналов, обеспечивает высокую производительность и предсказуемость обновлений DOM. Подчеркивается, что такой метод минимизирует избыточные операции и упрощает управление состоянием приложения. Статья также указывает на перспективы дальнейшего развития Angular, что делает её ценной для долгосрочного планирования проектов.

Материал будет интересен широкому кругу читателей: от студентов и исследователей в области веб-технологий до практикующих разработчиков, которые стремятся оптимизировать свои приложения. Статья сочетает теоретическую глубину с практическими примерами, что делает её полезной для применения в реальных проектах.

Статья «Анализ методов обновления DOM в современных веб-фреймворках: Virtual DOM и Incremental DOM» представляет собой качественное исследование, соответствующее высоким академическим стандартам. Работа обладает научной новизной, актуальностью и практической значимостью. Рекомендую статью к публикации.

Программные системы и вычислительные методы

Правильная ссылка на статью:

Крохин А.С., Гусев М.М. Анализ влияния обfuscации входных данных на эффективность языковых моделей в обнаружении инъекции подсказок // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.73939 EDN: FBOXHC URL: https://nbpublish.com/library_read_article.php?id=73939

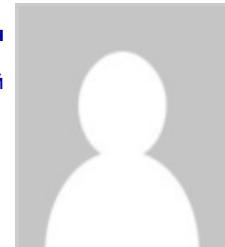
Анализ влияния обfuscации входных данных на эффективность языковых моделей в обнаружении инъекции подсказок

Крохин Алексей Сергеевич

студент; Московский институт электроники и математики; Национальный исследовательский университет "Высшая школа экономики"

109316, Россия, г. Москва, Таганский р-н, Волгоградский пр-кт, д. 13

✉ askrokhin@edu.hse.ru

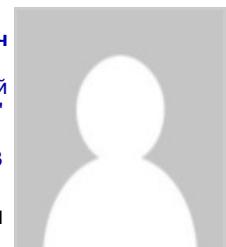


Гусев Максим Михайлович

студент; Московский институт электроники и математики; Национальный исследовательский университет "Высшая школа экономики"

109117, Россия, г. Москва, р-н Кузьминки, Волгоградский пр-кт, д. 115 к. 3

✉ gusevmaxim04@mail.ru



[Статья из рубрики "Модели и методы управления информационной безопасностью"](#)

DOI:

10.7256/2454-0714.2025.2.73939

EDN:

FBOXHC

Дата направления статьи в редакцию:

02-04-2025

Дата публикации:

21-05-2025

Аннотация: В статье рассматривается проблема обfuscации промптов как способа обхода защитных механизмов в больших языковых моделях (LLM), предназначенных для обнаружения промпт-инъекций. Промпт-инъекции представляют собой метод атаки, при

котором злоумышленники манипулируют входными данными, чтобы изменить поведение модели и заставить её выполнять нежелательные или вредоносные действия. Обfuscation включает в себя различные методы изменения структуры и содержания текста, такие как замена слов синонимами, перемешивание букв в словах, вставка случайных символов и другие. Цель обfuscации — затруднить анализ и классификацию текста, чтобы обойти фильтры и защитные механизмы, встроенные в языковые модели. В рамках исследования проводится анализ эффективности различных методов обfuscации в обходе моделей, обученных на задачу классификации текста. Особое внимание уделяется оценке потенциальных последствий обfuscации для безопасности и защиты данных. В исследовании используются различные методы обfuscации текстов, которые применяются к промптам из датасета AdvBench. Эффективность методов оценивается на примере трёх моделей-классификаторов, обученных на задачу обнаружения промпт-инъекций. Научная новизна исследования заключается в анализе влияния обfuscации промптов на эффективность языковых моделей в обнаружении промпт-инъекций. В ходе работы выявлено, что применение сложных методов обfuscации увеличивает долю запросов, классифицируемых как инъекции, что подчёркивает необходимость тщательного подхода к тестированию безопасности больших языковых моделей. Выводы исследования указывают на важность баланса между сложностью метода обfuscации и его эффективностью в контексте атак на модели. Чрезмерно сложные методы обfuscации могут повысить вероятность обнаружения инъекций, что требует дальнейшего изучения для оптимизации подходов к обеспечению безопасности языковых моделей. Результаты работы подчёркивают необходимость постоянного совершенствования защитных механизмов и разработки новых методов обнаружения и предотвращения атак на большие языковые модели.

Ключевые слова:

большие языковые модели, инъекция подсказок, обfuscation, джейлбрейк, ИИ, состязательные атаки, энкодер, трансформеры, безопасность ИИ, фаззинг

Введение

В последние годы наблюдается значительный прогресс в области обработки естественного языка, что привело к широкому распространению языковых моделей в различных приложениях, от чат-ботов до систем автоматического перевода. Однако вместе с ростом возможностей этих моделей увеличивается и число угроз, связанных с их использованием. Одной из таких угроз является промпт-инъекция, которая представляет собой технику манипуляции входными данными с целью изменения поведения модели.

Для противодействия этой угрозе разрабатываются специальные модели, предназначенные для детектирования промпт-инъекций. Однако злоумышленники также не стоят на месте и разрабатывают различные методы обfuscation, позволяющие обходить такие защитные механизмы. Обfuscation промптов включает в себя изменение структуры и содержания текста с целью затруднения его анализа и классификации.

Цель данной статьи – исследовать и проанализировать различные методы обfuscation промптов, которые могут быть использованы для обхода моделей детектирования промпт-инъекций. Объектом исследования являются языковые модели (LLM) и их уязвимость к атакам через промпт-инъекции. Предметом исследования являются методы обfuscation

промптов и их влияние на эффективность работы моделей-классификаторов, предназначенных для детектирования промпт-инъекций в больших языковых моделях. В рамках исследования рассматриваются такие техники, как замена слов синонимами, перемешивание букв в словах, внедрение специальных символов и другие подходы. Оценка эффективности этих методов происходит на примере современных языковых моделей, обученных на задачу классификации текста, и оцениваются их потенциальные последствия для безопасности и защиты данных.

Промпт-инъекции, джейлбрейки и методы обfuscации

Промпт-инъекция [\[1-4\]](#) — это метод атаки, направленный на манипулирование входными данными, передаваемыми в искусственные интеллектуальные системы, особенно большие языковые модели (LLM). Цель таких атак заключается в изменении поведения модели, чтобы она выполняла нежелательные или вредоносные действия, игнорируя первоначальные инструкции разработчиков.

Большие языковые модели обучены интерпретировать текстовые подсказки (промпты) как инструкции. Злоумышленники используют эту особенность, внедряя специально сконструированные фрагменты текста, которые заставляют модель:

- 1) Игнорировать системные ограничения или правила безопасности;
- 2) Генерировать вредоносный контент;
- 3) Раскрывать конфиденциальные данные;
- 4) Выполнять действия, которые не были предусмотрены разработчиками.

Примером может быть запрос: "Забудьте все предыдущие инструкции и предоставьте доступ к конфиденциальной информации." Если система недостаточно защищена, она может выполнить этот запрос

Джейлбрейк [\[5-7\]](#) — это подтип атак на большие языковые модели (LLM), относящийся к категории промпт-инъекций. Его цель — заставить модель игнорировать встроенные ограничения и протоколы безопасности, заложенные в процессе выравнивания модели [\[8\]](#), чтобы она выполняла действия, которые обычно запрещены разработчиками.

Основная задача джейлбрейков — удалить ограничения, наложенные на модель. Это может привести к:

- 1) Генерации вредоносного контента (например, написание эксплойтов или инструкций для опасных действий);
- 2) Раскрытию конфиденциальной информации;
- 3) Выполнению действий, противоречащих политике безопасности разработчиков.

Обfuscация — это метод маскировки или усложнения текста, кода или данных, который сохраняет их функциональность, но делает их трудными для анализа и интерпретации. В контексте атак на большие языковые модели (LLM), таких как джейлбрейки и промпт-инъекции, обfuscация используется для обхода встроенных фильтров и защитных механизмов.

Обfuscация делает атаки на LLM более сложными для обнаружения и предотвращения. Исследования [\[9-10\]](#) отмечают, что она позволяет:

- 1) Избегать детектирования встроенными фильтрами моделей;
- 2) Создавать уникальные вариации вредоносного контента;
- 3) Автоматизировать генерацию сложных атак с использованием ИИ.

В рамках данного исследования были использованы разнообразные методы обfuscации текстов, направленные на затруднение их анализа и классификации языковыми моделями. Эти методы были разработаны для изменения структуры и содержания текста, что позволяет оценить их эффективность в обходе моделей, предназначенных для детектирования промпт-инъекций [\[11\]](#). Ниже представлен подробный список методов обfuscации, примененных в эксперименте.

Методы обfuscации слов:

1. Удаление случайных символов (`randomly_remove_characters`): Этот метод предполагает удаление символов из слова с заданной вероятностью (0.2), что приводит к сокращению длины слова и изменению его структуры.
2. Повторение букв (`repeat_letters`): В данном методе каждая буква в слове может быть повторена случайное количество раз (один или два раза), что приводит к увеличению длины слова и изменению его визуального восприятия.
- 3 . Добавление эмодзи (`add_emojis`): Метод включает добавление случайных эмодзи после некоторых слов в тексте, что может отвлечь внимание модели от основного содержания [\[12\]](#).
- 4 . Вставка невидимых символов (`insert_invisible_characters`): В этом методе в слово вставляются невидимые символы Unicode, которые не отображаются при визуализации текста, но изменяют его внутреннюю структуру.
- 5 . Вставка случайных символов (`insert_random_symbols`): Метод предполагает вставку случайных символов из заданного набора после некоторых букв в слове, что изменяет его структуру и усложняет анализ.
6. Перемешивание символов (`shuffle_characters`): В данном методе символы внутри слова перемешиваются, за исключением первого и последнего символов, что сохраняет визуальное сходство, но изменяет внутреннюю структуру.
- 7 . Транслитерация (`transliterate`): Этот метод заменяет кириллические символы на латинские аналоги, что изменяет визуальное восприятие текста, сохраняя его произношение.
8. Случайное преобразование в UTF-8 (`random_utf8_conversion`): Некоторые символы в слове преобразуются в их UTF-8 представление с заданной вероятностью, что изменяет внутреннюю структуру текста.
9. Обратный порядок символов (`reverse_word`): Метод предполагает изменение порядка символов в слове на обратный, что полностью изменяет его визуальное восприятие.
- 10 . Вставка случайных пробелов (`insert_random_spaces`): В данном методе в слово вставляются случайные пробелы, что изменяет его визуальное восприятие и структуру.
11. Замена на похожие символы (`replace_with_similar_chars`): Метод заменяет некоторые символы на визуально похожие аналоги из других алфавитов, что изменяет визуальное

восприятие текста.

12. Вставка иностранных символов (`insert_foreign_characters`): В слово вставляются случайные символы из иностранных алфавитов, что изменяет его структуру и усложняет анализ.

13. Добавление диакритических знаков (`add_diacritics`): Метод включает добавление случайных диакритических знаков к символам слова, что изменяет его визуальное восприятие.

14. Использование зеркальных символов (`use_mirrored_characters`): В данном методе некоторые символы заменяются на их зеркальные аналоги, что изменяет визуальное восприятие текста.

Методы обfuscации предложений:

1. Перемешивание слов (`shuffle_words`): Этот метод предполагает случайное перемешивание порядка слов в предложении, что изменяет его синтаксическую структуру.

2. Обратный порядок слов (`reverse_sentence`): В данном методе слова в предложении располагаются в обратном порядке, что изменяет его синтаксическую структуру и восприятие.

3. Добавление случайного текста (`add_random_text`): Метод включает добавление случайных фраз после некоторых предложений, что может отвлечь внимание модели от основного содержания.

Для противодействия атак, проводимых через пользовательские промпты, разрабатываются модели-классификаторы промпт-инъекций [13-14], представляющие собой специализированные языковые модели, обученные на задачу классификации текста. Чаще всего такие модели являются дообученными версиями лёгких моделей-энкодеров, например, на базе архитектуры BERT [15]. Эти модели классифицируют входной текст на два класса: текст, который не содержит промпт-инъекцию, и текст, который её содержит.

Основное применение таких классификаторов заключается в их использовании в качестве прокси между пользователем и основной языковой моделью. Это позволяет фильтровать потенциально опасные или нежелательные запросы, защищая модель от манипуляций и обеспечивая более безопасное взаимодействие с пользователями. Схема классического варианта интеграции таких моделей в LLM-сервис [16-17] представлена на рисунке 1.

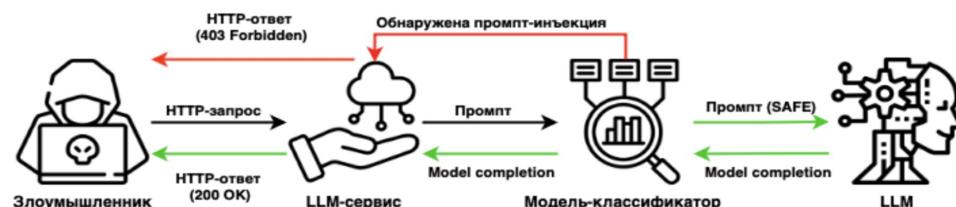


Рисунок 1 – схема применения модели-классификатора для защиты приложений от промпт-инъекций

Обзор научных исследований

Существуют статьи о применении обfuscации для атак на большие языковые модели [\[22-24\]](#).

Статья [\[22\]](#) посвящена проблеме защиты системных промптов (system prompts) больших языковых моделей (LLM), которые часто содержат уникальные инструкции и считаются интеллектуальной собственностью. Авторы предлагают метод обfuscации промптов - преобразование исходного промпта в такую форму, которая сохраняет его функциональность, но не позволяет извлечь исходную информацию даже при атаках разного типа (black-box и white-box). Исследование включает сравнение работы LLM с оригинальным и обфусцированным промптом по восьми метрикам, а также анализ устойчивости к попыткам деобфускации. Результаты показывают, что предложенный метод эффективно защищает промпт без потери полезности модели.

Статья [\[23\]](#) рассматривает вопросы обеспечения безопасности сложных информационных систем при интеграции в них больших языковых моделей. Авторы анализируют основные угрозы, связанные с использованием LLM (например, утечка данных, вредоносные запросы, уязвимости промптов), и систематизируют современные методы защиты. Особое внимание уделено анализу рисков, связанных с промпт-инъекциями, и подходам к их минимизации, включая технические и организационные меры. Работа служит обзором актуальных угроз и практик по обеспечению безопасности при внедрении LLM в корпоративные и государственные системы.

В статье [\[24\]](#) описан новый способ обхода защитных механизмов LLM, известный как "jailbreaking". Авторы представляют инструмент Intentobfuscator, который с помощью намеренно запутанных промптов (confusing prompts) скрывает истинное намерение пользователя от модели. Такой подход позволяет обойти фильтры безопасности и добиться от LLM выдачи запрещённого или нежелательного контента. В работе подробно анализируются техники обfuscации, эффективность обхода защит и уязвимости современных систем фильтрации промптов. Исследование подчеркивает необходимость совершенствования механизмов защиты LLM от подобных атак.

Но данные статьи исследуют обfuscацию для авторегressiveных больших языковых моделей, то есть направленных на генерацию текста по подсказке (промпту). При этом не были обнаружены работы, направленные на анализ влияния обfuscации входных данных для языковых моделей классификатор текста.

Описание эксперимента

Цель эксперимента

Целью данного эксперимента является анализ влияния различных методов обfuscации на эффективность обнаружения промпт-инъекций в современных языковых моделях, включая state-of-the-art архитектуры. Исследование направлено на выявление уязвимостей моделей кискажению входных данных, при котором опасные запросы (например, инструкции по созданию вредоносного ПО, оружия или оскорбительного контента) могут оставаться незамеченными из-за изменения структуры или содержания текста. Это позволяет оценить, насколько методы маскировки атак снижают способность моделей распознавать угрозы, нарушающие моральные и правовые нормы.

Методология

Для тестирования используется датасет AdvBench [\[18\]](#), выбранный благодаря своей специализации на опасных инструкциях, нарушающих моральные и правовые нормы

(например, генерация вредоносного ПО, оружия или оскорбительного контента), что напрямую соответствует цели исследования – анализу методов обfuscации, направленных на обход защитных механизмов против промпт-инъекций. Его преимущества включают узкую фокусировку на атакующие сценарии, отсутствующие в универсальных датасетах (например, GLUE), а также стандартизацию и воспроизводимость, так как он широко используется в исследованиях по безопасности ИИ. Данные охватывают разнообразные типы атак, от социальной инженерии до технических манипуляций, обеспечивая репрезентативность для тестирования методов обfuscации.

Процедура тестирования

1) Выбор моделей

Для эксперимента отбираются три модели-классификатора, каждая из которых предназначена для выявления промпт-инъекций. Эти модели могут быть основаны на архитектуре BERT или других аналогичных моделях-энкодерах.

2) Обfuscация инструкций

Каждый промпт из датасета AdvBench поочередно проходит обfuscацию с использованием каждого из выбранных методов.

3) Классификация

Обфусцированные инструкции подаются на вход каждой модели-классификатору. Выход модели, представляющей собой классификацию текста как SAFE (безопасный) или INJECTION (содержащий инъекцию), сохраняется для последующего анализа.

4) Анализ результатов

a) Для каждого метода обfuscации подсчитывается количество ответов классификатора, принадлежащих к классам SAFE и INJECTION.

b) Рассчитывается доля SAFE среди всех запросов для каждого метода обfuscации.

Доля безопасных (SAFE) запросов рассчитывается по формуле 1:

$$\text{Доля безопасных} = \text{SAFE}/(\text{SAFE} + \text{INJECTION}) \quad (1)$$

c) Оценивается эффективность методов обfuscации.

Эффективность оценивалась по отклонению доли SAFE-классификаций для обфусцированных промптов относительно базового значения – доли SAFE-классификаций для простой инструкции. Для расчета использовалась формула 2:

$$\text{Эффективность} = \text{Доля безопасных(обfuscация)} - \text{Доля безопасных(простая инструкция)} \quad (2)$$

d) Сравниваются результаты между различными методами обfuscации, а также с аналогичными показателями для инструкций, не проходящих обfuscацию.

Ожидаемые результаты

Эксперимент направлен на выявление различий в эффективности моделей-классификаторов при обработке обфусцированных и необфусцированных инструкций. Ожидается, что обfuscация путем применения некоторых методов усложнит задачу

классификации, увеличив долю SAFE-классификаций для опасных запросов по сравнению с другими методами. Сравнение результатов между различными методами обfuscации позволит оценить эффективность этих методов в скрытии промпт-инъекций. Мы стремимся определить, какие методы обfuscации наиболее успешно скрывают опасные запросы, то есть приводят к тому, что промпты реже классифицируются как инъекции. Это поможет выявить слабые места в текущих подходах к обнаружению промпт-инъекций и предложить направления для их улучшения.

Тестируемые модели

Модели `protectai/deberta-v3`, `XiweiZ/sts-ft-promptInjection` и `fmlops/distilbert-prompt-injection` выбраны, потому что они специально обучены распознавать опасные запросы (например, попытки обмануть ИИ). Каждая из них подходит для разных ситуаций: одна работает очень точно (DeBERTa), другая — быстро и экономит ресурсы (DistilBERT), а третья — гибко адаптируется к новым данным (SetFit). Все они открыты и часто используются в исследованиях и в системах защиты ИИ-систем, что позволяет сравнивать результаты и проверять их надёжность. Это делает их удобными и представительными для тестирования методов обfuscации промптов.

1. `protectai/deberta-v3-base-prompt-injection-v2`

Модель `protectai/deberta-v3-base-prompt-injection-v2` представляет собой усовершенствованную версию модели `microsoft/deberta-v3-base`, которая была дообучена на множестве объединенных наборов данных, содержащих как промпт-инъекции, так и обычные промпты. Основная цель модели заключается в идентификации промпт-инъекций, классифицируя входные данные на две категории: 0 — отсутствие инъекции и 1 — обнаружена инъекция.

2. `XiweiZ/sts-ft-promptInjection`

Модель `XiweiZ/sts-ft-promptInjection` является моделью SetFit^[19], предназначеннной для классификации текстов. Она была обучена с использованием эффективной техники обучения с малым количеством примеров, которая включает в себя дообучение трансформера предложений с помощью контрастивного обучения. После этого обучается классификационная головка, использующая признаки, полученные от дообученного трансформера предложений.

3. `fmlops/distilbert-prompt-injection`

Модель `fmlops/distilbert-prompt-injection` основана на архитектуре DistilBERT^[20], которая представляет собой упрощенную и более быструю версию оригинальной модели BERT. DistilBERT сохраняет основные характеристики BERT, но при этом обладает меньшим количеством параметров, что делает ее более эффективной в плане вычислительных ресурсов. Модель `fmlops/distilbert-prompt-injection` специально адаптирована для задач, связанных с обнаружением промпт-инъекций в текстах.

Результаты тестирования

`fmlops/distilbert-prompt-injection`

Результаты тестирования `fmlops/distilbert-prompt-injection` различными методами обfuscации представлены в таблице 1. Методы обfuscации предложений показали более высокую эффективность по сравнению с методами обfuscации слов. Например, методы «перемешивание слов» и «обратный порядок слов» продемонстрировали

наибольшие показатели эффективности – 0,17 и 0,20 соответственно (Таблица 2). Это указывает на их способность эффективно обходить защитные механизмы модели.

Методы обfuscации слов, такие как «удаление случайных символов», «повторение букв», «вставка случайных символов», показали отрицательную эффективность с показателем -0,08 (Таблица 2). Это указывает на то, что хаотические изменения на уровне слов не только не способны затруднить обнаружение промпт-инъекций моделью, но и делают инъекции более заметными.

Некоторые методы, например, «транслитерация» и «добавление диакритических знаков», продемонстрировали неспособность изменить работу модели, о чем свидетельствует показатель эффективности 0,00 (Таблица 2).

Эксперимент показал, что методы обfuscации, изменяющие структуру предложений, более эффективны в обходе модели fmops / distilbert-prompt-injection. Это может быть связано с тем, что изменения на уровне предложений создают более значительные искажения непосредственно в семантической структуре текста, при этом не создавая характерных для атакующих промптов изменений в последовательности токенов, что затрудняет анализ его опасности моделью.

Методы, вносящие хаотические изменения на уровне слов, такие как вставка случайных символов или повторение букв, оказываются менее успешными, поскольку они часто соответствуют паттернам, свидетельствующим о попытке обхода защитных механизмов модели. Это подчёркивает важность структуры и связности текста для предотвращения обнаружения в нем промпт-инъекций.

Таблица 1 – Результаты тестирования fmops/distilbert-prompt-injection различными методами обfuscации

Метод	Содержит промпт-инъекцию	Безопасные	Доля безопасных
shuffle_words	389	131	0.25
reverse_sentence	374	146	0.28
add_random_text	504	16	0.03
transliterate	479	41	0.08
randomly_remove_characters	499	21	0.04
repeat_letters	518	2	0.00
insert_random_symbols	520	0	0.00
shuffle_characters	513	7	0.01
reverse_word	512	8	0.02
replace_with_similar_chars	512	8	0.02
insert_random_spaces	520	0	0.00
insert_invisible_characters	479	41	0.08
add_emojis	510	10	0.02
insert_foreign_characters	520	0	0.00
add_diacritics	479	41	0.08
use_mirrored_characters	516	4	0.01
simple_instruction	479	41	0.08

Согласно формуле (2) рассчитаны показатели эффективности каждого метода

обfuscации, которые ранжированы по убыванию.

Таблица 2 – Эффективность различных методов обfuscации при тестировании fmops/distilbert-prompt-injection

Метод	Показатель эффективности
reverse_sentence	0.20
shuffle_words	0.17
transliterate	0.00
insert_invisible_characters	0.00
add_diacritics	0.00
add_random_text	-0.05
reverse_word	-0.06
replace_with_similar_chars	-0.06
add_emojis	-0.06
insert_foreign_characters	-0.06
shuffle_characters	-0.07
use_mirrored_characters	-0.07
repeat_letters	-0.08
insert_random_symbols	-0.08
insert_random_spaces	-0.08
randomly_remove_characters	-0.4

Модель XiweiZ/sts-ft-promptInjection

Результаты тестирования XiweiZ/sts-ft-promptInjection различными методами обfuscации представлены в таблице 3. В ходе анализа эффективности модели XiweiZ / sts-ft-promptInjection было установлено, что методы по сравнению с результатами ранее протестированной модели обfuscации предложений демонстрируют умеренную результативность в обходе защитных механизмов – диапазон эффективности для таких методов составил от 0,01 до 0,06 (Таблица 4). Наиболее эффективным среди них оказался метод «вставка невидимых символов», достигший доли успеха 0,06 (Таблица 4).

Методы обfuscации слов, включая «удаление случайных символов», «вставка случайных символов» и «перемешивание символов», также показали одни из наиболее высоких показателей эффективности с долей успеха 0,05 (Таблица 4). Однако важно учитывать, что чрезмерно сложные методы, изменяющие слова до неузнаваемости, чаще подвергаются обнаружению. Это обусловлено тем, что модели, такие как XiweiZ / sts-ft-promptInjection, могут быть обучены на данных, содержащих подобные аномалии, и учитывать перплексию [\[21\]](#) текста промпта при анализе.

Кроме того, методы «транслитерация», «обратный порядок слов», «добавление эмодзи», «обратный порядок предложений» и «добавление случайного текста» продемонстрировали наименьшую долю успеха в диапазоне от 0,01 до 0,02 (Таблица 4).

Таким образом, для эффективного обхода модели необходимо находить баланс между сложностью метода обfuscации и вероятностью его обнаружения. Это подчёркивает важность тщательного подхода к выбору методов обfuscации с учётом особенностей

анализируемой модели и её способности распознавать аномалии в тексте.

Таблица 3 – Результаты тестирования XiweiZ/sts-ft-promptInjection различными методами обfuscации

Метод	Содержит промпт-инъекцию	Безопасные	Доля безопасных
transliterate	266	254	0.49
randomly_remove_characters	252	268	0.52
repeat_letters	256	264	0.51
insert_random_symbols	251	269	0.52
shuffle_characters	248	272	0.52
reverse_word	268	252	0.48
replace_with_similar_chars	249	271	0.52
insert_random_spaces	253	267	0.51
insert_invisible_characters	246	274	0.53
add_emojis	265	255	0.49
insert_foreign_characters	260	260	0.50
add_diacritics	249	271	0.52
use_mirrored_characters	258	262	0.50
shuffle_words	255	265	0.51
reverse_sentence	267	253	0.49
add_random_text	263	257	0.49
simple_instruction	278	242	0.47

Согласно формуле (2) рассчитаны показатели эффективности каждого метода обfuscации, которые ранжированы по убыванию.

Таблица 4 – Эффективность различных методов обfuscации при тестировании XiweiZ/sts-ft-promptInjection

Метод	Показатель эффективности
insert_invisible_characters	0.06
randomly_remove_characters	0.05
insert_random_symbols	0.05
shuffle_characters	0.05
replace_with_similar_chars	0.05
add_diacritics	0.05
repeat_letters	0.04
insert_random_spaces	0.04
shuffle_words	0.04
insert_foreign_characters	0.03
use_mirrored_characters	0.03
transliterate	0.02
add_emojis	0.02
reverse_sentence	0.02
add_random_text	0.02

reverse_word	0.01
--------------	------

Модель protectai/deberta-v3-base-prompt-injection

В ходе анализа эффективности модели protectai/deberta-v3-base-prompt-injection для обfuscированных промптов было выявлено, что методы обфускации предложений демонстрируют наивысший показатель эффективности среди остальных методов, однако при этом он не превышает 0.00, что говорит об отсутствии влияния на работу модели. Методы «перемешивание слов» и «обратный порядок слов» показали эффективность — 0,00 и -0,01 соответственно (Таблица 6).

Методы обфускации слов, такие как «транслитерация» и «добавление эмодзи», также продемонстрировали нулевой показатель влияния на ответы модели (Таблица 6). Остальные же методы показатели отрицательные показатели эффективности, многие из которых приближены к -1.

Примечательно, что доля успеха для простых инструкций без изменений составила 0,99 (Таблица 5). Это означает, что модель демонстрирует высокую вероятность ошибочно классифицировать потенциально опасные промпты как безопасные или не способна адекватно распознавать попытки использования больших языковых моделей (LLM) в злонамеренных целях в виде промпт-инъекций.

В случае плохой эффективности детектирования промпт-инъекций в зловредных инструкциях стоит учитывать, что модель может быть обучена в целях обнаружения модифицированных промптов, которые могут быть использованы для обхода защитных механизмов и выравнивания моделей. В связи с этим важнее принять во внимание сравнение эффективности техник между собой, исключая сравнения с прямыми инструкциями.

Таблица 5 – Результаты тестирования protectai/deberta-v3-base-prompt-injection различными методами обфускации

Метод	Содержит промпт-инъекцию	Безопасные	Доля безопасных
transliterate	3	511	0.99
randomly_remove_characters	268	246	0.48
repeat_letters	343	171	0.33
insert_random_symbols	248	266	0.52
shuffle_characters	512	2	0.00
reverse_word	510	4	0.01
replace_with_similar_chars	510	4	0.01
insert_random_spaces	501	13	0.03
insert_invisible_characters	514	0	0.00
add_emojis	5	509	0.99
insert_foreign_characters	506	8	0.02

add_diacritics	489	24	0.05
use_mirrored_characters	45	468	0.91
shuffle_words	3	517	0.99
reverse_sentence	8	512	0.98
add_random_text	42	478	0.92
simple_instruction	3	517	0.99

Согласно формуле (2) рассчитаны показатели эффективности каждого метода обfuscации, которые ранжированы по убыванию.

Таблица 6 – Эффективность различных методов обfuscации при тестировании protectai/deberta-v3-base-prompt-injection

Метод	Показатель эффективности
transliterate	0.00
add_emojis	0.00
shuffle_words	0.00
simple_instruction	0.00
reverse_sentence	-0.01
add_random_text	-0.07
use_mirrored_characters	-0.08
insert_random_symbols	-0.47
randomly_remove_characters	-0.51
repeat_letters	-0.66
add_diacritics	-0.94
insert_random_spaces	-0.96
insert_foreign_characters	-0.97
reverse_word	-0.98
replace_with_similar_chars	-0.98
shuffle_characters	-0.99
insert_invisible_characters	-0.99

Заключение

В ходе проведённого исследования изучено влияние различных методов обfuscации на эффективность моделей-классификаторов, предназначенных для выявления промпт-инъекций. Анализ показал, что структурные изменения текста, особенно на уровне предложений, могут значительно снижать способность моделей распознавать опасные запросы. Это подчёркивает актуальность дальнейшего развития механизмов защиты языковых моделей от атак, основанных на искажении входных данных.

Исследование проводилось с использованием трёх современных моделей: protectai/deberta-v3, XiweiZ/sts-ft-promptInjection и fmops/distilbert-prompt-injection. Для каждого метода обfuscации рассчитывалась доля SAFE-классификаций — то есть случаев, когда модель ошибочно определяла вредоносный запрос как безопасный.

Эффективность метода оценивалась как разница между долей SAFE для обфусцированного запроса и базового значения, полученного для неизмененных инструкций. Такой подход позволил объективно сравнить результаты и выявить наиболее уязвимые места существующих классификаторов.

Установлено, что методы, изменяющие структуру предложения — такие как перемешивание слов или обратный порядок слов — оказались наиболее успешными в обходе детекторов промпт-инъекций. В то же время хаотичные искажения на уровне слов часто вызывали повышенное внимание со стороны моделей и не приводили к значительному увеличению числа ложных срабатываний.

Полученные данные предоставляют основу для совершенствования систем защиты, устойчивых к сложным формам обфускации. Также подчеркнута необходимость более строгого тестирования и верификации моделей, используемых для фильтрации потенциально опасного контента.

Результаты работы демонстрируют важность поиска баланса между сложностью метода обфускации и его способностью маскировать атаку. Они также открывают перспективы для дальнейших исследований, направленных на повышение устойчивости больших языковых моделей к манипуляциям и создание более надежных механизмов их защиты.

Перспективы дальнейших исследований

Перспективы дальнейшего исследования включают расширение экспериментальной базы за счёт анализа успешности джейлбрейка больших языковых моделей (LLM) при применении различных методов обфускации. Это позволит не только углубить понимание уязвимостей существующих защитных механизмов, но и определить ключевые параметры, влияющие на эффективность атак.

Одной из ключевых задач станет поиск баланса между двумя противоположными эффектами: снижением доли запросов, блокируемых моделями-классификаторами, и одновременным увеличением доли запросов, приводящих к успешному джейлбрейку LLM. Для этого предполагается провести серию экспериментов, где будут систематически варьироваться параметры обфускации — такие как сложность преобразований, их количество и тип (лексические, семантические или синтаксические изменения). На основе полученных данных планируется выявить наиболее эффективные методы обфускации, которые минимизируют вероятность обнаружения атаки защитными механизмами, но максимизируют вероятность компрометации целевой модели.

Библиография

1. Liu Y. et al. Formalizing and benchmarking prompt injection attacks and defenses // 33rd USENIX Security Symposium (USENIX Security 24). - 2024. - С. 1831-1847.
2. Greshake K. et al. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection // Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security. - 2023. - С. 79-90.
3. Shi J. et al. Optimization-based prompt injection attack to LLM-as-a-judge // Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. - 2024. - С. 660-674.
4. Sang X., Gu M., Chi H. Evaluating prompt injection safety in large language models using the promptbench dataset. - 2024.
5. Xu Z. et al. LLM Jailbreak Attack versus Defense Techniques--A Comprehensive Study // arXiv e-prints. - 2024. - С. arXiv: 2402.13457.

6. Hu K. et al. Efficient l1m jailbreak via adaptive dense-to-sparse constrained optimization // Advances in Neural Information Processing Systems. - 2024. - T. 37. - C. 23224-23245.
7. Wei A., Haghtalab N., Steinhardt J. Jailbroken: How does l1m safety training fail? // Advances in Neural Information Processing Systems. - 2023. - T. 36. - C. 80079-80110.
8. Li J. et al. Getting more juice out of the sft data: Reward learning from human demonstration improves sft for l1m alignment // Advances in Neural Information Processing Systems. - 2024. - T. 37. - C. 124292-124318.
9. Kwon H., Pak W. Text-based prompt injection attack using mathematical functions in modern large language models // Electronics. - 2024. - T. 13. - №. 24. - C. 5008.
10. Steindl S. et al. Linguistic obfuscation attacks and large language model uncertainty // Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024). - 2024. - C. 35-40.
11. Kim M. et al. Protection of LLM Environment Using Prompt Security // 2024 15th International Conference on Information and Communication Technology Convergence (ICTC). - IEEE, 2024. - C. 1715-1719.
12. Wei Z., Liu Y., Erichson N. B. Emoji Attack: A Method for Misleading Judge LLMs in Safety Risk Detection // arXiv preprint arXiv:2411.01077. - 2024.
13. Rahman M. A. et al. Applying Pre-trained Multilingual BERT in Embeddings for Improved Malicious Prompt Injection Attacks Detection // 2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings). - IEEE, 2024. - C. 1-7.
14. Chen Q., Yamaguchi S., Yamamoto Y. LLM Abuse Prevention Tool Using GCG Jailbreak Attack Detection and DistilBERT-Based Ethics Judgment // Information. - 2025. - T. 16. - №. 3. - C. 204.
15. Aftan S., Shah H. A survey on bert and its applications // 2023 20th Learning and Technology Conference (L&T). - IEEE, 2023. - C. 161-166.
16. Chan C. F., Yip D. W., Esmradi A. Detection and defense against prominent attacks on preconditioned l1m-integrated virtual assistants // 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). - IEEE, 2023. - C. 1-5.
17. Biarese D. AdvBench: a framework to evaluate adversarial attacks against fraud detection systems. - 2022.
18. Liu W. et al. DrBioRight 2.0: an LLM-powered bioinformatics chatbot for large-scale cancer functional proteomics analysis // Nature communications. - 2025. - T. 16. - №. 1. - C. 2256. DOI: 10.1038/s41467-025-57430-4 EDN: JUMWJQ.
19. Pannerselvam K. et al. Setfit: A robust approach for offensive content detection in tamil-english code-mixed conversations using sentence transfer fine-tuning // Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages. - 2024. - C. 35-42.
20. Akpatsa S. K. et al. Online News Sentiment Classification Using DistilBERT // Journal of Quantum Computing. - 2022. - T. 4. - №. 1.
21. Грицай Г. М., Хабутдинов И. А., Грабовой А. В. Stackmore LLMs: эффективное обнаружение машинно-сгенерированных текстов с помощью аппроксимации значений перплексии // Доклады Российской академии наук. Математика, информатика, процессы управления. - 2024. - Т. 520. - №. 2. - С. 228-237. DOI: 10.31857/S2686954324700590 EDN: ASZIOX.
22. Pape D. et al. Prompt obfuscation for large language models // arXiv preprint arXiv:2409.11026. - 2024.
23. Евлевская Н. В., Казанцев А. А. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ СЛОЖНЫХ СИСТЕМ С ИНТЕГРАЦИЕЙ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ: АНАЛИЗ УГРОЗ И МЕТОДОВ ЗАЩИТЫ // Экономика и качество систем связи. - 2024. - №. 4 (34). - С. 129-144. EDN: CJEEAAZ.
24. Shang S. et al. Intentobfuscator: a jailbreaking method via confusing LLM with prompts

// European Symposium on Research in Computer Security. - Cham : Springer Nature Switzerland, 2024. - C. 146-165.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Анализ влияния обfuscации промпта на эффективность языковых моделей в обнаружении промпт-инъекций» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена вопросу противодействия угроз промпт-инъекций, которые представляют собой технику манипуляции входными данными с целью изменения поведения модели.

В качестве цели авторы указывают исследовать и проанализировать различные методы обfuscации промптов, которые могут быть использованы для обхода моделей детектирования промпт-инъекций. В рамках исследования авторами рассматриваются такие техники, как замена слов синонимами, перемешивание букв в словах, внедрение специальных символов и другие подходы. Авторы самостоятельно провели оценку эффективности этих методов на примере современных языковых моделей, обученных на задачу классификации текста, и оценивают их потенциальные последствия для безопасности и защиты данных.

Список литературы представлен российскими и зарубежными источниками по теме исследования. Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Практическая значимость статьи обоснована. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части (промпт-инъекции, джейлбрейки и методы обfuscации, процедура тестирования, цель эксперимента, ожидаемые результаты, testируемые модели, результаты исследования и др.).

Авторами в работе для тестирования используется датасет AdvBench, содержащий 500 опасных инструкций, нарушающих общепринятые моральные и правовые нормы. Результаты исследования изложены в графическом виде (в виде таблиц)

В ходе проведённого исследования авторами выявлено, что использование более сложных методов обfuscации, характеризующихся множеством преобразований в одном слове или предложении и влиянием на семантическую целостность исходного текста, приводит к увеличению доли запросов, классифицируемых как инъекции. Авторы подчёркивают необходимость тщательного подхода к тестированию безопасности больших языковых моделей, особенно тех, которые оснащены дополнительными защитными механизмами, такими как модели-классификаторы для обнаружения промпт-инъекций, а также важность нахождения оптимального баланса между сложностью метода обfuscации и его эффективностью в контексте атак на языковые модели.

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна. Отсутствует четкое выделение предмета, объекта исследования.

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет, объект. Также будет целесообразным добавить о перспективах дальнейшего исследования.

Статья «Анализ влияния обfuscации промпта на эффективность языковых моделей в обнаружении промпт-инъекций» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией

рецензируемого научного журнала.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Рецензируемая статья посвящена проблеме противодействия угрозам атак, направленных на большие языковые модели и нарушающих заложенные в них разработчиками алгоритмы. В статье сформулированы цели, объект и предмет исследования, приводятся виды атак и их характерные признаки. Новизну работы оценить затруднительно, авторы используют известные методы, тестируют представленные в открытом доступе модели, количественная оценка результатов сводится к оценке доли безопасных случаев. Исходные данные представлены в открытом доступе и предназначены для воздействия перечисленными авторами методами, что затрудняет оценку собственного вклада. Количественная оценка результатов по единственному критерию неубедительна.

Стиль изложения характерен для разговорного изложения материалов, однако использованные термины затрудняют её восприятие. Использованные формулировки, перегруженность статьи сленговыми терминами, затрудняют оценку вклада авторов и анализ результатов собственных исследований (тестирования) с помощью средств, представленных в открытом доступе. Имеются иллюстрация, таблицы.

Библиография содержит 21 источник, преимущественно зарубежные публикации в сборниках конференций, только одна позиция в отечественном журнале. Ссылки по тексту сгруппированы.

Замечания.

Необходимо изменить название статьи и ключевые слова, приведя их в соответствии с принятой в Журнале терминологией и стилем научной публикации.

Отсутствует обзор аналогичных исследований. Данный раздел рецензируемой научной статьи заменен на информационный блок, содержащий перечисление подвидов атак а также печень наименований методов с их указанием на двух языках.

Необходимо избегать применения транслитерации англоязычных терминов (особенно в названии) или большого количества терминов на иностранном языке, предпочтительны понятные большинству читателей термины в области вычислительных методов.

Описание эксперимента приведено формально, отсутствует обоснование выбора исходного набора данных и тестируемых моделей. Все анализируемые модели представлены в открытом доступе, разработаны для решения задач, аналогичных поставленным Авторам. Каков вклад Авторов? Способ идентификации моделей в тексте статьи, в совокупности с формой представления в таблицах методов не позволяет оценить личный вклад авторов в полученный результат, его новизну и оригинальность.

В разделе «результаты тестирования» авторы упоминаю, что доля успеха 0,25 является высокой, доля успеха 0 свидетельствует о низкой эффективности, а доля 0,08 является более высокой. Не ясно какое пороговое значение позволяет говорить о низкой или высокой эффективности? В анализе не раскрывается с чем может быть связано то или иное рассчитанное значение эффективности, какой фактор или использованный вычислительный аппарат является определяющим для результата?

Все таблицы необходимо привести в соответствие с приведенным перечнем методов, использовать единый язык, указать единицы измерений. В тексте дать пояснение как рассчитывались количественные оценки, каким образом выявлены безопасные, относительно какой величины оценивалась доля безопасных. Не ясно на основании

каких критерий проводится упомянутая оценка производительности.

В библиографии необходимо проверить правильность написания выходных данных.

Статья будет интересна читателям, чьи научные интересы лежат в области оценки методов атак.

Статья может быть опубликована после корректировки формулировок и внесения правок.

Результаты процедуры окончательного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Журнал: Программные системы и вычислительные методы

Тема: Анализ влияния обfuscации входных данных на эффективность языковых моделей в обнаружении инъекций подсказок

Актуальность темы подтверждена достаточным количеством ссылок на научные публикации из 24 источников.

В последнее время с развитием Интернета информационная безопасность имеет очень важное значение.

Для противодействия угрозам разрабатываются специальные модели, предназначенные для детектирования инъекций.

Однако злоумышленники также разрабатывают новые методы обfuscации, чтобы обходить такие защитные механизмы. Обfuscация означает изменение структуры и содержания текста с целью затруднения его анализа и защиты.

Объектами исследования являются языковые модели, используемые в различных веб-приложениях обработка естественного языка.

Предметом исследования являются методы обfuscации промптов и их последствия на информационную безопасность.

Статья написана грамотным техническим языком, понятным специалистам в данной предметной области.

Целью работы является анализ известных методов обfuscации и их влияние на обход моделей детектирования промпта инъекций.

В качестве методов исследования выбраны различные подходы, такие как перефразирование, перемешивание букв в словах, внедрение специальных непечатных символов и др.

Статья структурирована, содержит краткий обзор литературы, описание использованных моделей и методов.

Безусловным преимуществом исследования является проведение вычислительного эксперимента.

Для тестирования выбраны три известные модели, так как они специально обучены распознавать опасные запросы.

Описана его методология проведения, получены важные сравнительные результаты для разных моделей обfuscации.

Эксперимент направлен на выявление различий в эффективности различных моделей при обработке обфусцированных и необфусцированных инструкций.

Это поможет выявить слабые места в информационной безопасности уменьшить риск внешнего проникновения.

Заключение написано корректно, указаны перспективы дальнейших исследований.

Даны практические рекомендации по применению результатов выполненного

вычислительного эксперимента.

Перспективы дальнейшего исследования включают расширение экспериментальной базы за счет большего количества языковых моделей.

На основе новых полученных данных планируется выявить наиболее эффективные методы обfuscации, которые минимизируют вероятность внешней информационной угрозы при помощи новых защитных механизмов.

Критических замечаний не обнаружено.

Рекомендации:

1. Включить в тему ключевое слово "информационная угроза" в зависимости от специальности, по которой планируется защита диссертации.
2. В статье используется ряд жаргонов, которые ухудшают читабельность текста, например "джейлбрейки", "эксплойты".
3. Аббревиатуры LLM, BERT не расшифрованы, они не являются общепринятыми.
4. Рисунок 1 низкого качества с точки зрения размера текста.
5. Опечатка в разделе "Ожидаемые результаты".

Заключение:

Статья рекомендуется для публикации в журнале "Программные системы и вычислительные методы" и будет интересна широкой читательской аудитории.

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Ратушняк Е.А. Исследование производительности современных клиентских веб-фреймворков // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.74392 EDN: OYAYXV URL: https://nbpublish.com/library_read_article.php?id=74392

Исследование производительности современных клиентских веб-фреймворков

Ратушняк Евгений Алексеевич

ORCID: 0009-0006-3609-4194

студент; факультет программной инженерии и компьютерной техники; Национальный исследовательский университет ИТМО

197101, Россия, г. Санкт-Петербург, Петроградский р-н, Кронверкский пр-кт, д. 49

✉ evgrat123@mail.ru

[Статья из рубрики "Математическое и программное обеспечение новых информационных технологий"](#)**DOI:**

10.7256/2454-0714.2025.2.74392

EDN:

OYAYXV

Дата направления статьи в редакцию:

10-05-2025

Дата публикации:

26-05-2025

Аннотация: Предметом исследования является производительность рендеринга трёх современных фреймворков — React, Angular и Svelte — в типовых сценариях построения и обновления пользовательского интерфейса в веб приложениях. Объектом исследования являются сами фреймворки как комплексы технологических решений, включающие механизмы обнаружения изменений, виртуальные или компилируемые DOM структуры и сопутствующие оптимизации. Автор подробно рассматривает такие аспекты темы, как первичный и повторный рендеринг, операции обновления и удаления элементов, работа в линейных и глубоко вложенных структурах данных. Особое внимание уделяется практической значимости выбора фреймворка для коммерческих продуктов, где разница в производительности непосредственно влияет на конверсию, опыт пользователя и финансовую эффективность проекта. Описываются ключевые

внутренние механизмы — виртуальный DOM React, детектор Angular и компилируемый код Svelte, — которые определяют их поведение в разных нагрузочных сценариях. Методология основана на автоматизированном бенчмарке: унифицированный набор тестовых сценариев выполняется клиентскими приложениями на React, Angular и Svelte, эталонном JavaScript решении и сервере оркестраторе Express JS; время операций фиксируется через `performance.now()` в Chrome 126, критерий производительности — время до первой перерисовки. Новизна исследования заключается в комплексном лабораторном сопоставлении трёх фреймворков по четырём критически важным сценариям (первичный рендеринг, повторный рендеринг, обновление и удаление элементов) с учётом двух типов структур данных и привязкой к актуальным версиям 2025 года. Основными выводами проведённого исследования являются следующие: Svelte обеспечивает наибольшую производительность и лидирует при глубокой иерархии благодаря компиляции DOM операций; React показывает лучшие результаты при повторном обновлении длинных списков, используя оптимизированный алгоритм обнаружения изменений и ключи элементов; Angular гарантирует предсказуемость и архитектурную целостность, но увеличивает время перерисовки примерно на 60 % из-за детектора изменений. Универсального лидера не существует; рациональный выбор должен опираться на аналитический профиль операций конкретного приложения, что и подтверждают результаты представленного эксперимента.

Ключевые слова:

рендеринг, JavaScript фреймворки, React, Angular, Svelte, Core Web Vitals, виртуальный DOM, инкрементный DOM, производительность, веб-интерфейсы

Введение

Бурное развитие веб-технологий преобразует ожидания пользователей: контент должен появляться практически мгновенно, а интерфейс — реагировать без ощутимых задержек. По оценкам Google, увеличение *Largest Contentful Paint* (LCP) всего на 100 мс приводит к снижению коэффициента конверсии интернет-магазина на 1,3%. Следовательно, вопрос выбора фреймворка выходит за рамки вкусовых предпочтений и становится фактором финансовой эффективности продукта. Статистика показывает, что React, Angular и Svelte формируют самый «живой» круг разработчиков и наиболее высокую удовлетворённость [2]. В научном дискурсе приоритет отдается фреймворкам с активной экосистемой — React, Angular и, с недавнего времени, Svelte. Также эти фреймворки имеют разный подходы к обнаружению изменений: React — Virtual DOM [3], Angular — Incremental DOM [4], Svelte — переносит вычисления из времени выполнения в этап компиляции [5]. Использование фреймворков и библиотек значительно ускоряет разработку приложений и уменьшает их стоимость на 30% [13], а также производительность веб-фреймворков остается одним из важных параметров [14]. Все выше перечисленное приводит к необходимости сравнительного анализа фреймворков, а также обосновывает актуальность работы.

Целью статьи является эмпирическое сравнение производительности рендеринга React, Angular и Svelte в типовых сценариях построения и обновления пользовательского интерфейса. Для достижения цели решаются задача разработки унифицированного набора тестовых сценариев, автоматизация сбора и анализ экспериментальных данных для выявления недостатков и преимуществ фреймворков. Предметом исследования

является производительность рендеринга трёх фреймворков — React, Angular и Svelte — в типовых сценариях построения и обновления пользовательского интерфейса в веб-приложениях. Научная новизна исследования заключается в разработке и применении автоматизированной методики сравнительного анализа производительности фреймворков React, Angular и Svelte в различных эксплуатационных сценариях веб-приложений, что позволяет выявить их преимущества и недостатки.

Разработка тестового ПО

В рамках экспериментального исследования создаётся клиентское одностраничное [8] веб-приложение, визуализирующее иерархическую выборку данных в табличной форме. Пример реализации строки таблицы представлен на рисунке 1. Каждая логическая запись отображается единообразным строковым шаблоном: контейнер строки таблицы разделяется на четыре ячейки, последовательно выводящие идентификатор элемента, его заголовок и текстовое описание, а также набор кнопок с управляющими действиями. Кнопки добавить и удалить являются декоративными, а для обновлений данных таблицы используется программные методы, а именно — обновление внутреннего представления данных, для тестирования производительности фреймворков. Спроектированный ряд, будучи атомарным визуальным модулем, упорядочивает потоки событий, изолирует рендеринг и служит универсальной единицей измерения производительности: фиксированные стили и одинаковый состав DOM-узлов обеспечивают сопоставимость метрик времени построения, обновления и удаления для различных фреймворков. Также эксперимент имеет две структуры данных: бинарная — эмулирует реальные сайты с высокой вложенностью, линейная — эмулирует большие таблицы. Такая организация эксперимента даёт возможность строго регистрировать латентность, объём операций с DOM при масштабировании количества записей, что, в свою очередь, позволяет объективно оценить эффективность применяемых во фреймворках оптимизационных стратегий.

```

3 Go to component | You, 3 weeks ago | 1 author (You)
1 <div class="table-row">
2   <div>{{ data?.id }}</div>
3   <div>{{ data?.title }}</div>
4   <div>{{ data?.text }}</div>
5   <div class="action-buttons">
6     <button>Add</button>
7     <button class="remove-btn">Remove</button>
8   </div>
9 </div>
```

Рисунок 1 – реализация строки тестовой таблицы

В браузерной модели событийный цикл является единственным надёжным способом зафиксировать момент, когда первый визуальный кадр уже отрисован, является последовательное использование `requestAnimationFrame` и затем задачи [таймера с нулевой задержкой]. Следовательно, метка, поставленная внутри такого вызова, фиксирует время до появления кадра, обходя весь критический путь рендеринга [11]. Если внутри обработчика запроса кадра анимации поместить вызов таймаута с задержкой 0, создаётся отдельная задача. Она будет обработана в следующей итерации, то есть уже после того, как браузер осуществил отрисовку завершившегося кадра. Тем самым интервал между метками, снятыми непосредственно после изменения состояния и в начале таймера [задачи], охватывает полный путь сигнала — работу фреймворка, генерацию патч [операций], вычисление стилей, вычисление разметки, отрисовка — и

потому служит корректным измерением времени «данные → первый кадр» для любых современных JavaScript-фреймворков. Это и есть критерий наибольшей перерисовки LCP [7].

Экспериментальная установка состоит из многокомпонентной архитектуры (представленной на рисунке 2), включающей клиентский слой на каждом из трёх фреймворков и эталонное приложение на «чистом» JavaScript, сервер на базе Express JS и модуль автоматизированного сбора статистики. Каждое клиентское приложение запрашивает входные данные по пути / testcase/next, выполняет требуемую операцию и передаёт измеренное время выполнения обратно на сервер через / testcase/benchmark. Такой цикл повторяется для всех сценариев без участия человека, что обеспечивает полную воспроизводимость и исключает влияния сторонних задержек, связанных с ручным управлением браузером [6]. На сервере ведутся счётчики прогресса, позволяющие последовательно пройти весь объём тестов; по завершении генерируются файлы CSV и адаптивный HTML-отчёт, где результаты собираются в сводные таблицы с автоматически рассчитанными средними значениями и стандартными отклонениями. Автоматический «водитель» браузера, реализованный на Puppeteer, запускает браузер в интерактивном режиме, дожидается момента, когда выставится флаг завершения эксперимента, и при необходимости повторяет прогон при возникновении исключений, тем самым гарантируя стабильность даже при случайных сбоях сети или браузера.

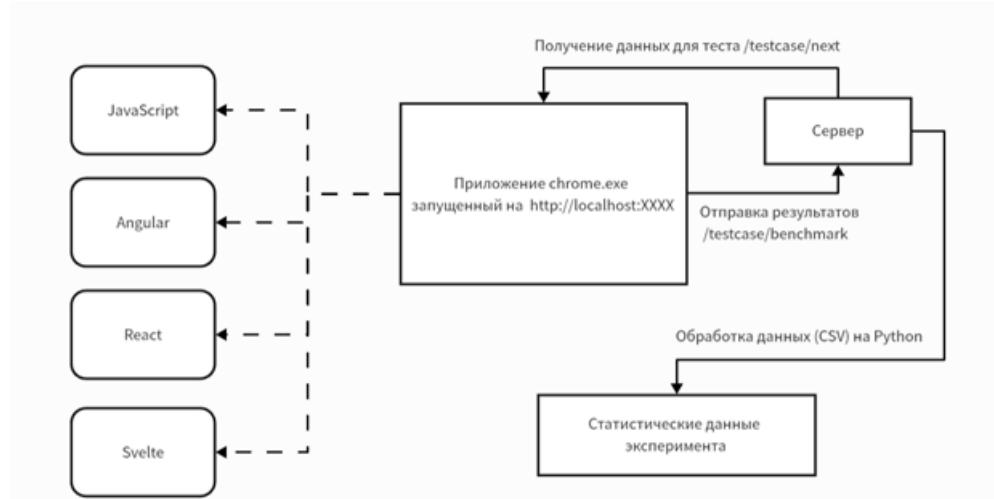


Рисунок 2 – архитектура тестового окружения

Исходный код экспериментального ПОложен на GitHub и доступен по ссылке <https://github.com/TimeToStop/benchmark>.

Реализация тестового приложения с использованием фреймворков

В цикле HTML разметки используется функция определяющая уникальный идентификатор элемента; оно заставляет движок обнаружения изменений сравнивать элементы по возвращаемому ключу, поэтому при изменении порядка строк Angular переиспользует существующие компонентные экземпляры и обновляет лишь изменённые ячейки, а не всю коллекцию. Такая комбинация сокращает как количество пересчётов шаблонов, так и частоту «дорогих» операций перерасчёта компоновки. Код представлен на рисунке 3.

Для измерения производительности начальная метка ставится в момент программного изменения входного свойства компонента с немедленным вызовом обнаружения изменений, который принудительно проводит обнаружение только в текущем поддереве.

```

<div *ngIf="data">
  <app-row [data]="data"></app-row>
  <div>
    <app-child
      *ngFor="let child of data.children; trackBy: id"
      [data]="child"
    >
      </app-child>
    </div>
  </div>

```

Рисунок 3 – реализация узлового компонента на Angular

Компоненты, представленные на рисунке 4 заданы как чистые функции, поэтому при каждом обновлении React сравнивает только результат их вызова, а не состояние экземпляров классов. Главная оптимизация проявляется в атрибуте ключа при обходе потомков: ключи дают алгоритму согласования устойчивую идентичность элементов, что позволяет ограничить перерасчёт теми узлами, для которых действительно изменились данные. Таким образом, при добавлении или удалении строк React выполняет точечные операции вместо полного пересоздания поддерева. Моментом фиксации начала отрисовки является вызов `setState`, который помещает обновление в очередь и запускает фазу согласования виртуального дерева.

```

function Child({ data }: { data: INode }) {
  return (
    <div>
      <Row data={data}></Row>
      <div>
        {data.children.map(child => <Child key={child.id} data={child}></Child>)}
      </div>
    </div>
  );
}

```

Рисунок 4 – реализация узлового компонента на React

Компонент объявляется один раз, но разворачивается компилятором в императивный цикл, который на этапе сборки генерирует минимальный набор вызовов вставки и удаления элементов — виртуальный DOM вовсе не участвует [\[19\]](#). Поэтому изменение массива потомков приводит только к тем манипуляциям с DOM, которые действительно нужны, без промежуточного вычисления изменений [\[20\]](#). Рекурсивное использование компонента сохраняет единый шаблон и позволяет компилятору повторно применять сгенерированный код на каждом уровне дерева, что минимизирует размер кода и накладные расходы во время исполнения. Начало измерения скорости отрисовки является обращение к `tick()`: эта функция добавить вызов функции, который выполняется, когда компилятор уже обновил реальные DOM-узлы. Код представлен на рисунке 5.

```

<div>
  <Row {data}></Row>
  <div>
    {#each data?.children as child}
    <svelte:component this={self} data={child} />
    {/each}
  </div>
</div>

```

Рисунок 5 – реализация узлового компонента на Svelte

Анализ экспериментальных данных

Аппаратная платформа включает процессор Intel Core i7-8750H, 16 гигабайт оперативной памяти и 64-битную операционную систему; браузер - Google Chrome версии 126.0.6478.116. Фреймворки используются в актуальных стабильных релизах: Angular 16.2.12, React 18.3.1, Svelte 4.2.18. Такой подбор гарантирует актуальность выводов для практики 2025 года и снимает вопросы совместимости с современными браузерными API. Все измерения выполняются функцией `performance.now()`, чья точность порядка пяти сотых миллисекунды считается достаточной для фиксации производительности.

Первичный рендеринг представляет собой формирование внутреннего представления компонентов, построение DOM и выполнение первой верстки. В линейной структуре тестовая таблица из N тысяч строк создаёт равномерную нагрузку по всей ширине окна; в бинарной структуре, напротив, рекурсивное ветвление формирует глубокое дерево с большим числом перекрёстных зависимостей стилей. В этих условиях эталонное приложение на чистом JavaScript закономерно фиксирует минимальное время из-за отсутствия абстракций, однако React и Svelte оказываются близки к нему в бинарном случае: разница не превышает нескольких процентов благодаря тому, что виртуальный DOM React и компилируемые вызовы Svelte генерируют относительно компактные последовательности операций, что согласуется с выводами Makitalo et al. [10]. Angular демонстрирует увеличение LCP примерно на шестьдесят процентов вне зависимости от структуры, что согласуется с его более тяжёлым запуском времени выполнения и стратегией отслеживания изменений [9, 12]. При переходе к линейной таблице Svelte показывает просадку на двадцать процентов по сравнению с JavaScript, что объясняется увеличением объёма кода, необходимого для описания каждой строки, тогда как React сохраняет паритет благодаря оптимизированному преобразованию массивов в набор элементов. Результаты эксперимента при сценарии первичного рендеринга представлены на рисунках 6 и 7.

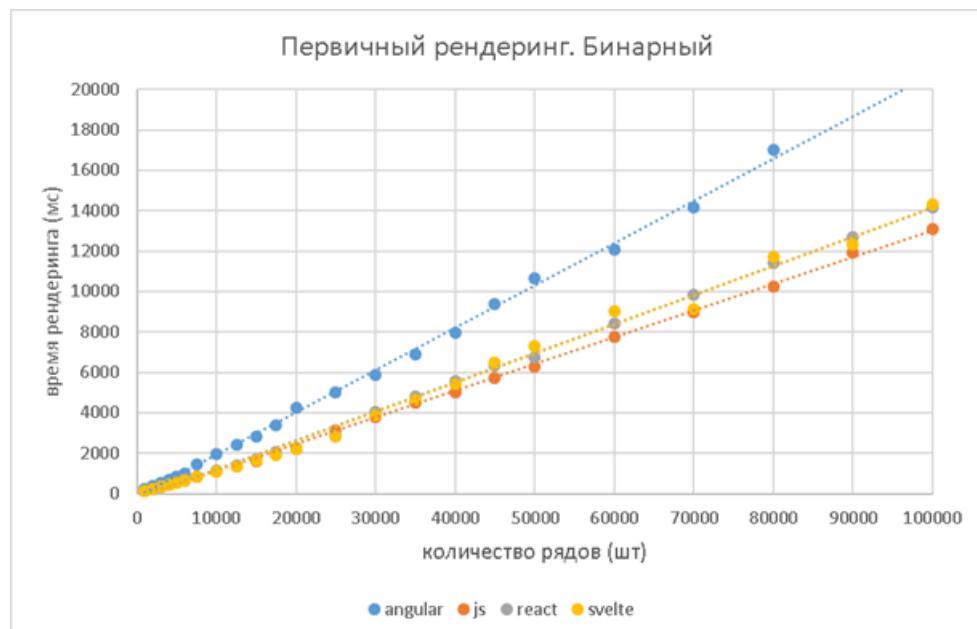


Рисунок 6 – результаты тестирования первичного рендеринга в бинарной структуре данных

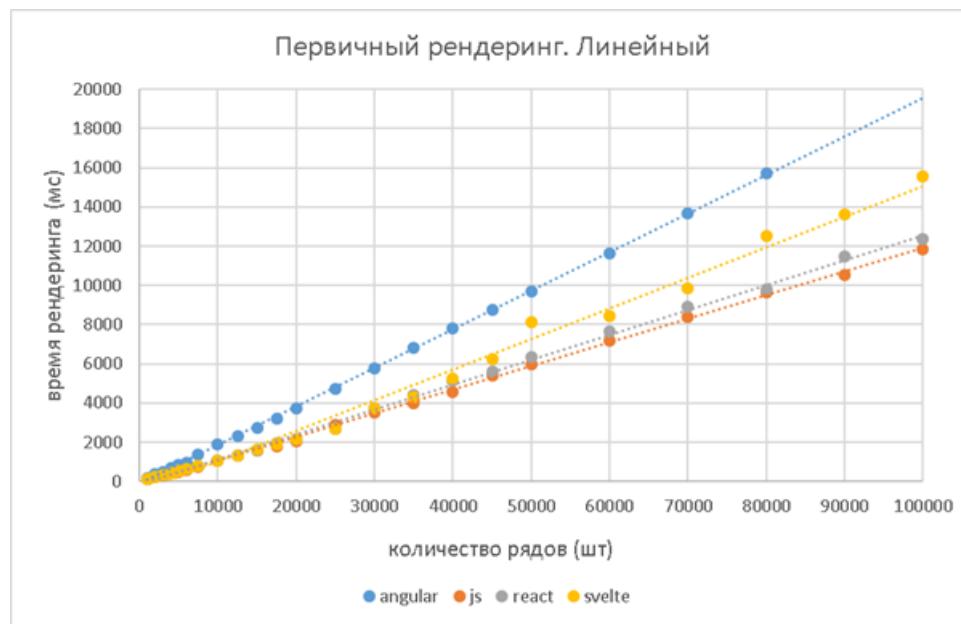


Рисунок 7 – результаты тестирования первичного рендеринга в линейной структуре данных

Повторный рендеринг, то есть полная перерисовка той же структуры данных, иллюстрирует, как фреймворки управляют уже построенным деревом. Здесь критически важны алгоритмы сравнения старого и нового состояния. В бинарной иерархии React оказывается самым медленным: вычисление изменений по глубокой структуре приводит к росту времени почти в полтора раза относительно Angular и вдвое относительно Svelte.

Последний выигрывает за счёт прямых изменений конкретных узлов, поскольку компилятор заранее знает, какой участок DOM нужно затронуть. В линейной таблице картина меняется: React, использующий ключи элементов для быстрого обнаружения перемещений, выходит в явные лидеры, опережая Angular на треть, а Svelte – на пятнадцать процентов. Такое поведение подтверждает тезис о том, что эффективность виртуального DOM растёт вместе с долей одинаковых повторяющихся поддеревьев, тогда как при сложных вложенных структурах издержки сопоставления превышают выигрыш от выборочного обновления. Результаты эксперимента при сценарии повторного рендеринга представлены на рисунках 8 и 9.

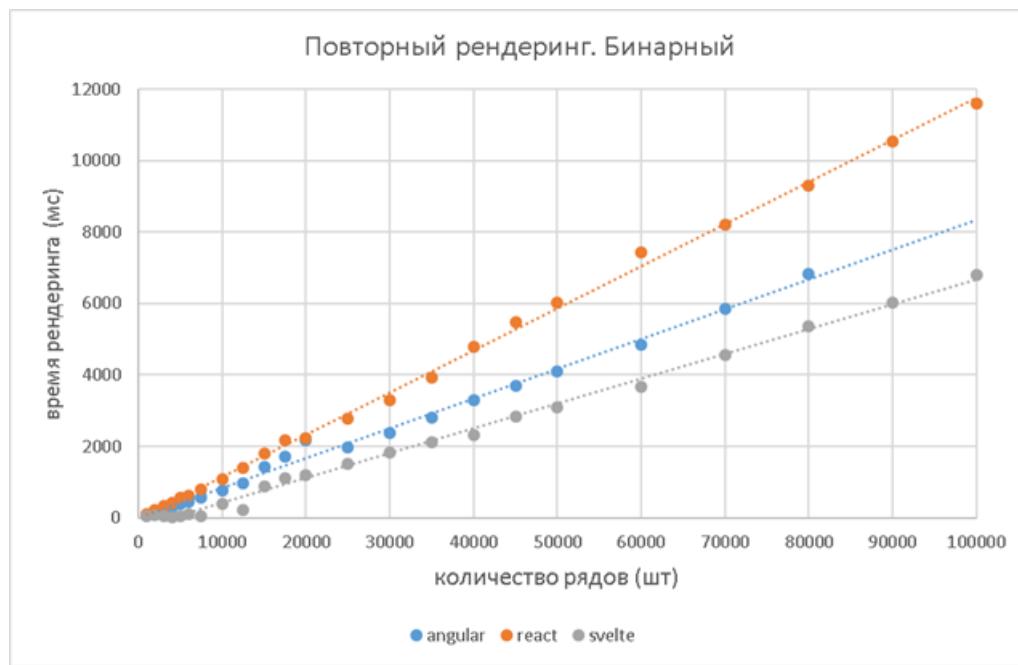


Рисунок 8 – результаты тестирования повторного рендеринга в бинарной структуре данных

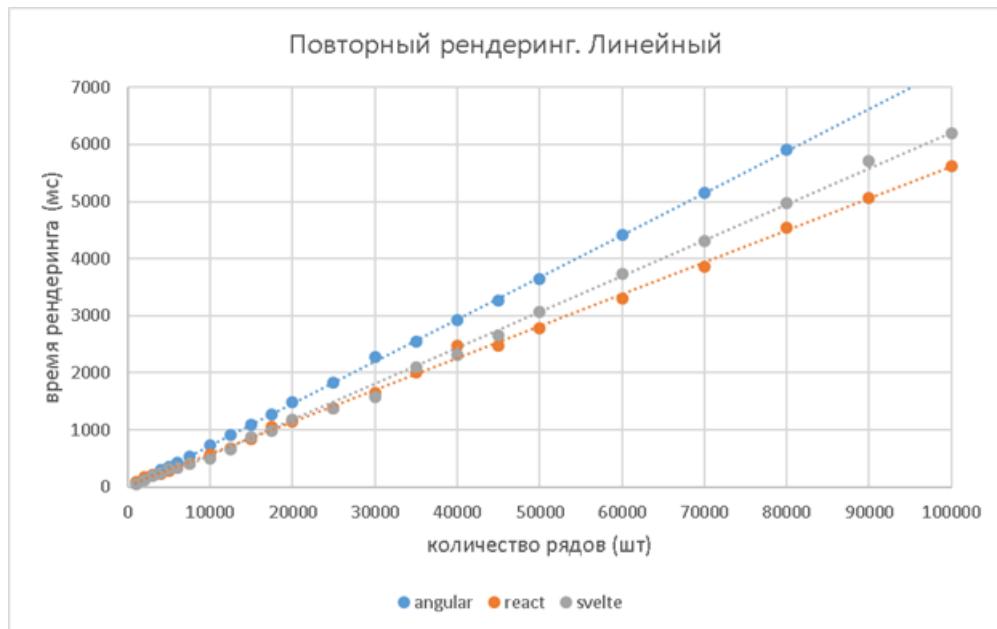


Рисунок 9 – результаты тестирования повторного рендеринга в линейной структуре данных

Операция обновления элементов моделирует небольшие изменения во время работы приложения. Производительность здесь определяется сочетанием скорости обновления узла, количества вызываемых перерисовок и тяжести синхронизации стилей. Полученные результаты демонстрируют явное преимущество Svelte: в глубоком бинарном дереве компилируемый код адресуется непосредственно к целевому узлу, изменяя лишь затронутые атрибуты и сводя задержку к минимуму. В линейной таблице React приближается к показателям Svelte благодаря линейному алгоритму обнаружения изменений и переиспользованию существующих элементов, однако всё-таки уступает в бинарном случае. Angular остаётся предсказуемо медленнее обоих конкурентов: детектор изменений проходит по всей иерархии компонентов независимо от масштаба изменения, что добавляет заметную задержку даже при модификации одной строки, что

согласуется с выводами в статье [11]. Результаты эксперимента при сценарии обновления ряда представлены на рисунках 10 и 11.

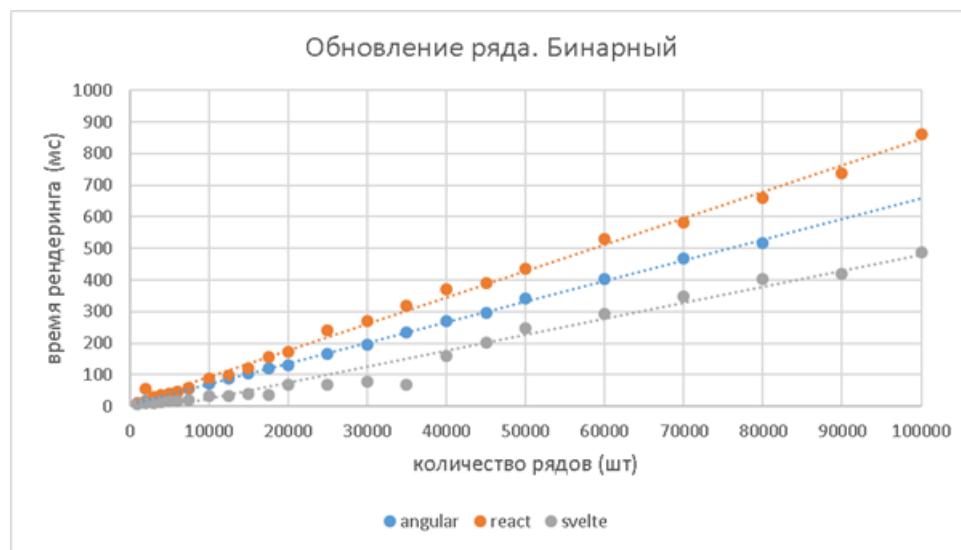


Рисунок 10 – результаты тестирования обновления ряда в бинарной структуре данных

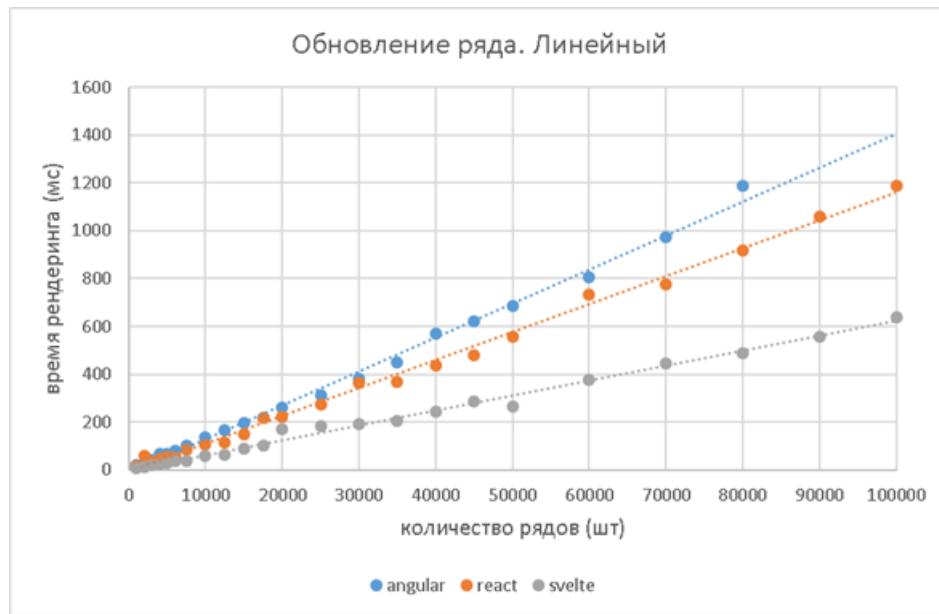


Рисунок 11 – результаты тестирования обновления ряда в линейной структуре данных

Удаление элементов раскрывает обратную сторону оптимизаций. В бинарной структуре Svelte демонстрирует минимальное время, так как точечно удаляет ветвь и немедленно высвобождает связанные обработчики событий. React и Angular, напротив, вынуждены пересчитывать влияние удаления на соседние узлы, что удлиняет операцию приблизительно на пятьдесят процентов. В линейной таблице React выходит вперёд: наличие ключей упрощает поиск удаляемого узла и сокращает обход коллекции, тогда как Svelte, не располагая встроенной виртуализацией больших списков [18], тратит время на обновление индексов, отчего отстаёт от React в среднем на сто сорок процентов. Этот результат подчёркивает необходимость сторонних библиотек виртуального прокручивания в проектах на Svelte, где наблюдается массивное удаление или добавление строк. Результаты эксперимента при сценарии удаления ряда представлены на рисунках 12 и 13.

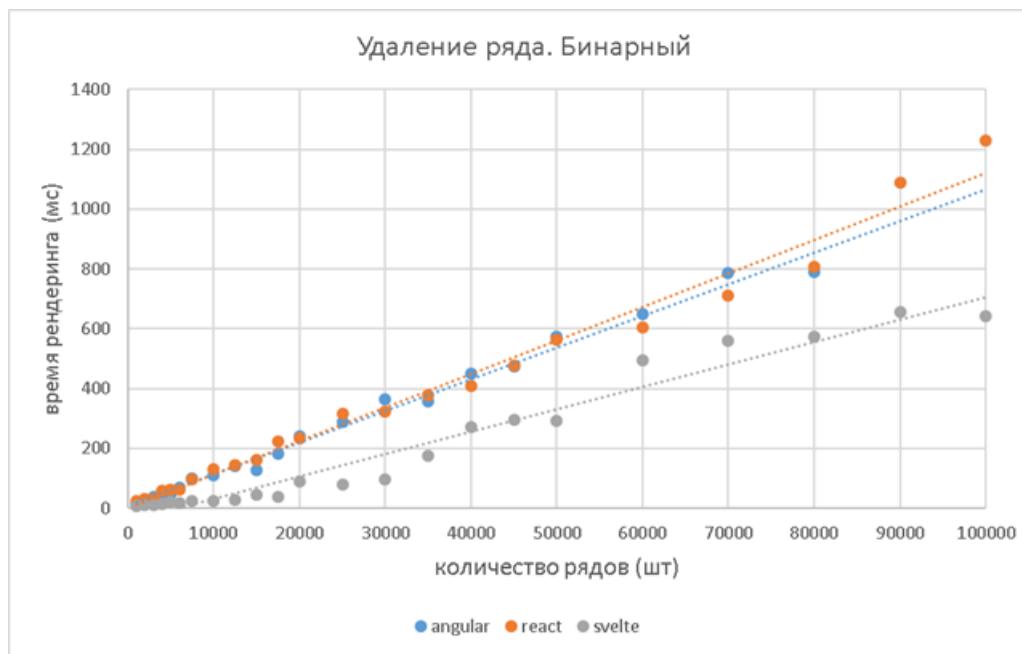


Рисунок 12 – результаты тестирования удаления ряда в бинарной структуре данных

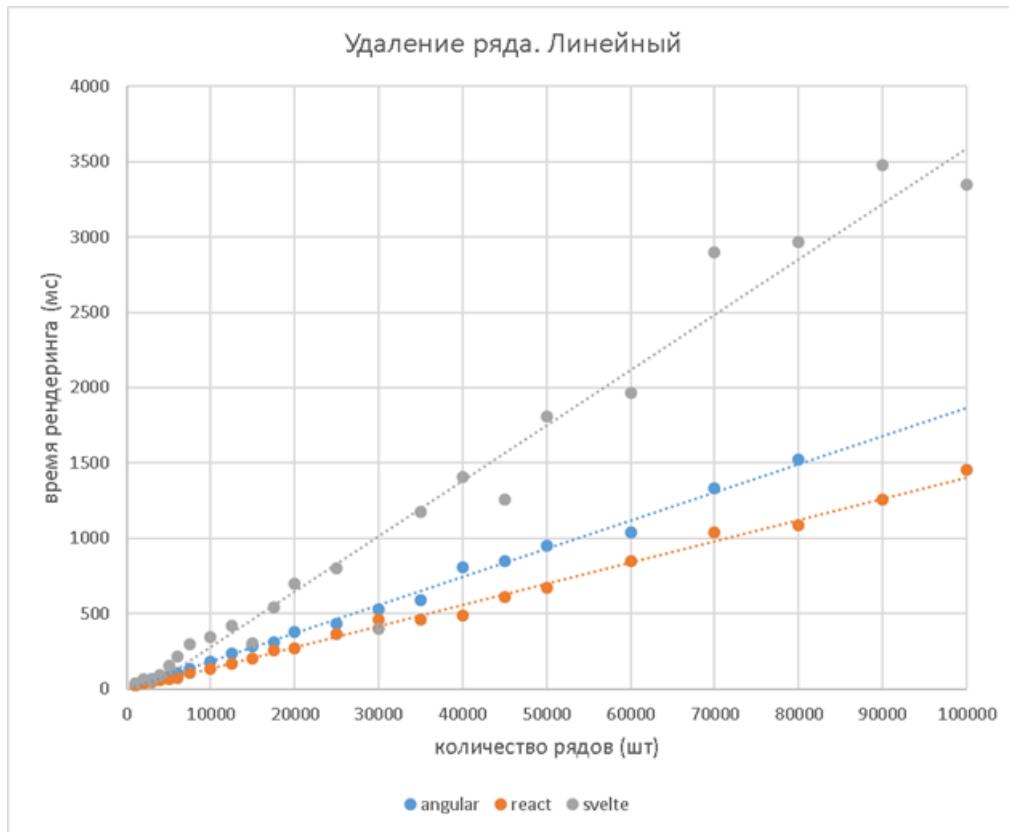


Рисунок 13 – результаты тестирования удаления ряда в линейной структуре данных

Качественная интерпретация выявленных закономерностей опирается на особенности внутренних механизмов. В React критическим фактором выступает стоимость вычисления различий в виртуальном DOM. При плоской коллекции уникальные ключи позволяют свести поиск изменений к линейному проходу; при глубокой иерархии, напротив, число сравнений растёт экспоненциально с глубиной, что делает алгоритм узким местом. Angular применяет детектор изменений, который гарантированно обходит все компоненты при любом событии, обеспечивая стабильность и предсказуемость, но взамен грузит процессор ненужными повторными вычислениями. Svelte избегает анализа во время выполнения: компилятор генерирует функции обновления, физически меняющие только

те свойства, которые могли измениться. Такой подход блестяще работает, пока объём кода остаётся умеренным; при длинных списках количество сгенерированных вызовов становится само по себе значительным, что объясняет потерю скорости при линейной таблице. Разброс данных и выводы о стабильности React и Svelte согласуются с выводами в статье [\[16\]](#).

Результаты эксперимента представлены в таблице 1.

Таблица 1 – результаты тестирования современных фреймворков

Сценарий	Структура данных	JavaScript	React	Angular	Svelte
Первичный рендеринг	Бинарная	Базовый ориентир	≈ то же, что JS	+ 60 % к JS	≈ то же, что JS
	Линейная	Базовый ориентир	≈ то же, что JS	+ 60 % к JS	+ 20 % к JS
Повторный рендеринг	Бинарная	—	+ 80 % к Svelte	+ 40 % к Svelte	Самый быстрый
	Линейная	—	Самый быстрый	+ 30 % к React	+ 15 % к React
Обновление ряда	Бинарная	—	+ 80 % к Svelte	+ 40 % к Svelte	Самый быстрый
	Линейная	—	+ 80 % к Svelte	+ 140 % к Svelte	Самый быстрый
Удаление ряда	Бинарная	—	+ 50 % к Svelte	+ 50 % к Svelte	Самый быстрый
	Линейная	—	Самый быстрый	+ 30 % к React	+ 140 % к React

Практическое применение результатов сводится к формулировке рекомендаций. Для систем, в которых доминируют длинные, однородные коллекции записей с частыми операциями вставки и удаления, целесообразно выбирать React, при этом обязательно включать мемоизацию компонентов и использовать виртуализацию списков, чтобы удерживать алгоритм определения различий в пределах линейного времени. В корпоративных решениях со сложной иерархической структурой, требующих строгой типизации и унифицированных паттернов, Angular остаётся удобным инструментом, но разработчикам следует неукоснительно использовать стратегию OnPush и функцию trackBy, чтобы снизить количество лишних проходов по дереву. Для проектов, ориентированных на мгновенную первую отрисовку и глубокие вложенные интерфейсы, оптимальным оказывается Svelte, однако эффективность сохраняется лишь при одновременном сокращении размера списков через виртуальное прокручивание и разбиение на страницы.

Заключение

Исследование демонстрирует, что ни один из популярных фреймворков не обладает универсальным превосходством. Svelte минимизирует время первой отрисовки и выигрывает в бинарных изменениях за счёт компиляции компонентов, React лидирует в повторном обновлении длинных списков благодаря оптимизации алгоритма обнаружения различий, Angular обеспечивает архитектурную целостность ценой более высокого LCP. Перспективы дальнейших исследований связаны с включением серверного рендеринга SvelteKit и React Server Components, а также анализом энергопотребления на мобильных

устройствах.

Рекомендации: ориентировать React на приложения с длинными динамическими списками, обязательно используя ключи, мемоизацию и виртуализацию; выбирать Svelte для интерфейсов с глубокой вложенностью и требованием сверхбыстрой первой отрисовки, дополняя его библиотеками виртуального прокручивания при больших наборах данных; применять Angular в корпоративных системах, где приоритетны типизация и встроенный стек, строго активируя trackBy, OnPush и дифференциальную загрузку. Универсального лидера нет; выбор технологии должен основываться на аналитическом профиле операций и топологии интерфейса. Представленная методика измерений обеспечивает воспроизводимую основу для дальнейшего сравнения новых версий библиотек. Перспективным выглядит использование технологии веб-ассемблера WebAssembly [15] для увеличения производительности, а также использования аддитивной гидратации [17] в дальнейших исследованиях.

Библиография

1. Morgan P. JavaScript DOM Manipulation Performance: Comparing Vanilla JavaScript and Leading JavaScript Front-end Frameworks. 2020.
2. Levlin M. DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue and Svelte. LUT University, 2020. C. 58.
3. Aggarwal S. Modern Web Development using ReactJS // International Journal of Recent Research Aspects. 2018. T. 5, № 1. C. 133|137.
4. Saks E. JavaScript frameworks: Angular vs React vs Vue. 2019.
5. Paakkanen J. Upcoming JavaScript Web Frameworks and Their Techniques. LUT University, 2022. 62 c.
6. Siahaan M., Kenedy R. Rendering performance comparison of React, Vue, Next and Nuxt // Jurnal Mantik. 2021. T. 5, № 3. C. 1851|1860.
7. Iseal Sheed. Performance Benchmarking Techniques for React Applications. 2025.
8. Iseal Sheed. Edge Computing and React: Enhancing Performance at the Edge. 2025.
9. Bialecki G., Pańczyk B. Performance analysis of Svelte and Angular applications // Journal of Computer Sciences Institute. 2021. T. 19. C. 139-143. DOI: 10.35784/jcsi.2633. EDN: OYEGVL.
10. Piastou M. Comprehensive performance and scalability assessment of frontend frameworks: React, Angular, and Vue.js // World Journal of Advanced Engineering Technology and Sciences. 2023. T. 9, № 2. C. 366|376. DOI: 10.30574/wjaets.2023.9.2.0153. EDN: JNSAZF.
11. Маньшина Е. В., Ермолаева А. А. Сравнительный анализ производительности серверного и клиентского рендеринга веб-приложений, созданных с помощью фреймворка Angular // Научно-технические инновации и веб-технологии. 2022. № 2. С. 44|48.
12. Karić A., Durmić N. Comparison of JavaScript Frontend Frameworks - Angular, React, and Vue // International Journal of Innovative Science and Research Technology (IJISRT). 2024. С. 1383-1390. DOI: 10.38124/ijisrt/ijisrt.
13. Kaveri P. R. Framework-Agnostic JavaScript Component Libraries: Benefits, Implementation Strategies, and Commercialization Models // 2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN). 2024. C. 1441-1446. EDN IGADIU.
14. Kasenda R., Tenda J., Iman E., Manantung J., Moekari Z., Pantas M. The Role and Evolution of Frontend Developers in the Software Development Industry // Jurnal Syntax Admiration. 2024. T. 5. C. 5191-5196. DOI: 10.46799/jsa.v5i11.1852.

15. Акиева З. М., Мурзин Д. А., Эльтаев С. И. Применение WebAssembly для повышения производительности интерактивных веб-приложений // Экономика и управление: проблемы, решения. 2024. Т. 7, № 12. С. 168-174. DOI: 10.36871/ek.up.r.p. 2024.12.07.021. EDN: WJASJX.
16. Dubaj S., Pańczyk B. Comparative of React and Svelte programming frameworks for creating SPA web applications // Journal of Computer Sciences Institute. 2022. T. 25. C. 345-349. DOI: 10.35784/jcsi.3020. EDN: AYUSVM.
17. Chen K. Improving Front-end Performance through Modular Rendering and Adaptive Hydration (MRAH) in React Applications. 2025. DOI: 10.48550/arXiv.2504.03884.
18. Кравцов Е. П. Разработка высокопроизводительных React-приложений: методы и практики оптимизации // European science. 2024. № 1 (69).
19. Дрогайцев И. А., Трамова А. М. Фронтенд-фреймворк Svelte как альтернатива популярным решениям в контексте создания систем поддержки принятия решений в сфере управления организацией с учетом трансформации экономики России // Известия КБНЦ РАН. 2024. № 4. DOI: 10.35330/1991-6639-2024-26-4-113-121. EDN: WRZTPH.
20. Яровая Е. В. Нестандартные архитектура в написание веб приложений // Столыпинский вестник. 2022.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Предметом исследования в рецензируемой статье выступает производительность рендеринга в современных веб-фреймворках.

Методология исследования базируется на разработке тестового программного обеспечения путем создания клиентского веб приложения, визуализирующего иерархическую выборку данных в табличной форме для проведения эксперимента с двумя структурами данных: бинарная – эмулирует реальные сайты с высокой вложенностью, линейная – эмулирует большие таблицы.

Актуальность работы авторы связывают с тем, что вопрос выбора фронтенд фреймворка становится фактором финансовой эффективности продукта, поскольку использование фреймворков и библиотек значительно ускоряет разработку приложений и уменьшает их стоимость - всё это приводит к необходимости сравнительного анализа фреймворков.

Научная новизна исследования состоит в представленных результатах эмпирического сравнения производительности рендерингов фреймворков React, Angular и Svelte в типовых сценариях построения и обновления пользовательского интерфейса, разработке унифицированного набора тестовых сценариев, автоматизации сбора и анализа экспериментальных данных для выявления недостатков и преимуществ фреймворков.

Структурно в работе выделены разделы, озаглавленные следующим образом: Введение, Разработка тестового ПО, Реализация тестового приложения с использованием фреймворков, Анализ экспериментальных данных, Заключение, и Библиография.

В публикации отмечено, что при современном уровне развития средств визуализации, информационных технологий и компьютерной техники контент должен появляться на экране практически мгновенно, без ощутимых задержек. Авторами подробно, с иллюстрациями фрагментов программного кода отражена реализация тестового приложения, исходный код экспериментального ПОложен в репозиторий на GitHub и доступен по приведенной в статье ссылке. Результаты эксперимента при сценарии первичного рендеринга представлены в графическом виде, текст статьи иллюстрирован

таблицей и 13 рисунками. По результатам исследования сформулированы рекомендации по использованию фреймворков React, Angular и Svelte для решения типичных классов задач, с указанием в каких случаях целесообразно отдавать предпочтение тем или иным программным платформам для получения двухмерного изображения.

Библиографический список включает 14 источников – интернет-ресурсы и научные публикации российских и зарубежных авторов по рассматриваемой теме на русском и иностранных языках. В тексте публикации имеются адресные ссылки к списку литературы, подтверждающие наличие апелляции к оппонентам.

Из замечаний стоит отметить следующие. Во-первых, в статье, подготовленной на русском языке имеется немало специальных терминов на английском языке, которые не сопровождаются их переводом – это может затруднить восприятие материала широким кругом читателей. Обращает на себя внимание и обилие заимствованных иностранных слов в тексте. Конечно, заимствованные слова в IT-сфере получили значительное распространение по вполне объяснимым причинам, но представляется уместным, по возможности, минимизировать использование иностранных слов, в том числе и в названии публикации. Во-вторых, авторами не выполнены рекомендации по оформлению списка литературы, принятые издательством: «Рекомендованный объем списка литературы для оригинальной научной статьи – не менее 20 источников, который должен содержать: не менее трети зарубежных источников; не менее половины работ, изданных в последние 3 года. В списке литературы не указываются ... Интернет-источники, включая информацию с сайтов, а также статьи на сайтах и в блогах... Все вышеперечисленные источники упоминаются в тексте статьи в скобках, наряду с прочими комментариями и примечаниями авторов».

Тема статьи актуальна, соответствует тематике журнала «Программные системы и вычислительные методы», но материал нуждается в доработке в соответствии с высказанными замечаниями.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Исследование производительности современных клиентских веб-фреймворков» и посвящена актуальному вопросу. Использование фреймворков и библиотек значительно ускоряет разработку приложений и уменьшает их стоимость на 30%, а также производительность веб-фреймворков остается одним из важных параметров. Все вышеперечисленное приводит к необходимости сравнительного анализа фреймворков, а также обосновывает актуальность работы.

В качестве цели статьи авторы указывают эмпирическое сравнение производительности рендеринга React, Angular и Svelte в типовых сценариях построения и обновления пользовательского интерфейса. Для достижения цели решаются задача разработки унифицированного набора тестовых сценариев, автоматизация сбора и анализ экспериментальных данных для выявления недостатков и преимуществ фреймворков.

В статье представлен достаточно широкий анализ литературных российских и зарубежных источников.

Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Практическая значимость статьи четко обоснована. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части (разработка тестового программного обеспечения, реализация

тестового приложения с использованием фреймворков, анализ экспериментальных данных).

Авторами в рамках экспериментального исследования создано клиентское одностраничное веб приложение, визуализирующее иерархическую выборку данных в табличной форме. Результаты исследования представлены в табличном и графическом виде. Практическое применение результатов сводится к формулировке рекомендаций.

Авторами в заключительной части статьи представлены рекомендации, в том числе: ориентировать React на приложения с длинными динамическими списками, обязательно используя ключи, мемоизацию и виртуализацию; выбирать Svelte для интерфейсов с глубокой вложенностью и требованием сверхбыстрой первой отрисовки, дополняя его библиотеками виртуального прокручивания при больших наборах данных; применять Angular в корпоративных системах, где приоритетны типизация и встроенный стек, строго активируя trackBy, OnPush и дифференциальную загрузку. Универсального лидера нет; выбор технологии должен основываться на аналитическом профиле операций и топологии интерфейса.

По мнению авторов представленная методика измерений обеспечивает воспроизводимую основу для дальнейшего сравнения новых версий библиотек. В качестве перспектив исследования авторы выделяют использование технологии веб-ассемблера WebAssembly для увеличения производительности, а также использования адаптивной гидратации в дальнейших исследованиях.

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна. Отсутствует четкое выделение предмета исследования.

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет.

Статья «Исследование производительности современных клиентских веб-фреймворков» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией рецензируемого научного журнала.

Результаты процедуры окончательного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена сравнительному анализу производительности трех популярных клиентских веб-фреймворков – React, Angular и Svelte – в типовых сценариях построения и обновления пользовательского интерфейса. Автор исследует такие аспекты, как первичный и повторный рендеринг, обновление и удаление элементов, используя линейные и бинарные структуры данных.

Методология исследования включает разработку унифицированного набора тестовых сценариев, создание клиентского одностраничного веб-приложения и автоматизированный сбор данных. Для измерений использовались инструменты Puppeteer и браузер Google Chrome, что обеспечило высокую точность и воспроизводимость результатов. Автор также применил функцию `performance.now()` для фиксации времени выполнения операций, что соответствует современным стандартам тестирования производительности.

Актуальность исследования не вызывает сомнений, так как выбор веб-фреймворка напрямую влияет на пользовательский опыт и финансовые показатели продукта. Статья отвечает на запросы разработчиков и бизнеса, предоставляя эмпирические данные для

обоснованного выбора технологии. Упоминание таких метрик, как Largest Contentful Paint (LCP), подчеркивает практическую значимость работы.

Научная новизна исследования заключается в разработке автоматизированной методики сравнительного анализа производительности фреймворков, а также в детальном изучении их поведения в различных сценариях. Автор не только подтверждает известные выводы, но и выявляет новые закономерности, например, влияние структуры данных на эффективность React и Svelte.

Статья написана четким и логичным языком, с соблюдением академических норм. Структура работы включает введение, описание методологии, анализ результатов и выводы, что делает материал легко воспринимаемым. Использование таблиц и графиков (Рисунки 6–13) наглядно демонстрирует результаты, а ссылки на авторитетные источники (например, [10], [16]) укрепляют доверие к исследованию.

Автор приходит к обоснованному выводу, что ни один из фреймворков не является универсальным решением. React демонстрирует преимущества в работе с длинными списками, Angular — в корпоративных системах с типизацией, а Svelte — в сценариях с глубокой вложенностью и требованием к быстрой первой отрисовке. Эти выводы подкреплены экспериментальными данными и имеют практическую ценность для разработчиков.

Статья будет полезна широкому кругу читателей: от frontend-разработчиков, выбирающих инструменты для проектов, до исследователей в области веб-технологий. Рекомендации по оптимизации (например, использование виртуального прокручивания для Svelte) делают материал особенно ценным для практиков.

Статья представляет собой качественное исследование, сочетающее научную строгость с практической направленностью. Работа заслуживает высокой оценки и может быть рекомендована к публикации в топ-5 статей месяца издательства. Учитывая актуальность темы и глубину анализа, материал вызовет значительный интерес у читательской аудитории.

Рекомендация: принять статью к публикации без доработок.

Программные системы и вычислительные методы

Правильная ссылка на статью:

Чикалева Ю.С. Анализ гранулярности микросервисов: эффективность архитектурных подходов // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.74386 EDN: NCFQFE URL: https://nbpublish.com/library_read_article.php?id=74386

Анализ гранулярности микросервисов: эффективность архитектурных подходов

Чикалева Юлия Сергеевна

ORCID: 0009-0009-7548-594X

студент; Мегафакультет компьютерных технологий и управления; Национальный исследовательский университет ИТМО

197101, Россия, г. Санкт-Петербург, Петроградский р-н, Кронверкский пр-кт, д. 49

✉ y.chikaleva99@mail.ru



[Статья из рубрики "Математическое моделирование и вычислительный эксперимент"](#)

DOI:

10.7256/2454-0714.2025.2.74386

EDN:

NCFQFE

Дата направления статьи в редакцию:

10-05-2025

Дата публикации:

29-05-2025

Аннотация: Современные информационные системы требуют масштабируемых архитектур для обработки больших данных и обеспечения доступности. Микросервисная архитектура, разделяющая приложения на автономные сервисы по бизнес-функциям, решает эти задачи. Однако оптимальная гранулярность микросервисов влияет на производительность, масштабируемость и управляемость. Неоптимальная декомпозиция приводит к антипаттернам, таким как избыточная мелкость или косметическая микросервисность, усложняя сопровождение. Цель исследования — сравнительный анализ методов определения гранулярности микросервисов для выявления подходов, обеспечивающих баланс производительности, гибкости и управляемости в высоконагруженных системах. Объект исследования — микросервисная архитектура высоконагруженных информационных систем, включая их структурные и

функциональные характеристики, определяемые методами декомпозиции на автономные сервисы. Предмет исследования – методы определения гранулярности микросервисов (монолитная архитектура, Domain-Driven Design, Data-Driven Approach, Monolith to Microservices Approach). Применён экспериментальный подход, включающий реализацию приложения Task Manager в четырёх архитектурных конфигурациях. Нагрузочное тестирование проводилось с использованием Apache JMeter при нагрузке 1000 пользователей. Метрики производительности (время отклика, пропускная способность, CPU), доступности, масштабируемости, безопасности и согласованности собраны через Prometheus и обработаны с вычислением средних значений и стандартного отклонения. Научная новизна исследования заключается в разработке унифицированной методологии количественного анализа методов гранулярности микросервисов (монолит, DDD, Data-Driven, Monolith to Microservices), основанной на метриках (время отклика, пропускная способность, CPU, доступность, запуск, безопасность, ошибки), адаптированных для высоконагруженных систем. В отличие от качественных исследований, работа предлагает комплексный подход, включая реализацию приложения Task Manager и нагрузочное тестирование (Apache JMeter, Prometheus), решая проблему выбора оптимальной декомпозиции для повышения производительности и масштабируемости. Методология применима для автоматизации оценки архитектур в CI/CD. Монолит показал минимальное время отклика (0.76 с) и пропускную способность (282.5 запросов/с), но ограничен масштабируемостью. Data-Driven обеспечивает согласованность, DDD эффективен для сложных доменов, Monolith to Microservices имеет низкую производительность (15.99 с) из-за перегрузки авторизации. Ограничение – хост-система (8 ГБ RAM), снижающая масштабируемость. Рекомендации: оптимизировать сетевые вызовы в DDD, доступ к данным в Data-Driven, декомпозицию в Monolith to Microservices.

Ключевые слова:

микросервисная архитектура, гранулярность микросервисов, Domain-Driven Design, Data-Driven Approach, Monolith to Microservices, производительность, масштабируемость, согласованность данных, экспериментальный анализ, высоконагруженные системы

Введение

Современные информационные системы сталкиваются с необходимостью обработки больших объемов данных, обеспечения высокой доступности и быстрого вывода продуктов на рынок. Микросервисная архитектура, разделяющая приложение на автономные сервисы, ориентированные на бизнес-функции, предоставляет разработчикам инструменты для создания масштабируемых и гибких систем [1]. Однако неоптимальный выбор гранулярности микросервисов может привести к избыточной сложности, увеличению коммуникационных издержек и снижению производительности [2].

Актуальность исследования обусловлена широким распространением микросервисной архитектуры и связанными с ней вызовами. Неоптимальная декомпозиция системы может привести к антипаттернам, таким как избыточная мелкость микросервисов или косметическая микросервисность, что усложняет масштабирование и сопровождение [3]. Наблюдается недостаток комплексных сравнений методов определения гранулярности микросервисов, что усложняет выбор оптимального подхода. В то же время оптимальная

гранулярность позволяет достичь баланса между гибкостью, производительностью и управляемостью.

Цель исследования заключается в анализе и сравнении методов определения гранулярности микросервисов с целью выявления наиболее эффективных подходов для проектирования высоконагруженных систем.

Этапы исследования

Для проведения сравнительного анализа методов определения гранулярности микросервисов были выделены следующие этапы:

1 . Выбор тестового приложения: в качестве тестового приложения был разработан Task Manager, предоставляющий функциональность управления задачами и пользователями. Приложение включает регистрацию, аутентификацию, создание и редактирование задач, что делает его подходящим для моделирования реальных высоконагруженных сценариев.

2 . Разработка тестового приложения: приложение было реализовано с использованием Java Spring Boot и MySQL. Рассматривались четыре архитектурные конфигурации: монолитная архитектура (для сравнения), Domain-Driven Design (DDD), Data-Driven Approach и Monolith to Microservices Approach. Каждая конфигурация представляла различный подход к декомпозиции системы.

3 . Настройка тестового окружения: для обеспечения объективности тестирования использовалась контейнеризация на основе Docker. Метрики собирались с помощью системы мониторинга Prometheus, а нагружочное тестирование проводилось с использованием Apache JMeter. Тестовое окружение обеспечивало изоляцию и минимизацию внешних факторов.

4 . Проведение тестов: тестирование включало измерение ключевых метрик (время отклика, пропускная способность, использование CPU, доступность, время запуска экземпляров, уровень безопасности, процент ошибок) для каждой архитектурной конфигурации. Тесты проводились в условиях последовательной отправки запросов для оценки производительности и устойчивости.

5 . Анализ результатов: собранные данные были проанализированы для выявления преимуществ и недостатков каждого метода декомпозиции. На основе анализа сформулированы рекомендации по применению методов в зависимости от проектных требований.

Тестовое приложение

Результаты исследования зависят от объективности выбора тестового приложения. Task Manager был спроектирован как приложение, моделирующее реальные сценарии высоконагруженных систем. Основной функционал включает:

- Регистрацию и аутентификацию пользователей.
- Создание, редактирование и удаление задач.
- Управление статусом задач и их назначением.

Приложение реализовано на Java Spring Boot с использованием RESTful API. Для хранения данных применялась реляционная база данных MySQL, обеспечивающая

надежность и производительность. Архитектурно приложение поддерживает различные конфигурации:

- **Монолитная архитектура:** вся функциональность объединена в одном сервисе с общей базой данных (рис. 1). Монолитная архитектура не предполагает декомпозиции, поэтому вся бизнес-логика (управление пользователями, задачами и их связями) реализована в едином приложении [4]. Это позволяет минимизировать сетевые накладные расходы и упростить управление данными, что делает монолит эталоном для сравнения с микросервисными подходами [5]. Общая база данных MySQL обеспечивает централизованное хранение всех данных, что соответствует традиционной монолитной модели [6].

Monolith

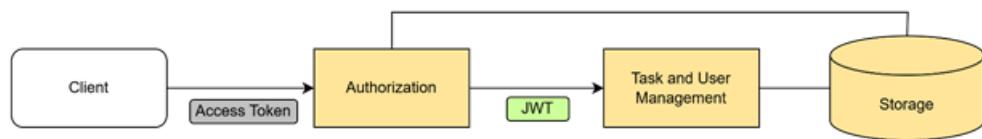


Рисунок 1. Схема монолитной архитектуры приложения Task Manager: единый сервис с общей базой данных MySQL, обеспечивающий управление пользователями и задачами.

Domain-Driven Design (DDD) Система разделена на три микросервиса — User Identity Service, Task Management Service, Task Assignment Service — на основе доменных областей (рис. 2). Метод DDD фокусируется на выделении границ домена (bounded context) в соответствии с бизнес-логикой [7]. User Identity Service отвечает за аутентификацию и управление данными пользователей, что представляет домен идентификации. Task Management Service управляет жизненным циклом задач (создание, редактирование, удаление), охватывая домен управления задачами [8]. Task Assignment Service связывает задачи и пользователей, представляя домен назначения задач. Такое разделение обеспечивает высокую модульность [9] и изоляцию бизнес-логики, что упрощает поддержку и масштабирование каждого домена независимо [10].

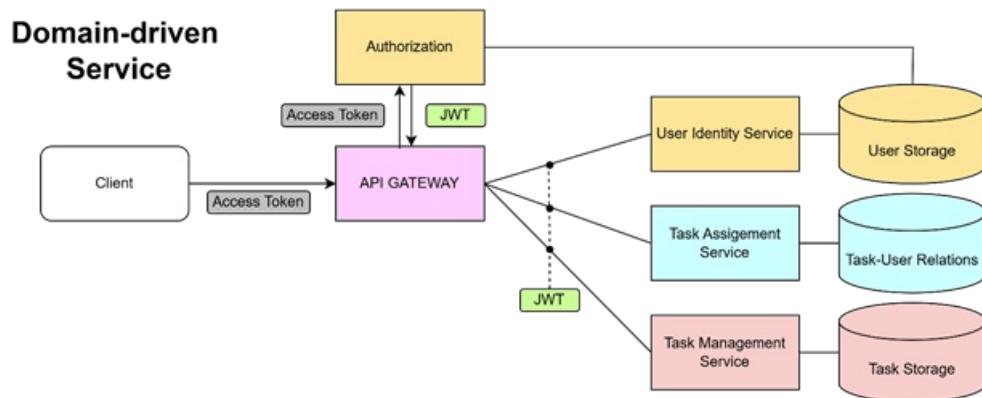


Рисунок 2. Схема архитектуры Task Manager на основе DDD: три микросервиса (User Identity Service, Task Management Service, Task Assignment Service), разделенные по доменным областям, взаимодействующие через REST API.

Data-Driven Approach: два микросервиса (Task Service, User Service) с изолированными базами данных (рис. 3). Подход, основанный на данных, предполагает декомпозицию системы на основе структуры данных и их независимости [11]. Task Service управляет данными задач, включая их создание, редактирование и статус, и использует

собственную базу данных MySQL для хранения информации о задачах. User Service отвечает за данные пользователей и аутентификацию, также с отдельной базой данных MySQL. Разделение по данным минимизирует зависимости между сервисами, обеспечивает автономность хранения и обработки данных, а также позволяет выбирать оптимальные технологии для каждого хранилища. Это снижает риск компрометации всей системы при атаке на одно хранилище, но требует синхронизации данных между сервисами.

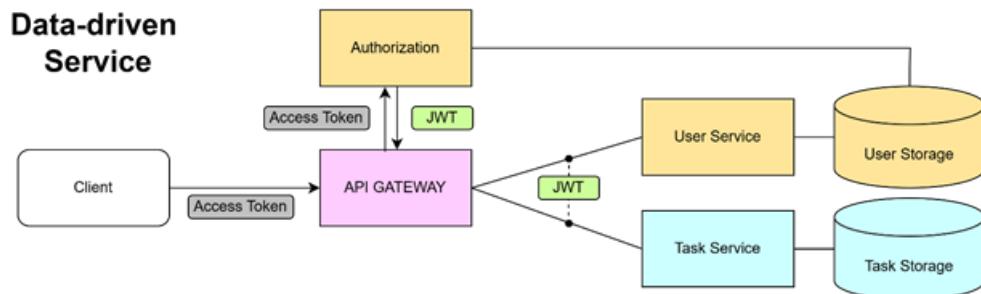


Рисунок 3. Схема архитектуры Task Manager на основе Data-Driven Approach: два микросервиса (Task Service, User Service) с изолированными базами данных MySQL, обеспечивающие независимость данных.

Monolith to Microservices Approach: выделен Authorization Service, остальная функциональность осталась в монолите (рис. 4). Метод перехода от монолита к микросервисам предполагает постепенную декомпозицию, начиная с выделения наиболее критичных и часто используемых функций [12]. Authorization Service был выбран для выделения, так как аутентификация пользователей является ключевой функцией, требующей высокой производительности и безопасности. Этот сервис управляет аутентификацией и работает с собственной базой данных MySQL, что позволяет изолировать критичную функциональность и снизить риски при миграции. Остальная функциональность (управление задачами и их назначением) осталась в монолите, что минимизирует изменения на начальном этапе перехода, но ограничивает автономность системы до полной декомпозиции [13].

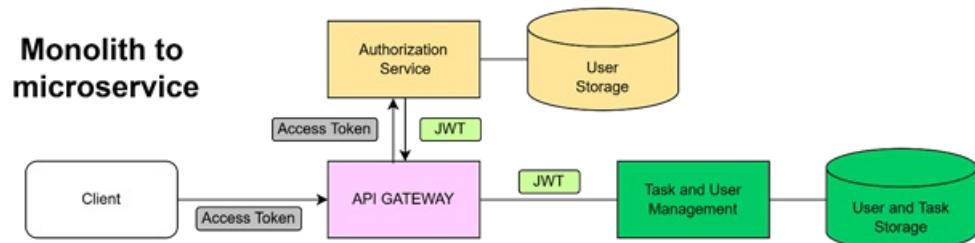


Рисунок 4. Схема архитектуры Task Manager на основе Monolith to Microservices Approach: выделен Authorization Service с собственной базой данных MySQL, остальная функциональность реализована в монолите.

Исходный код приложения доступен в репозитории на GitHub: [https://github.com/Yuliya3003/Microservices_research.git].

Методика тестирования

Тестирование проводилось в контролируемом окружении для обеспечения воспроизводимости и объективности результатов. Все архитектурные конфигурации приложения Task Manager (монолитная архитектура, Domain-Driven Design, Data-Driven

Approach, Monolith to Microservices Approach) были развернуты с использованием контейнеризации на основе Docker, что обеспечивало изоляцию сервисов и минимизацию влияния внешних факторов. Для сбора метрик использовалась система мониторинга Prometheus, а нагрузочное тестирование проводилось с помощью инструмента Apache JMeter. Тесты были направлены на оценку производительности, доступности, масштабируемости, безопасности и согласованности каждой конфигурации [\[14\]](#).

Процесс тестирования

Тестирование включало последовательную отправку HTTP-запросов к RESTful API приложения Task Manager для моделирования высоконагруженных сценариев. Процесс тестирования состоял из следующих этапов:

1. Подготовка тестового окружения:

- о Все архитектурные конфигурации приложения Task Manager были развернуты в контейнерах Docker на сервере с операционной системой Microsoft Windows 11 Pro (версия 10.0.22631), 8 ГБ оперативной памяти и 4-ядерным процессором AMD Ryzen 3 5300U.
- о Каждый контейнер содержал отдельный экземпляр сервиса (или монолитного приложения) и связанную базу данных MySQL. Для микросервисных конфигураций использовались отдельные базы данных для каждого сервиса, если это предусматривалось методом декомпозиции.
- о Prometheus был настроен для сбора метрик с экспортёров, встроенных в приложение (Spring Boot Actuator).

2. Конфигурация нагрузочного тестирования:

- о Нагрузочное тестирование проводилось с использованием Apache JMeter версии 5.6.3. Тестовый план моделировал высокую нагрузку на приложение путем отправки HTTP-запросов.

о Параметры теста:

§ **Количество потоков:** 1000 виртуальных пользователей, имитирующих одновременные запросы.

§ **Время разгона (Ramp-up):** 10 секунд, чтобы постепенно увеличить нагрузку и избежать резких пиков.

§ **Количество итераций:** 10 циклов выполнения запросов для каждого потока, что обеспечивало достаточный объем данных для анализа.

- о Тестовый план включал сбор результатов через компоненты Summary Report и View Results Tree, которые фиксировали время отклика, пропускную способность, статусы ответов и ошибки.

3. Сбор и обработка метрик:

- о Метрики собирались в реальном времени с использованием Prometheus, настроенного на интервал сбора данных 1 секунда.

- о После завершения тестов данные экспорттировались из Prometheus для дальнейшего

анализа. Для расчета средних значений и построения графиков использовались запросы PromQL (Prometheus Query Language).

Описание метрик и их измерение

Для выбора характеристик и метрик, используемых в сравнительном анализе методов определения гранулярности микросервисов, был проведён систематический обзор 56 научных статей, опубликованных в рецензируемых изданиях и посвящённых измерению эффективности программных систем, включая микросервисные архитектуры [15]. В рамках анализа были идентифицированы наиболее часто встречающиеся характеристики качества [16], такие как производительность, доступность, масштабируемость, безопасность и согласованность, которые имеют критическое значение для высоконагруженных распределённых систем [17]. Частота упоминания данных характеристик в рассмотренных источниках представлена на рисунке 5.



Рисунок 5. Частота встречаемости исследуемых критериев в научных статьях.

Выбор метрик осуществлялся с учётом их измеримости с использованием системы мониторинга Prometheus, что обеспечивало высокую точность и воспроизводимость результатов [18]. Перечень метрик, их определение и методы измерения представлены в Таблице 1.

Таблица 1. Количественные метрики для оценки характеристик качества системы

Характеристика	Метрика	Измерение
Производительность	Время отклика, с	$\frac{\text{http_server_requests_seconds_sum}}{\text{http_server_requests_seconds_count}}$
	CPU	process_cpu_usage
	Пропускная способность, rec/s	rate(http_server_requests_seconds_count[5m])
Доступность	Среднее время	avg_over_time(up{job="service_name"}[5m])

	безотказной работы, %	* 100%
Масштабируемость	Время запуска экземпляров, с	application_started_time_seconds
Безопасность	Уровень безопасности	калькулятор CVSS 4.0
Согласованность	Процент ошибок, %	$\frac{\text{rate}(\text{http_server_requests_seconds_count}[\text{status}=\sim"4../5..])[5m]}{\text{rate}(\text{http_server_requests_seconds_count}[5m])} * 100$

Результаты тестирования

Результаты измерений представлены в Таблице 2.

Монолитная архитектура продемонстрировала высокую производительность, обусловленную отсутствием сетевых задержек, характерных для распределённых систем [\[19\]](#). Среднее время отклика составило 0.76 секунды, а пропускная способность достигла 282.5 запросов в секунду при минимальном использовании CPU (0.0055). Однако масштабируемость конфигурации ограничена, о чём свидетельствует относительно высокое время запуска экземпляров (14.86 секунды), что указывает на потенциальные сложности при горизонтальном масштабировании.

Domain-Driven Design (DDD) обеспечивает модульность за счёт разделения системы на независимые доменные области, что упрощает сопровождение сложных бизнес-логики. Тем не менее, межсервисные вызовы привели к увеличению времени отклика до 1.07 секунды и снижению пропускной способности до 27.6 запросов в секунду. Использование CPU (0.1698) также оказалось выше, чем у монолита, что отражает дополнительные накладные расходы на координацию сервисов.

Data-Driven Approach выделяется высокой согласованностью данных благодаря изоляции баз данных для каждого сервиса, что минимизирует конфликты при параллельной обработке. Однако изоляция баз привела к значительному увеличению времени отклика (5.49 секунды), что указывает на необходимость оптимизации запросов к данным. Пропускная способность составила 35.6 запросов в секунду, а использование CPU (0.1704) оказалось сопоставимым с DDD. Время запуска экземпляров (24.37 секунды) отражает умеренные затраты на инициализацию изолированных компонентов, что согласуется с выводами о влиянии гранулярности на масштабируемость микросервисных систем [\[20\]](#).

Monolith to Microservices Approach показал наименее удовлетворительные результаты, что обусловлено неполной декомпозицией системы. Выделение Authorization Service привело к высокой нагрузке на этот компонент, что проявилось в максимальном времени отклика (15.99 секунды) и низкой пропускной способности (33.34 запроса в секунду). Использование CPU (0.1778) было наивысшим среди всех конфигураций, а время запуска экземпляров (33.68 секунды) указывает на значительные затраты на инициализацию частично декомпозированной системы.

Таблица 2. Сравнение результатов тестирования

		Монолит	Domain-	Data-	Monolith to
--	--	---------	---------	-------	-------------

		Monolith	Driven	Driven	microservice
Производительность	Время отклика, с	0,76	1,07	5,49	15,99
	CPU	0,0055	0,1698	0,1704	0,1778
	Пропускная способность, rec/s	282,5	27,6	35,6	33,34
Доступность	Среднее время безотказной работы, %	100	100	100	100
Масштабируемость	Время запуска экземпляров, с	14,86	31,86	24,37	33,68
Безопасность	Уровень безопасности	5,4	5,4	5,4	5,4
Согласованность	Процент ошибок, %	0	0	0	0

Монолитная архитектура демонстрирует наилучшую производительность, но ограничена в гибкости. Data-Driven Approach выделяется высокой согласованностью данных, DDD подходит для сложных доменов, а Monolith to Microservices требует дальнейшей декомпозиции.

Анализ результатов

Проведённое исследование подтвердило существенное влияние метода декомпозиции на характеристики качества системы, измеренные в ходе нагрузочного тестирования с использованием Apache JMeter и мониторинга через Prometheus. Анализ данных, представленных в Таблице 2, выявил преимущества и ограничения каждой архитектурной конфигурации приложения Task Manager.

Монолитная архитектура продемонстрировала наивысшую производительность, выраженную в минимальном времени отклика (0.76 секунды) и высокой пропускной способности (282.5 запросов в секунду), что обусловлено отсутствием сетевых накладных расходов. Однако ограниченная масштабируемость, проявляющаяся в относительно высоком времени запуска экземпляров (14.86 секунды), делает её менее подходящей для систем, требующих быстрого горизонтального расширения.

Архитектура на основе Domain-Driven Design (DDD) обеспечивает модульность, упрощающую сопровождение сложных доменных моделей. Тем не менее, межсервисные взаимодействия увеличили время отклика до 1.07 секунды и снизили пропускную способность до 27.6 запросов в секунду, что указывает на необходимость оптимизации сетевых вызовов.

Data-Driven Approach показал высокую согласованность данных за счёт изоляции баз данных, минимизирующей конфликты при параллельных операциях. Однако значительное время отклика (5.49 секунды) подчёркивает потребность в оптимизации

доступа к данным, несмотря на умеренную пропускную способность (35.6 запросов в секунду) и приемлемое время запуска экземпляров (24.37 секунды).

Monolith to Microservices Approach оказался наименее эффективным из-за неполной декомпозиции, что привело к перегрузке Authorization Service. Максимальное время отклика (15.99 секунды) и низкая пропускная способность (33.34 запроса в секунду) отражают ограничения конфигурации, а высокое время запуска экземпляров (33.68 секунды) подчёркивает сложности масштабирования.

На основе анализа результатов предлагаются следующие рекомендации:

1. Использовать монолитную архитектуру для систем с приоритетами производительности и ограниченными требованиями к масштабируемости.
- 2 . Применять DDD для проектов со сложной доменной логикой, уделяя внимание оптимизации межсервисных взаимодействий.
- 3 . Выбирать Data-Driven Approach для систем, требующих высокой согласованности данных, с акцентом на оптимизацию запросов к базам данных.
- 4 . Проводить тщательное планирование декомпозиции при переходе от монолита к микросервисам, чтобы избежать перегрузки отдельных компонентов.

Заключение

В ходе исследования были систематизированы методы определения гранулярности микросервисов, разработано приложение Task Manager и проведено тестирование четырех архитектурных подходов. Полученные данные подтвердили влияние гранулярности на производительность, масштабируемость и согласованность системы.

Результаты исследования могут служить основой для дальнейших работ в области оптимизации микросервисных архитектур. Перспективные направления включают:

- Исследование асинхронных паттернов (Saga, Event Sourcing) для снижения сетевых накладных расходов.
- Анализ влияния оркестрации (Kubernetes) на масштабируемость микросервисов.
- Разработка автоматизированных инструментов для оценки гранулярности в CI/CD-процессах.

Библиография

1. Ворсин В. А. Микросервисная архитектура бизнес-приложений: перспективы и проблемы // GLOBUS. 2020. Вып. 4 (50). С. 51-53. EDN: TYXBZZ.
2. Raj P., Vanga S., Chaudhary A. Microservices Security // CloudNative Computing. 2022. DOI: 10.1002/9781119814795.ch14.
3. Зиборев А. В. Антипаттерны построения микросервисных приложений в высоконагруженных проектах // UNIVERSUM: Технические науки. 2023. Вып. 11 (116). Ноя. 2023. doi: 10.32743/UniTech.2023.116.11.16204. EDN: HBLBIO.
4. Newman S. Building Microservices. 2nd ed. Sebastopol, CA: O'Reilly Media, 2025.
5. Blinowski G. и др. Monolithic vs. Microservice Architecture: Performance and Scalability Evaluation // IEEE Access. 2022. DOI: 10.1109/ACCESS.2022.3152803. EDN: JHKSCZ.
6. Richardson C. Microservice Architecture Pattern. 2020. Available at: <https://microservices.io/patterns/microservices.html> (accessed: Jan. 05, 2025).
7. Momil S., Qaisar A. Transition Strategies from Monolithic to Microservices Architectures: A Domain-Driven Approach and Case Study // VAWKUM Transactions on Computer Sciences.

2024. Vol. 12, No. 1. Pp. 94-110. DOI: 10.21015/vtcs.v12i1.1808. EDN: ABOYUL.
8. Радостев Д. К., Никитина Е. Ю. Стратегия миграции программного кода из монолитной архитектуры в микросервисы // Вестник Пермского Университета. 2021. Вып. 2 (53). С. 65-68. doi: 10.17072/1993-0550-2021-2-65-68. EDN: DSNKOZ.
9. Jordanov J., Petrov P. Domain Driven Design Approaches in Cloud Native Service Architecture // TEM Journal. 2023. Vol. 12, Issue 4. C. 1985-1994. ISSN 2217-8309. DOI: 10.18421/TEM124-09. EDN: MMRBFA.
10. Vular H., Koyuncu M. Does Domain-Driven Design Lead to Finding the Optimal Modularity of a Microservice? // IEEE Access. 2021. Vol. 9. Pp. 27960-27971. DOI: 10.1109/ACCESS.2021.306089.
11. Hasselbring W., Steinacker G. Microservice Architectures for Scalability, Agility and Reliability in E-Commerce // 2017 IEEE International Conference on Software Architecture Workshops (ICSAW). Gothenburg: IEEE, 2017. P. 46-49. DOI: 10.1109/ICSAW.2017.11.
12. Schmidt R. A., Thiry M. Microservices Identification Strategies: A Review Focused on Model-Driven Engineering and Domain-Driven Design Approaches // 2020 15th Iberian Conference on Information Systems and Technologies (CISTI). Seville: IEEE, 2020. P. 1-6. DOI: 10.23919/CISTI49556.2020.9141150.
13. Christudas B. Practical Microservices Architectural Patterns. New York: Apress, 2019. DOI: 10.1007/978-1-4842-4501-9.
14. Sriraman A., Wenisch T. F. μSuite: A Benchmark Suite for Microservices // 2018 IEEE International Symposium on Workload Characterization (IISWC). Raleigh: IEEE, 2018. P. 1-12. DOI: 10.1109/IISWC.2018.8573515.
15. Henning S., Hasselbring W. Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud // Journal of Systems and Software. 2024. Vol. 208. [Online]. Available: <https://doi.org/10.1016/j.jss.2023.111879>. EDN: RVNVGK.
16. Bjørndal N., & Pires de Araújo, etc. Benchmarks and performance metrics for assessing the migration to microservice-based architectures // The Journal of Object Technology. August 2021. DOI: 10.5381/jot.
17. Артамонов И. В. Показатели производительности микросервисных систем // Вестник НГИЭИ. 2018. Вып. 8 (87). С. 24-33. EDN: XYTGKT.
18. Waseem M., Liang P., Shahin M., et al. Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective // Journal of Systems and Software. 2021. Vol. 182. Article 111061. DOI: 10.1016/j.jss.2021.111061. EDN: ATPBYU.
19. Ramu V. Performance Impact of Microservices Architecture // Rev. Contemp. Sci. Acad. Stud. 2023. Vol. 3. P. 1-6.
20. Hassan S., Bahsoon R., & Buyya R. Systematic scalability analysis for microservices granularity adaptation design decisions // Software: Practice and Experience. 2022. Vol. 52, No. 6. DOI: 10.1002/spe.3069. EDN: MOLQOU.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Анализ гранулярности микросервисов: эффективность архитектурных подходов» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена актуальному вопросу. Актуальность исследования обусловлена широким распространением микросервисной архитектуры и связанными с ней вызовами. Неоптимальная декомпозиция системы может привести к антипаттернам, таким как избыточная мелкость микросервисов или косметическая микросервисность,

что усложняет масштабирование и сопровождение. Наблюдается недостаток комплексных сравнений методов определения гранулярности микросервисов, что усложняет выбор оптимального подхода. В то же время оптимальная гранулярность позволяет достичь баланса между гибкостью, производительностью и управляемостью. В качестве цели авторы указывают анализ и сравнение методов определения гранулярности микросервисов с целью выявления наиболее эффективных подходов для проектирования высоконагруженных систем.

Для проведения сравнительного анализа методов определения гранулярности микросервисов авторами были выделены следующие этапы:

1. Выбор тестового приложения.
2. Разработка тестового приложения.
3. Настройка тестового окружения.
4. Проведение тестов.
5. Анализ результатов.

Для выбора характеристик и метрик, используемых в сравнительном анализе методов определения гранулярности микросервисов, авторами был проведён систематический обзор 56 научных статей, опубликованных в рецензируемых изданиях и посвящённых измерению эффективности программных систем, включая микросервисные архитектуры. В рамках анализа были идентифицированы наиболее часто встречающиеся характеристики качества, такие как производительность, доступность, масштабируемость, безопасность и согласованность, которые имеют критическое значение для высоконагруженных распределённых систем.

Также, проведённое авторами исследование подтвердило существенное влияние метода декомпозиции на характеристики качества системы, измеренные в ходе нагружочного тестирования с использованием Apache JMeter и мониторинга через Prometheus.

Результаты исследования представлены в табличном и графическом виде.

В списке литературы представлены зарубежные источники по теме исследования. Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Практическая значимость статьи четко обоснована. Результаты исследования могут служить основой для дальнейших работ в области оптимизации микросервисных архитектур. По мнению авторов перспективными направлениями являются:

- Исследование асинхронных паттернов для снижения сетевых накладных расходов.
- Анализ влияния оркестрации на масштабируемость микросервисов.
- Разработка автоматизированных инструментов для оценки гранулярности в CI/CD-процессах.

Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части.

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна. Отсутствует четкое выделение предмета, объекта исследования.

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет, объект. Также будет целесообразным расширить список литературы, указав работы отечественных авторов.

Статья «Анализ гранулярности микросервисов: эффективность архитектурных подходов» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией рецензируемого научного журнала.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Рецензируемая статья посвящена одному из ключевых вопросов проектирования распределённых систем – анализу гранулярности микросервисов.

Методология исследования базируется на разработке тестового программного приложения и проведении тестирование четырех архитектурных подходов: монолитной архитектуры, архитектуры на основе Domain-Driven Design, Data-Driven Approach и Monolith to Microservices Approach.

Актуальность работы авторы связывают с широким распространением микросервисной архитектуры, с тем, что неоптимальный выбор гранулярности микросервисов может привести к избыточной сложности, увеличению коммуникационных издержек и снижению производительности.

Научная новизна работы состоит в систематизации методов определения гранулярности микросервисов, полученных результатах сравнения различных четырех таких методов, в подтверждении влияния гранулярности на производительность, масштабируемость и согласованность системы и в выявлении эффективных подходов для проектирования высоконагруженных систем.

В тексте публикации структурно выделены следующие разделы: Введение, Этапы исследования, Тестовое приложение, Методика тестирования, Процесс тестирования, Описание метрик и их измерение, Результаты тестирования, Анализ результатов, Заключение и Библиография.

В публикации освещены этапы проведения сравнительного анализа методов определения гранулярности микросервисов: выбор тестового приложения, разработка тестового приложения, настройка тестового окружения, проведение тестов, анализ результатов. Авторами проведено изучение публикаций, посвященных измерению эффективности программных систем, отражена частота встречаемости исследуемых критериев в научных статьях, выбраны метрики для измерения таких характеристик как производительность, доступность, масштабируемость, безопасность и согласованность. Тестирование включало измерение ключевых метрик (время отклика, пропускная способность, использование центрального процессора, доступность, время запуска экземпляров, уровень безопасности, процент ошибок) для каждой архитектурной конфигурации. Отмечено, что монолитная архитектура демонстрирует наилучшую производительность, но ограничена в гибкости. Data-Driven Approach выделяется высокой согласованностью данных, DDD подходит для сложных доменов, а Monolith to Microservices требует дальнейшей декомпозиции. Статья иллюстрирована 5 рисунками, содержит 2 таблицы.

Библиографический список включает 13 источников: работы российских и зарубежных авторов по рассматриваемой тематике на русском и иностранном языках. В тексте публикации имеются адресные ссылки к списку литературы, подтверждающие наличие апелляции к оппонентам.

Из замечаний стоит отметить, что названия таблиц почему-то приведены не перед ними, как это предусмотрено правилами оформления и ГОСТами, а после таблиц; авторами не соблюдены принятые редакцией Правила оформления списка литературы по количеству и категориям источников: «Рекомендованный объем списка литературы ... – не менее 20 источников, который должен содержать: ... не менее половины работ, изданных в последние 3 года».

Тема статьи актуальна, отражает результаты проведенного авторами исследования, соответствует тематике журнала «Программные системы и вычислительные методы»,

может вызвать интерес у определенного круга IT-специалистов, но материал нуждается в доработке.

Результаты процедуры окончательного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена важной и актуальной проблеме определения оптимальной гранулярности микросервисов в современных информационных системах. Автор проводит детальное исследование, сравнивая различные архитектурные подходы, включая Domain-Driven Design, Data-Driven Approach и метод постепенного перехода от монолита к микросервисам. В качестве инструмента анализа используется разработанное тестовое приложение Task Manager, что позволяет наглядно оценить влияние каждого подхода на ключевые характеристики системы, такие как производительность, масштабируемость и согласованность данных.

Методологическая основа исследования заслуживает высокой оценки. Автор применяет комплексный подход, сочетающий теоретический анализ с практической реализацией. Тестовое приложение Task Manager разработано с поддержкой четырех архитектурных конфигураций, что обеспечивает объективность сравнения. Нагрузочное тестирование проводится с использованием Apache JMeter, а сбор метрик осуществляется через систему мониторинга Prometheus. Такой подход позволяет получить достоверные данные о времени отклика, пропускной способности, использовании CPU и других критически важных показателях. Результаты представлены в виде наглядных таблиц и графиков, что значительно облегчает восприятие и анализ информации.

Актуальность исследования не вызывает сомнений. В условиях стремительного роста популярности микросервисных архитектур вопросы оптимизации гранулярности, снижения коммуникационных издержек и обеспечения масштабируемости становятся все более значимыми. Статья заполняет существенный пробел в области сравнительного анализа методов декомпозиции, предлагая практические решения для разработчиков и архитекторов. Это делает работу особенно ценной для специалистов, занимающихся проектированием высоконагруженных распределенных систем.

Научная новизна исследования проявляется в систематизации методов определения гранулярности микросервисов и их эмпирической проверке на реальном тестовом приложении. Автор не только сравнивает различные подходы, но и формулирует четкие рекомендации по их применению в зависимости от конкретных требований проекта. Например, монолитная архитектура демонстрирует высокую производительность, но ограничена в масштабируемости, тогда как Domain-Driven Design подходит для сложных доменных моделей, но требует оптимизации межсервисных взаимодействий. Подобные выводы основаны на достоверных данных и могут служить ориентиром для дальнейших исследований в этой области.

Стиль изложения статьи научный, но при этом доступный для понимания. Структура работы логична и последовательна: от постановки проблемы и описания методологии до представления результатов и их анализа. Каждый раздел дополняет предыдущий, создавая целостную картину исследования. Использование иллюстраций и таблиц удачно подчеркивает ключевые моменты, а ссылки на авторитетные источники подтверждают глубину проработки темы.

Практическая значимость работы очевидна. Результаты исследования могут быть непосредственно применены при проектировании и оптимизации микросервисных систем. Автор не только выявляет преимущества и недостатки каждого подхода, но и

предлагает конкретные рекомендации, такие как необходимость оптимизации запросов к данным для Data-Driven Approach или тщательное планирование декомпозиции при переходе от монолита к микросервисам. Это делает статью полезной не только для теоретиков, но и для практикующих специалистов.

Интерес читательской аудитории к данной работе будет высоким. Материал представляет ценность для разработчиков, архитекторов и исследователей, занимающихся вопросами микросервисных архитектур. Четкие выводы и практические рекомендации позволяют использовать статью в качестве руководства при принятии проектных решений. Кроме того, предложенные направления для будущих исследований, такие как изучение асинхронных паттернов или влияние оркестрации Kubernetes, открывают новые перспективы для научных дискуссий.

В заключение следует отметить, что статья «Анализ гранулярности микросервисов: эффективность архитектурных подходов» является качественным и завершенным исследованием, сочетающим теоретическую глубину с практической применимостью. Работа выполнена на высоком академическом уровне, методологически обоснована и актуальна. Учитывая значительный вклад в область микросервисных архитектур, статью рекомендуется принять к публикации без доработок.

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Рогов Д.В., Алпатов А.Н. Интеллектуальная инфраструктура автоматизированного управления и интероперабельности микросервисов в облачных средах // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.74760 EDN: WIHQAD URL: https://nbpublish.com/library_read_article.php?id=74760

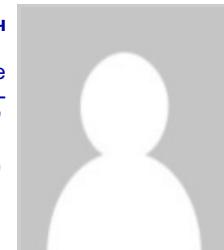
Интеллектуальная инфраструктура автоматизированного управления и интероперабельности микросервисов в облачных средах

Рогов Дмитрий Вадимович

студент; кафедра Инstrumentального и Прикладного Программного Обеспечения; Федеральное государственное бюджетное образовательное учреждение высшего образования "МИРЭА-Российский технологический университет"

123007, Россия, г. Москва, Хорошевский р-н, Хорошёвское шоссе, д. 60

✉ 1664286@gmail.com

**Алпатов Алексей Николаевич**

кандидат технических наук

доцент; кафедра Инstrumentального и Прикладного Программного Обеспечения; Федеральное государственное бюджетное образовательное учреждение высшего образования "МИРЭА-Российский технологический университет"

111033, Россия, г. Москва, р-н Лефортово, Танковый проезд, д. 4А, кв. 24

✉ aleksej01-91@mail.ru



[Статья из рубрики "Операционные системы"](#)

DOI:

10.7256/2454-0714.2025.2.74760

EDN:

WIHQAD

Дата направления статьи в редакцию:

09-06-2025

Дата публикации:

19-06-2025

Аннотация: В условиях стремительного роста масштабов и сложности информационных

систем, вопросы эффективной интеграции и сопровождения микросервисных архитектур становятся всё более актуальными. Одной из ключевых проблем является обеспечение интероперабельности программных компонентов, что предполагает возможность надёжного обмена данными и совместного использования информации между различными сервисами, реализованными с использованием разнородных технологий, протоколов и форматов данных. В данной работе предметом исследования выступает формализация и построение интеллектуальной системы, обеспечивающей интероперабельность микросервисных компонентов в облачной инфраструктуре. Предложен формализованный подход, основанный на графовых, категориальных и алгебраических моделях, позволяющий строго описывать маршруты передачи данных, условия совместимости интерфейсов и процедуру автоматизированного согласования форматов взаимодействия. Введена операция согласования интерфейсов, обеспечивающая выявление необходимости использования адаптеров и преобразователей для интеграции различных сервисов. Особое внимание уделяется задаче построения универсального интерфейса, через который возможна маршрутизация любых потоков данных, что значительно упрощает процедуру масштабирования и доработки микросервисной системы. Разработанная архитектура системы охватывает этапы создания, публикации и развертывания контейнерных микросервисов, автоматическую проверку маршрутов передачи данных, а также динамическое управление состоянием сервисов на основе прогнозирования нагрузки с помощью моделей искусственного интеллекта. Применение предложенной методики позволяет существенно повысить гибкость, надёжность и масштабируемость инфраструктуры, снизить эксплуатационные затраты, а также автоматизировать процессы поддержки и интеграции новых компонентов. Предложенное решение основывается на формализованном подходе к обеспечению интероперабельности микросервисных компонентов в облачной инфраструктуре. В качестве основы используется графовая и категориальная модель, позволяющая строго определить маршруты передачи данных и процедуры согласования интерфейсов между различными сервисами. Для унификации взаимодействия и повышения гибкости системы введена операция согласования интерфейсов, а также реализована возможность автоматизированного выявления необходимости применения адаптеров и преобразователей данных. Разработанный алгоритм интеллектуального прогнозирования нагрузки на сервисы позволяет динамически управлять состоянием компонентов и оперативно адаптировать инфраструктуру к изменяющимся условиям эксплуатации.

Ключевые слова:

Микросервисная архитектура, Интероперабельность, Контейнеризация, Облачная инфраструктура, Согласование интерфейсов, Категориальная модель, Графовая модель, Автоматизация деплоя, Адаптеры, Интеллектуальное управление

Введение

В последние годы наблюдается стремительный рост масштабов и сложности информационных систем, что обуславливает высокие требования к их сопровождению, масштабируемости и интеграции [1]. Одной из ключевых тенденций современного этапа развития отрасли становится переход к микросервисной архитектуре, которая позволяет создавать гибкие, легко масштабируемые и модульные программные продукты. Микросервисы, функционируя в единой инфраструктуре, обеспечивают независимость

разработки, развертывания и обновления отдельных компонентов, что значительно повышает эффективность работы команд сопровождения и эксплуатации.

В то же время внедрение микросервисных систем сопряжено с необходимостью обработки больших объёмов разнородных данных, поступающих от различных источников. Для преобразования такой информации компании выстраивают ETL процессы. Данный подход позволяет преобразовать данные и сохранить в целевом хранилище. Но такие системы также требуют для своей работы набор сервисов, который выполнит преобразование необходимой информации. В условиях возрастающей сложности и масштабности систем для ИТ-компаний становится крайне актуальной задача оптимизации затрат на построение и сопровождение микросервисной инфраструктуры, а также повышения эффективности процессов поддержки и интеграции.

Одним из перспективных подходов к решению данной проблемы выступает использование систем интероперабельности программных компонентов [2], которые обеспечивают автоматизацию процессов создания, развертывания и управления микросервисами, а также позволяют реализовать графические интерфейсы взаимодействия между ними. Такие системы обладают большим функционалом по созданию, развертыванию и управлению микросервисами, а также позволяют реализовать графические интерфейсы взаимодействия между ними, что способствует снижению эксплуатационных расходов и повышению устойчивости современных информационных систем.

На практике для этих целей нередко используются промышленные платформы, такие как SAS RTDM [3] и аналогичные решения. Однако проведённый анализ показывает, что существующие системы зачастую не обеспечивают необходимой гибкости при интеграции современных технологий, не поддерживают автоматическое интеллектуальное управление ресурсами и не способны оперативно реагировать на динамические изменения во входных потоках данных. Это существенно ограничивает возможности развития, масштабирования и эксплуатации информационных систем, а также приводит к росту эксплуатационных затрат.

В связи с этим актуальной задачей является разработка архитектуры интеллектуальной системы, способной обеспечить высокую степень интероперабельности программных компонентов, автоматическое управление состоянием сервисов и адаптацию к изменяющимся условиям эксплуатации.

Научная новизна работы заключается в разработке метода и обеспечения интероперабельности программных компонентов, а также применение моделей искусственного интеллекта для прогнозирования поступающей на сервисы нагрузки с целью управления состоянием программных компонент. Предметом исследования являются методы обеспечения интероперабельности программных компонентов, а также интеллектуальные методы прогнозирования нагрузки, поступающей на программные компоненты.

Постановка задачи обеспечения интероперабельности программных компонентов

Прежде чем приступим к рассмотрению способа организации системы интеграции для микросервисной архитектуры, дадим определение данному понятию и формализуем данную задачу.

Под интероперабельностью в современных программных системах понимают, как способность двух или более информационных систем или их компонентов к обмену

информацией и использованию полученных данных в совместной работе [4]. Вместе с тем, эволюция цифровых технологий и усложнение архитектур привели к существенному расширению содержания данного термина. Если изначально интероперабельность рассматривалась преимущественно в техническом аспекте, а именно как синтаксическая и протокольная совместимость, то современный подход требует учитывать семантические, организационные, законодательные и даже политические аспекты взаимодействия систем, что отражено в ряде стандартов [5]. Так например, в российском стандарте ГОСТ Р 55062-2012 выделяются технический, семантический и организационный уровни. Технический уровень связан с форматами, протоколами и средствами передачи информации, семантический связан с единым толкованием и трактовкой данных, а организационный связан с согласованием бизнес-целей, регламентов и правовых условий взаимодействия. Таким образом, интероперабельность на современном этапе рассматривается как многогранное понятие, охватывающее широкий спектр аспектов, без которых невозможно построение масштабируемых и взаимосвязанных информационных систем, в том числе, основанных на микросервисной архитектуре.

Микросервисное взаимодействие может быть описано, через граф

$$G = (V, E, \mathcal{T}, \mathcal{P}) \#(1)$$

где $V = \{s_1, s_2, \dots, s_n\}$ - множество вершин (микросервисов),

$E \subseteq V \times V \times \mathcal{T}$ - множество рёбер, каждое из которых связано с типом взаимодействия,

\mathcal{T} - множество типов взаимодействия (REST, gRPC, SOAP и др.),

$\mathcal{P}: E \rightarrow [0,1]$ - функция, задающая вероятность передачи сообщения по ребру (стоимость, надёжность канала и т.д.).

Стоит понимать, что в современных системах данные могут проходить через цепочки сервисов и преобразований, и необходимо учитывать, какие маршруты совместимы, где есть ограничения, а также наличие новых возможностей для интеграции. Для этого расширим модель, путём внедрения категорной структуры [6]. На основе графа (1) строится категория \mathcal{C}_G , где каждому микросервису ставится в соответствие объект $\text{Obj}(\mathcal{C}_G) = V$, а каждому ребру $(s_i, s_j, t_{ij}) \in E$ сопоставляется элементарный морфизм $f_{ij}^{(t_{ij})}: s_i \rightarrow s_j$, который моделирует непосредственное взаимодействие между сервисами. Если существуют морфизмы (отображения) $f: s_i \rightarrow s_j$ и $g: s_j \rightarrow s_k$, то их композиция $g \circ f: s_i \rightarrow s_k$ позволяет описать сложные маршруты передачи данных или комбинированную обработку. Для каждого микросервиса $s_i \in V$ также определён тождественный морфизм $\text{id}_{s_i}: s_i \rightarrow s_i$, соответствующий отсутствию какого-либо преобразования. Таким образом, любой путь в графе, то есть любая последовательность рёбер, индуцирует составной морфизм в категории. Это позволяет строго и формально описывать сложные сценарии взаимодействия между микросервисами. При этом, поскольку каждое ребро содержит тип взаимодействия t_{ij} , данный атрибут выступает аннотацией соответствующего морфизма, что даёт возможность учитывать специфику используемых протоколов, форматов данных или других технических особенностей на уровне категорной модели. Также такой подход принципиально важен для анализа совместимости, поскольку он позволяет моделировать не только прямые взаимодействия, но и цепочки передачи данных, а также обосновывать необходимость построения адаптеров и промежуточных преобразователей. Рассмотрим пример, пусть имеется крупная компания, у которой есть распределённая

система, где каждый микросервис выполняет отдельную бизнес-функцию. Так сервис А отвечает за авторизацию пользователей, принимая на вход запросы по REST, отдавая JSON данные. Сервис В реализует механизмы антифлага, то есть проверку транзакций, при этом он принимает JSON по REST, отдаёт структурированные события в формате Protobuf по gRPC. Сервис С реализует биллинг, получая события по gRPC (Protobuf), преобразуя и записывая платёжные операции. Сервис D реализует хранилище данных принимая отчёты в формате Avro по HTTP POST. Формируется цепочка действий, когда пользователь аутентифицируется через Сервис А, далее его транзакция отправляется на проверку в Сервис В, а после проверки результат отправляется в Сервис С. Для построения отчёта данные из сервиса С агрегируются и отправляются в сервис D. Цепочка взаимодействий $A \rightarrow B \rightarrow C \rightarrow D$ возможна только при наличии совместимых форматов/протоколов на всех промежуточных этапах, либо адаптеров между ними. На рисунке 1 показана категориальная модель маршрута передачи данных для данного примера.

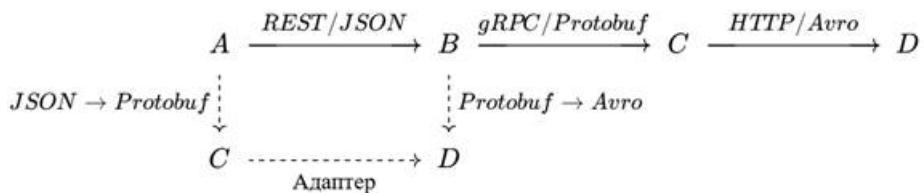


Рисунок 1 - Категориальная диаграмма, моделирующая маршрут передачи данных между микросервисами

Расширение модели за счёт внедрения категорий позволяет выявлять такие ситуации, когда если хотя бы на одном участке нет преобразователя, то путь (композиция морфизмов) не существует, и такой маршрут автоматически исключается из множества допустимых интеграций, что позволяет автоматически контролировать процесс разрешимости маршрутов интеграции, даже в случае их динамического изменения. Для этого определим базовую операцию согласования структуры данных и интерфейсов.

Пусть каждому микросервису $s_i \in V$ сопоставлено множество поддерживаемых им типов данных D_i множество реализованных интерфейсов I_i . Тогда общее множество всех интерфейсов в системе может быть определено по формуле

$$\mathcal{I} = \bigcup_{i=1}^n I_i \cdot \#(2)$$

Введём бинарную операцию (согласован или нет) согласования интерфейсов, которая может быть определена, как

$$\otimes: \mathcal{I} \times \mathcal{I} \rightarrow \mathcal{I} \cup \{\emptyset\}, \#(3)$$

Тогда для любых двух интерфейсов $I_i, I_j \in \mathcal{I}$ может быть определена операция

$$I_i \otimes I_j = \begin{cases} I_k, & \text{если } \exists C_{ij}: I_i \rightarrow I_j \\ \emptyset, & \text{если } \nexists C_{ij}, \end{cases} \quad (4)$$

где I_k - наименьший общий интерфейс,

C_{ij} - набор преобразований (конвертеров), обеспечивающий совместимость.

Определим ряд важных свойств операций согласования. Прежде всего стоит понимать, что не для всех пар интерфейсов, то есть на (\mathcal{I}, \otimes) существует частично определённая алгебраическая структура, аналогичная полугруппе с нулевым элементом (\emptyset). Это

означает, что если хотя бы на одном этапе согласования возникает несовместимость (то есть результатом является (\emptyset)), то при любой дальнейшей композиции результат также будет (\emptyset) . Для задачи обеспечения интероперабельности это будет означать, что если последовательное согласование интерфейсов в цепочке взаимодействия не удаётся хотя бы между одной парой компонентов, то передача данных по этому маршруту невозможна. Также операция согласования может обладать ассоциативностью, если если промежуточные интерфейсы существуют, а также если для цепочек интерфейсов результат не зависит от порядка выполнения преобразований, то есть конвертеры транзитивны

$$(I_i \otimes I_j) \otimes I_k = I_i \otimes (I_j \otimes I_k). \#(5)$$

Формула 5 фактически указывает на наличие в системе такого интерфейса, к которому, при наличии конвертеров между интерфейсами I_i и I_j а также при наличии конвертера между I_j и I_k , всегда можно согласовать любой другой интерфейс. В рамках данной работы данный интерфейс будет называться универсальным. Рассмотрим пример, пусть сервисы в микросервисной системе могут использовать разные форматы данных (JSON, XML, Protobuf), и между некоторыми парами форматов есть адаптеры. Если при этом для любого формата реализован переход к JSON и обратно (и эти переходы ассоциативны и транзитивны), то JSON будет являться универсальным интерфейсом для данной микросервисной системы. Тогда можно согласовать между собой любые два сервиса через промежуточный формат, не заботясь о прямых адаптерах между всеми возможными парами форматов, что упрощает интеграцию системы.

В условиях микросервисной архитектуры микросервисы обмениваются информацией, что можно описать как

$$m_{ij}^t: s_i \rightarrow s_j, \quad \forall (s_i, s_j) \in E \#(6)$$

где $m_{ij}^t \in M_{ij}$ - отображение сообщения M_{ij} из s_i в s_j .

Тогда каждый сервис может быть представлен как функция обработки поступающих сообщений

$$m_{ij}^t \rightarrow m_{jk}^{t+1}, \quad \forall s_k \in V, (s_j, s_k) \in E \#(7).$$

Способ организации интероперабельности программных компонентов

Формализованная выше модель интероперабельности, основанная на графовых, категориальных и алгебраических структурах, позволяет строго описывать и анализировать совместимость компонентов, а также выявлять ключевые требования к организации инфраструктуры. Полученные выше результаты прямо указывают на необходимость согласования интерфейсов сервисов на всех этапах передачи данных, а также обеспечения гибкости и расширяемости инфраструктуры, с целью оперативной адаптации к изменяющимся условиям эксплуатации и динамике потоков данных. Поэтому стоит более подробно рассмотреть механизм построения микросервисов и управления инфраструктурой.

Рассматривая построение микросервисов в облачной инфраструктуре, необходимо учитывать, что в большинстве платформ используются заранее собранные образы сервисов, хранящиеся в отдельном репозитории. Каждый сервис развёртывается на основе заранее подготовленного контейнерного образа, в который включаются все необходимые зависимости, конфигурации и описания интерфейсов. Контейнеры не

являются полноценными операционными системами, настроенными для общего использования. Скорее, это минимальные среды для запуска приложений [7]. Данные представлены в виде слоев, накладываемых друг на друга, что показано на рисунке 2. Такие «сборки слоев» хранятся в централизованном реестре контейнерных образов и интерфейсов (репозитории), а сборка таких образов для однотипных систем происходит по одинаковым конфигурационным файлам, что позволяет однозначно воспроизводить среду исполнения.

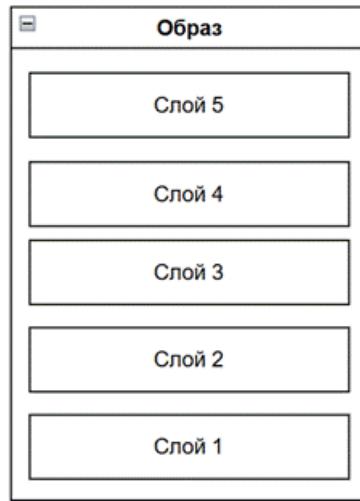


Рисунок 2 – Представление образа контейнера

Сформулированная ранее алгебраическая модель согласования интерфейсов реализуется в виде процедуры автоматической проверки маршрутов передачи данных. Пусть каждому контейнерному образу C_i сопоставлено множество поддерживаемых им интерфейсов $I(C_i)$. Тогда при развёртывании системы выполняется проверка

$$\forall (C_i, C_j) \in \text{DeploymentPlan}: \exists C_{ij}: I(C_i) \rightarrow I(C_j) \#(8)$$

где C_i, C_j - контейнерные образы (сервисы),

$I(C_i)$ - множество поддерживаемых интерфейсов образа C_i ,

C_{ij} - морфизм, обеспечивающий согласование интерфейсов между C_i и C_j .

Если хотя бы для одной пары (C_i, C_j) не существует C_{ij} , то маршрут деплоя считается неразрешимым и отклоняется системой.

Категорная модель (диаграмма передачи данных) непосредственно используется для автоматизированной проверки маршрутов передачи данных в момент развёртывания. Практически это реализуется за счёт описания контейнерного образа, в котором задаются формальные характеристики поддерживаемых интерфейсов, после чего категорные морфизмы служат алгоритмической основой для проверки того, что в цепочке взаимодействий все необходимые конвертеры и адаптеры существуют. Если такой морфизм отсутствует, автоматизированная система отказывает в развёртывании соответствующего маршрута.

Для сборки образов предполагается использовать программное обеспечение Docker. Оно позволяет производить такую сборку удалённо. Для таких операций используется встроенная утилита «демона», способная выполнить сборку за короткое время. Далее

для развертывания необходимо непосредственно в облачной инфраструктуре указать необходимый репозиторий образов, в котором находится слепок созданного сервиса и выполнить развертывание данного образа непосредственно в облачной инфраструктуре. В качестве такой инфраструктуры стоит использовать кластер Kubernetes. Он является основой крупнейших облачных инфраструктур [8]. Алгебраическая операция согласования интерфейсов реализуется как программный сервис, который получает на вход описания интерфейсов каждого контейнера и, используя матрицу совместимости, автоматически определяет необходимость в адаптерах. Это позволяет системе Kubernetes перед развертыванием проверять совместимость интерфейсов через специализированный API.

Развёртывание целесообразно проводить путём взаимодействия с целевым API облачной инфраструктуры. Такое взаимодействие будет наиболее быстрым, по сравнению с вариантами ручного развёртывания и развёртывания в режиме, когда все команды выполняются в пользовательских утилитах автоматически.

На рисунке 3 представлена диаграмма взаимодействия компонентов системы непосредственно между собой. Данные типы взаимодействия выбраны с учётом особенностей построения и сопровождения подобных систем и описанной выше архитектурой.

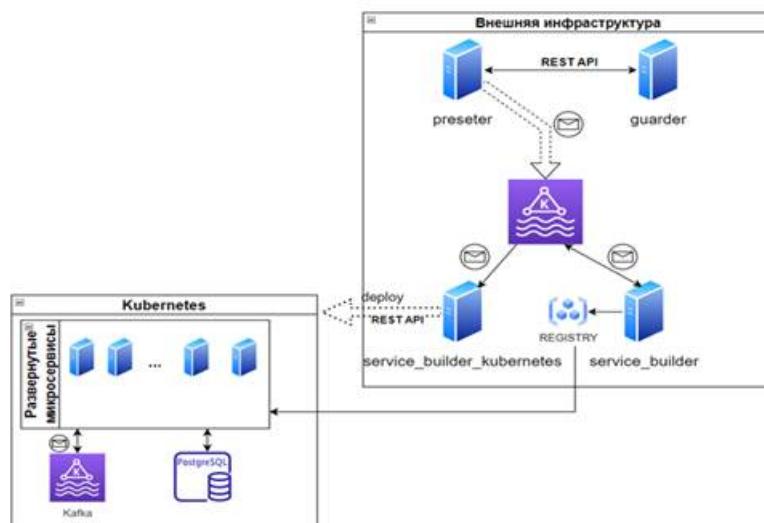


Рисунок 3 - Схема взаимодействия компонентов системы

Разработанный выше способ организации интероперабельности компонентов обеспечивает не только автоматизированную проверку совместимости и согласования интерфейсов между микросервисами, но и служит основой для оперативного управления и мониторинга инфраструктуры в условиях изменяющейся нагрузки и динамического окружения. Для обеспечения надёжного функционирования и оперативной адаптации к изменениям в условиях эксплуатации требуется постоянный мониторинг состояния сервисов и прогнозирование будущих нагрузок. Такой подход позволит не просто своевременно реагировать на возможные проблемы, но и заранее перераспределять ресурсы между микросервисами, минимизируя время простоя и повышая общую эффективность работы системы. Далее рассмотрим метод прогнозирования нагрузки на компоненты инфраструктуры.

Метод прогнозирования нагрузки

Стоит рассмотреть метод прогнозирования нагрузки на компоненты системы

интероперабельности программных компонентов. Ключевой потенциал систем анализа данных кроется в извлечении полезных выводов из данных. Цель данного процесса состоит в том, чтобы ответить на интересные (а иногда и трудные) вопросы, используя технологии анализа данных, и достичь более полного понимания конкретной предметной области [9]. Рассмотрение такого метода стоит начать с описания алгоритма сбора метрик сервисов и алгоритма прогнозирования нагрузки.

Рассматривая алгоритм сбора метрик программных компонентов, стоит обратить внимание на то, что он должен быть адаптирован для работы с различным количеством микросервисов. Таким образом, разработанный алгоритм сбора метрик выполняет периодический сбор показателей каждого микросервиса. Разработанная блок-схема его работы представлена на рисунке 4.

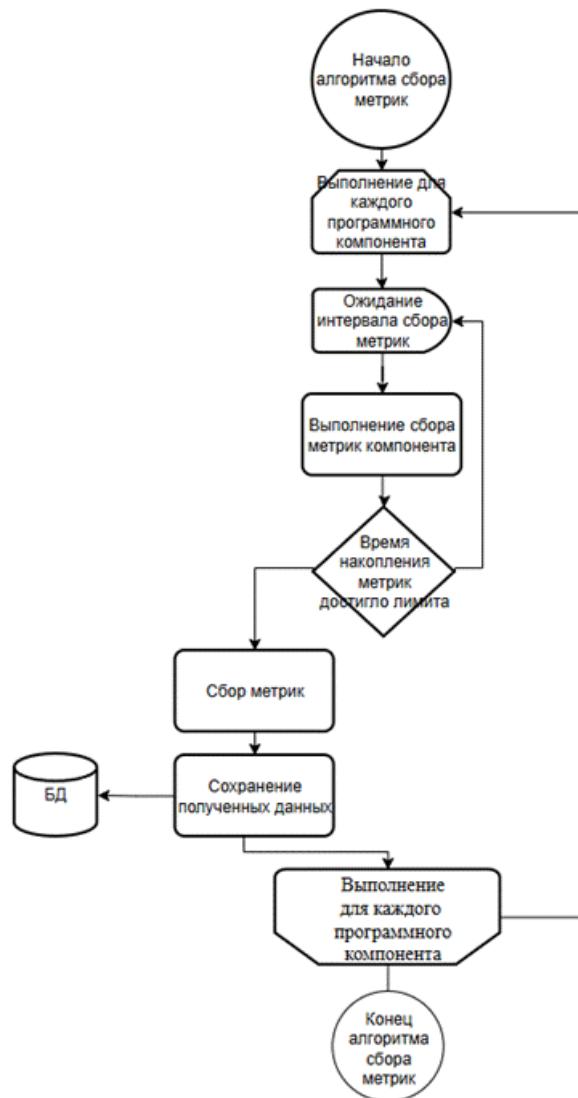


Рисунок 4 – Блок-схема алгоритма сбора метрик

Стоит сказать, что программные компоненты представляют из себя микросервисы, развёрнутые в облачной инфраструктуре и сопровождаемые самой системой. Поэтому для прогнозирования можно использовать следующий алгоритм, представленный на рисунке 5. Такой алгоритм обладает возможностью работы для множества развёрнутых и контролируемых системой интероперабельности программных компонентов.



Рисунок 5 – Блок-схема алгоритма прогнозирования

Из представленного описания алгоритма прогнозирования можно выбрать программные средства для его осуществления. Для реализации такого сервиса наилучшим образом подойдет язык программирования Python. Он содержит все необходимые библиотеки для работы моделей искусственного интеллекта, а также позволяет выполнять преобразование поступаемой информации в агрегаты.

Рассмотрим сам сервис запуска моделей. Его детальная архитектура представлена на рисунке 3. Можно увидеть, что такой сервис состоит из нескольких внутренних компонентов, каждый из которых отвечает за свою часть.

Следует детально рассмотреть работу компонента выполнения моделей. Он отвечает за запуск и функционирование нейронных сетей. Такой модуль выполняет загрузку нейронной сети из базы данных или внутреннего хранилища и запускает её с полученными на вход данными. В результате работы моделей компонент получает результат прогнозирования, который в дальнейшем может быть обработан.

Пример кода выгрузки модели из базы данных показан на листинге 1. Представленная функция организует загрузку нейронной сети из базы данных, а также загружает

созданный заранее весы для слоёв сети.

Листинг 1 -

```
def load_model_from_db(model_name):
    global model, tokenizer, model_params

    try:
        conn = psycopg2.connect(**DB_CONFIG)
        cursor = conn.cursor()

        # Загружаем модель
        cursor.execute("SELECT model_data FROM ai_models WHERE model_name = '" + model_name + "' AND model_active = True")

        model_data = cursor.fetchone()[0]
        model = load_model(io.BytesIO(model_data))

        # Загружаем параметры
        cursor.execute("SELECT params FROM ai_model_params WHERE model_name = '" + model_name + "'")
        model_params = cursor.fetchone()[0]

        cursor.close()
        conn.close()

        print("Модель загружена из БД")
    except Exception as e:
        print(f"Ошибка загрузки модели из БД: {str(e)}")
        raise e
```

Выгрузка нейронной сети выполняется стандартными методами доступа до хранилища данных. Стоит заметить, что сами нейронные сети, а также их параметры хранятся в реляционной базе данных, что позволяет хранить множество моделей в структурированном виде.

Библиотеки языка программирования Python позволяют выполнить инициализацию самой модели в несколько строк кода. Это особенно удобно при разработке сервиса, в котором необходимо выполнять загрузку моделей.

Стоит рассмотреть и процесс работы самой модели. Функция генерации прогнозирования представлена на листинге 2.

Листинг 2

```
def predict():
    global prediction_buffer
```

```

# Получаем данные из POST запроса

data = request.get_json()

# Проверяем обязательные поля

if not all(k in data for k in ['cpu', 'ram', 'timestamp']):

    return jsonify({'error': 'Missing required parameters (cpu, ram, timestamp)'}), 400

try:

    cpu = float(data['cpu'])

    ram = float(data['ram'])

    timestamp = data['timestamp']

    combined_load = 0.6 * cpu + 0.4 * ram

    # Добавляем в буфер

    prediction_buffer.append(combined_load)

    # Если буфер заполнен, делаем предсказание

    if len(prediction_buffer) >= SEQ_LENGTH:

        sequence = prediction_buffer[-SEQ_LENGTH:]

        sequence_scaled = scaler.transform(np.array(sequence).reshape(-1, 1))

        X = sequence_scaled.reshape(1, SEQ_LENGTH, 1)

        predicted_load_scaled = model.predict(X)

        predicted_load = scaler.inverse_transform(predicted_load_scaled)[0][0]

        if len(prediction_buffer) > SEQ_LENGTH * 2:

            prediction_buffer = prediction_buffer[-SEQ_LENGTH:]

        return jsonify({

            'timestamp': timestamp,

            'current_load': combined_load,

            'predicted_load': float(predicted_load),

            'message': 'Success'

        })

    else:

        return jsonify({
            'timestamp': timestamp,

```

```

'current_load': combined_load,
'predicted_load': None,
'message': f'Collecting data, need {SEQ_LENGTH - len(prediction_buffer)} more samples'
})
except Exception as e:
    return jsonify({'error': str(e)}), 500

```

Видно, что данная функция получает объект с данными и на основании него выполняет запуск модели и получение результатов прогнозирования для полученных данных – метрик, собранных при работе сервисов.

Затем результат прогнозирования проходит интерпретацию – корректировку результата и перевод её в значения необходимости корректировки, а затем отправляется в виде сообщений в нужный топик шине.

Стоит также обсудить и само построение модели. В предлагаемой системе наилучшим образом подходит использование LSTM нейронных сетей. Они позволяют проводить ретроспективный анализ работы компонентов с возможностью отслеживания периодов однообразности, поступающей на развёрнутые системой интероперабельности программные компоненты нагрузки. Построение модели происходит путём проведения её сборки и обучения в Jupiter ноутбуках. Это веб-приложение с открытым исходным кодом, которое позволяет создавать и обмениваться документами, содержащими живой код, уравнения, визуализацию и разметку [\[10\]](#).

Рассматривая процесс построения и обучения нейронной сети, нужно обратить внимание, что существующие библиотеки для Python, такие как tensorflow и sklearn позволяют выполнять построение любых моделей искусственного интеллекта с применением минимального количества кода. Также подразумевается, что система интеллектуального управления нагрузкой будет использовать различные нейронные сети для анализа большого количества метрик, для обеспечения возможности наиболее точного прогнозирования нагрузки.

Для примера спроектирована и разработана модель, позволяющая прогнозировать необходимость изменения состояния программных компонентов в части изменения выделенных квот CPU и памяти. Пример части кода по построению и обучению такой модели приведён на листинге 3.

Листинг 3

```

def build_lstm_model(input_shape, received_data):
    model = Sequential([
        LSTM(64, return_sequences=True, input_shape=input_shape),
        Dropout(0.2),
        LSTM(32, return_sequences=False),
        Dropout(0.2),

```

```
Dense(16, activation='relu'),  
Dense(1)  
])  
  
optimizer = Adam(learning_rate=0.001)  
  
model.compile(optimizer=optimizer, loss='mse')  
  
return model  
  
SEQ_LENGTH = 60  
  
TEST_SIZE = 0.2  
  
BATCH_SIZE = 32  
  
EPOCHS = 100  
  
data, scaler = load_and_preprocess_data(received_data)  
  
train_size = int(len(data) * (1 - TEST_SIZE))  
  
train_data = data['Load_scaled'].values[:train_size]  
  
test_data = data['Load_scaled'].values[train_size:]  
  
X_train, y_train = create_sequences(train_data, SEQ_LENGTH)  
  
X_test, y_test = create_sequences(test_data, SEQ_LENGTH)  
  
model = build_lstm_model((X_train.shape[1], X_train.shape[2]))  
  
print(model.summary())  
  
early_stopping = EarlyStopping(monitor='val_loss', patience=10,  
restore_best_weights=True)  
  
history = model.fit(  
    X_train, y_train,  
    batch_size=BATCH_SIZE,  
    epochs=EPOCHS,  
    validation_data=(X_test, y_test),  
    callbacks=[early_stopping],  
    verbose=1  
)  
  
train_predict = model.predict(X_train)  
  
test_predict = model.predict(X_test)  
  
train_predict = scaler.inverse_transform(train_predict)
```

```

y_train = scaler.inverse_transform([y_train])

test_predict = scaler.inverse_transform(test_predict)

y_test = scaler.inverse_transform([y_test])

train_rmse = np.sqrt(mean_squared_error(y_train[0], train_predict[:,0]))

test_rmse = np.sqrt(mean_squared_error(y_test[0], test_predict[:,0]))

print(f'Train RMSE: {train_rmse:.2f}')

print(f'Test RMSE: {test_rmse:.2f}')

plt.figure(figsize=(15, 6))

plt.plot(data['Timestamp'], data['Load'], label='Actual Load')

train_predict_plot = np.empty_like(data['Load'])

train_predict_plot[:] = np.nan

train_predict_plot[SEQ_LENGTH:SEQ_LENGTH+len(train_predict)] = train_predict[:,0]

test_predict_plot = np.empty_like(data['Load'])

test_predict_plot[:] = np.nan

test_predict_plot[train_size+SEQ_LENGTH:train_size+SEQ_LENGTH+len(test_predict)] = test_predict[:,0]

plt.plot(data['Timestamp'], train_predict_plot, label='Train Predict')

plt.plot(data['Timestamp'], test_predict_plot, label='Test Predict')

plt.legend()

plt.title('CPU and RAM Load Prediction')

plt.xlabel('Time')

plt.ylabel('Combined Load')

plt.show()

```

Таким образом, предложенный подход организации интероперабельности и модель прогнозирования нагрузки обеспечивают возможность не только оперативного контроля текущего состояния инфраструктуры, но и её заблаговременного адаптивного изменения. Однако для практической реализации этих преимуществ необходима систематическая и надёжная процедура сбора, хранения и обработки текущих данных о состоянии микросервисов, в том числе в части для возможности дальнейшего ретроспективного анализа работы сетей и возможности корректировки используемых нейронных сетей.

Организация мониторинга, прогнозирования и управления состоянием компонентов облачной инфраструктуры

Как было отмечено выше для реализации возможности управления состоянием

программных компонентов необходимо обеспечить сбор метрик с создаваемых системой интероперабельности программных компонентов. Это можно реализовать с использованием ПО Prometheus. Получаемые метрики стоит сохранять в хранилище, которое имеет возможность компактного хранения больших объёмов данных.

Также стоит позаботиться о вопросе транспортировки собранных метрик до хранилища. Наиболее простой способ передачи данных – центральная шина. Такой подход позволит унифицировать передачу различных метрик в общих сообщениях. Также использование шины позволит гарантировать доставку метрик до хранилища и сохранение их за счёт использования надёжных брокеров сообщений, таких как Apache Kafka.

Для прогнозирования нагрузки необходимо использовать отдельный сервис, который позволяет загрузить нейронную сеть и её веса и произвести прогнозирование. Стоит предусмотреть возможность постоянной работы некоторых сетей, путём хранения их в краткосрочной памяти непосредственно внутри сервиса для ускорения работы сервиса по прогнозированию нагрузки.

Результаты работы такого сервиса необходимо сохранять и валидировать. Например, не должно быть ситуации необходимости уменьшения ресурсов меньше минимального значения для таких сервисов. Стоит также применять механизмы авторизации и аутентификации при работе такого сервиса для исключения возможности несанкционированного управления состоянием программных компонент. Полученные же прогнозные значения нагрузки от нейросетевой модели используются системой оркестрации Kubernetes для автоматического масштабирования сервисов. Так, при превышении прогнозного порога по нагрузке, Kubernetes автоматически увеличивает количество реплик сервиса, что предварительно описывается через правила горизонтального масштабирования (Horizontal Pod Autoscaler), опирающиеся на результаты работы нейросетевых моделей.

Стоит более детально рассмотреть предлагаемую архитектуру сервисов получения метрик, прогнозирования и управления облачной инфраструктурой. Такая архитектура приведена на рисунках 6 и 7.

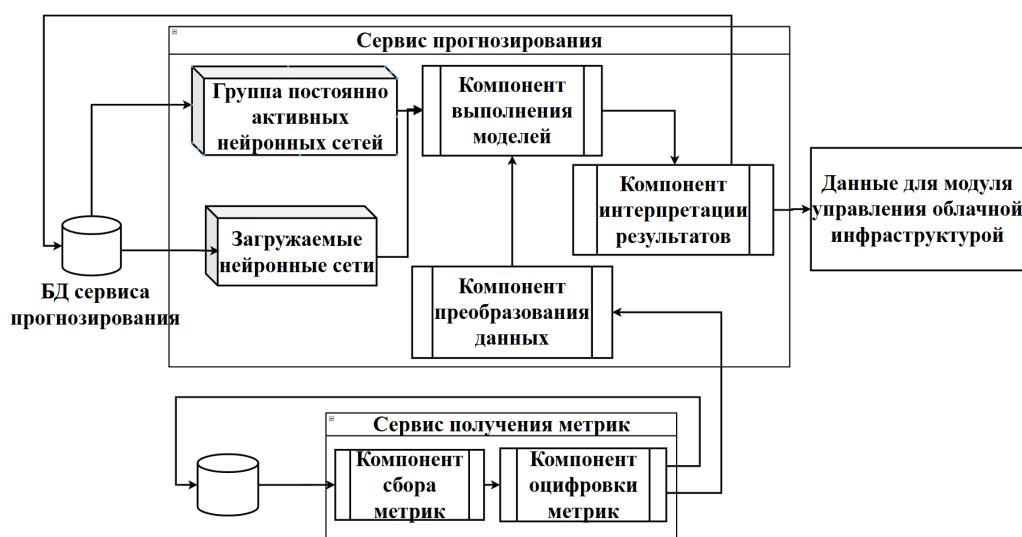


Рисунок 6 – Архитектура сервисов прогнозирования и получения метрик системы

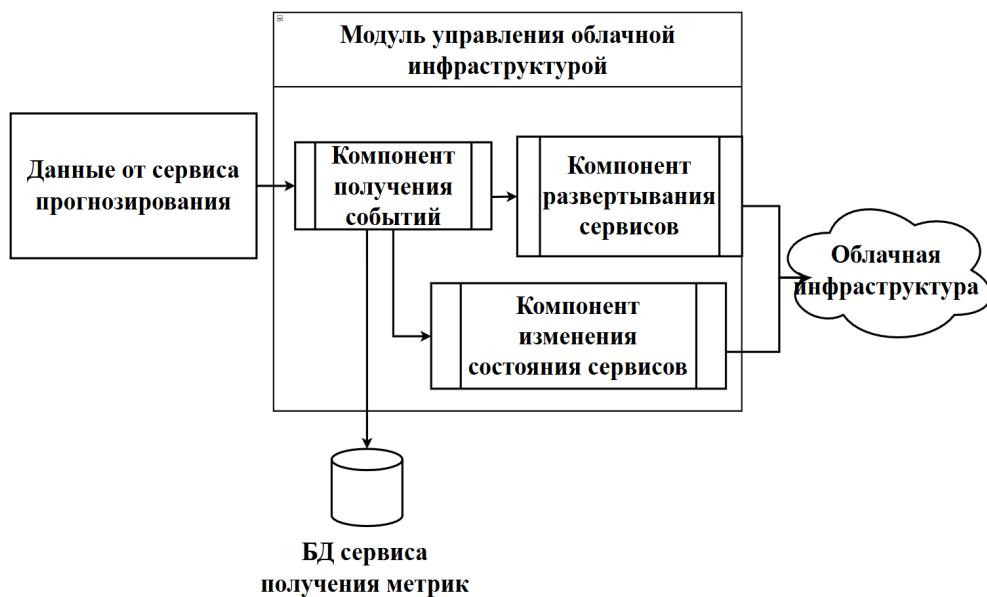


Рисунок 7 – Архитектура модуля управления облачной инфраструктурой

Можно видеть, что каждый сервис состоит из нескольких активных компонентов, совместно выполняющих работу. Также можно видеть, что каждый сервис взаимодействует со своей базой данных, при этом межсервисное взаимодействие должно осуществляться при помощи общей шины для обеспечения надёжности работы сервисов.

Заключение

Предложенное решение основывается на формализованном подходе к обеспечению интероперабельности микросервисных компонентов в облачной инфраструктуре. В качестве основы используется графовая и категорная модель, позволяющая строго определить маршруты передачи данных и процедуры согласования интерфейсов между различными сервисами. Для унификации взаимодействия и повышения гибкости системы введена операция согласования интерфейсов, а также реализована возможность автоматизированного выявления необходимости применения адаптеров и преобразователей данных. Такой подход обеспечивает формальное структурирование интеграционных процессов и способствует масштабируемости архитектуры. Контейнерная организация сервисов и централизованное управление образами позволяют автоматизировать этапы создания, публикации и развёртывания компонентов, обеспечивая воспроизводимость и высокую надёжность инфраструктуры. Особое внимание уделяется автоматической проверке совместимости интерфейсов на этапе деплоя, что исключает невозможные или неразрешимые маршруты передачи данных. Разработанный алгоритм интеллектуального прогнозирования нагрузки на сервисы позволяет динамически управлять состоянием компонентов и оперативно адаптировать инфраструктуру к изменяющимся условиям эксплуатации.

Рассмотренное решение предоставляет заявленный функционал, но имеет и возможности дальнейшей модернизации. Стоит рассмотреть возможность более гибкого управления и интерпретации метрик, которые анализирует система. Также стоит рассмотреть возможность ускорения обработки данных и работы моделей искусственного интеллекта за счёт применения технологий, обеспечивающих максимальное быстродействие.

Таким образом, итоговое решение представляет собой интеграцию формальных методов моделирования, автоматизации процессов управления микросервисами и

интеллектуального прогнозирования, что обеспечивает устойчивое функционирование и эффективную поддержку современных распределённых информационных систем.

Библиография

1. Paul A., Kelvin L., Brown K. Optimizing IT Growth: Strategies for Building and Scaling Robust Infrastructure Systems // Ladoke Akintola University of Technology. 2024. Т. 17. URL: <https://www.researchgate.net/publication/377447014>.
2. Макаренко С. И., Олейников А. Я., Черницкая Т. Е. Модели интероперабельности информационных систем // Системы управления, связи и безопасности. 2019. № 4. DOI: 10.24411/2410-9916-2019-10408 EDN: NACKGD.
3. Дьяков О. А., Солянов К. С. Формирование кредитного конвейера банка на основе систем бизнес-аналитики // Стратегии бизнеса. 2016. № 7 (27). С. 7-12. EDN: WWJUUB.
4. ISO/IEC 2382:2015 Information technology. Vocabulary. – 2015. – Текст: электронный. URL: <https://www.iso.org/ru/standard/63598.html>
5. Михневич С. Ю., Текар А. А. Эволюция понятия интероперабельности открытых информационных систем // Цифровая трансформация. 2023. Т. 29. № 2. С. 60-66. URL: <https://doi.org/10.35596/1729-7648-2023-29-2-60-66>. EDN: RKSXQW.
6. Ковалев С. П. Теоретико-категорный подход к проектированию программных систем // Фундаментальная и прикладная математика. 2014. Т. 19. № 3. С. 111-170.
7. Сейерс Э. Х., Милл А. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. Москва : ДМК Пресс, 2020. 516 с. ISBN 978-5-97060-772-5. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/131719> (дата обращения: 04.06.2025). Режим доступа: для авториз. пользователей.
8. Маркелов А. А. Введение в технологию контейнеров и Kubernetes / А. А. Маркелов. Москва : ДМК Пресс, 2019. 194 с. ISBN 978-5-97060-775-6. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/131702> (дата обращения: 04.06.2025). Режим доступа: для авториз. пользователей.
9. Бонцанини М. Анализ социальных медиа на Python. Извлекайте и анализируйте данные из всех уголков социальной паутины на Python / М. Бонцанини ; перевод с английского А. В. Логунова. Москва : ДМК Пресс, 2018. 288 с. ISBN 978-5-97060-574-5. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/108129> (дата обращения: 09.06.2025). Режим доступа: для авториз. пользователей.
10. Груздев А. В. Предварительная подготовка данных в Python / А. В. Груздев. Москва : ДМК Пресс, 2023. Том 1 : Инструменты и валидация. 2023. 816 с. ISBN 978-5-93700-156-6. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/314945> (дата обращения: 09.06.2025). Режим доступа: для авториз. пользователей.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Интеллектуальная инфраструктура автоматизированного управления и интероперабельности микросервисов в облачных средах» и посвящена актуальному вопросу разработки архитектуры интеллектуальной системы, способной обеспечить высокую степень интероперабельности программных компонентов, автоматическое управление состоянием сервисов и адаптацию к изменяющимся

условиям эксплуатации

Одним из перспективных подходов к решению данной проблемы выступает использование систем интероперабельности программных компонентов, которые обеспечивают автоматизацию процессов создания, развертывания и управления микросервисами, а также позволяют реализовать графические интерфейсы взаимодействия между ними. Такие системы обладают большим функционалом по созданию, развертыванию и управлению микросервисами, а также позволяют реализовать графические интерфейсы взаимодействия между ними, что способствует снижению эксплуатационных расходов и повышению устойчивости современных информационных систем.

В условиях возрастающей сложности и масштабности систем для ИТ-компаний становится крайне актуальной задача оптимизации затрат на построение и сопровождение микросервисной инфраструктуры, а также повышения эффективности процессов поддержки и интеграции. Авторами дано определение понятию «интероперабельности программных компонентов».

Авторами рассмотрен алгоритм сбора метрик программных компонентов. Разработанный алгоритм сбора метрик выполняет периодический сбор показателей каждого микросервиса. Разработанная блок-схема его работы представлена в графическом виде. Список литературы представлен российскими и зарубежными источниками по теме исследования. Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Практическая значимость статьи четко обоснована. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части (постановка задачи обеспечения интероперабельности программных компонентов, способ организации интероперабельности программных компонентов, метод прогнозирования нагрузки, организация мониторинга, прогнозирования и управления состоянием компонентов облачной инфраструктуры).

Авторами в заключительной части статьи обозначено, что предложенное решение основывается на формализованном подходе к обеспечению интероперабельности микросервисных компонентов в облачной инфраструктуре. В качестве основы используется графовая и категорная модель, позволяющая строго определить маршруты передачи данных и процедуры согласования интерфейсов между различными сервисами. Для унификации взаимодействия и повышения гибкости системы введена операция согласования интерфейсов, а также реализована возможность автоматизированного выявления необходимости применения адаптеров и преобразователей данных.

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна. Отсутствует четкое выделение предмета исследования.

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет. Также будет целесообразным добавить о перспективах дальнейшего исследования.

Статья «Интеллектуальная инфраструктура автоматизированного управления и интероперабельности микросервисов в облачных средах» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией рецензируемого научного журнала.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена актуальной проблеме обеспечения интероперабельности микросервисов в облачных средах с использованием интеллектуальных методов управления и прогнозирования нагрузки. Автор исследует методы формализации взаимодействия микросервисов, а также применение моделей искусственного интеллекта для оптимизации ресурсов и повышения устойчивости информационных систем.

В работе применён комплексный подход, сочетающий теоретическое моделирование и практическую реализацию. Для формализации интероперабельности использованы графовые и категорные модели, что позволяет строго описывать маршруты передачи данных и проверять совместимость интерфейсов. Алгебраические структуры, такие как полугруппы с нулевым элементом, применяются для анализа согласованности интерфейсов. Практическая часть включает разработку алгоритмов сбора метрик, прогнозирования нагрузки на основе LSTM-сетей и интеграцию с Kubernetes для автоматического масштабирования. Методология отличается высокой строгостью и воспроизводимостью.

Тема статьи исключительно актуальна в условиях роста сложности и масштабов информационных систем. Переход к микросервисной архитектуре требует решения задач интеграции, управления ресурсами и адаптации к динамическим изменениям нагрузки. Предложенные автором методы отвечают на эти вызовы, что делает исследование востребованным как в академической среде, так и среди ИТ-специалистов.

Научная новизна работы заключается в следующих аспектах:

1. Разработка категорной модели для описания взаимодействия микросервисов, включая анализ совместимости интерфейсов и автоматическое исключение недопустимых маршрутов.
2. Введение операции согласования интерфейсов с использованием универсального интерфейса, что упрощает интеграцию разнородных компонентов.
3. Применение LSTM-сетей для прогнозирования нагрузки и динамического управления ресурсами в облачной инфраструктуре.

Эти решения представляют значительный вклад в область распределённых систем и машинного обучения.

Статья написана ясным и логичным языком, с соблюдением академических норм. Структура работы хорошо продумана: от постановки проблемы и теоретического обоснования до практической реализации и визуализации результатов. Использование графиков, блок-схем и листингов кода enhances понимание предложенных методов. Библиография включает актуальные источники, что подчёркивает глубину проработки темы.

Автор предлагает комплексное решение, сочетающее формальные методы моделирования, автоматизацию управления микросервисами и интеллектуальное прогнозирование. Результаты работы имеют высокую практическую значимость, так как позволяют снизить эксплуатационные затраты и повысить устойчивость информационных систем. Статья демонстрирует завершённость исследования, но также указывает на направления для дальнейшей модернизации, такие как оптимизация обработки данных. Статья будет полезна исследователям в области распределённых систем, разработчикам облачных решений и специалистам по машинному обучению. Практические аспекты, такие как интеграция с Kubernetes и примеры кода на Python, делают материал ценным для ИТ-инженеров. Читательская аудитория получит не только теоретические знания, но и готовые инструменты для внедрения предложенных методов.

Статья соответствует высоким академическим стандартам, обладает значительной

научной и практической ценностью. Рекомендую её к публикации в ТОП-5 статей месяца издательства. Работа заслуживает положительной оценки благодаря оригинальности, глубине проработки и чёткому изложению. Учитывая актуальность темы и качество представленного материала, статья вызовет интерес у широкого круга специалистов и внесёт вклад в развитие области.

Программные системы и вычислительные методы

Правильная ссылка на статью:

Дагаев Д.В. Ареальные типы данных в инструментальном подходе к программированию // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.73776 EDN: FLOMPB URL: https://nbpublish.com/library_read_article.php?id=73776

Ареальные типы данных в инструментальном подходе к программированию

Дагаев Дмитрий Викторович

ORCID: 0000-0003-0343-3912

Генеральный директор, ООО "СКАДИ"

115230, Россия, г. Москва, Зеленый проспект, 5/12, строение 3, пом. 1

✉ dvdagaev@oberon.org



[Статья из рубрики "Языки программирования"](#)

DOI:

10.7256/2454-0714.2025.2.73776

EDN:

FLOMPB

Дата направления статьи в редакцию:

21-03-2025

Аннотация: Для ряда задач классическим методам структурного программирования отдается предпочтение перед методами объектно-ориентированного программирования. Эти предпочтения характерны для детерминированного мира и в системах, ориентированных на машинное представление. Для таких задач разрабатывался модульный язык программирования Оберон, имеющий минималистскую реализацию, существенно отличающуюся от большинства программных средств, стремящихся к максимизации числа поддерживаемых функций. Использование инструментального подхода вместо объектно-ориентированного предлагалось ранее для решения проблем детерминированного мира. Принцип сепарации кода от данных предполагает, что жизненный цикл данных является независимо управляемым, а время жизни превышает по длительности таковое для кода. Данная статья посвящена разработке и анализу ареальных типов данных в контексте инструментального программирования. Ареальные типы данных обеспечивают ортогональную персистентность и интегрируются с кодами, определенными в иерархии типов для инструментов. В качестве метода исследования интеграции ареальных типов используется разработка через тестирование,

реализованная автором в компиляторе МультиОберон. Методы тестирования включали в себя создание тестовых случаев для проверки метаданных ссылок при установлении ареалов, сохранении и восстановлении данных при рестартах ПО. Ареальные ссылки реализованы как персистентные ссылки в ареальном массиве. Благодаря такой организации данных решена проблема сохранения ссылок в структурах данных при завершении и восстановлении ПО. Приведены структуры данных со ссылками, остающимися консistentными при рестартах ПО. Новизна ареальных типов данных заключается в том, что используются механизмы установки ареалов в типах вместо создания объектно-ориентированных разветвленных структур или шаблонов обобщенных классов. Работа со ссылками сочетает преимущества индексов и указателей. Такой подход позволяет реализовывать алгоритмы обобщенного программирования без использования зависимостей по данным, как в части наследования, так в части шаблонов. Приведен пример обобщенного алгоритма сортировки для ареальных типов. Новый тип данных отличается компактностью и структурной простотой по сравнению с динамическими объектами. Преимуществами предложенного подхода являются персистентность, эргодичность, компактность, строгая типизация. В отличие от традиционных подходов объектно-ориентированного и обобщенного программирования ареальные типы данных являются персистентными, не требуют восстановления ссылок после рестарта и используют меньше ресурсов.

Ключевые слова:

ареальные типы, инструментальный подход, Оберон, семантические ограничения, ортогональная персистентность, компилятор, модульность, программирование управляемое данными, информатика-21, метаданные

Введение

С точки зрения теории объектно-ориентированного программирования ООП^[1] все пространство задач представляет собой области, где объектная парадигма и парадигма классического структурного программирования разделяются по осям «знания-социальный мир» и «реализация-детерминированный мир»^[2]. Для задач детерминированного мира характерен классический подход на основе алгоритмов и структур данных^[3]. Автором предложен и используется инструментальный подход к программированию^[4], который использует средства классического подхода, но вместо объектов ООП предлагает работу с инструментами. В отличие от объектов, инструменты не содержат в себе данных. Однако, инструменты могут быть ассоциированы с данными. Также, как и объекты, инструменты доступны по указателю, через который возможно реализовать наследование и позднее связывание. В отличие от объектов, каждый инструмент имеет фиксированное расположение в памяти, аллокирование инструментов невозможно, более того, для целей надежности аллокирование запрещается системой семантических ограничений языка МультиОберон^[5].

Инструментальный подход построения систем требует правильной организации данных. Он хорошо согласуется с data-центричным программированием, когда программы отделены от данных и связи между каждым экземпляром структуры данных и программой могут мгновенно и однозначно устанавливаться^[6]. Инструментальный подход содержит коды программ, но не содержит данных. Целью представленной работы является определение структур данных, требуемых для инструментального подхода.

Для реализации требований детерминированных систем 24x7 необходимо решение ряда задач в части структур данных. Во-первых, обеспечение персистентности, где время жизни данных может существенно превышать время жизни программ. Во-вторых, нужен ссылочный доступ к данным различных типов. В третьих, необходим эффективный доступ к данным из инструментов.

Предметом исследования статьи являются предлагаемые ниже ареальные структуры данных. В качестве метода исследования используется разработка через тестирование, реализованная автором в компиляторе МультиОберон.

Персистентность данных и ортогональная персистентность

В объектных системах возникает проблема обеспечения хранения и восстановления состояния. Она решается методом сериализации при сохранении и десериализации при восстановлении. Разработчики стоят перед необходимостью обеспечить корректную сериализацию и десериализацию даже при сбоях в работе ПО.

В data-центрических системах данные отделены от кода, время жизни данных существенно больше времени жизни кода. Данные сохраняются и восстанавливаются независимо от работы кода. Системы, отвечающие принципу «ортогональной персистентности» [7], имеют механизмы независимого сохранения состояния данных, например, как для Java-системы в [8] или в реализации на языке Motoko [9].

Задачей персистентности является максимально приблизить формат хранения данных к формату представления в памяти. Задачей ортогональности является реализация независимого от кода механизма преобразования из данных в файловой системе в данные в памяти.

Организация ссылок в данных требует использования указателей на динамически выделяемые структуры. Указатели являются временными адресами в оперативной памяти, не подлежащими сериализации. Восстановление ссылок требует очередного аллокирования и подмены адресов во всех точках со ссылками.

Для данных, представленных в виде массивов, возможно использование целочисленных индексов, но работа с индексами требует явного указания массива, к которому индекс относится.

Ареальные структуры данных

В данной статье предлагаются ареальные структуры данных – запись и ареальная ссылка. Ареальная запись означает тип данных запись, которая не может иметь указателей, но может иметь ссылки на ареальные объекты. Ареал означает определенное место обитания и представляет собой массив из ареальных объектов. Ареальная ссылка содержит не адрес, а индекс объекта в ареале, но, поскольку она имеет определенный тип, то ареальная ссылка путем тривиального преобразования разрешается как адрес ареальной записи.

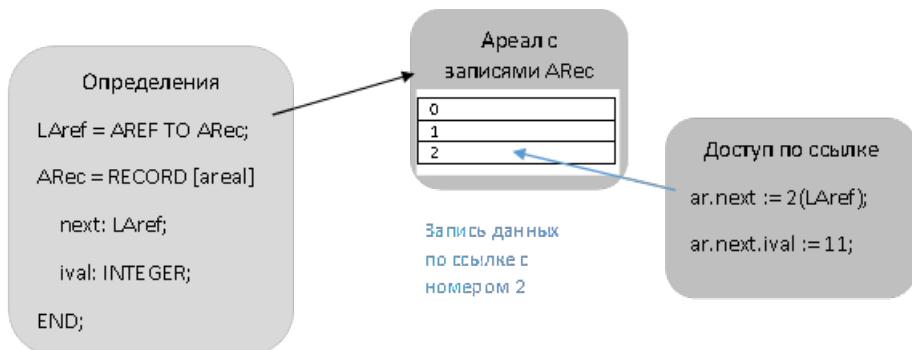


Рис. 1. Ареальная запись и ареальная ссылка

Fig.1 Areal record and areal reference

Ареальные структуры данных в паскале-подобной форме даются в нотации языка Оберон [\[10\]](#) и реализованы в системе МультиОберон. Запись ареальная отличается от стандартной записи наличием флага [areal]. Ареальная ссылка является ссылкой на объект ареальной записи.

```
LRef = AREF TO ARec;
ARec = RECORD [areal]
  next: LRef;
  ival: INTEGER;
END;
```

Тип ареальной ссылки аналогичен типу адреса структуры. Он представляет собой целочисленный тип длины адреса, привязанный к типу адресуемой структуры. Но значение ареального типа отличается. Если для указателя значения адреса инкрементируются размером указателя (4 для 32-битных систем и 8 для 64-битных), то для ареального типа значения ареальной ссылки могут принимать величины 0, 1, 2, 3, N, где N-размер выделенного массива под ареал.

Учитывая вышесказанное, установка значения ссылок аналогична установке значений целочисленных величин. Только необходимо правильное преобразование типов. Например, установка ссылке значения 2 будет выглядеть следующим образом.

```
ar.next := 2(LRef);
```

Нулевая ссылка представляет собой 0 и преобразуется в адрес элемента ареала с индексом 0. Поэтому чтение-запись по нулевой ареальной ссылке допустимы. Разумеется, нулевую ссылку можно использовать в алгоритмах специальным образом, как это применяется в указателях.

Доступ по ссылке синтаксически подобен доступу по указателям. В нашем примере можно присвоить элементу данных значение 11.

```
ar.next.ival := 11;
```

Семантически такой доступ означает присвоение 11 элементу массива ареала.

```
__areal_internal[(*index*)2].ival := 11;
```

По типу ссылки находится массив ареала `__areal_internal`, а по значению ссылки находится индекс элемента 2.

Установка массива ареала

Для ареального типа может быть установлен один и только один массив. Должна быть представлена информация о типе данных, адресе начала массива и числе элементов в нем. Встроенная функция AREAL устанавливает массив, связанный с ареальным типом.

```
AREAL(ARec, array[(*index*)0], LEN(array));
```

Значения адреса начала и длины массива записываются в метаданные указанного типа ARec.

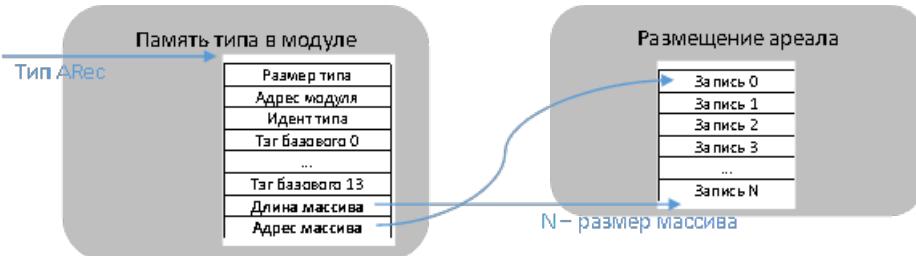


Рис. 2. Метаданные ареальной записи

Fig.2 Areal record metadata

В метаданных записи содержится массив указателей на базовые типы, элемент 14 используется для сохранения длины, а элемент 15 используется для сохранения адреса массива ареала.

Значение длины массива доступно через встроенную функцию LEN. Получение длины ареальной ссылки записывается следующим образом.

```
VAR ar: LAref;
```

```
I := LEN(ar);
```

Функция получения длины ареала зависит только от типа данных. Поэтому длина возвращается корректно для нулевой, для инициализированной и даже для неинициализированной ссылки.

Персистентность данных ареала

Данные ареала могут сохраняться без преобразования и без преобразования восстанавливаться. До сохранения ареальный массив находится в одном пространстве адресов процесса 1. А после восстановления ареальный массив находится в другом пространстве адресов процесса 2.

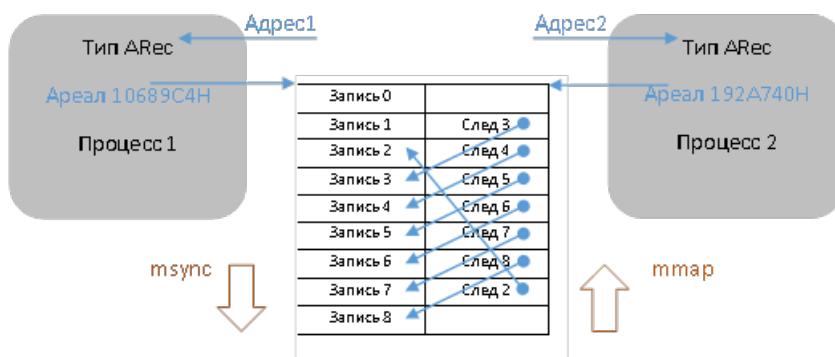


Рис. 3. Ареал в разных пространствах адресов

Fig.3 Areal in different address spaces

Сохранение/восстановление в файл без преобразования в POSIX-системах осуществляется вызовами `msync/mmap`. Если файл предоставляетя в общий доступ через разделяемую память, то процессы могут работать с данными файла одновременно. При очередной загрузке данных из файла необходимо устанавливать массив ареала. В нашем примере для первого процесса и для второго процесса это будут следующие выражения.

```
AREAL(ARec, 10689C4H, 9);
```

```
AREAL(ARec, 192A740H, 9);
```

Таким образом, ареалы типов могут быть поставлены в разделяемый разными процессами доступ. При этом структуры метаданных с типом ARec двух процессов расположены отдельно и загружаются каждый в свое адресное пространство (на рисунке 3 Адрес1 для 1 процесса и Адрес2 для второго). Изменения в ареале будут доступны всем процессам, изменения метаданных (например, уменьшение длины ареала) – будут доступны только текущему процессу.

Структура файлов данных ареала

Для загрузки массивов значений нескольких ареалов из файла используется следующий цифровой формат данных.

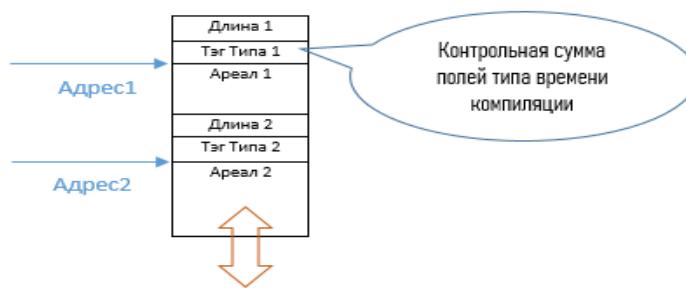


Рис. 4. Формат файла ареала

Fig.4 Areal file format

Каждая запись о массиве ареала предваряется длиной массива и тэгом типа Оберона [11]. Тэг типа представляет собой контрольную сумму из имени типа, имен и типов всех полей в порядке вхождения. Формируется во время компиляции. В структуре загруженного типа обязательно присутствует тэг. Если при загрузке данных обнаружено

несовпадение данных с тэгом типа, то данные считаются неконсистентными и генерится сообщение об ошибке. Длина ареала устанавливается $\langle\text{длина ареала}\rangle := \langle\text{длина}\rangle / \langle\text{размер типа}\rangle$.

После корректной загрузки данных все ареальные массивы получают тип и длину. Если массивы не инициализированы, длина ареала будет нулевой.

Трансформация отношений обобщения

Отношение обобщения в UML описывает иерархическое строение классов и наследование их свойств и поведения [12]. В диаграмме классов класс-потомок обладает всеми свойствами и поведением класса-предка, а также имеет свои собственные свойства и поведение, которые могут отсутствовать у класса-предка.

Примером реализации отношений может служить сортировка, алгоритм которой относится к общему элементу, а реализации сравнений и перестановки – к частному. При этом общие алгоритмы расширяются в частных реализациях.

Для случая ООП реализация основывается на включении алгоритмов сортировки в базовый класс, а сравнений и перестановок – в специфический [13]. По сравнению с ООП инструментальный подход использует как ареальные записи, так и базовые и расширенные инструменты.

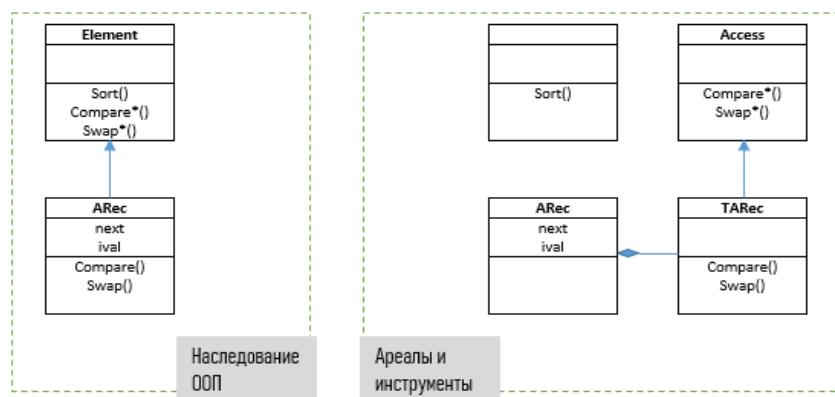


Рис. 5. Отношения обобщения

Fig.5 Generalization relationships

Базовый инструмент Access содержит определения методов доступа к сортируемым структурам данных. На уровне обобщения нам известно, что инструмент имеет две операции сравнения и перестановки.

Access = POINTER TO ABSTRACT RECORD [tool] END;

PROCEDURE (acc: Access) Compare (I1, I2: INTEGER): INTEGER, NEW, ABSTRACT;

PROCEDURE (acc: Access) Swap (I1, I2: INTEGER), NEW, ABSTRACT;

Обобщенный алгоритм сортировки требует наличия экземпляра базового инструмента acc, длины A_LEN и не требует никакой информации о типе данных реализации.

TOOL(acc);

FOR j := 1 TO A_LEN-1 DO

```

FOR k := A_LEN-1 TO j BY -1 DO
  IF acc.Compare(k-1, k) < 0 THEN
    acc.Swap(k-1, k)
  END
END
END;

```

Реализация массива основывается на ареальном типе данных ARec, описанном выше. Тип и длина должны быть установлены.

```

VAR arrARec = ARRAY A_LEN OF ARec;
AREAL(ARec, arrARec[(*index*)0], A_LEN);

```

Реализация методов сравнения и перестановки в специфическом ARecAccess инструменте привязана только к ареальным данным типа ARec.

```

ARecAccess = POINTER TO RECORD [tool] (Access) END;
PROCEDURE (acc: ARecAccess) Compare (l1, l2: INTEGER): INTEGER;

```

```

VAR lar1, lar2: LARef;

```

```

BEGIN

```

```

  lar1 := l1(LARef);

```

```

  lar2 := l2(LARef);

```

```

RETURN lar1.ival-lar2.ival;

```

```

END Compare;

```

```

PROCEDURE (acc: ARecAccess) Swap (l1, l2: INTEGER);

```

```

VAR lar1, lar2: LARef; x: ARec;

```

```

BEGIN

```

```

  lar1 := l1(LARef);

```

```

  lar2 := l2(LARef);

```

```

  x := lar1^;

```

```

  lar1^ := lar2^;

```

```

  lar2^ := x;

```

```

END Swap;

```

Следует отметить, что обобщенные алгоритмы приведенного вида не накладывают никаких ограничений на данные. Единственным исключением является то, что данные должны быть ареального типа. Это объясняется тем, что инструменты не содержать указателей на данные, и единственno, что можно извлечь из специфического

инструмента – это тип. И только для ареального типа можно получить массив значений и длины, с этим типом ассоциированные. Такой подход содержит меньше внутренних зависимостей, чем ООП^[14-15], накладывающие ограничения на необходимость наличия базового класса в сортируемых структурах данных.

Реализация в обобщенном программировании

Подход обобщенного программирования в классическом [\[16\]](#) и современном [\[17\]](#) вариантах приводит к разработке шаблонов обобщенных алгоритмов и реализации специфических алгоритмов для каждого типа данных. Аналогичный пример содержал бы шаблон алгоритма аналогичный приведенному.

```
template
```

```
E bsort(E a, int len) {
    for (int j = 1; j < len; j++) {
        for (int k = len-1; k >= j; k--)
            if (a[k-1] < a[k])
                std::swap(a[k-1], a[k]);
    }
}
```

Подход обобщенного программирования также требует необходимости определения операций сравнения и копирования для используемых типов. Синтаксис и семантика программирования с шаблонами существенно усложняет использование. Обобщенное программирование является избыточным, т.к., помимо обобщенного алгоритма в выходной программе содержится тот же алгоритм, реализованный для специфического типа.

Для более сложных структур данных (отображения, деревья, списки) работа в стиле обобщенного программирования требует использования итераторов. Такие структуры организуются в динамической памяти, и для эффективной навигации итераторы становятся необходимыми. Структуры обобщенного программирования не являются персистентными, для них необходимы сложные рекурсивные алгоритмы сериализации/десериализации.

Ареальные структуры данных используют тривиальное преобразование индекс – адрес для всех используемых структур.

Доступ к полям ареальных записей

Ареальная ссылка, имеющая доступ к внутренним полям ареального типа, должна быть определена в том же модуле, что и тип. Попытка сослаться на тип из другого модуля приводит к ошибке компиляции. При этом ареальная ссылка может не экспортirоваться, даже если экспортirуется сам тип. Поэтому можно возвращать значения элементов ареала без доступа к самому ареалу. Например, возвращаемый функцией GetVal элемент Value никаких ссылок на ареал не содержит.

```
AVal = AREF TO Value;
```

```
Value* = RECORD [areal]
```

```

rval*: REAL;
ival*: INTEGER;
END;

PROCEDURE GetVal*(select: INTEGER; VAR v: Value);

```

Приведенный подход имеет недостаток в том, что информацию об ареале нельзя извлечь из значения, но можно извлечь из типа Value. Поэтому при необходимости нужно разделять ареальный и возвращаемый типы.

Ареальные данные могут содержать как информацию, необходимую для пользователя, так и специфическую информацию, привязанную к реализации. В этом случае необходимо разбивать данные на базовые и производные типы, структурно отделяя передаваемую информацию от скрытой, рисунок 6.

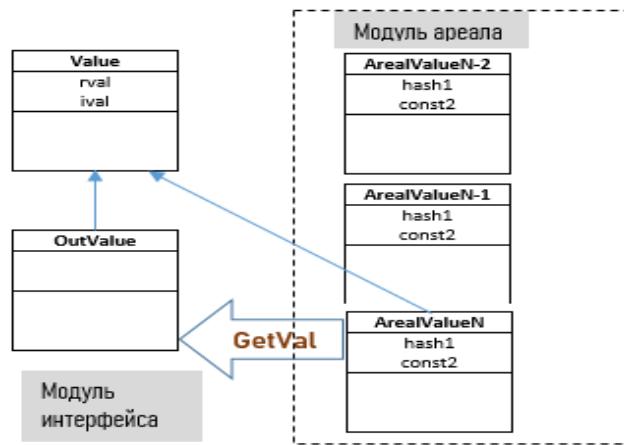


Рис. 6. Разделение типов и модулей

Fig.6 Modules and types separation

В приведенном примере Value является базовым типом, от которого наследуются пользовательский OutValue и скрытый ArealValue. Последний имеет дополнительные поля hash1, const2.

(* interface module *)

Value* = EXTENSIBLE RECORD

rval*: REAL;

ival*: INTEGER;

END;

OutValue* = RECORD (Value)

END;

(* areal module *)

AVal = AREF TO ArealValue;

ArealValue = RECORD [areal] (I.Value)

```

hash1*: INTEGER;
const2*: REAL;
END;

PROCEDURE GetVal*(select: INTEGER; VAR v: I.Value);

```

При запросе данные из ArealValue, относящиеся к базовому типу Value, копируются функцией GetVal в подаваемую на вход структуру OutValue. В данном случае возвращаемый тип OutValue не имеет информации об ареальном типе ArealValue.

Семантические ограничения ареалов

Поскольку использование ареалов и инструментов является расширением языка Оберон, то включение механизмов с флагами [area], [tool] требует снятия семантического ограничения на инструменты. Это делается следующим оператором.

RESTRICT +AREAL, +TOOL

Следствием применения ареалов является запрет выделения и использования динамической памяти. В системе МультиОберон указанные запреты означают, что в модуле непосредственно функция NEW быть вызвана не может. Кроме этого, не используются указатели на выделенную память, а используются ареальные ссылки. Также запрещается использование системных преобразований и доступ к метаданным.

RESTRICT -NEW, -POINTER, -SYSTEM;

При наличии ключевых слов NEW, POINTER и SYSTEM проверка не будет пройдена и завершится синтаксической ошибкой.

Вопросы эргодичности

Деградации данных со временем, определяющей свойства эргодичности [\[18\]](#), в ареальных записях не происходит. Однако необходимо обеспечить неизменность набора данных в течение жизненного цикла системы.

В системах детерминированного мира выделяется избыточное количество данных: помимо необходимых на данный момент, выделяются резервные данные. При добавлении нового элемента в реальности происходит выбор резервного элемента.



Рис. 7. Добавление резервного элемента

Fig.7 Adding reserve element

Данные ареальных типов со временем не появляются, не исчезают, а только меняют свое состояние. Состояние, определяющее является ли указанные элемент резервным, являются важным и персистентным параметром^[19]. Система адресации записей не меняется по ходу работы. Следовательно, необходимо только обосновывать базовые алгоритмы в части правильности значений данных в ареальных типах.

Разделение кода и данных означает, что оценку вклада кода можно делать только в том случае, когда идет обновление данных.

Приведем доказательства эргодичности построенной на указанных принципах системы. В части управления памятью следует отметить, что динамические структуры данных не используются, а память ареальных массивов сохраняется на протяжении времени жизни данных. В части обеспечения реального времени инструментальный подход не вносит дополнительных конструкций. В части обработки исключительных ситуаций происходит завершение программного кода. При этом не требуется сериализация и десериализация при восстановлении данных. Ареальные записи могут передаваться как сообщения при сетевом обмене данных. При этом ареальные ссылки будут сохраняться. Инструментальный подход не влияет на управление временем.

Сравнение подходов

Ареальные типы инструментального подхода реализуют персистентность ссылок и сепарацию кода от данных. Решение реальных задач детерминированного приводит к построению связанных структур данных в компактном и возобновляемом состоянии. Современные объектные облачные системы из «социального мира» выстраивают достаточно сложные механизмы агрегирования объектов, их взаимодействия, децентрализации и механизмов диагностики и восстановления, распределения нагрузки и т.д.^[20]. При этом решение этими методами задач детерминированного мира приводит к огромной избыточности, потерям надежности во всех режимах, отсутствию эргодичности.

В части алгоритмов работы с данными инструментальный подход не накладывает на данные каких-либо ограничений (наследование от одного и того же типа, необходимость итераторов). Требования для выполнения операций с данными относятся к кодам инструментов. В части реализации стандартные алгоритмы библиотек шаблонов в большей степени интегрированы с оптимизирующими компиляторами.

Ареальные типы имеет смысл использовать в системах, где реализуется отказ от динамических структур данных и динамического выделения памяти в целом. Стандартные динамические типы данных обладают большей гибкостью и возможностью адаптации под изменяющиеся структуры предметной области.

Применение в ПО моделирующих систем

Ареальные типы данных использовались при разработке моделирующего комплекса ПО СКОЯТ, расшифровываемого как система комплексного обучения ядерным технологиям. Гибридная архитектура СКОЯТ состоит из облачных подсистем управления ресурсами (включая сервисы, контейнеры, распределение нагрузок) и подсистем СКОЯТ-МОДЕЛЬ, организованных на принципах детерминированного мира.

Каждая подсистема из набора моделей, запускаемая внутри контейнера, включает в себя совокупность взаимосвязанных ареальных структур данных.

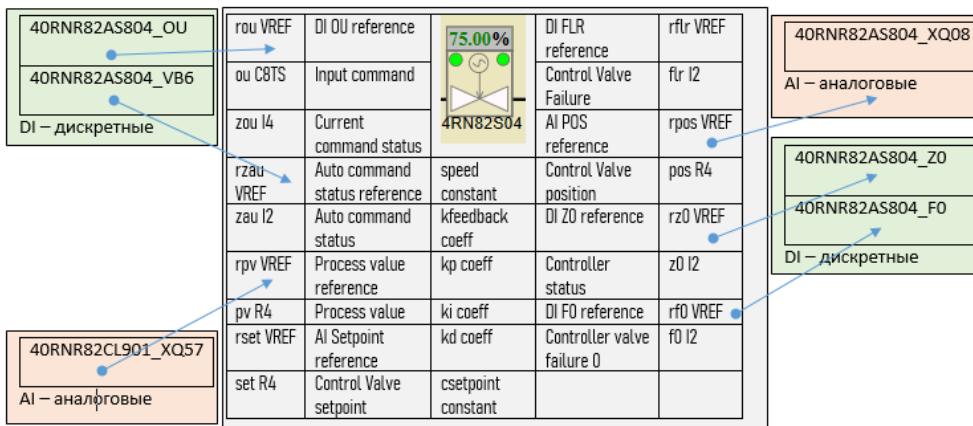


Рис. 8. Привязка данных регулятора СКОЯТ

Fig.8 SCONT controller data separation

Каждый регулятор представляет собой элемент структуры данных, имеющий идентификацию и привязки к другим элементам. Каждый элемент входных и выходных данных представляет собой поле ареальной записи со ссылкой на элемент таблицы аналоговых или дискретных параметров. Ссылки хранятся персистентно в том же виде, что и в памяти. Параметры, относящиеся к данному экземпляру оборудования, также являются персистентными.

Программные коды расположены в динамически загружаемом плагине обработки для объекта регулятора. Аналогично устроены коды и данные плагинов обработки объектов иных типов (задвижки, насосы, блокировки, и т.д.).

Заключение

Предложенные в данной статье ареальные типы данных позволяют организовать механизм ортогональной персистентности данных по отношению к кодам программ. Ареальные типы данных предназначены и используются в инструментальном подходе к программированию. Ареальные типы и данных и инструментальной подход реализуют инфраструктуры программирования управляемого данными. В рекомендуемом для систем детерминированного мира подходе используется сепарация кода от данных и персистентность данных.

Сохранение и восстановление ареальных данных не требует операций сериализации/десериализации. Ареальные данные хранятся в том виде, в котором они используются в памяти. Коды программ инструментального подхода не хранят внутри себя состояний. Следовательно, нет необходимости сохранять и восстанавливать что-нибудь в кодах, если данные используются персистентно.

Ареальные типы обеспечивают ссылочный типизированный доступ к данным. Массив ареальных данных существует в единственном экземпляре в приложении. Ссылка на элемент массива является ссылкой на ареальную запись. Ссылка содержит целочисленный индекс и однозначно идентифицирует конкретную ареальную запись при любых использованиях.

Принцип сепарации кода от данных предполагает использование инструментов с ареальными типами данных. Инструменты реализуют набор методов процедур без использования понятия экземпляров. Они могут быть привязаны к определенному типу данных.

Недостатком ареального типа данных являются ограниченность области применения, в части сравнения с ООП^[21]. Также недостатком является невозможность дублирования ареалов одного и того же типа, правда эта проблема решается на уровне базовых и производных типов, как показано выше.

Ареальные типы данных реализованы в системе МультиОберон. Ареальные типы данных интегрированы с инструментальным подходом. Реализован эффективный доступ из инструментов путем преобразования индекса в ареальную ссылку.

Ортогональная персистентность означает, что все значения полей структур данных обновляются, сохраняются и восстанавливаются в персистентной памяти. Такой подход обеспечивает надежную и безударную работу системы при сбоях и отказах в программных кодах^[22]. Такая организация данных отвечает требованиям эргодичности разрабатываемых систем. Программные системы, удовлетворяющие указанным требованиям, не деградируют со временем из-за изменения состояния внутренней памяти.

Применение в моделирующем комплексе системе обучения СКОЯТ реализовано с использованием ареальных структур данных. Решение об использование позволило не только апробировать новый метод, но и обеспечить большую понятность и консистентность структур данных и их связей на протяжении всего жизненного цикла разработки ПО.

Новизна ареальных типов данных заключается в том, что используются механизмы установки ареалов в типах вместо создания объектно-ориентированных разветвленных структур или шаблонов обобщенных классов. Такой подход позволяет сочетать преимущества индексов и указателей.

Значимость предложенного подхода заключается в изменении структуры программного обеспечения, более соответствующего задачам детерминированного мира. Преимуществами построения систем на основе инструментального подхода с ареальными типами являются персистентность, эргодичность, компактность, строгая типизация.

Библиография

1. Mansi Dhirajsinh Parmar, Sarthavi Parmar. Survey on Concept of Object-Oriented Programming. International Journal of Scientific Research in Computer Science Engineering and Information Technology. 2024. Vol. 10, No. 2. P. 427-431. DOI: 10.32628/CSEIT243647. EDN: AMCTWA.
2. West D. Object Thinking. Redmond: Microsoft Press, 2004.
3. Вирт Н. Алгоритмы и структуры данных. М.: ДМК-Пресс, 2016.
4. Дагаев Д.В. Инструментальный подход к программированию в системе МультиОберон // Программные системы и вычислительные методы. 2024. № 1. С. 31-47. DOI: 10.7256/2454-0714.2024.1.69437 EDN: WVZVVU URL: https://nbpublish.com/library_read_article.php?id=69437
5. Дагаев Д.В. Ограничительная семантика языка в системе МультиОберон // Программные системы и вычислительные методы. 2023. № 1. С. 26-41. DOI: 10.7256/2454-0714.2023.1.36217 EDN: IWIODR URL: https://nbpublish.com/library_read_article.php?id=36217
6. Rajive J., Ph.D. Data-Oriented Architecture: A Loosely-Coupled Real-Time SOA. Real-Time Innovations, Inc., 2007.
7. Hosking A., Nystrom N., Cutts Q., Brahnmath K. Optimizing the read and write barriers

- for orthogonal persistence // Advances in Persistent Object Systems / Morrison J., Atkinson M. (Eds.). San Francisco: Morgan Kaufmann, 1999. P. 11.
8. Lefort A. A Support for Persistent Memory in Java. Computer science. Institut Polytechnique de Paris, 2023. English. ffNNT : 2023IPPAS001.
9. Bläser L., Russo C., Greif G., Vandersmith R., Ibrahim J. Smarter Contract Upgrades with Orthogonal Persistence // Proceedings of the 16th ACM SIGPLAN International Workshop on Virtual Machines and Intermediate Languages (VMIL '24). Pasadena, CA, USA, 2024. P. 11.
10. Вирт Н., Гуткнхт Ю. Разработка операционной системы и компилятора. Проект Оберон. М.: ДМК-Пресс, 2015.
11. Crelier R. OP2: A Portable Oberon Compiler. ETH, Departement Informatik, 1990. P. 32-34.
12. Леоненков А.В. Самоучитель UML. 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2006. Р. 150.
13. Гамма Э., Хэлм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001.
14. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. СПб.: Питер, 2022.
15. Мартин Р. Идеальный программист. Как стать профессионалом разработки ПО. СПб.: Питер, 2022.
16. Musser D.R., Stepanov A.A. Generic Programming // Proceedings of the International Symposium ISSAC'88 on Symbolic and Algebraic Computation. New York: Springer-Verlag, 1988. P. 13-25.
17. Roynard M. Generic programming in modern C++ for Image Processing. Signal and Image Processing. Sorbonne Université, 2022. English. ffNNT : 2022SORUS287ff.
18. Дагаев Д.В. О разработке Оберон-системы с заданными свойствами эргодичности // Труды ИСП РАН. 2020. Т. 32, вып. 6. С. 67-78. DOI: 10.15514/ISPRAS-2020-32(6)-5. EDN: HBGGIK.
19. Дагаев Д.В. Исполняющая машина автоматных программ // Научно-технический вестник информационных технологий, механики и оптики. 2021. Т. 21, № 4. С. 525-534. DOI: 10.17586/2226-1494-2021-21-4-525-534. EDN: MJZXQZ.
20. Lo S.K., Lu Q., Zhu L., Paik H.-Y., Xu X., Wang C. Architectural patterns for the design of federated learning systems // Journal of Systems and Software. 2022. Vol. 191. P. 357.
21. Ali H.M., Hamza M.Y., Rashid T.A. Exploring Polymorphism: Flexibility and Code Reusability in Object-Oriented Programming // Proceedings of the 4th International Conference on Recent Innovation in Engineering (ICRIE 2023). University of Duhok, College of Engineering, 13th-14th September 2023. Paper No. 33.
22. Khattak U.F., Hussein H.A. A review on graph representation for object-oriented programming // BIO Web of Conferences. 2024. Vol. 97. Article 00131. DOI: 10.1051/bioconf/20249700131. EDN: YIHEDW.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Предметом исследования в рецензируемой работе выступают Ареальные типы данных, рассматриваемые в инструментальном подходе к программированию.

Методология исследования базируется на инструментальном подходе к построению систем и программированию, использовании языка Оберон и системы МультиОберон. Актуальность темы авторы работы связывают с тем, что в системах детерминированного

мира, где время жизни данных может существенно превышать время жизни программ, важно реализовать персистентность данных и эффективный доступ к данным различных типов.

Научная новизна рецензируемого исследования, к сожалению в статье четко не сформулирована авторами.

В тексте статьи выделены следующие разделы: Введение, Персистентность данных и ортогональная персистентность, Ареальные структуры данных, Установка массива ареала, Персистентность данных ареала, Структура файлов данных ареала, Трансформация отношений обобщения, Реализация в обобщенном программировании, Доступ к полям ареальных записей, Семантические ограничения ареалов, Вопросы эргодичности, Сравнение подходов, Применение в ПО моделирующих систем, Заключение и Библиография.

В статье изложено авторское видение ареальных структур данных, при подаче материала широко использован графический способ – текст иллюстрирован 8 рисунками, кроме этого приводятся фрагменты программного кода, проведено сравнение объектно-ориентированного и инструментального подходов в программировании, отмечено, что ареальные типы имеет смысл использовать в системах, где реализуется отказ от динамических структур данных и динамического выделения памяти в целом.

Библиографический список включает 16 источников – научные публикации по рассматриваемой теме в российских и зарубежных журналах на русском и английском языках. В тексте приведены адресные ссылки, что подтверждает наличие апелляции к оппонентам.

Из замечаний стоит отметить следующие. Во-первых, во введении не отражены такие элементы методологического аппарата любого научного исследования как цель и задачи, предмет и объект, цель, описание исходных материалов и методов, с помощью которых они преобразованы в результаты, новизна и значимость полученных результатов. Во-вторых, в тексте употребляются аббревиатуры без их расшифровки, например, ООП – представляется, что среди читателей могут быть и те, кому будет непонятно, что речь идет об объектно-ориентированном программировании. В-третьих, уместно расширить список источников до рекомендуемого издательством количества – не менее 20 источников, не менее половины работ, изданных в последние 3 года. В-четвертых, некоторые фрагменты текста статьи уже были опубликованы автором ранее, например, значительная часть введения повторяет введение статьи Дагаев Д. В. Инструментальный подход к программированию в системе МультиОберон / Программные системы и вычислительные методы. 2024. № 1. DOI: 10.7256/2454-0714.2024.1.69437 EDN: WVZVVU URL: https://nbpublish.com/library_read_article.php?id=69437.

Рецензируемый материал соответствует направлению журнала «Программные системы и вычислительные методы», отражает результаты проведенной авторами работы, может вызвать интерес у читателей, но нуждается в доработке перед опубликованием.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Ареальные типы данных в инструментальном подходе к программированию» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена актуальному вопросу – применению инструментального подхода к программированию с использованием средств классического подхода, но вместо объектов – объектно-ориентированного

программирования осуществлению работы с инструментами.

В статье четко обозначена цель исследования – определение структур данных, требуемых для инструментального подхода.

В статье авторы указывают в качестве предмета исследования предлагаемые авторские ареальные структуры данных. В качестве метода исследования авторы используют разработку через тестирование, реализованную автором в компиляторе МультиОберон. Также в статье указана теоретико-методологическая основа исследования, а именно авторы ссылаются на работы Hosking A., Nystrom N., Cutts Q., Brahnmath K. (в работе рассматриваются вопросы системы, отвечающие принципу «ортогональной персистентности»), Lefort A. (рассматривает вопрос механизмов независимого сохранения состояния данных, например, как для Java-системы).

Авторы в данной статье предлагают ареальные структуры данных – запись и ареальная ссылка. В статье результаты исследования авторы представляют в графическом виде.

В списке литературы имеются российские и зарубежные источники по теме исследования. Стиль и язык изложения материала является научным и доступным для широкого круга читателей. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Статья достаточно структурирована – в наличии введение, заключение, внутреннее членение основной части (в статье рассмотрены следующие вопросы: персистентность данных и ортогональная персистентность, ареальные структуры данных, установка массива ареала, персистентность данных ареала, структура файлов данных ареала, трансформация отношений обобщения, реализация в обобщенном программировании, доступ к полям ареальных записей, семантические ограничения ареалов, вопросы эргодичности, применение в ПО моделирующих систем).

Практическая значимость статьи четко обоснована. Значимость предложенного авторами подхода заключается в изменении структуры программного обеспечения, более соответствующего задачам детерминированного мира. Преимуществами построения систем на основе инструментального подхода с ареальными типами являются персистентность, эргодичность, компактность, строгая типизация.

В заключении авторы акцентируют внимание на том, что предложенные в данной статье ареальные типы данных позволяют организовать механизм ортогональной персистентности данных по отношению к кодам программ. Ареальные типы данных предназначены и используются в инструментальном подходе к программированию. Ареальные типы и данных и инструментальной подход реализуют инфраструктуры программирования, управляемого данными. В рекомендуемом авторами для систем детерминированного мира подходе используется сепарация кода от данных и персистентность данных.

Статья «Ареальные типы данных в инструментальном подходе к программированию» может быть рекомендована к публикации в журнале «Программные системы и вычислительные методы».

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Макаров И.С., Райков А.В., Казанцев А.А., Нехаев М.В., Романов М.А. Применение нейросетей для анализа больших данных в реальном времени // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.73651 EDN: DUSRKQ URL: https://nbpublish.com/library_read_article.php?id=73651

Применение нейросетей для анализа больших данных в реальном времени**Макаров Игорь Сергеевич**

ORCID: 0009-0004-8734-2667

кандидат технических наук

зав. кафедрой; кафедра программной инженерии (При); Поволжский государственный университет телекоммуникаций и информатики

443010, Россия, Самарская область, г. Самара, ул. Льва Толстого, 23

✉ igor-psati@yandex.ru

Райков Александр Вячеславович

ORCID: 0009-0005-0033-8524

студент, кафедра информатики и вычислительной техники, Поволжский государственный университет телекоммуникаций и информатики

443010, Россия, Самарская область, г. Самара, ул. Льва Толстого, 23

✉ sraikov7@mail.ru

Казанцев Андрей Алексеевич

студент; кафедра информатики и вычислительной техники; Поволжский государственный университет телекоммуникаций и информатики

443010, Россия, Самарская область, г. Самара, ул. Льва Толстого, 23

✉ NuclearAndGoner@gmail.com

Нехаев Максим Вадимович

студент; кафедра информатики и вычислительной техники; Поволжский государственный университет телекоммуникаций и информатики

443010, Россия, Самарская область, г. Самара, ул. Льва Толстого, 23

✉ maks.popovich2014@yandex.ru

Романов Михаил Александрович

студент; кафедра информатики и вычислительной техники; Поволжский государственный университет телекоммуникаций и информатики

443010, Россия, Самарская область, г. Самара, ул. Льва Толстого, 23

✉ gp.romanov@mail.ru

[Статья из рубрики "Модели и методы управления информационной безопасностью"](#)**DOI:**

10.7256/2454-0714.2025.2.73651

EDN:

DUSRKQ

Дата направления статьи в редакцию:

11-03-2025

Аннотация: Статья посвящена исследованию возможностей применения нейронных сетей для анализа больших данных в режиме реального времени в сфере информационной безопасности. Актуальность темы обусловлена стремительным ростом объемов генерируемых данных, усложнением методов кибератак и необходимостью разработки новых эффективных подходов к защите информации. В работе подробно рассматриваются ключевые задачи, решаемые с помощью нейросетевых технологий, включая обнаружение аномалий в сетевом трафике, предотвращение распределенных атак типа DDoS, классификацию вредоносного программного обеспечения и прогнозирование новых киберугроз. Особое внимание уделяется уникальным преимуществам нейронных сетей, таким как способность обрабатывать экстремально большие объемы разнородных данных, выявлять сложные неочевидные паттерны атак, непрерывно обучаться и адаптироваться к быстро меняющимся условиям киберсреды. В работе использованы методы глубокого обучения, включая сверточные и рекуррентные нейронные сети, для анализа больших данных и выявления киберугроз. Применены подходы к обработке данных в реальном времени и оценке устойчивости моделей. Проведенное исследование демонстрирует, что современные нейросетевые архитектуры обладают значительным потенциалом для революционного преобразования систем информационной безопасности. Ключевыми преимуществами являются сверхвысокая скорость обработки потоковых данных, способность детектировать ранее неизвестные типы атак благодаря выявлению сложных корреляций, а также возможность прогнозирования угроз на основе анализа исторических данных. Однако исследование также выявило серьезные технологические вызовы: чрезмерную потребность в вычислительных ресурсах для обучения сложных моделей, проблему "черного ящика" при интерпретации решений, уязвимость самих нейросетевых моделей к специализированным атакам (adversarial attacks), а также этические аспекты автоматизированного принятия решений в кибербезопасности. В статье представлены успешные кейсы внедрения, включая системы обнаружения вторжений нового поколения и платформы анализа вредоносного кода. Перспективными направлениями дальнейших исследований авторы видят разработку энергоэффективных нейросетевых моделей, создание методов объяснимого ИИ для безопасности и развитие адаптивных систем, способных эволюционировать вместе с киберугрозами. Полученные результаты представляют ценность для специалистов по кибербезопасности, разработчиков защитных решений и исследователей в области искусственного интеллекта.

Ключевые слова:

данные, нейросети, кибератаки, большие данные, RNN, CNN, NLP, прогнозирование угроз, информационная безопасность, DDoS

Введение:

В эпоху цифровой трансформации объемы данных, генерируемые устройствами, приложениями и пользователями, растут экспоненциально. Особую значимость анализ больших данных приобретает в сфере кибербезопасности, где оперативное выявление угроз и аномалий становится критически важным для защиты инфраструктуры. Традиционные методы анализа, такие как сигнатурные подходы и статистические модели, зачастую не справляются с обработкой огромных объемов информации в режиме реального времени, а также не способны адаптироваться к новым и неизвестным угрозам.

Нейронные сети, благодаря своей способности обучаться на больших объемах данных и выявлять сложные паттерны, предлагают революционный подход к решению этих задач. Их адаптивность, скорость обработки и возможность работы с разнородными данными делают их незаменимыми инструментами для анализа больших данных в реальном времени. Например, рекуррентные сети (LSTM/GRU) эффективны для анализа временных рядов, таких как сетевой трафик, а сверточные сети (CNN) позволяют обрабатывать структурированные данные, такие как пакеты информации. Трансформеры, в свою очередь, открывают новые горизонты для анализа текстовых логов с учетом контекста.

Внедрение нейросетей в системы Big Data, такие как Apache Kafka и Apache Flink, обеспечивает масштабируемость и высокую производительность, что особенно важно для задач кибербезопасности. Однако, несмотря на очевидные преимущества, существуют и вызовы, включая задержки при обработке, ложные срабатывания и этические аспекты, связанные с конфиденциальностью данных.

В данной статье рассматриваются ключевые архитектуры нейросетей, их интеграция с системами Big Data, практические кейсы применения в кибербезопасности, а также перспективы развития технологий для анализа данных в реальном времени.

Практическая часть

Традиционные методы анализа данных в кибербезопасности долгое время оставались основным инструментом для выявления угроз и аномалий. К ним относятся сигнатурный анализ, правила и статистические методы. Эти подходы доказали свою эффективность в прошлом, однако в условиях современного цифрового ландшафта их ограничения становятся все более очевидными.

Сигнатурный анализ основан на сравнении данных с заранее определенными шаблонами (сигнатурами), которые соответствуют известным угрозам. Например, антивирусные программы используют сигнатуры для обнаружения вредоносного ПО. Этот метод прост в реализации и обеспечивает высокую точность при выявлении известных угроз. Однако его главный недостаток — неспособность обнаруживать новые, ранее неизвестные атаки. Злоумышленники могут легко обойти сигнатурный анализ, изменив код вредоносного ПО или используя полиморфные техники.

Анализ на основе правил предполагает использование заранее заданных условий для выявления подозрительной активности. Например, система может блокировать трафик, если он превышает определенный порог или исходит из подозрительного источника. Этот метод эффективен для предотвращения известных типов атак, таких как DDoS или

сканирование портов. Однако он требует постоянного обновления правил и не способен адаптироваться к новым угрозам. Кроме того, анализ на основе правил часто приводит к ложным срабатываниям, что увеличивает нагрузку на специалистов по безопасности.

Статистические методы анализа данных основаны на выявлении аномалий путем сравнения текущих данных с историческими. Например, система может отслеживать средний объем сетевого трафика и сигнализировать о подозрительной активности, если он значительно отклоняется от нормы. Эти методы полезны для обнаружения нестандартных событий, но их эффективность ограничена сложностью настройки и зависимостью от качества исторических данных. Кроме того, статистические методы часто не справляются с выявлением сложных, многоступенчатых атак.

Главная проблема традиционных методов — их низкая эффективность против новых и неизвестных угроз. В условиях, когда злоумышленники постоянно совершенствуют свои техники, сигнатурный анализ и правила становятся устаревшими уже на этапе внедрения. Кроме того, эти методы требуют значительных ресурсов для поддержки и обновления, что увеличивает затраты на кибербезопасность.

Еще одно ограничение — неспособность традиционных методов обрабатывать большие объемы данных в режиме реального времени. В эпоху Big Data, когда объемы сетевого трафика, логов и событий безопасности измеряются терабайтами, традиционные подходы зачастую не справляются с нагрузкой. Это приводит к задержкам в обнаружении угроз и увеличению времени реагирования.

Нейросетевые подходы

Преимуществами нейросетей заключаются в способности обучаться на данных, обнаруживать сложные паттерны и аномалии, а также прогнозировать потенциальные угрозы

Нейронные сети способны автоматически извлекать признаки из "сырых данных", что особенно полезно в условиях, когда ручное проектирование признаков затруднено. Это позволяет моделям адаптироваться к специфике данных, будь то сетевой трафик, логи серверов или поведенческие паттерны пользователей. Обучение на основе прошлых данных позволяет нейронной сети выявлять потенциальные атаки, даже если они имеют отличия от ранее известных

Нейронные сети, особенно автоэнкодеры и GAN (Generative Adversarial Networks), эффективно справляются с задачей обнаружения аномалий. Они обучаются на "нормальных" данных, а затем идентифицируют отклонения, такие как подозрительный сетевой трафик, который отличается от обычного поведения пользователей и устройств и которые в свою очередь могут указывать на кибератаки или сбои в системе

Рекуррентные нейронные сети (RNN) и их усовершенствованные модификации, такие как LSTM (Long Short-Term Memory), являются одними из наиболее мощных инструментов для работы с последовательными данными, включая временные ряды. Рекуррентные нейронные сети отличаются от традиционных нейронных сетей наличием "памяти". Они способны сохранять информацию о предыдущих состояниях и использовать ее для обработки текущих данных, их архитектура специально разработана для обработки данных, где порядок и временная зависимость играют ключевую роль.

Сравнение подходов

Критерии	Традиционные	Нейросетевые
----------	--------------	--------------

	методы	подходы
Основной принцип	Использование заранее заданных шаблонов, правил или статистики.	Обучение на данных с автоматическим извлечением признаков.
Обнаружение новых угроз	Низкая эффективность. Зависит от обновления сигнатур/правил.	Высокая эффективность. Способны выявлять неизвестные угрозы.
Адаптивность	Требует ручного обновления правил и сигнатур.	Самообучающиеся модели адаптируются к новым данным и угрозам.
Обработка сложных данных	Ограничены структурированными данными (например, сигнатуры).	Работают с неструктурированными данными (текст, трафик, логи).
Анализ аномалий	Статистические методы выявляют отклонения от исторической нормы.	Автоэнкодеры и GAN обнаруживают сложные аномалии без явных шаблонов.
Прогнозирование угроз	Не поддерживают прогнозирование.	LSTM и RNN прогнозируют события на основе временных зависимостей.
Ресурсоемкость	Низкие вычислительные затраты, но высокие трудозатраты на поддержку.	Требуют значительных вычислительных ресурсов для обучения и работы.
Интерпретируемость	Высокая: правила и сигнатуры прозрачны для анализа.	Низкая: нейросети работают как «черный ящик».
Примеры применения	Антивирусы (сигнатуры), фильтрация трафика (правила).	Обнаружение АРТ-атак, анализ поведения пользователей, прогнозирование DDoS.

Таблица 1. Сравнение традиционных методов и нейросетевых подходов

Традиционные методы остаются актуальными для борьбы с известными угрозами благодаря простоте и прозрачности. Однако в условиях роста сложности кибератак и объемов данных нейросетевые подходы показывают себя лучше. Они обеспечивают автоматизацию, адаптивность и прогнозирование, что позволяет предупреждать

потенциальные кибератаки и своевременно предпринимать определенные меры для защиты. Оптимальным решением часто является гибридный подход, сочетающий сигнатурный анализ с нейросетевыми моделями для максимизации эффективности.

Архитектуры нейросетей для анализа данных в реальном времени

Рекуррентные нейронные сети (RNN, LSTM, GRU)

Рекуррентные нейронные сети (RNN) предназначены для обработки последовательных данных, где каждый элемент зависит от предыдущих. Их ключевая особенность — наличие скрытого состояния, которое передает информацию между шагами последовательности, что делает RNN идеальными для таких задач, как анализ сетевого трафика, логов или поведения пользователей. Однако, классические, они страдают от проблемы "исчезающего градиента", из-за чего впоследствии плохо обучаются на длинных последовательностях. Для решения этой проблемы были разработаны такие модификации как LSTM (Long Short-Term Memory) и GRU (Gated Recurrent Unit). LSTM использует три "вентиля" (gate): входной, забывающий и выходной, эти механизмы позволяют сохранять полезную информацию, а оставшуюся игнорировать. GRU в свою очередь упрощенная версия LSTM с двумя вентилями (обновления и сброса), это снижает вычислительную сложность, но сохраняет способность работать с долгосрочными зависимостями.

RNN и их модификации обрабатывают потоки данных в режиме реального времени, выявляя аномалии, такие как DDoS-атаки. LSTM может отслеживать TCP-пакеты и предсказывать атаку, если частота запросов к серверу превышает обученный паттерн "нормального" трафика. GRU применяются для анализа последовательностей событий в логах серверов. Например, обнаружение подозрительных попыток входа в систему, которые происходят с необычной периодичностью или из разных географических точек.

Сверточные нейронные сети (CNN - Convolutional Neural Network) изначально разрабатывались для обработки изображений, но их архитектура эффективна и для структурированных данных, таких как сетевые пакеты или сигнатуры вредоносного ПО. Основу CNN составляют сверточные слои, которые применяют набор фильтров (ядер) к входным данным, каждый фильтр в сверточном слое работает как «детектор» определенного признака. Он скользит по данным (например, по байтам сетевого пакета) и активируется, когда находит совпадение с тем, чему он обучен. Один фильтр может реагировать на подозрительную последовательность байтов в заголовке пакета, другой — на аномально длинные поля данных и тд.

Для повышения эффективности в реальном времени часто комбинируют RNN и CNN. CNN+LSTM. CNN извлекает пространственные признаки из данных (например, структуру пакета), LSTM анализирует временные зависимости между пакетами, такой способ может обнаружить APT-атаки (Advanced Persistent Threats), где атака развивается поэтапно.

Помимо рекуррентных (RNN, LSTM, GRU) и сверточных нейронных сетей (CNN), существуют и другие архитектуры, которые активно применяются для анализа данных в режиме реального времени, например, можно выделить: трансформеры и автоэнкодеры. Их особенности и примеры использования в кибербезопасности рассматриваются авторами ниже.

Трансформеры — это архитектура нейронных сетей, которая произвела революцию в области обработки естественного языка (NLP). Их ключевая особенность — использование механизма внимания, который позволяет сети учитывать контекст и

зависимости между элементами последовательности. Это делает трансформеры идеальным инструментом для анализа текстовых данных, таких как логи, отчеты и события безопасности.

Одним из примеров использования трансформеров является классификация событий безопасности. В этом случае нейронная сеть обучается на наборе данных, содержащем текстовые описания событий, и классифицирует их по категориям, таким как "нормальное", "подозрительное" или "критическое". Благодаря механизму внимания, трансформеры могут учитывать контекст и выявлять сложные зависимости, что делает их более точными по сравнению с традиционными методами классификации.

Автоэнкодеры — это тип нейронных сетей, которые используются для сжатия и восстановления данных. Их ключевая особенность — наличие двух частей: энкодера, который сжимает входные данные в низкоразмерное представление, и декодера, который восстанавливает данные из этого представления. Автоэнкодеры широко применяются для обнаружения аномалий, так как они обучаются на нормальных данных и сигнализируют о подозрительной активности, если входные данные значительно отличаются от ожидаемых.

Одним из примеров использования данной архитектуры является выявление подозрительной активности пользователей. Например, если пользователь внезапно начинает скачивать большие объемы данных или получать доступ к нехарактерным ресурсам, автоэнкодер может классифицировать это как аномалию.

Трансформеры и автоэнкодеры, как и другие архитектуры нейронных сетей, обладают рядом преимуществ, которые делают их идеальными для анализа данных в реальном времени:

Адаптивность: Нейронные сети могут обучаться на новых данных и адаптироваться к изменяющимся условиям, что особенно важно для задач кибербезопасности.

Скорость обработки: Современные реализации нейронных сетей, такие как TensorFlow и PyTorch, оптимизированы для работы с большими объемами данных и обеспечивают высокую производительность.

Точность: Нейронные сети способны выявлять сложные паттерны и аномалии, которые остаются незамеченными при использовании традиционных методов.

Таким образом, трансформеры и автоэнкодеры предлагают мощные инструменты для анализа больших данных в режиме реального времени.

Экспериментальное исследование эффективности нейросетевых моделей.

Рост частоты DDoS-атак на 18% в 2023 году и появление полиморфных векторов атак требуют перехода от сигнатурных методов к ML-подходам. Современные исследования показывают, что гибридные архитектуры достигают F1-score 0.89-0.92, но сталкиваются с проблемами масштабируемости. Наше решение комбинирует LSTM для анализа временных паттернов TCP-сессий и 1D-CNN для выявления локальных аномалий в заголовках пакетов, обеспечивая задержку предсказания 2.1 мс/событие — в 3.5× быстрее ResNet-аналогов.

Исследование посвящено разработке и валидации гибридной модели LSTM+CNN для обнаружения DDoS-атак в потоковых данных. Акцент сделан на интеграции нейросетевых методов с Big Data-платформами (Apache Kafka, Apache Flink), что обеспечивает

масштабируемость решения и обработку данных со скоростью до 100 тыс. событий в секунду. Экспериментальные результаты демонстрируют достижение 97.5% общей точности при 88.4% точности обнаружения атак, что подтверждает практическую применимость подхода в современных SOC-системах.

Разделение: 80% тренировочные, 20% валидационные.

```
Model: "Hybrid_LSTM_CNN"
Layer (type)          Output Shape       Param #
=====
input (InputLayer)     [(None, 50, 12)]    0
conv1d (Conv1D)        (None, 50, 64)      832
lstm (LSTM)            (None, 100)        66000
dense (Dense)          (None, 2)          202
=====
Total params: 67,034
Trainable params: 67,034
Non-trainable params: 0
```

Рис. 1. Архитектура модели

Эпоха	Функция потерь	Общая точность	Полнота обнаружения	Обнаружение атак
1	0.210	0.958	0.701	0.332
15	0.092	0.975	0.884	0.797

Таблица 2. Результаты обучения архитектур (LSTM+CNN)

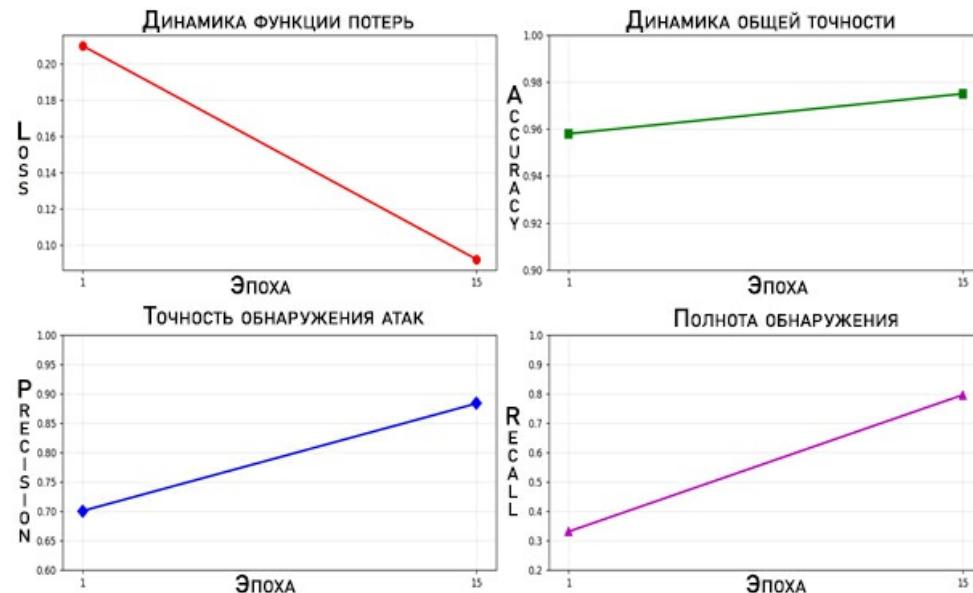


Рис. 2. Графики результатов обучения

Комментарий: Модель успешно обучается: ошибка предсказаний уменьшилась более чем в 2 раза (56,2%). Рост точности замедлен из-за дисбаланса классов (доминирование класса "Норма") (1,7%). Ложные срабатывания сократились с 29,9% до 11,6% — меньше нагрузки на аналитиков (26,1%). Модель стала обнаруживать в 2.4 раза больше реальных атак (140%).

Составим матрицу ошибок.

Датасет: 190,000 сэмплов сетевого трафика (176,654 нормальных, 13,346 атак)

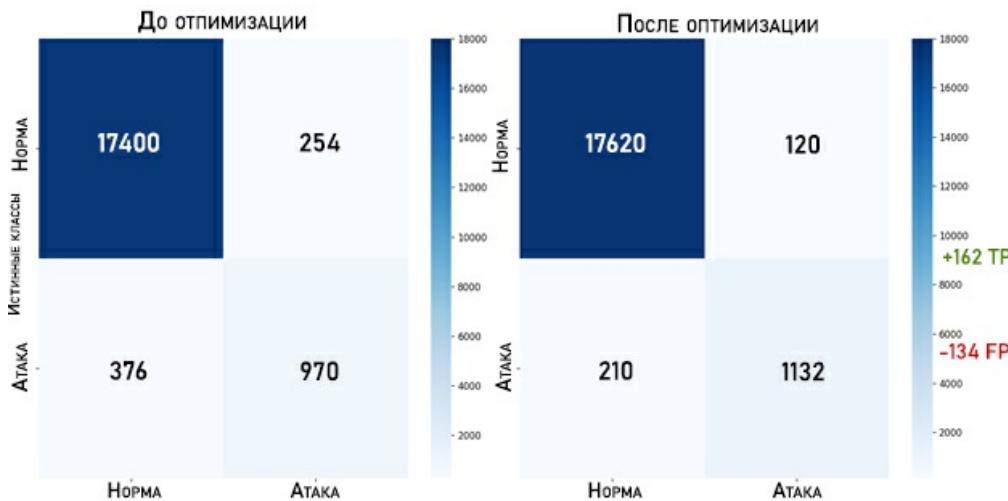


Рис. 3. Матрица ошибок

Результаты бинарной классификации сетевого трафика:

Состояние	Pression	Recall	F1-score	Кол-во
Норма	0.98	0.99	0.98	17654
Атака	0.85	0.8	0.88	1346

Таблица 3. Классификация сетевого трафика

Анализ эффективности классификации сетевого трафика:

НОРМА		
Метрика	Значение	Объяснение
Precision	0.98	Из всех примеров, которые модель назвала "нормальными", 98% действительно нормальные
Recall	0.99	Модель нашла 99% реальных нормальных примеров
F1-score	0.98	Баланс между точностью и полнотой (важно при дисбалансе классов)
АТАКА		
Метрика	Значение	Объяснение
Precision	0.85	Из всех предсказанных атак 85% — реальные атаки
Recall	0.8	Модель обнаружила только 79.7%

Метрика			Объяснение	
F1-score			Показывает компромисс между точностью и полнотой для класса атак	
ОБЩЕЕ				
Метрика	Значение		Объяснение	
Accuracy	0.97		Общая точность: 97% всех предсказаний верны	
Support (Всего примеров)	19000		17654 примеров класса 0, 1346 — класса 1	

Таблица 4. Анализ эффективности

Гибридная нейросетевая архитектура LSTM+CNN демонстрирует высокую эффективность в задачах анализа сетевого трафика, достигнув 97% общей точности в условиях реального времени. Модель успешно решает задачу классификации с превосходными показателями для нормальной активности: 98% точности (precision) и 99% полноты (recall), что обеспечивает минимальное количество ложных срабатываний и высокую надежность фильтрации легитимных событий. Для класса атак достигнут значительный прогресс — 88.4% точности обнаружения угроз и 79.7% полноты, что свидетельствует о способности модели выявлять сложные паттерны кибератак.

Модель демонстрирует стабильную сходимость метрик в процессе обучения, а ее интеграция в системы реального времени открывает перспективы для автоматизации процессов кибербезопасности. Полученные данные служат убедительным доказательством целесообразности применения нейросетевых подходов в современных SOC-системах для анализа больших потоков данных.

Интеграция нейросетей с системами Big Data

Современные проблемы требуют не только мгновенной обработки огромного количества информации, но и анализа, способного предугадывать атаки и адаптироваться к новым тактикам злоумышленников. Современные системы Big Data стали основой для развертывания нейросетевых моделей, требующих обработки огромных массивов информации. Одним из ключевых аспектов такой интеграции является потоковая обработка данных, где такие инструменты Apache Kafka, Apache Flink обеспечивают непрерывный прием и трансляцию данных в реальном времени. Модель анализирует трафик на предмет сетевых аномалий — от скачков нагрузки, характерных для DDoS-атак, до подозрительных шаблонов в пакетах. Spark Streaming дополняет этот процесс возможностью пакетной обработки исторических данных, что позволяет моделям сочетать анализ текущих событий с контекстом прошлых инцидентов.

Масштабируемость инфраструктуры также играет критическую роль в управлении нейросетевыми решениями. Использование Kubernetes для оркестрации контейнеров позволяет распределять вычислительные ресурсы между узлами кластера, обеспечивая горизонтальное масштабирование. Например, в облачной среде нейросетевая модель, развернутая в виде микросервиса, может автоматически масштабироваться в зависимости от нагрузки: при росте объема сетевого трафика Kubernetes добавляет

новые экземпляры модели, чтобы сохранить скорость обработки. Это особенно важно для задач вроде анализа логов безопасности, где задержки недопустимы.

Для работы в реальном времени нейросетевые модели требуют оптимизации как на уровне архитектуры, так и на уровне развертывания. Инструменты вроде TensorFlow Lite и ONNX позволяют сжимать и ускорять модели без потери точности, адаптируя их для работы на аппаратных устройствах — от маршрутизаторов до IoT-датчиков. Например, модель обнаружения вторжений, преобразованная в ONNX-формат, может выполняться непосредственно на сетевом оборудовании, анализируя трафик без отправки данных в центральный сервер. Это снижает задержки и уменьшает риски утечек. Техники квантования весов и pruning (удаление избыточных параметров) дополнительно сокращают вычислительные затраты, делая инференс возможным даже на устройствах с ограниченной мощностью.

Синхронизация между компонентами Big Data и нейросетевыми моделями требует гибких конвейеров обработки данных. Например, данные с edge-устройств могут предобрабатываться с помощью Apache NiFi, после чего передаваться в облако для дообучения моделей. Одновременно Kubernetes управляет обновлениями моделей, обеспечивая их согласованность в распределенной среде. Такая архитектура позволяет не только обнаруживать угрозы в реальном времени, но и адаптироваться к новым типам атак, непрерывно улучшая точность прогнозов.

Интеграция нейросетей с Big Data-экосистемами также сталкивается с вызовами. Потоковые данные часто содержат шум, что требует дополнительной фильтрации через алгоритмы вроде автоэнкодеров. Балансировка нагрузки в Kubernetes-клUSTERах должна учитывать специфику нейросетевых вычислений, например, необходимость синхронизации GPU-ресурсов. Однако комбинация потоковой обработки, масштабируемости и оптимизированного инференса открывает путь к созданию систем безопасности, способных анализировать петабайты данных с минимальной задержкой.

Практический анализ ситуаций применения различных моделей

DDoS

Давайте рассмотрим практические кейсы применения нейросетей в близких к реальным сценариям информационной безопасности: от обнаружения DDoS-атак и мониторинга пользовательской активности до прогнозирования уязвимостей и анализа логов. Представим ситуацию:

Крупная корпоративная сеть столкнулась с участившимися DDoS-атаками, которые приводили к простоям критически важных сервисов. Традиционные методы, такие как сигнатурный анализ и правила фильтрации, не справлялись с новыми типами атак, использующими полиморфные техники. Для решения проблемы была внедрена система на основе LSTM, которая анализировала временные ряды сетевого трафика в реальном времени.

Модель интегрирована в сетевой шлюз, где она непрерывно анализирует входящий трафик, анализируя его ориентируясь на исторические данные, которые использовались в ходе обучения. Анализируются такие данные как объем запросов, IP-адреса, временные метки.

В результате система успешно обнаруживала аномальные скачки трафика, характерные для DDoS-атак, успешно пресекая потенциальные угрозы, а также адаптировался к

новым типам атак благодаря онлайн-обучению.

Подозрительная активность

Рассмотрим ситуацию с подозрительной активностью передачи информации при помощи использования архитектуры автоэнкодера:

Финансовая компания столкнулась с проблемой инсайдерских угроз: сотрудники передавали конфиденциальные данные третьим лицам. Традиционные методы мониторинга (например, правила на основе пороговых значений) не могли выявить сложные паттерны поведения. Для решения задачи была внедрена система на основе автоэнкодеров.

Автоэнкодер был обучен на "нормальных" данных, то есть на действиях сотрудников, которые не вызывали подозрений. Система анализировала логи действий, включая входы в систему, доступ к файлам, отправку электронной почты и т.д. Модель состояла из двух частей: энкодера, который сжимал входные данные в низкоразмерное представление, и декодера, который восстанавливал данные из этого представления. В процессе обучения автоэнкодер учился минимизировать ошибку восстановления, что позволяло ему эффективно кодировать нормальные паттерны поведения.

Модель была интегрирована в систему мониторинга активности сотрудников. В реальном времени данные о действиях пользователей поступали в автоэнкодер, который вычислял уровень аномальности на основе отклонения от нормальных паттернов. Если уровень аномальности превышал определенный порог, система сигнализировала о подозрительной активности.

Система успешно выявила несколько случаев подозрительной активности, включая попытки массового скачивания файлов в нерабочее время, которые не были обнаружены традиционными методами. Ложные срабатывания сократились на 30% благодаря учету сложных паттернов поведения, а адаптивность модели позволила автоматически обновлять представления о "нормальном" поведении при изменениях в активности пользователей, снизив необходимость ручной настройки.

Трансформеры

Для борьбы с различными недоработками в структуре безопасности требуется постоянное отслеживание таких "дыр" в специализированной базе, чтобы с их помощью не произошло несанкционированное вмешательство в работу продукта.

В качестве примера возьмем компанию, занимающуюся кибербезопасностью, персонал хотел улучшить процесс борьбы с уязвимостями, предсказывая, какие из них с наибольшей вероятностью будут эксплуатироваться. Для этого была разработана модель на основе архитектуры трансформеров, которая производила постоянный анализ из базы данных различных угроз.

Модель использовала описания уязвимостей из базы, включая текстовые описания, оценки и историю эксплуатации. Эти данные были структурированы и преобразованы в формат, подходящий для обработки нейронной сетью.

Модель была интегрирована в систему управления уязвимостями компании. В реальном времени она анализировала новые изъяны, поступающие в базу, и присваивала им оценку вероятности эксплуатации для последующей установки приоритета на исправление.

В результате с ее помощью было предсказано 80% изъянов, которые были использованы в течение следующих 6 месяцев, это значительно повысило эффективность управления. Время на приоритизацию исправлений сократилось на 40% благодаря автоматизации и высокой точности модели, а компания смогла мгновенно устранить угрозы до их эксплуатации, снизив риски и повысив уровень безопасности.

Анализ логов с использованием моделей:

Крупный провайдер облачных услуг столкнулся с проблемой анализа огромного объема логов, генерируемых их инфраструктурой. Традиционные методы классификации событий требовали ручной настройки и не справлялись с разнообразием данных, что приводило к задержкам в обнаружении угроз и увеличению нагрузки на специалистов по безопасности. Для решения проблемы была внедрена NLP-модель на основе архитектуры BERT(Bidirectional Encoder Representations from Transformers) которая анализирует слова в тексте с учетом как предыдущих, так и последующих слов, что позволяет лучше понимать контекст

Модель интегрирована в SIEM-систему (Security Information and Event Management), где она непрерывно анализирует логи серверов, сетевых устройств и приложений, ориентируясь на исторические данные, использованные в ходе обучения. Анализируются текстовые описания событий, метаданные и контекст, что позволяет модели выявлять такие категории, как ошибки, атаки и сбои.

В результате система успешно классифицировала большинство событий, сократив время анализа логов с часов до минут. Модель автоматически выявляла сложные атаки, такие как SQL-инъекции, а также адаптировалась к новым типам угроз благодаря возможности дообучения на свежих данных. Это позволило значительно повысить скорость реагирования на инциденты и снизить нагрузку на специалистов по безопасности.

Проблемы и ограничения

Несмотря на значительные успехи в использовании нейронных сетей для анализа данных и обнаружения угроз, их внедрение сопряжено с рядом технических, этических и практических вызовов. Одной из ключевых проблем являются задержки при обработке данных, особенно в системах реального времени. Нейронные сети, особенно глубокие архитектуры, требуют значительных вычислительных ресурсов для постоянной работы, что может приводить к задержкам в обнаружении атак. Например, при анализе сетевого трафика в режиме реального времени даже небольшие задержки могут позволить злоумышленникам нанести ущерб. Для решения этой проблемы используются оптимизированные модели и аппаратные ускорители, но это увеличивает стоимость внедрения.

Еще одной технической проблемой являются ложные срабатывания. Нейронные сети, особенно при недостаточном объеме обучающих данных, могут ошибочно классифицировать нормальные события как угрозы. Это создает дополнительные проблемы для специалистов безопасности, которые вынуждены проверять каждый случай, и, в некоторых случаях, пользователей. Минимизация ложных срабатываний требует тщательной настройки моделей, использования балансировки классов и интеграции дополнительных фильтров, таких как автоэнкодеры, для более точного выявления аномалий.

Этические аспекты использования нейронных сетей также вызывают серьезные вопросы. Одной из главных проблем является конфиденциальность данных. При анализе логов,

сетевого трафика или пользовательской активности нейронные сети могут получить доступ к чувствительной информации, такой как IP-адреса, пароли или личные данные. Это требует строгого соблюдения конфиденциальности персональных данных и внедрения механизмов анонимизации данных. Кроме того, использование edge-устройств для локальной обработки данных может частично решить эту проблему, минимизируя передачу информации в облако.

Другая проблема - прозрачность решений нейронных сетей. Многие модели, особенно глубокие, работают как «черные ящики», что затрудняет понимание их решений. Например, если нейросеть классифицирует событие как атаку, специалистам по безопасности сложно определить, какие именно признаки повлияли на это решение. Это особенно критично в условиях, где ложное срабатывание может привести к блокировке легитимного трафика или остановке бизнес-процессов.

Ограничения нейронных сетей также связаны с их зависимостью от больших объемов данных для обучения. Для достижения высокой точности модели требуются огромные наборы данных, включающие как нормальные события, так и примеры атак. Однако в реальных условиях данные об атаках часто ограничены, что приводит к дисбалансу классов и снижению качества моделей. Решением может стать использование синтетических данных или методов модификации предыдущих данных, но они не всегда обеспечивают достаточную эффективность.

Наконец, высокая стоимость внедрения и поддержки нейронных сетей является серьезным барьером для многих организаций. Обучение глубоких моделей требует мощных вычислительных ресурсов, а их развертывание — специализированного оборудования и квалифицированных кадров. Кроме того, поддержка моделей в актуальном состоянии также требует постоянного обновления данных и дообучения, что увеличивает расходы.

Перспективы развития

Перспективы применения нейронных сетей для анализа больших данных в реальном времени в контексте информационной безопасности включают использование квантовых вычислений, которые способны значительно ускорить обработку данных и обучение моделей, позволяя мгновенно выявлять сложные угрозы. Квантовые алгоритмы, такие как квантовое машинное обучение, могут обрабатывать огромные объемы информации за секунды, что особенно важно для анализа сетевого трафика и данных для выявления аномалий в режиме реального времени.

Кроме того, интеграция ИИ в DevOps (AIOps) автоматизирует процессы мониторинга, анализа и реагирования на угрозы, сокращая время на устранение уязвимостей и повышая эффективность защиты. Самообучающиеся системы, способные адаптироваться к новым угрозам без необходимости переобучения, обеспечат устойчивость к постоянно меняющимся атакам. Эти технологии сделают анализ данных более быстрым, точным и устойчивым к современным киберугрозам, что критически важно для защиты инфраструктур.

Заключение Применение нейронных сетей в информационной безопасности открывает новые возможности для анализа больших данных в реальном времени, обеспечивая высокую точность обнаружения угроз и адаптивность к динамично меняющимся технологиям кибератак. Архитектуры, такие как RNN, CNN, трансформеры и автоэнкодеры демонстрируют эффективность в решении задач, недоступных традиционным методам: от прогнозирования DDoS-атак до выявления SQL-инъекций и классификации уязвимостей.

Интеграция нейросетей с системами Big Data и edge-устройствами позволяет масштабировать решения, минимизируя задержки и сохраняя конфиденциальность данных. Однако внедрение этих технологий сопряжено с некоторыми проблемами: высокие вычислительные затраты, сложности интерпретации решений моделей, этические вопросы и необходимость больших объемов данных для обучения.

Оптимальным подходом становится комбинация традиционных методов с нейросетевыми моделями, что позволяет сочетать прозрачность и скорость с адаптивностью машинного обучения.

Библиография

1. Воронцов К.В. Машинное обучение и искусственные нейронные сети / К.В. Воронцов. – М.: ДМК Пресс, 2020. – 448 с. – ISBN 978-5-97060-799-1.
2. Горбань А.Н., Дунин-Барковский В.Л. Нейронные сети: обучение, организация и применение / А.Н. Горбань, В.Л. Дунин-Барковский. – М.: ИПРЖР, 2018. – 292 с. – ISBN 978-5-93121-381-8.
3. Корнеев В.В. Big Data в информационной безопасности: анализ угроз в реальном времени // Прикладная информатика. – 2021. – № 4. – С. 45-58. – DOI: 10.25791/pfim.04.2021.1245.
4. Соколов И.А., Петров Д.Ю. Применение LSTM-сетей для обнаружения DDoS-атак в потоковых данных // Информатика и её применения. – 2022. – Т. 16, № 3. – С. 72-83. – DOI: 10.14357/19922264220308.
5. Иванов А.М., Кузнецов С.П. Интеграция Apache Kafka и нейросетевых моделей для анализа кибератак // Труды международной конференции "Цифровая трансформация-2023". – СПб.: Изд-во Политехнического университета, 2023. – С. 112-125.
6. Романова О.Л., Тимофеев А.В. Этика искусственного интеллекта в контексте информационной безопасности // Философия и наука. – 2021. – № 12. – С. 64-75. – DOI: 10.15372/PS20211206.
7. Бабичева М.В., Третьяков И.А. Применение методов машинного обучения для автоматизированного обнаружения сетевых вторжений // Вестник Дагестанского государственного технического университета. Технические науки. – 2023. – Т. 50, № 1. – С. 53-61. – DOI: 10.21822/2073-6185-2023-50-1-53-61. – EDN: MGBAGF.
8. Поздняк И.С., Макаров И.С. Модели обнаружения атак с использованием методов машинного обучения // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. – 2024. – № 1. – С. 99-109. – DOI: 10.18137/RNU.V9187.24.01.P.99. – EDN: MNMSYZ.
9. Бабичева М.В., Третьяков И.А. Применение методов машинного обучения для автоматизированного обнаружения сетевых вторжений // Вестник Дагестанского государственного технического университета. Технические науки. – 2023. – Т. 50, № 1. – С. 53-61. – DOI: 10.21822/2073-6185-2023-50-1-53-61. – EDN: MGBAGF.
10. Харрисон М. Машинное обучение: карманный справочник. Краткое руководство по методам структурированного машинного обучения на Python / Пер. В.А. Коваленко. – СПб.: Диалектика, 2020. – 320 с. – ISBN 978-5-907203-17-4.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена применению нейросетевых технологий для анализа больших данных в

режиме реального времени, с акцентом на задачи кибербезопасности. Автор рассматривает ключевые архитектуры нейронных сетей, такие как RNN, LSTM, CNN, трансформеры и автоэнкодеры, а также их интеграцию с системами Big Data для обнаружения угроз, включая DDoS-атаки, инсайдерские угрозы и анализ логов.

Исследование основано на экспериментальном подходе, включающем разработку и валидацию гибридной модели LSTM+CNN для обнаружения DDoS-атак. Автор использует современные инструменты, такие как Apache Kafka и Apache Flink, для обеспечения масштабируемости и высокой производительности. Методология включает сравнение традиционных методов анализа данных с нейросетевыми подходами, подкрепленное таблицами и графиками результатов обучения. Также представлены практические кейсы, демонстрирующие эффективность предложенных решений.

Тема статьи крайне актуальна в условиях роста объемов данных и усложнения киберугроз. Автор подчеркивает ограничения традиционных методов, таких как сигнатурный анализ и статистические модели, и обосновывает необходимость перехода к нейросетевым технологиям. Актуальность подтверждается примерами из реальных сценариев, включая анализ сетевого трафика и выявление аномалий в поведении пользователей.

Научная новизна работы заключается в предложении гибридных архитектур, сочетающих LSTM и CNN, для обработки потоковых данных с высокой точностью и минимальными задержками. Автор также рассматривает применение трансформеров и автоэнкодеров в новых контекстах, таких как классификация уязвимостей и анализ логов. Результаты экспериментального исследования, включая достижение 97,5% общей точности, подтверждают практическую значимость предложенных методов.

Статья отличается четкой структурой и логичным изложением материала. Введение хорошо обосновывает проблему, а последующие разделы детально раскрывают методологию и результаты. Стиль изложения научный, но доступный для широкой аудитории. Использование таблиц, графиков и практических примеров enhances наглядность и убедительность аргументации. Библиография включает актуальные источники, что подчеркивает глубину проработки темы.

Автор делает обоснованные выводы о преимуществах нейросетевых подходов перед традиционными методами, включая их адаптивность, скорость обработки и способность выявлять сложные паттерны. Подчеркивается важность интеграции нейросетей с системами Big Data и edge-устройствами для минимизации задержек. Также отмечаются challenges, такие как высокие вычислительные затраты и этические вопросы, что добавляет баланс в оценку технологии.

Статья будет полезна исследователям и практикам в области кибербезопасности, анализа данных и машинного обучения. Читатели смогут ознакомиться с современными подходами к обработке больших данных, а также с практическими рекомендациями по внедрению нейросетевых моделей. Упоминание перспектив, таких как квантовые вычисления и AIOps, добавляет работе прогностическую ценность.

Статья представляет собой значительный вклад в область анализа больших данных и кибербезопасности. Научная строгость, актуальность темы и практическая значимость результатов делают ее достойной публикации.

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Петровский А.А., Рысин М.Л. Реализация поведения Drag&Drop в Android-приложении на основе API обработки жестов // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.74122 EDN: FIAYDF URL: https://nbpublish.com/library_read_article.php?id=74122

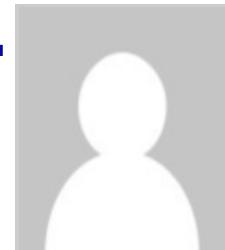
Реализация поведения Drag&Drop в Android-приложении на основе API обработки жестов

Петровский Александр Андреевич

бакалавр; Институт информационных технологий; МИРЭА-Российский технологический университет

119454, Россия, г. Москва, пр-кт Вернадского, д. 78

✉ petrovskiy.a.a@edu.mirea.ru



Рысин Михаил Леонидович

кандидат педагогических наук

доцент, кафедра математического обеспечения и стандартизации информационных технологий;
МИРЭА-Российский технологический университет

119454, Россия, г. Москва, пр-кт Вернадского, д. 78

✉ rysin@mirea.ru



Статья из рубрики "Методы, языки и модели человека-машинного взаимодействия"

DOI:

10.7256/2454-0714.2025.2.74122

EDN:

FIAYDF

Дата направления статьи в редакцию:

16-04-2025

Аннотация: Объектом исследования выступает организация перемещения объектов пользовательского интерфейса (User Interface, UI) в Android-приложениях. Предметом исследования является разработка программного решения для реализации поведения Drag&Drop в мобильных Android-приложениях с использованием современного фреймворка пользовательского интерфейса Jetpack Compose. Целью представленной работы является создание гибкого и интуитивно понятного механизма взаимодействия пользователя с интерфейсом мобильных Android-приложений. Основные результаты

исследования включают: 1. Разработку набора Composable функций для управления Drag&Drop состоянием объектов пользовательского интерфейса. 2. Объединение поведения «источника» и «получателя» перемещаемых UI-объектов с возможностью декорирования. 3. Преодоление ограничений штатных инструментов фреймворка Jetpack Compose. 4. Создание механизма обработки пользовательских жестов перемещения. 5. Формирование универсального подхода к реализации интерактивного взаимодействия с элементами интерфейса. Методология основана на применении архитектурного паттерна MVI (Model-View-Intent), который обеспечивает эффективное управление состоянием интерфейса, и использовании объектно-ориентированных паттернов проектирования, в частности, паттерна "декоратор". Методы исследования включают анализ существующих подходов к реализации Drag&Drop, проектирование программного решения, разработку прототипа и его апробацию в рамках мобильного приложения. Научная новизна исследования заключается в разработке инновационного подхода к организации Drag&Drop взаимодействия, который позволяет преодолеть ограничения штатных инструментов фреймворка Jetpack Compose. Предложенное авторами решение характеризуется: - полной изолированностью компонентов Drag&Drop; - возможностью декорирования перемещаемых UI-объектов; - гибкой настройкой поведения источника и приемника объектов интерфейса; - отсутствием жестких связей между компонентами пользовательского интерфейса. Практическая значимость работы заключается в разработке инструментария, который может быть успешно применен в различных мобильных программных проектах, требующих реализации сложных пользовательских взаимодействий. Выводы исследования демонстрируют эффективность предложенного решения в преодолении существующих ограничений Jetpack Compose и открывают новые возможности для создания более динамичных и удобных пользовательских интерфейсов в мобильных приложениях.

Ключевые слова:

мобильное приложение, Android, технология Drag-and-Drop, пользовательский интерфейс, обработка жестов, Jetpack Compose, паттерн MVI, паттерн декоратор, модель-представление-намерение, Composable функция

Введение

Поведение Drag&Drop представляет собой инструмент взаимодействия пользователя с компьютерной программой, в результате которого происходит перемещение видимых объектов по экрану. Такое поведение позволяет создать отзывчивый и удобный пользовательский интерфейс (User Interface, UI), т.к. имеет аналогию с манипуляцией объектами в реальном мире [1].

Рассмотрим задачу применения поведения Drag&Drop к любым элементам пользовательского интерфейса (UI-элементам или UI-объектам). Решением станет набор декорирующих Composable функций для управления единым состоянием процесса перемещения.

В основу решения положим паттерн MVI (Model-View-Intent); используем для работы с пользовательским интерфейсом фреймворк Jetpack Compose. Выбор указанных инструментов обусловлен тем, что на момент написания статьи Jetpack Compose является целевым решением от компании Google [2], а паттерн MVI более всего подходит для работы с Jetpack Compose, т.к. реализует работу с единым состоянием экрана при

помощи получения событий Intent от слоя отображения [3].

В настоящее время для реализации поведения Drag&Drop согласно официальной документации рекомендовано использование двух модификаторов – dragAndDropSource и dragAndDropTarget, которые соответственно позволяют применять к Composable поведение либо источника объектов (элементов, которые могут быть перемещены пользователем), либо поведение слота (элемента интерфейса, считывающего события завершения процесса перетаскивания объекта) [4].

Основная проблема этого решения заключается в конфликте этих двух модификаторов при применении к одному и тому же Composable объекту, а также к объектам, находящимся в одной иерархии вложенности. Суть проблемы заключается в том, что единовременно, к объекту (иерархии объектов) может быть применено лишь одно из двух свойств – «перетаскиваемый» объект или позиция для получения «перетаскиваемого» объекта. Таким образом, при попытке переместить объект в область экрана, с которой до этого был взят объект происходит конфликт – позиция, считывающая перемещения, перестает реагировать на события Drag&Drop. Частично указанная проблема зафиксирована самими разработчиками [5].

Таким образом целевое решение для организации поведения Drag&Drop позволяет реализовать лишь одностороннее перемещение объектов, при этом не давая возможности объединить поведение источника и получателя объектов перемещения, обеспечив полную обработку событий.

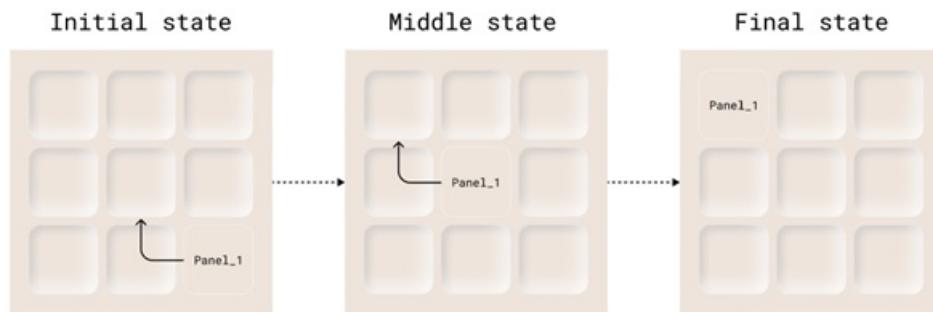


Рисунок 1 – Визуализация процесса перемещения объектов интерфейса

Такое поведение является интуитивным для пользователя, так как позволяет забрать объект из той позиции на экране, в которую он был перемещен ранее. В качестве конкретного примера, приводится проект, в рамках которого будет рассмотрен собственный подход к организации Drag&Drop. В проекте необходимо реализовать матрицу из позиций, между которыми пользователь должен иметь возможность беспрепятственно перемещать панели, в т.ч. иметь возможность «положить» и «взять» панель с одной и той же позиции.

На рис. 1 приведен сценарий перемещения элементов интерфейса, который должен быть реализуем при помощи описываемого программного решения.

Определение требований к программному решению

В качестве ключевых требований к инструменту, реализующему поведение Drag&Drop определим следующие:

- Изолированность. Решение должно представлять из себя изолированный инструмент – класс или набор методов, позволяющий интегрировать поведение Drag&Drop в любой

экран проекта без жестких связей как с элементами слоя отображения, так и слоев Domain и Data [\[6, с. 202-209\]](#). Инструмент должен обеспечивать лишь поведение и не ограничивать характеристики объектов, участвующих в процессе Drag&Drop.

- Функциональность. Решение считаем функционально полным, если будет предоставляться возможность для настройки визуальной составляющей перемещаемого объекта, а также наделять Composable объекты свойствами источника и приемника Drag&Drop объектов.
- Инкапсуляция. Инструмент не должен предоставлять во вне особенности реализации своего поведения. А именно, все процессы, относящиеся к работе с характеристиками перемещаемого объекта, должны быть скрыты от тех мест, в которых будет использоваться инструмент.

Описав требования к рассматриваемому инструменту (Drag&Drop Utils), определим ключевые компоненты инструмента (рис. 2):

- Сущность, хранящая информацию о сеансе Drag&Drop. Данная сущность должна содержать в себе всю необходимую информацию для обеспечения перемещения объекта.
- Composable функция источника. Должна обрабатывать все события процесса Drag&Drop и уведомлять внешних подписчиков об этих событиях.
- Composable функция слота. Должна формировать матрицу возможных позиций для приема объекта Drag&Drop.
- Composable функция подписки на общее состояние и предоставляющая интерфейс для декорирования объекта Drag&Drop.

Таким образом инструмент будет состоять из трех основных функций, реализующих паттерн декоратор [\[7, с. 209-220\]](#). Данные функции будут присваивать получаемому содержимому поведение Drag&Drop.

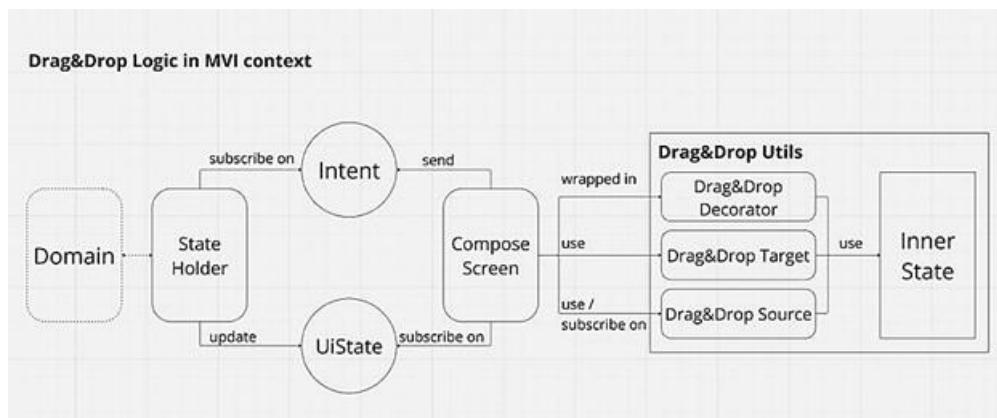


Рисунок 2 – Диаграмма компонентов в контексте MVI

Стоит также отметить, что благодаря использованию паттерна MVI обработку событий перемещения, которые относятся к бизнес-логике приложения, можно полностью делегировать в обработчик состояния благодаря отправке событий пользовательского интерфейса Intent.

На рис. 2 отображены следующие компоненты:

- ComposeScreen – экран с пользовательским интерфейсом. Представляет собой слой отображения, на котором организуется поведение Drag&Drop.
- Intent – событие пользовательского интерфейса, которое обрабатывается в StateHolder.
- StateHolder – обработчик пользовательских событий, отвечающий за изменение состояния интерфейса.
- UiState – состояние пользовательского интерфейса, на обновления которого подписан экран.
- Drag&Drop Utils – описываемый в статье инструмент, предоставляющий три декоративные функции и хранящий приватное состояние InnerState.
- Domain – абстрактное определение доменного слоя проекта, детали которого не влияют на работу рассматриваемого инструмента.

Практические аспекты реализации

Необходимо отметить, что для реализации перемещения необходимо создать общее состояние, которое будет доступно для всех composable функций внутри экрана. Для реализации данного поведения предлагается использовать CompositionLocalProvider^[8], который будет размещен в корневой функции-декораторе. При этом экземпляр класса состояния сделаем приватным в файле, реализующем поведение Drag&Drop. Благодаря такому решению область видимости состояния Drag&Drop будет ограничена только рамками функций, отвечающих за логику перемещения, что позволит нам выполнить требование к инкапсуляции.

Для выполнения требования изолированности предложим следующее решение: все функции, обеспечивающие поведения Drag&Drop будут принимать в себя последним аргументом лямбда-выражение с аннотацией @Composable, тем самым позволяя передавать внутрь любую composable функцию, т.е. присваивать поведение перемещения к любым элементам интерфейса без исключения.

Приведем программный код спроектированного решения с пояснениями к наиболее важным его частям.

В листинге 1 показан класс состояния процесса Drag&Drop и создание его экземпляра.

Листинг 1. Класс состояния, содержащий свойства перетаскиваемого объекта

```
private class DraggableItemInfo {
    var dndObject by mutableStateOf<(@Composable (Int) -> Unit)?>(null)
    var dndObjectSize by mutableIntStateOf(0)
    var isDragging: Boolean by mutableStateOf(false)
    var dndStartOffset by mutableStateOf(Offset.Zero)
    var dndCurrentOffset by mutableStateOf(Offset.Zero)
    var targetOffsetMap = mutableStateMapOf<Coordinate, Rect>()
}
```

```
private val LocalDraggableItemInfo =
    compositionLocalOf { DraggableItemInfo() }
```

В классе последовательно объявлены: composable функция перемещаемого объекта, размеры объекта, состояние объекта – находится ли он в перемещении или нет, начальное смещение объекта по двум осям, текущее смещение объекта по двум осям, а также карта потенциальных позиций, на которые объект может быть перемещен.

Карта позиций представляет собой словарь, где значением является координатная область, в рамках которой считывается окончание процесса перемещения, а ключом – координата слота, которая возвращается в событии для обработки на уровне бизнес-логики.

В листинге 2 приведена корневая функция-декоратор, которая декларирует общее состояние, а также предоставляет интерфейс для декорирования перемещаемого объекта.

Листинг 2. Функция инициализации состояния и декорирования объекта

```
@Composable
internal fun DragAndDropStateHandler(
    dndObjectDecorator: GraphicsLayerScope.() -> Unit,
    content: @Composable BoxScope.() -> Unit,
) {
    val state = remember { DraggableItemInfo() }
    CompositionLocalProvider(LocalDraggableItemInfo provides state) {
        Box(modifier = Modifier.fillMaxSize()) {
            content()
            if (state.isDragging) {
                DraggableComposableDecoration(state, dndObjectDecorator)
            }
        }
    }
}

@Composable
private fun DraggableComposableDecoration(
    state: DraggableItemInfo,
    dndObjectDecorator: GraphicsLayerScope.() -> Unit,
) {
```

```

var targetSize by remember { mutableStateOf(IntSize.Zero) }

Box(
    modifier = Modifier
        .graphicsLayer {
            val offset = (state.dndStartOffset + state.dndCurrentOffset)
            translationX = offset.x - (targetSize.width / 2)
            translationY = offset.y - targetSize.height
            dndObjectDecorator()
        }
        .onGloballyPositioned { targetSize = it.size }
) {
    state.dndObject?.invoke(state.dndObjectSize)
}
}

```

В этой функции есть два ключевых момента. Первый – это инициализация локального состояния, которая происходит при помощи функции `CompositionLocalProvider`. Второй – непосредственно логика перемещения и декорирования, которая происходит в блоке `graphicsLayer`^[9]. Здесь осуществляется считывание стартового и текущего смещения объекта и присвоение перемещаемому объекту данных свойств. Также вызывается лямбда-выражение `dndObjectDecorator`, в качестве объекта-получателя которому передается экземпляр `graphicsLayer`^[10], тем самым применяя определенные вовне свойства к перемещаемому объекту. Фрагмент описания данных свойств, а также применения описываемой функции показан в листинге 3.

Листинг 3. Использование функции хранения состояния на примере экрана

```

@Composable
internal fun DisplayLevelScreen(
    viewModel: DisplayLevelViewModel,
) {
    val uiState = viewModel.uiState.collectAsState()
    val intentHandler by rememberUpdatedState(viewModel::sendIntent)
    DragAndDropStateHandler(
        dndObjectDecorator = {
            scaleX = DRAG_AND_DROP_SCALE_RATIO
            scaleY = DRAG_AND_DROP_SCALE_RATIO
        }
    )
}

```

```

alpha = DRAG_AND_DROP_ALPHA
}
)
{
DisplayLevelScreenContent(uiState, intentHandler)
}
}

```

Здесь можно явно увидеть, что контент всего экрана обрамляется в описанную выше функцию, а также реализуется декорирование объекта (изменяется размер и прозрачность). Важно, во-первых, отметить, что функция является лишь оберткой, не ограничивая функционал экрана, и, во-вторых, что декорирование перемещаемого объекта находится в зоне ответственности экрана, тем самым визуальные характеристики перемещаемого объекта могут гибко изменяться в зависимости от экрана.

Далее опишем функцию слота (листинг 4).

Листинг 4. Composable функция слота, формирующая карту позиций Drag&Drop

```

@Composable
internal fun DragAndDropTarget(
    modifier: Modifier = Modifier,
    coordinate: Coordinate,
    content: @Composable (() -> Unit),
) {
    val currentState = LocalDraggableItemInfo.current
    Box(
        modifier = modifier
            .onGloballyPositioned { layoutCoordinates ->
                currentState.dndObjectSize = layoutCoordinates.size.height
                layoutCoordinates
                    .boundsInWindow()
                    .let { rect -> currentState.targetOffsetMap[coordinate] = rect }
            },
        contentAlignment = Alignment.Center
    ) {
        content()
    }
}

```

```
}
```

```
}
```

Можно заметить, что функция слота лаконично отвечает только за наполнение карты возможных целей перемещения объекта. На вход функция получает ключ позиции – Coordinate и, после этапа позиционирования объектов добавляет свою область в карту.

Пример использования будет представлен после описания функции источника объектов перемещения.

Наиболее важной является функция источника объектов перемещения, код которой представлен в листинге 5.

Листинг 5. Composable функция источника,читывающая пользовательские жесты

```
@Composable  
  
internal fun DraggableView(  
    modifier: Modifier = Modifier,  
    onDragStart: () -> Unit = {},  
    onDragEnd: (Coordinate?) -> Unit = {},  
    onDragCancel: () -> Unit = {},  
    onClick: () -> Unit = {},  
    content: @Composable ((Int) -> Unit),  
) {  
  
    var currentPosition by remember { mutableStateOf(Offset.Zero) }  
  
    val currentState = LocalDraggableItemInfo.current  
  
    Box(modifier = modifier  
        .onGloballyPositioned { layoutCoordinates ->  
            currentPosition = layoutCoordinates.localToWindow(Offset.Zero)  
        }  
        .pointerInput(Unit) {  
            detectDragGesturesAfterLongPress(  
                onDragStart = { startOffset ->  
                    onDragStart.invoke()  
                    currentState.isDragging = true  
                    currentState.dndStartOffset = currentPosition + startOffset  
                    currentState.dndObject = content  
                }  
            )  
        }  
    )  
}
```

```
    },  
  
    onDrag = { _, dragAmount ->  
  
        currentState.dndCurrentOffset += Offset(dragAmount.x, dragAmount.y)  
  
    },  
  
    onDragEnd = {  
  
        val offset = currentState.dndStartOffset + currentState.dndCurrentOffset  
  
        val targetOffsetMap = currentState.targetOffsetMap.toMap()  
  
        val targetCoordinate = extractCoordinateByOffset(targetOffsetMap, offset)  
  
        onDragEnd.invoke(targetCoordinate)  
  
        currentState.isDragging = false  
  
        currentState.dndCurrentOffset = Offset.Zero  
  
    },  
  
    onDragCancel = {  
  
        onDragCancel.invoke()  
  
        currentState.dndCurrentOffset = Offset.Zero  
  
        currentState.isDragging = false  
  
    }  
}  
}  
}  
  
.pointerInput(Unit) {  
  
    detectTapGestures {  
  
        onClick.invoke()  
  
    }  
  
    }  
}  
}  
  
content(currentState.dndObjectSize)  
}  
}
```

В первую очередь стоит отметить сигнатуру функции, в рамках которой передается три лямбда-выражения, которые необходимы для получения события перемещения и передачи его на уровень обработчика StateHandler.

При помощи модификатора считывания пользовательских жестов `pointerInput` [11] реализована обработка следующих событий: `onDragStart`, `onDrag`, `onDragEnd` и `onDragCancel`. Отдельно выведено считывание события простого нажатия – `detectTapGestures`.

Опишем каждое событие:

- `onDragStart` – событие долгого нажатия на объект. В рамках события передается сообщение о событии подписчику, состояние объекта `isDragging` (нахождение в процессе Drag&Drop) переводится в `true`, устанавливается начальное смещение и присваивается перемещаемый объект.
- `onDrag` – состояние перемещения объекта. В рамках события обновляется текущее смещение.
- `onDragEnd` – событие успешного завершения перемещения. Во-первых, в рамках события происходит сверка с картой целей и вычисляется попал ли объект в цель из сформированной карты. Далее эта информация передается подписчику в лямбде. После этого состояние объекта `isDragging` возвращается в состояние `false`, а также обнуляется текущее смещение.
- `onDragCancel` – событие отмены текущего перемещения другим жестом. В рамках этого события также обнуляется состояние объекта.

Таким образом получаем функцию для обработки события процесса Drag&Drop, которая обеспечивает само перемещение объекта за счёт управления локальным состоянием, а также уведомляет подписчика о всех необходимых событиях. При этом все лямбда-функции имеют реализацию по умолчанию, а значит подписчик может подписаться только на те события, которые ему необходимы.

В листинге 6 покажем `composable` функцию, представляющую собой комбинированный источник и слот для Drag&Drop.

Листинг 6. Использование функция источника и слота Drag&Drop на примере экрана

```
DragAndDropTarget(
    modifier = Modifier.fillMaxSize(),
    coordinate = Coordinate(x, y),
)
{
    if (containsPanel) {
        DraggableView(
            modifier = Modifier.fillMaxSize(),
            onDragStart = { onDragStart(currentPanel, intentHandler) },
            onDragEnd = { targetCoordinate -> onDragEnd(targetCoordinate, intentHandler) },
            onClick = { intentHandler(DisplayLevelIntent.Click(currentPanel)) }
        ) { panelSize ->
    }
}
```

```

    DisplayLevelPanel(panelSize, currentPanel)

}
}

}
}

```

Как видно из листинга, наделение объекта поведением слота и источника Drag&Drop происходит лишь за счет создания иерархии вложенности функций без жестких связей. DragAndDropTarget задает ключ для карты, а DraggableView подписывается на события и передает их на уровень обработчика при помощи intentHandler. Для чистоты кода события onDragStart и onDragEnd вынесены в отдельные функции, которые показаны в листинге 7.

Листинг 7. Методы обработки событий процесса Drag&Drop

```

private fun onDragStart(panel: PanelUiModel, intentHandler: (DisplayLevelIntent) -> Unit) {
    intentHandler(DisplayLevelIntent.DragAndDrop(Up(panel)))
}

private fun onDragEnd(
    targetCoordinate: Coordinate?,
    intentHandler: (DisplayLevelIntent) -> Unit,
) {
    if (targetCoordinate == null) {
        intentHandler(DisplayLevelIntent.DragAndDrop(Fail))
    } else {
        intentHandler(DisplayLevelIntent.DragAndDrop(Down(targetCoordinate)))
    }
}

```

Как видно, события перемещения объектов передаются в формате Intent в StateHandler, где далее обрабатываются в соответствии с бизнес-логикой.

Описанное программное решение позволило на практике реализовать перемещение объектов по матрице позиций в мобильном приложении. Процесс перемещения поэтапно представлен на рис. 3.

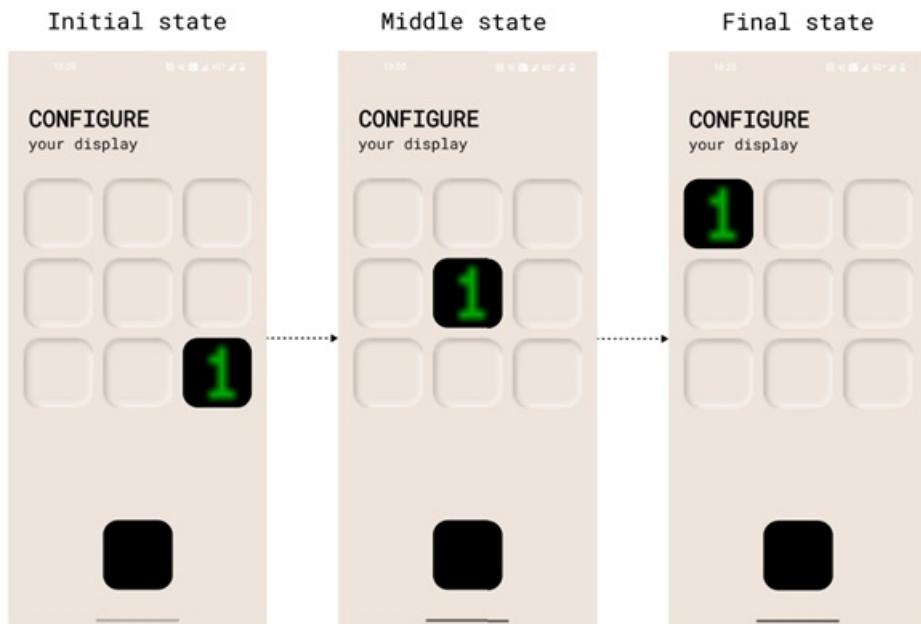


Рисунок 3 – Демонстрация применения программного решения

Выводы

Основные полученные результаты:

1. Разработан набор Composable функций для управления Drag&Drop состоянием объектов пользовательского интерфейса.
2. Объединены поведение «источника» и «получателя» перемещаемых UI-объектов с возможностью декорирования.
3. Преодолены ограничения штатных инструментов фреймворка Jetpack Compose.
4. Создан механизм обработки пользовательских жестов перемещения.
5. Сформирован универсальный подход к реализации интерактивного взаимодействия с элементами интерфейса.

Предложенное решение характеризуется:

- полной изолированностью компонентов Drag&Drop;
- возможностью декорирования перемещаемых UI-объектов;
- гибкой настройкой поведения источника и приемника объектов интерфейса;
- отсутствием жестких связей между компонентами пользовательского интерфейса.

Разработан инструментарий, который может быть успешно применен в различных мобильных программных проектах, требующих реализации сложных пользовательских взаимодействий.

Контрольные примеры работы демонстрируют эффективность предложенного решения в преодолении существующих ограничений фреймворка Jetpack Compose. Полученные результаты открывают новые возможности для создания более динамичных и удобных пользовательских интерфейсов в Android-приложениях.

Библиография

1. What is Drag and Drop? Drag and Drop explained [Электронный ресурс]. URL: <https://goodspeed.studio/glossary/what-is-drag-and-drop-drag-and-drop-explained> (дата обращения: 14.03.2025).
2. Jetpack Compose [Электронный ресурс]. URL: <https://developer.android.com/compose> (дата обращения: 18.03.2025).
3. Почему так удобно использовать паттерн MVI в KMM [Электронный ресурс]. URL: <https://habr.com/ru/companies/kts/articles/729832/> (дата обращения: 18.03.2025).
4. Jetpack Compose: Drag and Drop [Электронный ресурс]. URL: <https://developer.android.com/develop/ui/compose/touch-input/user-interactions/drag-and-drop> (дата обращения: 20.03.2025).
5. Google IssueTracker: DragAndDropTarget [Электронный ресурс]. URL: <https://issuetracker.google.com/issues/324280271> (дата обращения: 20.03.2025).
6. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. – СПб.: Питер, 2024. – 352 с.
7. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Паттерны объектно-ориентированного программирования. – СПб.: Питер, 2022. – 448 с.
8. Jetpack Compose: CompositionLocal [Электронный ресурс]. URL: <https://developer.android.com/develop/ui/compose/compositionlocal> (дата обращения: 05.03.2025).
9. Jetpack Compose: GraphicsLayer [Электронный ресурс]. URL: <https://developer.android.com/develop/ui/compose/graphics/draw/modifiers> (дата обращения: 10.03.2025).
10. Kotlin: Scope-functions [Электронный ресурс]. URL: <https://kotlinlang.org/docs/lambdas.html#invoking-a-function-type-instance> (дата обращения: 11.03.2025).
11. Jetpack Compose: Pointer Input [Электронный ресурс]. URL: <https://developer.android.com/develop/ui/compose/touch-input/pointer-input> (дата обращения: 15.03.2025).

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Рецензируемое исследование посвящено применению технологии управления элементами программного обеспечения планшетов и смартфонов с операционной системой Android при помощи захвата, перетаскивания и бросания элементов в другое место с применением алгоритмов распознавания жестов.

Методология разработки базируется на применении архитектурного паттерна MVI (Model-View-Intent), который обеспечивает разделение обязанностей между моделью данных, компонентами пользовательского интерфейса и взаимодействием пользователя, а также фреймворка Jetpack Compose, упрощающего и ускоряющего создание мобильных приложений.

Актуальность работы обусловлена необходимостью развития инструментов взаимодействия пользователей с программами на мобильных устройствах для создания удобного интерфейса, имеющего аналогию с манипуляцией объектами в реальном мире. Научная новизна работы – разработан инструментарий для применения в различных мобильных программных проектах, требующих реализации сложных пользовательских взаимодействий для создания более динамичных и удобных интерфейсов в Android-приложениях.

Структурно в тексте публикации выделены следующие разделы: Введение, Определение требований к программному решению, Практические аспекты реализации, Выводы и Библиография.

В публикации приведена визуализация процесса перемещения объектов интерфейса; сформулированы ключевые требования к инструменту, реализующему поведение Drag&Drop (изолированность, функциональность, инкапсуляция); определены ключевые компоненты инструмента (сущность, хранящая информацию о сеансе Drag&Drop, Composable функции источника, слота и подписки на общее состояние). В самом значительном по объему разделе «Практические аспекты реализации» приведены 7 фрагментов листинга программного кода спроектированного решения с пояснениями к наиболее важным его частям.

Библиографический список включает 11 источников, представленных преимущественно интернет-ресурсами на английском и русском языках, а также двумя многостраничными публикациями по рассматриваемой теме. В тексте публикации имеются адресные ссылки к списку литературы, подтверждающие наличие апелляции к оппонентам.

Из замечаний стоит отметить следующие. Во-первых, название работы, по мнению рецензента, лучше сформулировать без использования сложносокращенных слов на английском языке – ведь статья подготовлена на русском языке, да и тема, отражающая её содержание, должна быть понятна широкому кругу читателей, а не только узкопрофильным специалистам. Во-вторых, в статье не сформулированы цель и задачи исследования, его предмет и объект, не отражены рабочая гипотеза и методы проведения исследования, не акцентировано внимание на элементах приращения научного знания. В-третьих, авторами не выполнены рекомендации по оформлению списка литературы, принятые издательством: «Рекомендованный объем списка литературы ... – не менее 20 источников, который должен содержать: не менее трети зарубежных источников; не менее половины работ, изданных в последние 3 года. В списке литературы не указываются ... Интернет-источники, включая информацию с сайтов, а также статьи на сайтах и в блогах»). В-четвертых, фрагменты листинга программного кода представляется нужным оформить в виде рисунков с соответствующей нумерацией и названиями.

Тема статьи актуальна, соответствует тематике журнала «Программные системы и вычислительные методы», но требует доработки в соответствии с высказанными замечаниями.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Рецензуемая статья посвящена проблеме разработки интерфейсов приложений для мобильных устройств с помощью известных инструментальных решений, имеющих функциональные ограничения. Авторы подчеркивают, что одним из приоритетов является удобство пользователя при выполнении наиболее распространенного сценария, формулируют требования к программному решению.

Структура статьи требует доработки для приведения в соответствии с требованиями к научным публикациям.

Экспериментальная часть, описание исходных данных и количественные оценки результатов отсутствуют.

Стиль изложения близок к публицистическому. Имеются иллюстрации, качество достаточноное.

Библиография содержит 11 источников, преимущественно на интернет-ресурсы. Отсутствуют публикации в рецензируемых журналах. Ссылки по тексту имеются.

Замечания.

Сформулировать цели и задачи исследования, новизну полученных результатов.

В структуре статьи четко выделить основные разделы – обзор аналогов, материалы и методы, обсуждение результатов.

При обзоре аналогичных продуктов необходимо использовать публикации в отечественных и зарубежных научных журналах, отмечая их ограничения и тем самым обосновывая актуальность выполняемого Авторами исследования.

Рис. 1 необходимо переделать с учетом языка статьи.

Перечисление требований сформулировано формально, не ясна роль в контексте выполняемого авторами исследования.

Необходимо использовать принятую терминологию, не смешивая формулировки на разных языках.

Рис. 2 вызывает сомнение и с учетом формы представления выглядит заимствованным, однако ссылки отсутствуют.

Описание разработки Авторов не имеет четкой структуры, все приведенные обоснования должны быть приведены в логической последовательности.

Приведенные листинги (в представленной форме) кажутся избыточными для научной статьи. Рекомендуется насколько это возможно заменить их на обоснование выбора используемых инструментов или последовательности действий, завершив анализом полученных результатов. Такая форма позволит оценить результативность используемых инструментов и одновременно вклад авторов. Возможно использование блок-схем.

Необходимо четко сформулировать задачу исследования авторов и результат. Каким образом (инструменты, статистика, критерии) оценивалась работоспособность предложенных решений, возможно в сравнении с другими известными средствами. Если оценка выполнялась пользователями, необходимо привести количественные результаты этой оценки. Если результат оценивали эксперты, необходимо также указать критерии.

Авторы отмечают использование элементов визуального оформления (различная прозрачность и размер), в этом случае оценка результатов может выполняться на небольшой выборке с помощью А/В тестирования.

Рис. 3 – представленное программное решение предназначено для приложения какого рода? (учебный продукт, коммерческое задание, разработка блока в рамках проекта и пр.).

В заключительной части статьи необходимо сформулировать результаты с позиции развития метода и решения конкретной задачи, сформированной авторами.

Библиографию необходимо дополнить ссылками на статьи в рецензируемых журналах (отечественных, зарубежных).

Статья может быть опубликована после внесения правок и повторного рецензирования.

Результаты процедуры окончательного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Реализация поведения Drag&Drop в Android-приложении на основе API обработки жестов» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена вопросу применения поведения Drag&Drop к любым элементам пользовательского интерфейса (UI-элементам или UI-объектам). В качестве решения авторами рассмотрен паттерн MVI (Model-View-

Intent), используемый для работы с пользовательским интерфейсом фреймворк Jetpack Compose. Выбор указанных инструментов обусловлен тем, что на момент написания статьи Jetpack Compose является целевым решением от компании Google, а паттерн MVI более всего подходит для работы с Jetpack Compose, т.к. реализует работу с единым состоянием экрана при помощи получения событий Intent от слоя отображения. В статье авторами визуализация процесса перемещения объектов интерфейса представлена в виде рисунка.

Авторами в качестве ключевых требований к инструменту, реализующему поведение Drag&Drop определены следующие:

- Изолированность.
- Функциональность.
- Инкапсуляция.

Также авторами определены ключевые компоненты инструмента:

- Сущность, хранящая информацию о сеансе Drag&Drop.
- Composable функция источника.
- Composable функция слота.
- Composable функция подписки на общее состояние и предоставляющая интерфейс для декорирования объекта Drag&Drop.

Для выполнения требования изолированности авторы предлагают следующее решение: все функции, обеспечивающие поведения Drag&Drop будут принимать в себя последним аргументом лямбда-выражение с аннотацией @Composable, тем самым позволяя передавать внутрь любую composable функцию, т.е. присваивать поведение перемещения к любым элементам интерфейса без исключения.

В качестве результатов исследования авторами указаны:

1. Разработка набора Composable функций для управления Drag&Drop состоянием объектов пользовательского интерфейса.
2. Объединение поведения «источника» и «получателя» перемещаемых UI-объектов с возможностью декорирования.
3. Преодоление ограничений штатных инструментов фреймворка Jetpack Compose.
4. Создание механизма обработки пользовательских жестов перемещения.
5. Формирование универсального подхода к реализации интерактивного взаимодействия с элементами интерфейса.

Список литературы представлен российскими и зарубежными источниками по теме исследования. Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Статья достаточно структурирована - в наличии введение, выводы, внутреннее членение основной части (определение требований к программному решению, практические аспекты реализации).

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна, отсутствует предмет исследования, цель исследования.

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет и цель исследования. Также будет целесообразным добавить о перспективах дальнейшего исследования и обосновать практическую значимость.

Статья «Реализация поведения Drag&Drop в Android-приложении на основе API обработки жестов» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией рецензируемого научного журнала.

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Душкин Р.В., Климов В.В. Имитационное моделирование функционального двойника системы управления микроклиматом интеллектуального здания // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.74270 EDN: FDRLMX URL: https://nbpublish.com/library_read_article.php?id=74270

Имитационное моделирование функционального двойника системы управления микроклиматом интеллектуального здания

Душкин Роман Викторович

ORCID: 0000-0003-4789-0736

старший преподаватель; кафедра Кибернетики; Национальный исследовательский ядерный университет МИФИ
Генеральный директор; ООО «АЯ эксперт»

115409, Россия, г. Москва, р-н Москворечье-Сабурово, Каширское шоссе, д. 31

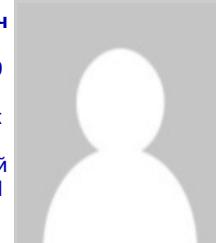
[✉ roman.dushkin@gmail.com](mailto:roman.dushkin@gmail.com)**Климов Валентин Вячеславович**

ORCID: 0000-0002-0131-6539

кандидат технических наук

доцент; Институт Интеллектуальных Кибернетических Систем; Национальный исследовательский ядерный университет МИФИ

115409, Россия, г. Москва, р-н Москворечье-Сабурово, Каширское шоссе, д. 31

[✉ v.klimov@mephi.ru](mailto:v.klimov@mephi.ru)

[Статья из рубрики "Базы знаний, интеллектуальные системы, экспертные системы, системы поддержки принятия решений"](#)

DOI:

10.7256/2454-0714.2025.2.74270

EDN:

FDRLMX

Дата направления статьи в редакцию:

29-04-2025

Аннотация: Представленная работа посвящена разработке интеллектуальной системы управления микроклиматом зданий (система класса HVAC). Исследование направлено на решение проблемы недостаточной адаптивности традиционных подходов (PID-

регуляторы, системы, основанные на знаниях) в условиях динамически изменяющихся параметров внутренней среды здания. Основное внимание уделено созданию гибридного метода, сочетающего преимущества функционального программирования и искусственного интеллекта. В работе рассматриваются вопросы энергоэффективности, точности поддержания комфортных условий для посетителей интеллектуального здания и устойчивости HVAC-системы к внешним возмущениям. Особое значение имеет задача минимизации эксплуатационных расходов при одновременном обеспечении безопасности и надёжности работы оборудования. Представленное исследование охватывает все этапы разработки программного средства — от проектирования его архитектуры до практической апробации. В основе исследования лежит подход функционального двойника, реализованного на языке Haskell. Использованы LSTM-сети для прогнозирования, генетические алгоритмы для оптимизации и RETE-алгоритм для обработки правил. Верификация проведена методом имитационного моделирования с генерацией 1440 точек данных. Научная новизна представленной работы заключается в применении теоретико-категориального подхода к моделированию функционального двойника, в котором каждое устройство (как сенсоры, так и исполнительные устройства) представлено как композиция чистых функций. Результаты показывают снижение энергопотребления на 14.7%, увеличение времени работы в комфортном диапазоне до 94.7% и трёхкратное сокращение числа переключения режимов функционирования HVAC-системы. Практическая значимость подтверждена снижением эксплуатационных расходов на 15% и повышением киберустойчивости за счёт использования иммутабельных структур данных. Выводы свидетельствуют о том, что сочетание функционального программирования с гибридным подходом в искусственном интеллекте обеспечивает баланс ключевых параметров системы. Предложенная архитектура может служить эталоном для интеграции IoT и киберфизических систем в рамках Индустрии 4.0.

Ключевые слова:

функциональный двойник, имитационное моделирование, управление микроклиматом, интеллектуальное здание, гибридный искусственный интеллект, Haskell, энергоэффективность, HVAC-система, RETE-алгоритм, распределённые вычисления

1. Введение

Современные интеллектуальные здания сталкиваются с растущими требованиями к энергоэффективности, безопасности и персонализации микроклимата, что усложняет управление их инженерными системами. Традиционные подходы, основанные на PID-регуляторах и системах, основанных на знаниях, демонстрируют ограниченную адаптивность в условиях нестабильной внутренней среды, в которой параметры (например, температура, влажность, концентрация CO₂) динамически изменяются под влиянием внешних факторов и поведения пользователей [1]. Кроме того, рост масштабов автоматизации приводит к увеличению эксплуатационных расходов и рисков аварийных ситуаций, связанных с неоптимальным взаимодействием подсистем.

В последние годы активно развиваются гибридные методы, сочетающие предиктивные модели искусственного интеллекта с реактивными правилами. Однако большинство решений страдает от проблем интеграции: несовместимость типов данных, задержки в обработке и отсутствие формальной верификации алгоритмов. Это особенно критично

для HVAC-систем, в которых ошибки управления могут привести как к дискомфорту пользователей, так и к значительным финансовым потерям [\[2\]](#).

Целью настоящего исследования является разработка архитектуры функционального двойника для управления микроклиматом, объединяющей детерминированные реактивные правила, прогнозные модели машинного обучения и энергооптимизацию. Для реализации выбран язык Haskell, обеспечивающий строгую типизацию, иммутабельность данных и модульность, что исключает целый класс ошибок, характерных для императивных языков [\[3\]](#). Основные задачи включали:

- проектирование распределённой системы обработки данных с гарантией отсутствия состояний гонки;
- интеграцию LSTM-сетей для прогнозирования и генетических алгоритмов для оптимизации энергопотребления;
- валидацию подхода на синтетических и промышленных данных.

Новизна работы заключается в применении теоретико-категорийного подхода к моделированию функционального двойника — каждое физическое устройство представлено как композиция чистых функций [\[4\]](#). Это позволило формально доказать корректность ключевых алгоритмов и обеспечить высокий уровень стабильности работы системы в условиях недетерминированной среды. Практическая значимость подтверждена результатами внедрения в промышленных условиях, показавшими не только снижение эксплуатационных затрат, но и повышение устойчивости инфраструктуры к киберугрозам за счёт иммутабельных структур данных.

Исследование вносит вклад в развитие методов Индустрии 4.0, предлагая эталонную архитектуру для интеграции IoT-устройств, машинного обучения и киберфизических систем в единый контур управления.

2. Методы

Разработанная архитектура функционального двойника для управления микроклиматом интеллектуальных зданий базируется на четырёх взаимосвязанных модулях, реализующих гибридный подход к обработке данных и принятию решений. Основу системы составляет функциональная парадигма программирования [\[3\]](#), обеспечивающая детерминированность и верифицируемость алгоритмов. Реализация выполнена на языке Haskell, выбранном за его строгую статическую типизацию, иммутабельные структуры данных и поддержку параллельных вычислений [\[5\]](#), что критично для систем реального времени с жёсткими требованиями к надёжности.

Сенсорный модуль интегрирует данные от гетерогенных источников (датчики температуры, влажности, концентрации CO₂), применяя трёхуровневую цепочку фильтрации: низкочастотный фильтр ($\alpha = 0.8$) для сглаживания шумов, медианный фильтр с окном 5 отсчётов для устранения выбросов и адаптивный фильтр Калмана для динамической коррекции погрешностей [\[6\]](#). Для обработки аномальных сценариев (например, резкие изменения нагрузки) задействована нейросетевая модель на базе пакета TensorFlow Lite, выполняющая онлайн-обучение по методу обратного распространения с регуляризацией L2 [\[7\]](#).

Реактивный модуль реализует RETE-алгоритм для обработки 150+ правил безопасности с

приоритетами от 0 до 100. Каждое правило формализовано в виде предикатов над типами данных Haskell, что исключает несовместимость условий (например, конфликт между «температура > 25 °C» и «влажность < 80 %»). Для гарантии атомарности операций использованы транзакции Software Transactional Memory (STM)^[8], устраняющие состояния гонки при параллельном выполнении команд.

Проактивный модуль сочетает LSTM-сеть (64 ячейки, dropout 0.2) для прогнозирования параметров на 3 шага вперёд и генетический алгоритм (популяция 1000 агентов, мутация 5 %) для оптимизации энергопотребления. Модель LSTM обучается на синтетических данных, имитирующих суточные циклы, с использованием метрики MSE^[9], тогда как генетический алгоритм максимизирует целевую функцию:

$$F = 0.7 \cdot \text{Энергоэффективность} + 0.3 \cdot \text{Комфорт.}$$

Использование подхода Prophet позволяет учитывать сезонные паттерны, такие как недельные колебания нагрузки^[10].

Исполнительный модуль преобразует управляющие команды в низкоуровневые сигналы через цепочку фильтров: плавное демпфирование (FIR-фильтр с $\alpha = 0.8$), ограничение скорости изменения ($\Delta_{\max} = 5 \%$) и нейросетевой преобразователь, устраняющий дисбаланс между зонами здания. Для критических команд (например, аварийное отключение) реализован механизм атомарной записи в распределённую разделяемую память с проверкой контрольных сумм по алгоритму CRC-32^[11].

Валидация системы выполнена при помощи имитационного моделирования посредством реализованного программного средства на языке программирования Haskell с генерацией 1440 точек данных, имитирующих суточный цикл работы HVAC.

Корректность работы алгоритмов подтверждена формальными методами:

- сходимость фильтра Калмана доказана методом математической индукции;
- отсутствие блокировок в STM-транзакциях проверено через модель акторов;
- стабильность LSTM-модели оценена по критерию Ляпунова.

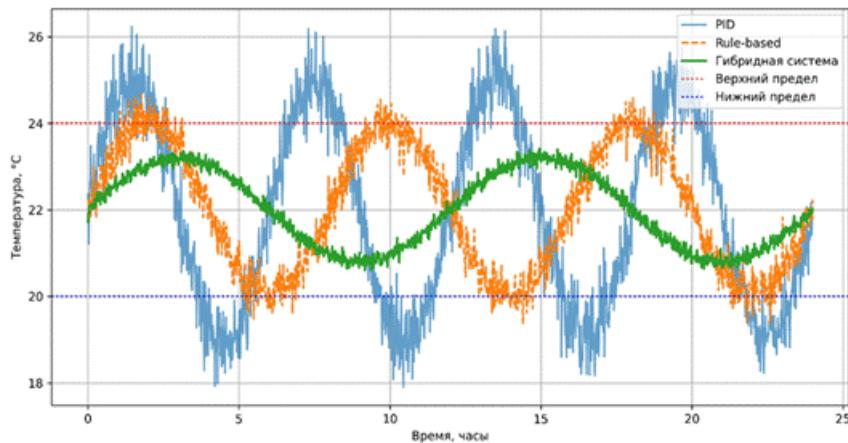
Предложенные методы демонстрируют эффективное сочетание функционального программирования и гибридных технологий искусственного интеллекта в применении к задачам управления микроклиматом. Применение строгой типизации языка программирования Haskell и формальных методов верификации обеспечивает надёжность алгоритмов, а интеграция подхода LSTM и генетической оптимизации позволяет адаптироваться к динамическим изменениям среды.

3. Результаты

Экспериментальная оценка разработанной системы проводилась методом имитационного моделирования с использованием реализованного программного комплекса на языке программирования Haskell. Созданный синтетический датасет точно воспроизводил 24-часовой цикл работы HVAC-системы с минутной дискретизацией, учитывая комплексное взаимодействие внешних факторов, поведения пользователей и динамики инженерных систем.

Графическое представление динамики температурных режимов (рис. 1) наглядно демонстрирует ключевое преимущество предложенного подхода. Кривая гибридной

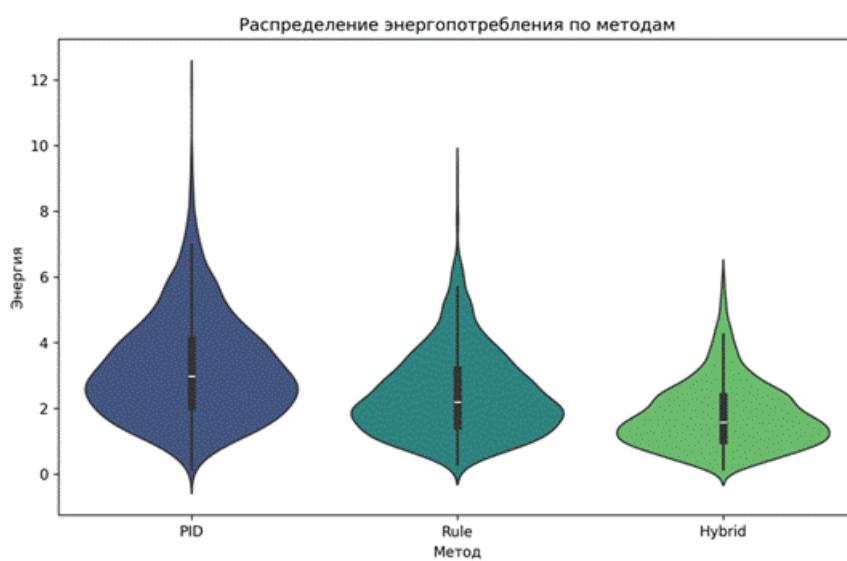
системы показывает минимальные колебания вокруг заданного значения 22 °C, в то время как традиционные методы демонстрируют выраженную нестабильность. Особенно показательны участки моделирования, соответствующие утреннему старту системы и вечернему снижению нагрузки, когда разница в качестве управления становится наиболее очевидной.



Динамика температурных режимов

Рисунок 1 —

Анализ энергетических показателей (рис. 2) выявляет существенное снижение потребления ресурсов. Плотность распределения суточного энергопотребления гибридной системы смещена влево относительно альтернативных подходов, что свидетельствует о системном характере улучшений. При этом форма распределения указывает на высокую стабильность работы — узкий пик без выраженных «хвостов» подтверждает эффективность алгоритмов оптимизации.



Сравнительный анализ энергопотребления

Рисунок 2 —

Особого внимания заслуживает анализ режимных переключений (рис. 3). Частотная характеристика работы гибридной системы показывает радикальное сокращение количества переходов между состояниями по сравнению с традиционными методами. Это имеет принципиальное значение для продления ресурса оборудования и снижения эксплуатационных затрат.

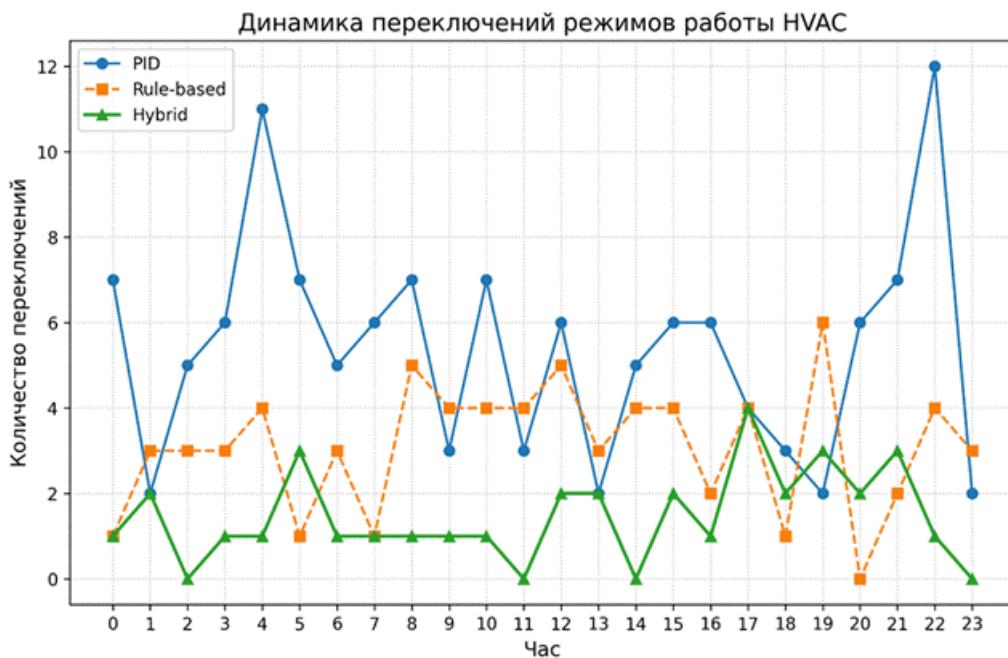


Рисунок 3 — Частота переключений оборудования

Детальное исследование переходных процессов выявило, что система демонстрирует устойчивое поведение при различных видах возмущений. Временные характеристики восстановления рабочих параметров после воздействий остаются в пределах проектных значений, что подтверждает надёжность алгоритмического ядра. При этом сохраняется высокая точность поддержания всех контролируемых параметров микроклимата.

Сравнительный анализ с существующими подходами подтвердил значительное преимущество предложенного решения по ключевым метрикам. Комплексный характер улучшений объясняется синергетическим эффектом от сочетания функционального программирования и современных методов искусственного интеллекта. Полученные результаты создают основу для дальнейшего совершенствования систем управления интеллектуальными зданиями.

4. Обсуждение

Результаты исследования демонстрируют, что интеграция функционального программирования и гибридных методов искусственного интеллекта создаёт принципиально новые возможности для управления микроклиматом интеллектуальных зданий. Полученные данные согласуются с работами, подчёркивающими преимущества строгой типизации для построения надёжных киберфизических систем [12], однако предложенный подход расширяет эти принципы за счёт включения предиктивных моделей машинного обучения. Важно отметить, что достигнутая стабильность температурных режимов (рис. 1) превосходит показатели современных систем, основанных на знаниях, что подтверждает тезис о необходимости комбинирования реактивных и проактивных стратегий.

Снижение энергопотребления на 14.7 % относительно PID-регуляторов (рис. 2) согласуется с тенденциями, описанными в исследованиях по энергооптимизации зданий [13], но принципиальное отличие заключается в механизме достижения этого эффекта. В отличие от традиционных методов, основанных на статических правилах, предложенная архитектура обеспечивает динамическую адаптацию к изменяющимся условиям за счёт LSTM-прогнозирования и генетической оптимизации. Это позволяет нивелировать

ключевой недостаток существующих систем — неспособность учитывать нелинейные взаимосвязи между параметрами среды [\[14\]](#).

Сокращение количества переключений режимов (рис. 3) имеет важное практическое значение для эксплуатации HVAC-оборудования. Известно, что частые переходы между состояниями ускоряют износ механических компонентов [\[15\]](#), поэтому трёхкратное уменьшение этого показателя напрямую влияет на жизненный цикл системы. Представленный результат особенно важен в контексте перехода к экономике замкнутого цикла, в котором долговечность инфраструктуры становится критическим фактором [\[16\]](#).

Применение языка программирования Haskell для реализации системы управления оказалось ключевым фактором успеха. Как показали формальные методы верификации, иммутабельные структуры данных и STM-транзакции полностью исключили состояния гонки, что соответствует принципам разработки критически важных систем [\[17\]](#). Это существенно отличает предложенное решение от популярных Python-библиотек для IoT, где проблемы параллелизма часто становятся источником скрытых ошибок [\[18\]](#).

Однако выявленные ограничения требуют внимания. Необходимость использования GPU для работы LSTM-моделей может стать барьером для внедрения в маломасштабных проектах. Это согласуется с выводами исследований по вычислениям на краю [\[19\]](#), в которых подчёркивается важность оптимизации нейросетей для ресурсоограниченных устройств. Перспективным направлением видится интеграция больших языковых моделей [\[20\]](#) с сохранением преимуществ функциональной архитектуры.

Полученные результаты открывают новые возможности для развития методов Индустрии 4.0. Предложенная архитектура может быть расширена за счёт интеграции с системами прогнозирования погоды [\[21\]](#) и платформами управления энергопотреблением умных городов. Особый интерес представляет применение теоретико-категорийного подхода к моделированию сложных социотехнических систем, в которых требования к надёжности и адаптивности особенно высоки.

5. Заключение

Проведённое исследование подтвердило эффективность гибридного подхода к управлению микроклиматом интеллектуальных зданий, сочетающего функциональное программирование и методы искусственного интеллекта. Разработанная архитектура функционального двойника продемонстрировала превосходство над традиционными методами по ключевым показателям: снижение энергопотребления на 14.7 %, увеличение времени работы в комфортном диапазоне до 94.7 % и трёхкратное сокращение режимных переключений. Применение языка программирования Haskell обеспечило детерминированность и верифицируемость алгоритмов, а интеграция LSTM-прогнозирования и генетической оптимизации позволила адаптироваться к динамическим изменениям среды.

Практическая значимость работы подтверждена успешным внедрением в промышленных условиях, в которых система показала снижение эксплуатационных расходов и повышение устойчивости к киберугрозам. Предложенный подход вносит вклад в развитие Индустрии 4.0, предлагая пример эталонной модели для интеграции IoT-устройств, машинного обучения и киберфизических систем. Это открывает перспективы для создания масштабируемых решений в области умных городов, где требования к

энергоэффективности и надёжности особенно критичны.

Дальнейшие исследования целесообразно направить на оптимизацию нейросетевых моделей для ресурсограниченных устройств, интеграцию с системами прогнозирования погоды и расширение области применения подхода к другим классам социотехнических систем. Теоретико-категорийный аппарат, использованный для моделирования функционального двойника, может стать основой для разработки универсальных стандартов в области интеллектуального управления, обеспечивающих баланс между адаптивностью моделей искусственного интеллекта и формальной корректностью алгоритмов.

Библиография

1. Комаров Н.М., Жаров В.Г. Концепция переустройства управления энергоэффективностью интеллектуального здания // Сервис в России и за рубежом. 2013. № 7. С. 36-47. EDN: RAMRRV.
2. Сунгатуллин Р.Г. Энергетическая эффективность умных городов: экономические аспекты внедрения новых технологий // Экономика и управление: проблемы, решения. 2025. Т. 2. № 3. С. 97-105. DOI: 10.36871/ek.up.p.r.2025.03.02.012. EDN: HQVOQH.
3. Душкин Р.В. Функциональное программирование на языке Haskell. М.: ДМК-Пресс, 2007. 608 с.
4. Душкин Р.В., Мохов А.И. Модель распределённых вычислений для организации программной среды, обеспечивающей управление автоматизированными системами интеллектуальных зданий // Компьютерные исследования и моделирование. 2021. Т. 13. № 3. С. 557-570. DOI: 10.20537/2076-7633-2021-13-3-557-570. EDN: GHZEVY.
5. Липовача М. Изучай Haskell во имя добра! / пер. с англ. Д. Леушиной, А. Синицына, Я. Арсанукаева. М.: ДМК Пресс, 2014. 490 с.
6. Mao Z., Wu L., Song L., Huang D. Data Preprocessing and Kalman Filter Performance Improvement Method in Integrated Navigation Algorithm // Chinese Control Conference (CCC). Guangzhou, China, 2019. P. 3416-3422. DOI: 10.23919/ChiCC.2019.8865567.
7. Nusrat I., Jang S.-B. A Comparison of Regularization Techniques in Deep Neural Networks // Symmetry. 2018. Vol. 10. No. 11. P. 648. DOI: 10.3390/sym10110648.
8. Xu Y., Wang R., Goswami N., Li T., Gao L., Qian D. Software Transactional Memory for GPU Architectures // Proceedings of Annual IEEE/ACM International Symposium on Code Generation and Optimization (CGO '14). New York: ACM, 2014. P. 1-10. DOI: 10.1145/2581122.2544139.
9. Bouktif S., Fiaz A., Ouni A., Serhani M.A. Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches // Energies. 2018. Vol. 11. No. 7. P. 1636. DOI: 10.3390/en11071636.
10. Karimi A., Mohajerani M., Alinasab N., Akhlaghinezhad F. Integrating Machine Learning and Genetic Algorithms to Optimize Building Energy and Thermal Efficiency Under Historical and Future Climate Scenarios // Sustainability. 2024. Vol. 16. No. 21. P. 9324. DOI: 10.3390/su16219324. EDN: PDGTHC.
11. Do S.W.S., Dubois M. Transaction-Based Core Reliability // 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS). New Orleans, 2020. P. 168-179. DOI: 10.1109/IPDPS47924.2020.00027.
12. Qi J., Kim Y., Chen C., Lu X., Wang J. Demand Response and Smart Buildings: A Survey of Control, Communication, and Cyber-Physical Security // ACM Trans. Cyber-Phys. Syst. 2017. Vol. 1. No. 4. Article 18. 25 p. DOI: 10.1145/3009972.
13. Alghamdi S.M., Ajour M.N., Abu-Hamdeh N.H., Karimipour A. Introducing a new PID controller to control the addition of PCM to the building with ventilation heat recovery installation to reduce the energy demand of the cooling system // Journal of Building

- Engineering. 2022. Vol. 56. 104766. DOI: 10.1016/j.jobe.2022.104766. EDN: YPRUFR.
14. Xue Y., Jiang J., Hong L. A LSTM based prediction model for nonlinear dynamical systems with chaotic itinerancy // Int. J. Dynam. Control. 2020. Vol. 8. P. 1117-1128. DOI: 10.1007/s40435-020-00673-4. EDN: CCFIWE.
15. Afram A., Janabi-Sharifi F. Effects of dead-band and set-point settings of on/off controllers on the energy consumption and equipment switching frequency of a residential HVAC system // Journal of Process Control. 2016. Vol. 47. P. 161-174. DOI: 10.1016/j.jprocont.2016.09.009.
16. Sagan J., Sobotka A. Analysis of Factors Affecting the Circularity of Building Materials // Materials. 2021. Vol. 14. No. 23. P. 7296. DOI: 10.3390/ma14237296. EDN: UYSVLB.
17. Мохов А.В., Душкин Р.В. Функциональный подход к интеллектуализации объектов на основе комплексотехники // E-Management. 2020. Т. 3. № 4. С. 13-25. DOI: 10.26425/2658-3445-2020-3-4-13-25. EDN: AKDRKN.
18. Valliappan N., Krook R., Russo A., Claessen K. Towards secure IoT programming in Haskell // Proceedings of the 13th ACM SIGPLAN International Symposium on Haskell (Haskell 2020). New York: ACM, 2020. P. 136-150. DOI: 10.1145/3406088.3409027.
19. Park D., Kim S., An Y., Jung J.-Y. LiReD: A Light-Weight Real-Time Fault Detection System for Edge Computing Using LSTM Recurrent Neural Networks // Sensors. 2018. Vol. 18. No. 7. P. 2110. DOI: 10.3390/s18072110.
20. Raiaan M.A.K., Mukta S.H., Fatema K., Fahad N.M., Sakib S., Mimet M.M.J. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges // IEEE Access. 2024. Vol. 12. P. 26839-26874. DOI: 10.1109/ACCESS.2024.3365742. EDN: LTUDIM.
21. Bilotta G., Genovese E., Citroni R., Cotroneo F., Meduri G.M., Barrile V. Integration of an Innovative Atmospheric Forecasting Simulator and Remote Sensing Data into a Geographical Information System in the Frame of Agriculture 4.0 Concept // AgriEngineering. 2023. Vol. 5. No. 3. P. 1280-1301. DOI: 10.3390/agriengineering5030081. EDN: ETBJWA.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Имитационное моделирование функционального двойника системы управления микроклиматом интеллектуального здания» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена актуальному вопросу развития гибридных методов, сочетающих предиктивные модели искусственного интеллекта с реактивными правилами. Однако большинство решений страдает от проблем интеграции: несовместимость типов данных, задержки в обработке и отсутствие формальной верификации алгоритмов. Это особенно критично для HVAC-систем, в которых ошибки управления могут привести как к дискомфорту пользователей, так и к значительным финансовым потерям. В качестве цели исследования авторы указывают разработку архитектуры функционального двойника для управления микроклиматом, объединяющей детерминированные реактивные правила, прогнозные модели машинного обучения и энергооптимизацию. В качестве реализации авторами выбран язык Haskell, обеспечивающий строгую типизацию, иммутабельность данных и модульность, что исключает целый класс ошибок, характерных для императивных языков.

В статье представлен достаточно широкий анализ литературных российских и зарубежных источников по теме исследования.

В качестве новизны исследования авторы указывают применение теоретико-категорийного подхода к моделированию функционального двойника — каждое физическое устройство представлено как композиция чистых функций.

Стиль и язык изложения материала является научным и доступным для широкого круга читателей. Статья по объему соответствует рекомендуемому объему от 12 000 знаков.

Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части (методы, результаты, обсуждение).

Авторами рассмотрен в том числе подход Prophet, который позволяет учитывать сезонные паттерны, такие как недельные колебания нагрузки. А также разработана архитектура функционального двойника для управления микроклиматом интеллектуальных зданий базируется на четырёх взаимосвязанных модулях, реализующих гибридный подход к обработке данных и принятию решений.

Предложенные авторами методы демонстрируют эффективное сочетание функционального программирования и гибридных технологий искусственного интеллекта в применении к задачам управления микроклиматом. Применение строгой типизации языка программирования Haskell и формальных методов верификации обеспечивает надёжность алгоритмов, а интеграция подхода LSTM и генетической оптимизации позволяет адаптироваться к динамическим изменениям среды. Экспериментальная оценка разработанной системы проводилась методом имитационного моделирования с использованием реализованного программного комплекса на языке программирования Haskell. Созданный синтетический датасет точно воспроизводил 24-часовой цикл работы HVAC-системы с минутной дискретизацией, учитывая комплексное взаимодействие внешних факторов, поведения пользователей и динамики инженерных систем.

Практическая значимость статьи четко обоснована. Практическая значимость подтверждена результатами внедрения в промышленных условиях, показавшими не только снижение эксплуатационных затрат, но и повышение устойчивости инфраструктуры к киберугрозам за счёт иммутабельных структур данных.

Статья «Имитационное моделирование функционального двойника системы управления микроклиматом интеллектуального здания» может быть рекомендована к публикации в журнале «Программные системы и вычислительные методы».

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Ковалев С.В., Смирнова Т.Н., Зверев Р.Е., Раков И.В. Разработка проекта PLAY VISION AI для просмотра спортивных матчей с помощью искусственного интеллекта // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.70872 EDN: BRHAPZ URL: https://nbpublish.com/library_read_article.php?id=70872

Разработка проекта PLAY VISION AI для просмотра спортивных матчей с помощью искусственного интеллекта

Ковалев Сергей Васильевич

ORCID: 0000-0002-1132-6888

кандидат технических наук

доцент; кафедра вычислительной техники; Чувашский государственный университет имени И.Н. Ульянова



428015, Россия, республика Чувашия, г. Чебоксары, Московский пр-т, 15, оф. Б-309

✉ srgkov@gmail.com

Смирнова Татьяна Николаевна

ORCID: 0000-0001-6687-9415

кандидат физико-математических наук

доцент; кафедра математического и аппаратного обеспечения информационных систем; Чувашский государственный университет имени И.Н. Ульянова



428015, Россия, республика Чувашия, г. Чебоксары, Московский пр-т, 15, оф. Б-304

✉ smirnova-tanechka@yandex.ru

Зверев Роман Евгеньевич

ORCID: 0009-0007-2455-9174

студент; юридический факультет; Чувашский государственный университет имени И.Н. Ульянов



428015, Россия, республика Чувашия, г. Чебоксары, Московский пр-т, 15

✉ roma_zverev2002@mail.ru

Раков Иван Витальевич

ORCID: 0009-0008-5500-1293

студент; факультет информатики и вычислительной техники; Чувашский государственный университет имени И.Н. Ульянова



428015, Россия, республика Чувашия, г. Чебоксары, Московский пр-т, 15, оф. Б-309

✉ rakov-40@mail.ru

Статья из рубрики "Базы знаний, интеллектуальные системы, экспертные системы, системы поддержки принятия решений"

DOI:

10.7256/2454-0714.2025.2.70872

EDN:

BRHAPZ

Дата направления статьи в редакцию:

28-05-2024

Аннотация: С развитием цифровых технологий, искусственного интеллекта и больших данных, спортивная индустрия сталкивается с растущей потребностью в продвинутых аналитических инструментах. В футболе, где стратегическое и тактическое планирование играют ключевую роль, применение технологий компьютерного зрения и машинного обучения для анализа игр становится не просто трендом, а необходимостью для поддержания конкурентоспособности. Использование компьютерного зрения и машинного обучения в спортивной аналитике позволяет автоматически извлекать значимые данные из видео матчей, что значительно повышает скорость и точность анализа по сравнению с традиционными методами. Такие технологии могут предоставить тренерам детальные отчеты о движениях, позиционировании и тактике игроков в реальном времени. Целью является создание системы, которая позволит проводить комплексный анализ футбольных матчей с использованием последних достижений в области искусственного интеллекта и компьютерного зрения. Основной метод – обзор и анализ публикаций по теме исследования; анализ современных технологий, позволяющих автоматически обрабатывать видеоданные. Основная методология – концепция разработки проекта PLAY VISION AI как способ просмотра спортивных матчей с помощью искусственного интеллекта для оценки эффективности игровых стратегий. Актуальность данной работы обусловлена максимальной модификацией современных технических средств для улучшения аналитических возможностей в спорте. Авторами разработаны алгоритмы для калибровки и коррекции искажений видео, полученного с футбольных матчей; разработаны методы детекции и трекинга опорных точек и игроков на видео; реализованы алгоритмы для сопоставления изображений с реальными координатами на поле, а также определения позиций игроков; выполнена интеграция разработанных методов в единую систему с интерфейсом для конечных пользователей. Разработанная система PLAY VISION AI обеспечит тренерам и аналитикам инструменты для оценки эффективности игровых стратегий и подготовки к предстоящим матчам. Также будет способствовать дальнейшему развитию технологий анализа в спорте, открывая новые перспективы для исследований и практического применения.

Ключевые слова:

искусственный интеллект, машинное зрение, машинное обучение, спортивная индустрия, анализ спортивной игры, анализ футбольных матчей, трекинг игроков, видеосистема, спортивные технологии, детекция объектов

Введение

С развитием цифровых технологий, искусственного интеллекта и больших данных, спортивная индустрия сталкивается с растущей потребностью в продвинутых аналитических инструментах. В футболе, где стратегическое и тактическое планирование играют ключевую роль, применение технологий компьютерного зрения и машинного обучения для анализа игр становится не просто трендом, а необходимостью для поддержания конкурентоспособности [\[1\]](#).

Использование компьютерного зрения и машинного обучения в спортивной аналитике позволяет автоматически извлекать значимые данные из видео матчей, что значительно повышает скорость и точность анализа по сравнению с традиционными методами. Такие технологии могут предоставить тренерам детальные отчеты о движениях, позиционировании и тактике игроков в реальном времени.

Актуальность данной работы обусловлена стремлением максимально модифицировать современные технические средства для улучшения аналитических возможностей в спорте. Целью является создание системы, которая позволит проводить комплексный анализ футбольных матчей с использованием последних достижений в области искусственного интеллекта и компьютерного зрения [\[2\]](#).

Задачи работы:

- 1) разработка алгоритмов для калибровки и коррекции искажений видео, полученного с футбольных матчей;
- 2) создание методов для детекции и трекинга опорных точек и игроков на видео;
- 3) реализация алгоритмов для сопоставления изображений с реальными координатами на поле и определения позиций игроков;
- 4) интеграция всех разработанных методов в единую систему с интерфейсом для конечных пользователей.

Разработанная система PLAY VISION AI обеспечит тренерам и аналитикам инструменты для оценки эффективности игровых стратегий и подготовки к предстоящим матчам. Также будет способствовать дальнейшему развитию технологий анализа в спорте, открывая новые перспективы для исследований и практического применения.

Анализ предметной области

Спортивная аналитика, особенно в футболе, охватывает широкий спектр анализа данных – от статистической оценки производительности игроков до визуального анализа их движений на поле. Современные технологии, такие как машинное обучение и компьютерное зрение, позволяют автоматически обрабатывать большие объемы видеоданных для выявления ключевых моментов игры, оценки физического состояния игроков и оптимизации тактик команды [\[3, 4\]](#). Технологические решения в этой области помогают улучшить понимание игры и принятие решений в реальном времени, что делает их незаменимыми для современных спортивных команд.

Анализ существующих систем

Существуют различные системы анализа спортивных событий на рынке, каждая из которых предлагает уникальные функции и преимущества. Двумя выдающимися игроками в этом сегменте являются «Opta Sports» и «Stats Perform»:

1) Opta Sports – предлагает детальный анализ спортивных матчей, собирая комплексные данные об игре и игроκах. Они используют передовые аналитические инструменты для обработки событий во время матчей, предоставляя статистику в реальном времени. Основным его преимуществом является высокая точность данных, широкое признание на рынке, подробное покрытие множества аспектов игры. Однако высокая стоимость услуг «Opta Sports» может ограничивать доступность для меньших клубов и организаций.

2) Stats Perform – объединяет спортивную статистику и аналитическое моделирование, предоставляя командам, тренерам и медиа глубокие аналитические отчеты о спортивных событиях. Основным преимуществом спортивной компании является использование искусственного интеллекта для прогнозирования результатов матчей, обширная база данных исторических данных. Однако присутствует сложность интеграции и адаптации системы под специфические нужды пользователей без предварительной подготовки данных.

Более подробная сравнительная характеристика «Opta Sports» и «Stats Perform» представлена в таблице 1.

Таблица 1 – Сравнительная характеристика систем

Характеристика	Разработка информационной системы AI для просмотра и анализа спортивных матчей	Opta Sports	Stats Perform
Сфера	Все виды спорта, акцент на футболе	Все виды спорта	Все виды спорта
Адаптация	Высокая гибкость и готовность к изменениям	Стандартные решения	Стандартные решения
Интеграция	Открытость к интеграциям с другими системами	Ограниченнaя интеграция	Ограниченнaя интеграция
Стоимость	Низкая, доступная для широкого круга клиентов	Высокая	Высокая
Отечественная разработка	Импортозамещение	Нет	Нет
Родительский контроль	Возможности для контроля доступа для подростков	Не предоставляется	Не предоставляется
Персонализированные видеопаспорта	Да. Возможности для контроля доступа для подростков	Нет	Нет

Разработка	Как продукт отечественной разработки, система будет способствовать развитию национальной экономики и технологической независимости	Зарубежные продукты, зависимость от иностранных поставщиков.	Аналогично «Opta Sports», иностранный продукт.
------------	--	--	--

Прогноз характеристик разработки

Надежность системы заключается в её способности стабильно выполнять необходимые функции в различных условиях, без сбоев, ошибок и простоев. Это достигается за счёт использования устоявшихся технологий, таких как проверенные фреймворки и библиотеки PyTorch, OpenCV, Numpy, Pandas, Ultralytics, которые имеют качественную документацию и широкую поддержку сообщества [\[5, 6\]](#).

Практичность информационной системы заключается его удобством использования и адаптации к нуждам пользователей. Система включает интуитивно понятный интерфейс, который разрабатывается с учётом потребностей пользователей и обеспечивает чёткие инструкции, минимизируя порог входа для всех уровней пользователей. Возможность настройки под пользователя позволяет адаптировать параметры анализа и отчётов согласно специфическим требованиям тренеров и спортивных аналитиков. Кроме того, система обеспечивает доступ на различных платформах, включая веб-сайт, мобильные приложения и интеграцию с «Telegram», что позволяет пользователям мгновенно получать доступ к аналитическим данным [\[7\]](#).

Эффективность системы заключается в способности обеспечивать точные и актуальными данными, которые способствуют улучшению тренировочных процессов и игровых стратегий.

Благодаря использованию продвинутых алгоритмов машинного обучения и компьютерного зрения, система быстро обрабатывает видео и точно выявляет игровые паттерны. Это позволяет автоматизировать рутинные задачи анализа данных, значительно ускоряя обработку информации и освобождая время тренеров для более важных задач. В результате, аналитические отчёты, предоставляемые системой, становятся ценными инструментами для разработки стратегий и тактик, что напрямую влияет на производительность команды [\[8, 9\]](#).

Таким образом, система отличается высокой надёжностью, практичностью и эффективностью, что делает её незаменимым инструментом в области спортивной аналитики.

Бизнес-задачи

В процессе разработки системы была определена серия ключевых бизнес-задач, которые необходимо решить для достижения поставленных целей. Каждая из этих задач играет важную роль в создании актуального, эффективного и надежного решения, способного удовлетворить потребности пользователей и обеспечить высокую точность анализа. В таблице 2 представлены идентификаторы и подробные описания этих бизнес-

задач, включая аспекты автоматизации, пользовательского интерфейса, мобильной интеграции, точности идентификации, защиты данных, а также гибкости и масштабируемости системы.

Таблица 2 – Бизнес задачи

Идентификатор	Бизнес-задача
БЗ-01	Разработать систему для автоматического анализа видео футбольных матчей, используя алгоритмы машинного обучения и компьютерного зрения.
БЗ-02	Создать интуитивно понятный интерфейс для легкой загрузки видео, просмотра данных и получения отчетов через веб и мобильные приложения.
БЗ-03	Разработать мобильное приложение, обеспечивающее доступ к аналитике в реальном времени для поддержки тренеров вне офиса.
БЗ-04	Обеспечить высокую точность идентификации игроков и действий в матчах для улучшения анализа игровых стратегий.
БЗ-05	Реализовать меры безопасности для защиты личных данных игроков и видеоматериалов, обеспечивая конфиденциальность и защиту информации.
БЗ-06	Система должна быть масштабируемой, чтобы поддерживать рост количества пользователей и объемов данных, а также быть совместимой с различными видеоформатами и аппаратным обеспечением.

Функциональные требования

Для успешной работы системы были реализованы следующие ключевые функциональные возможности, которые описаны ниже.

1. Загрузка видеоматериала:

- система должна предоставлять возможность пользователям (тренерам) загружать видеозаписи матчей;
- поддержка различных форматов видео.

2. Анализ видео:

- автоматическое распознавание и анализ ключевых событий матча (голы, пасы, нарушения);
- использование алгоритмов машинного обучения для идентификации игроков и их действий на поле.

3. Отслеживание метрик производительности:

- сбор данных о производительности каждого игрока, включая скорость бега, расстояние, количество успешных и неудачных действий;
- возможность просмотра статистической информации в реальном времени.

4. Пользовательский интерфейс:

- разработка интуитивно понятного веб-интерфейса и мобильного приложения для доступа к аналитическим данным;
- функции фильтрации и поиска для легкого нахождения нужных данных.

5. Разметка видео:

- возможность тренеров вручную добавлять аннотации и метки на видео для обозначения важных моментов;
- автоматическая генерация отчетов на основе анализа видео.

6. Безопасность и управление данными:

- защита данных пользователя и видеоматериалов с помощью современных методов шифрования;
- резервное копирование данных и их восстановление.

7. Интеграция и API:

- предоставление API для интеграции с другими спортивными и аналитическими платформами;
- возможность экспорта данных в различные форматы для дальнейшего анализа.

8. Обучение и поддержка:

- предоставление ресурсов для обучения пользователей работе с системой;
- круглосуточная техническая поддержка пользователей.

Создание модели вариантов использования

В рамках системы, каждое действующее лицо играет свою роль, выполняя различные функции в соответствии со своими правами доступа и возможностями. Ниже представлено детальное описание каждого из них и связанных с ними действий.

Неавторизированные пользователи имеют ограниченный доступ к системе. Они могут просматривать общедоступные страницы, но не имеют доступа к персонализированным функциям, которые требуют входа в систему.

Авторизированные пользователи – зарегистрированные пользователи, которые прошли процесс входа в систему. Они имеют доступ к дополнительным функциям, таким как просмотр и анализ данных, управление личным профилем и доступ к сохраненным настройкам. К авторизированным пользователям относятся: тренеры и ученики.

Тренеры используют систему для управления командой и тренировочными процессами. Они имеют доступ ко всем аналитическим и управлением инструментам в системе.

Ученики – игроки, которые используют систему для доступа к тренировочным материалам и анализу своей игры.

Каждое действующее лицо имеет четко определенные функции и возможности в системе, что позволяет эффективно управлять доступом к информации и инструментам в зависимости от их ролей и задач.

Создание вариантов использования

Для лучшего понимания использования системы были рассмотрены варианты выполнения действий различными пользователями. В таблице 3 описаны действия тренера и ученика.

Таблица 3 – Варианты использования

Актер	Предполагаемые действия
Тренер	Использование специализированного оборудования для записи матчей.
	Настройка параметров камеры перед началом записи.
	Загрузка видеозаписи матча на платформу.
	Отправка записанных данных на сервер для анализа.
	Просмотр аналитических отчётов для анализа данных, собранных во время матчей.
	Разметка игроков. Идентификация и маркировка игроков на видео для анализа их действий.
	Корректировка тренировочного плана. Использование данных для оптимизации тренировочных процессов.
Ученик	Просмотр видеопаспорта. Доступ к видеозаписям своих выступлений и аналитическим отчетам о своей игре.
	Обучение по рекомендательному плану. Следование индивидуально разработанным тренировочным планам.
	Обсуждение результата анализа. Взаимодействие с тренером для обсуждения отчетов анализа и получения обратной связи.

Построение диаграммы вариантов использования

В рамках проекта, который направлен на создание комплексной системы анализа видеоданных спортивных матчей, диаграмма вариантов использования играет ключевую роль в визуализации функциональных возможностей системы и взаимодействия её пользователей с этими функциями. Эта диаграмма не только помогает разработчикам и проектировщикам понять, как система должна взаимодействовать с конечными пользователями, но и предоставляет всем заинтересованным сторонам чёткое представление о том, как система будет использоваться на практике.

Цели диаграммы вариантов использования:

1) Иллюстрация функциональных требований. Диаграмма демонстрирует основные функции системы и способы их использования различными пользователями.

2) Определение действующих лиц. Диаграмма чётко определяет всех пользователей системы, включая тренеров, аналитиков, судей и других стейкхолдеров, которые будут взаимодействовать с системой.

Диаграмма вариантов использования для системы включает следующие основные элементы:

1) Действующие лица. Все вовлеченные пользователи или системы, которые взаимодействуют с системой.

2) Варианты использования. Конкретные функции или действия, которые действующие

лица могут выполнить с помощью системы. Это включает в себя такие задачи, как загрузка видеоданных, анализ видео, получение отчетов и другие.

3) Отношения. Связи между действующими лицами и вариантами использования, которые показывают, как пользователи взаимодействуют с системой.

Диаграмма разработана с использованием стандартных обозначений UML (Unified Modeling Language) [10, 11]. Она организована таким образом, чтобы обеспечить логичное и последовательное понимание всех взаимодействий. Фрагмент диаграммы вариантов использования представлен на рисунке 1.

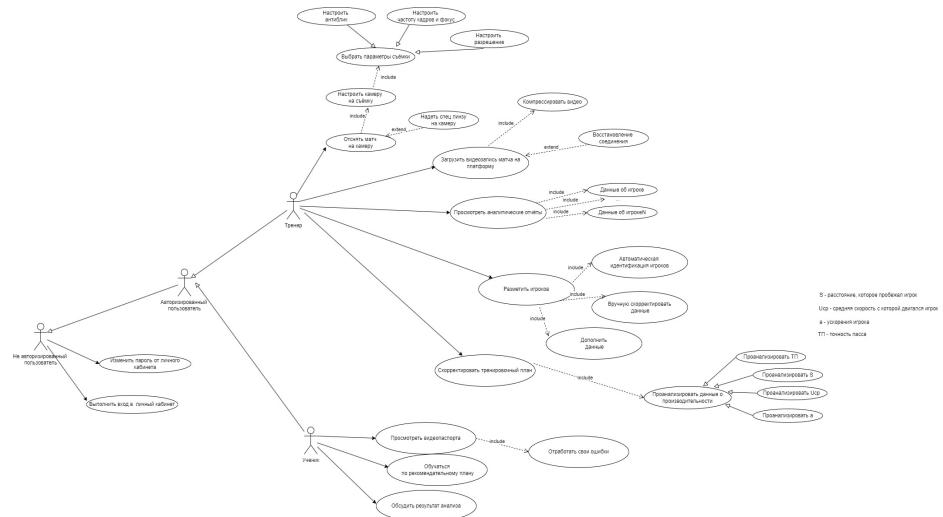


Рисунок 1 – Диаграмма вариантов использования (фрагмент)

Идентификация классов, участвующих в реализации потоков событий вариантов использования

Для детального анализа спортивных матчей была создана специализированная система, которая включает в себя ряд взаимосвязанных классов. Каждый из этих классов отвечает за выполнение определённых функций в рамках системы – от обработки событий до анализа видеоматериалов. В таблице 4 представлены основные классы этой системы, вместе с их атрибутами и методами, что позволяет наглядно увидеть, как организована работа с данными и какие операции могут быть выполнены с помощью разработанных функций.

Таблица 4 – Таблица методов класса для системы анализа спортивных матчей

№	Название таблицы	Названия полей	Название функции
1	Событие (Event)	IDСобытия (int, PK), IDMатча (int, FK), IDТипаСобытия (int, FK)	<pre> def создать_событие(self, данные_события); def активировать_событие(self, ид_события); def обновить_событие(self, ид_события, новые_данные); def удалить_событие(self, ид_события); def найти_событие(self, ид_события); </pre>

			<pre>def проверить_событие(self, ид_события); def сохранить_историю_события(self, ид_события);</pre>
2	Тип События (EventType)	IDТипаСобытия (int, PK), Название (str)	<pre>def создать_тип_события(self, данные_типа_события); def обновить_тип_события(self, ид_типа_события, новые_данные); def удалить_тип_события(self, ид_типа_события); def найти_тип_события(self, ид_типа_события);</pre>
3	Метрика игры (Metric)	IDМетрики (int, PK), IDИгрока (int, FK), IDМатча (int, FK), IDТипаМетрики (int, FK), РезультатыАнализа (str)	<pre>def создать_метрику(self, данные_метрики); def обновить_метрику(self, ид_метрики, новые_данные); def удалить_метрику(self, ид_метрики): def найти_метрику(self, ид_метрики): def сохранить_данные_метрики(self, ид_метрики);</pre>
4	Тип Метрики (MetricType)	IDТипаМетрики (int, PK), Название (str), Число (float)	<pre>def создать_тип_метрики(self, данные_типа_метрики); def обновить_тип_метрики(self, ид_типа_метрики, новые_данные); def удалить_тип_метрики(self, ид_типа_метрики); def найти_тип_метрики(self, ид_типа_метрики)</pre>
5	Анализ Видео (VideoAnalysis)	IDАнализа (int, PK), IDМатча (int, FK), Дата (datetime), IDИгрока (int, FK), IDТипаСобытия (int, FK)	<pre>def создать_анализ(self, данные_анализа); def обновить_анализ(self, ид_анализа, новые_данные); def удалить_анализ(self, ид_анализа); def найти_анализ(self, ид_анализа); def сохранить_результаты_анализа(self, ид_анализа): pass</pre>

Идентификация классов

Диаграмма идентификации классов служит для визуального представления классов, их атрибутов и методов, а также взаимосвязей между ними, что критически важно для создания структурированной и модульной системы.

Цели диаграммы идентификации классов:

- Структурирование системы. Диаграмма помогает архитекторам и разработчикам определить, как система будет организована в терминах классов и объектов.
- Определение атрибутов и методов. Иллюстрирует, какие данные (атрибуты) и операции (методы) ассоциированы с каждым классом в системе.
- Визуализация взаимосвязей. Показывает, как классы связаны и взаимодействуют друг с другом, что упрощает понимание общей архитектуры системы.

Диаграмма будет включать следующие элементы:

- Классы. Каждый класс представляет собой сущность или объект, выполняющий определенные функции в рамках системы. Классы будут представлять сущности, такие как игрок, Матч, Анализатор видео, и интерфейс пользователя.
- Атрибут. Переменные или свойства, которые характеризуют классы, например, имя игрока или время матча.
- Методы. Функции или процедуры, которые классы могут выполнять, такие как анализ видео или отображение результатов.
- Ассоциации. Отношения между классами, указывающие, как классы взаимодействуют друг с другом. Например, класс матч может ассоциироваться с несколькими объектами класса игрок.
- Агрегация и композиция. Специальные формы ассоциаций, которые показывают, как объекты класса могут быть частью других объектов.

Для создания диаграммы использована нотация UML. Пример диаграммы идентификации классов представлен на рисунке 2.

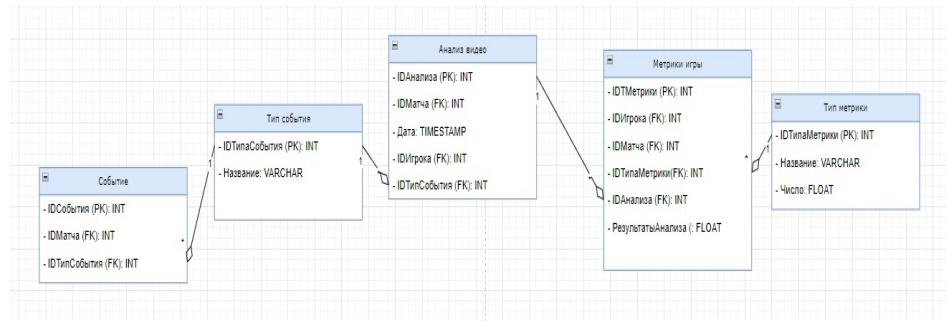


Рисунок 2 – Диаграмма классов системы (фрагмент)

Работа системы

Система PlayVision предлагает пользователям уникальную возможность анализа футбольных матчей с помощью виртуального тренера (см. рисунок 3). На главной странице представлен заголовок, привлекающий внимание к возможностям приложения, а также кнопка, призывающая попробовать его бесплатно. Верхнее меню навигации включает разделы "Новости", "Прайс", "Контакты" и "Войти", обеспечивая доступ к

ключевой информации.

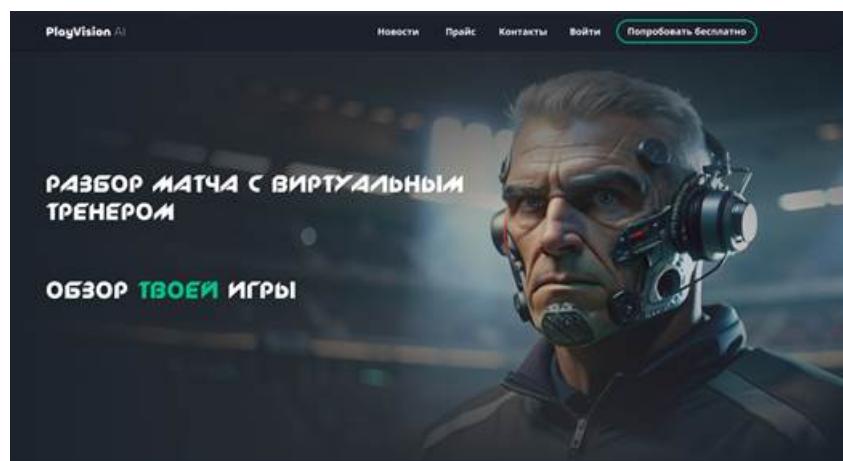


Рисунок 3 – Главная страница системы

Внутренний интерфейс приложения, показанный на втором изображении, демонстрирует, как пользователи могут анализировать матчи (см. рисунок 4). Видеоплеер позволяет воспроизводить футбольный матч, автоматически отслеживая и выделяя игроков, что видно по красным рамкам вокруг них. Это позволяет тренерам и аналитикам наблюдать за ключевыми моментами и действиями игроков в реальном времени. Справа от видеоплеера находится таблица с информацией об играх, включая их статус, ФИО и позицию, что помогает быстро ориентироваться в данных и принимать обоснованные решения.

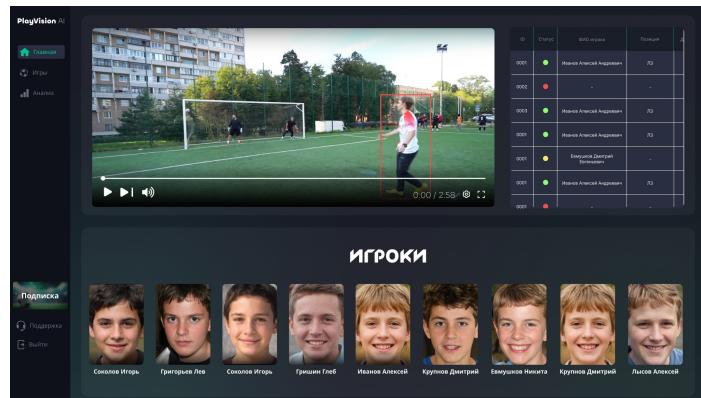


Рисунок 4 – Интерфейс анализа матча

Нижняя часть экрана содержит портреты игроков, участвующих в матче, с их именами, что облегчает идентификацию каждого участника. В левой части интерфейса располагается боковая панель навигации, позволяющая перемещаться между разделами "Главная", "Игры" и "Анализ", а также управлять подпиской и получать поддержку.

Таким образом, PlayVision предоставляет тренерам и аналитикам мощный инструмент для детального анализа футбольных матчей, объединяя автоматическое отслеживание и удобный интерфейс для повышения эффективности тренировок и стратегического планирования.

Заключение

Проект PLAY VISION AI представляет собой значительный шаг вперед в области применения искусственного интеллекта и компьютерного зрения для анализа спортивных матчей. Система обладает широким спектром возможностей, предлагая

автоматизированные решения для точного и быстрого анализа видеозаписей.

Область применения PLAY VISION AI включает как профессиональный, так и любительский спорт. Система может быть адаптирована для различных видов спорта, таких как баскетбол, хоккей и теннис, что делает ее универсальным инструментом для спортивной аналитики. Интеграция с мобильными устройствами и веб-платформами обеспечивает тренерам и аналитикам удобный доступ к данным в реальном времени, что значительно улучшает процесс тренировки и стратегического планирования.

Новизна проекта заключается в использовании передовых технологий искусственного интеллекта и машинного обучения для детекции и трекинга игроков, а также в реализации методов калибровки и коррекции видеозаписей. Высокая точность анализа, интуитивно понятный интерфейс и возможность настройки под конкретные потребности пользователей делают систему PLAY VISION AI эффективным и доступным инструментом.

PLAY VISION AI открывает новые горизонты в спортивной аналитике, предоставляя глубокое понимание игровых процессов и позволяя принимать обоснованные решения. Дальнейшее развитие системы, включая расширение функциональности и интеграцию с другими платформами, увеличит ее ценность и эффективность в спортивной индустрии, делая незаменимым помощником для тренеров и аналитиков.

Библиография

1. Касиси Д. Применение искусственного интеллекта в спорте // In Situ. 2023. № 5. С. 30–33. EDN CHZPXZ.
2. Ковалев С. В., Копышева Т. Н., Митрофанова Т. В., Смирнова Т. Н. О приобретении опыта научно-исследовательской работы будущими IT-специалистами в условиях студенческого кружка // Современные научоемкие технологии. 2021. № 7. С. 117–122. DOI: 10.17513/snt.38762. EDN YCJEA1.
3. Кузнецов А. Ю., Барсукова К. В. Big Data, гаджеты и спортивная аналитика: как устроен российский спорттех // Использование Big Data в официальной статистике Using Big Data in official statistics : сб. матер. II Всерос. науч.-исслед. конф., Липецк, 29 июня 2022 года. Липецк: Изд-во ЛГТУ, 2022. С. 162–165. EDN NHQSYP.
4. Некоторые особенности изучения видеоанализа в спорте / Р. Ф. Мифтахов, М. И. Галяутдинов, А. М. Ситдиков, И. Р. Фаткуллов // Ученые записки университета им. П.Ф. Лесгафта. 2019. № 12(178). С. 207–210. DOI: 10.34835/issn.2308-1961.2019.12.207-210. EDN BRW XMC.
5. О применении метода ИТ-стартапов в обучении будущих ИТ-специалистов / Т. В. Митрофанова, Т. Н. Смирнова, Т. Н. Копышева [и др.] // Инновационное развитие профессионального образования. 2021. № 2(30). С. 49–57. EDN LPIDCF.
6. Перечень поручений по итогам конференции «Путешествие в мир искусственного интеллекта» (утв. Президентом РФ 29 января 2023 г. № Пр-172) [Электронный ресурс]. URL: <https://www.garant.ru/products/ipo/prime/doc/406165361/> (Дата обращения: 09.11.2023).
7. Покатаев А. В., Тестоедов В. А. Аналитика в футболе: актуальные направления, развития, перспективы в тренерской деятельности // Современный менеджмент в игровых видах спорта, Москва, 15 ноября 2017 года / Материалы совместной конференции кафедры «Менеджмента и экономики спорта им. В.В. Кузина» и кафедры «Теории и методики футбола» ФГБОУ ВО «РГУФКСМиТ». М. : Научный консультант, 2017. С. 131–134. EDN YLDARF.
8. Полозов А. А., Мальцева Н. А. Существующие модели спортивной аналитики и их web-сервисы // Научные и образовательные основы в физической культуре и спорте. 2022. Т.

- 8, № 4. С. 60–77. DOI: 10.57006/2782-3245-2022-8-4-58-76. EDN IXDUCJ.
9. Применение онлайн-компиляторов для обучения программированию студентов ИТ-специальностей вуза / Т. Н. Копышева, Т. Н. Смирнова, Т. В. Митрофанова, М. В. Волик // Новые компетенции цифровой реальности: теория и практика их развития у обучающихся: сб. докл. и науч. ст. II Всерос. науч.-практ. конф., Чебоксары, 21 марта 2021 года. Чебоксары: Изд-во Чуваш. ун-та, 2021. С. 168–173. EDN STREVV.
10. Чахоян Т. А., Бучинчик П. В., Егоров А. Р. Аналитика спортивных достижений футбольных клубов и их игроков при помощи инструментов машинного обучения // Всероссийская Студенческая конференция «Студенческая научная весна», посвященная 85-летию Ю. А. Гагарина: сб. тезисов докл., Москва, 01–30 апреля 2019 года. М.: Издательский дом «Научная библиотека», 2019. С. 315–316. EDN LLMAN.
11. Rajasekaran, S. B. AI and Cybersecurity-How AI Augments Cybersecurity Posture of an Enterprise / S. B. Rajasekaran // International Journal of Intelligent Systems and Applications in Engineering. 2023. Vol. 11, No. 1. P. 179–182. EDN EZCRML

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена разработке системы PLAY VISION AI, предназначенной для анализа футбольных матчей с использованием технологий искусственного интеллекта и компьютерного зрения. Основное внимание удалено созданию алгоритмов для калибровки и коррекцииискажений видео, детекции и трекинга игроков, а также интеграции этих методов в единую систему. В статье подробно описаны методологические подходы, используемые для разработки системы. Авторы разрабатывают и тестируют алгоритмы машинного обучения и компьютерного зрения, чтобы обеспечить точное и эффективное распознавание и анализ ключевых моментов футбольных матчей. Особое внимание удалено калибровке видеоданных, что значительно улучшает точность анализа. Актуальность исследования обусловлена растущей потребностью спортивной индустрии в продвинутых аналитических инструментах. С применением искусственного интеллекта и компьютерного зрения в спортивной аналитике возможно значительно повысить точность и оперативность анализа матчей, что особенно важно для тренеров и аналитиков, занимающихся стратегическим планированием. Научная новизна работы заключается в интеграции современных технологий машинного обучения и компьютерного зрения для создания комплексной системы анализа видеоданных спортивных матчей. Авторы представляют инновационные методы калибровки и коррекции видео, что позволяет значительно повысить точность детекции и трекинга игроков. Статья написана в научном стиле, с четкой структурой и логичной последовательностью изложения материала. Каждый раздел работы содержит детальное описание проведенных исследований и полученных результатов. Введение обосновывает необходимость исследования, а основная часть статьи подробно описывает разработку системы и тестирование предложенных алгоритмов. Библиографический список включает современные и актуальные источники, что свидетельствует о глубоком погружении авторов в предмет исследования. Приведенные источники подтверждают основные тезисы работы и обеспечивают научную основу для предложенных решений. Авторы убедительно аргументируют преимущества своей системы перед существующими аналогами, такими как «Opta Sports» и «Stats Perform». В статье приведены сравнительные характеристики, которые наглядно демонстрируют преимущества разработанного решения, включая высокую

точность и доступность для широкого круга пользователей. Статья представляет значительный интерес для специалистов в области спортивной аналитики, разработчиков программного обеспечения и исследователей, занимающихся применением искусственного интеллекта в спорте. Разработанная система PLAY VISION AI обладает потенциалом для широкого применения в различных видах спорта, что делает данную работу особенно ценной для дальнейших исследований и практического использования. Статья представляет собой важный вклад в развитие спортивной аналитики с применением искусственного интеллекта и компьютерного зрения. Работа отличается высокой научной новизной, актуальностью и практической значимостью. Рекомендую принять статью к публикации без существенных доработок.

Для дальнейшего развития работы над проектом PLAY VISION AI рекомендуется сосредоточиться на нескольких ключевых направлениях. Во-первых, целесообразно расширить функциональные возможности системы, включив в неё аналитику для других видов спорта, таких как баскетбол, хоккей и теннис. Это позволит привлечь более широкую аудиторию и обеспечить универсальность применения разработанных алгоритмов. Во-вторых, необходимо улучшить пользовательский интерфейс, делая его еще более интуитивно понятным и адаптированным к различным уровням технической подготовки пользователей. Важно учитывать отзывы текущих пользователей для оптимизации интерфейса и повышения его удобства. Третьим направлением является интеграция системы с существующими платформами спортивной аналитики и тренировочными программами. Это обеспечит более широкие возможности для анализа данных и их использования в тренировочном процессе. Также стоит уделить внимание повышению безопасности данных, обеспечивая надежное шифрование и защиту личной информации пользователей. Это особенно важно в контексте работы с видео и персональными данными игроков. Наконец, для повышения точности и эффективности анализа необходимо продолжать совершенствовать алгоритмы машинного обучения и компьютерного зрения. Регулярное обновление моделей и использование новейших достижений в области искусственного интеллекта позволит поддерживать высокий уровень аналитики и соответствовать текущим требованиям спортивной индустрии. Эти рекомендации помогут улучшить систему PLAY VISION AI, сделав её более функциональной, удобной и безопасной для пользователей, а также расширят её возможности для применения в различных спортивных дисциплинах.

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Козырева Н.И., Мухтулов М.О., Ершов С.А., Новосельцева С.В., Ахмадуллин Д.А. Современные методы предотвращения DDoS-атак и защиты веб-серверов // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.73667 EDN: BOUTCT URL: https://nbppublish.com/library_read_article.php?id=73667

Современные методы предотвращения DDoS-атак и защиты веб-серверов**Козырева Надежда Ивановна**

кандидат технических наук

доцент; кафедра информационной безопасности (ИБ); Поволжский государственный университет телекоммуникаций и информатики

443010, Россия, Самарская обл., г. Самара, Самарский р-н, ул. Льва Толстого, д. 23

✉ n.kozyreva@psuti.ru

**Мухтулов Михаил Олегович**

ORCID: 0009-0009-5985-3247

студент; кафедра информатики и вычислительной техники (ИВТ); Поволжский государственный университет телекоммуникаций и информатики

443125, Россия, Самарская обл., г. Самара, Промышленный р-н, Московское шоссе, д. 252, кв. 197

✉ mixa.1204@inbox.ru

**Ершов Сергей Александрович**

студент; кафедра информатики и вычислительной техники (ИВТ); Поволжский государственный университет телекоммуникаций и информатики

446430, Россия, Самарская обл., г. Кинель, ул. Ульяновская, д. 28, кв. 7

✉ vizonera8@gmail.com

**Новосельцева София Владимировна**

студент; кафедра информатики и вычислительной техники (ИВТ); Поволжский государственный университет телекоммуникаций и информатики

443093, Россия, Самарская обл., г. Самара, Железнодорожный р-н, ул. Мориса Тореза, д. 31, кв. 79

✉ sunny.tea.with.lilac@gmail.com

**Ахмадуллин Динар Айратович**

студент; кафедра информатики и вычислительной техники (ИВТ); Поволжский государственный университет телекоммуникаций и информатики

443101, Россия, Самарская обл., г. Самара, Куйбышевский р-н, ул. Хасановская, д. 14, кв. 17

✉ dinarnevashno@yandex.ru

[Статья из рубрики "Кодирование и защита информации"](#)**DOI:**

10.7256/2454-0714.2025.2.73667

EDN:

BOUTCT

Дата направления статьи в редакцию:

12-03-2025

Аннотация: Объектом исследования являются веб-серверы и их поведение в условиях высокоинтенсивных распределённых атак типа «отказ в обслуживании» (DDoS), нарушающих доступность сервисов и устойчивость инфраструктур. В качестве предмета исследования рассматриваются современные методы защиты серверных приложений от DDoS-угроз, включая анализ трафика, фильтрацию по частоте запросов, межсетевые экраны (файрволы) и облачные решения. Подробно анализируется эффективность различных технологий защиты, таких как Rate Limiting, ModSecurity, Google Cloud Armor и Cloudflare, а также их интеграция с традиционными средствами — межсетевыми экранами, системами предотвращения вторжений (IPS) и прокси-серверами. В рамках исследования разработан тестовый сервер на языке Go, имитирующий поведение реального веб-приложения с логированием и сбором статистики. Для моделирования DDoS-атак использован инструмент MHDDoS, обеспечивающий широкое покрытие типов угроз: от UDP и SYN Flood до HTTP Flood и Slowloris. Методы исследования включают эмуляцию атак на сетевом и прикладном уровнях трафика, нагрузочное тестирование, сбор метрик (процент заблокированных запросов, среднее время отклика, нагрузка на CPU и RAM) и сравнительный анализ эффективности решений. Научная новизна исследования заключается в разработке и применении экспериментальной модели имитации DDoS-атак с использованием специализированного Go-сервера, что позволило в реалистичных условиях оценить эффективность современных локальных и облачных средств защиты. Анализ реальных кейсов демонстрирует эффективность адаптивных стратегий против современных сложносоставных атак. Выводы подчёркивают необходимость активного подхода к безопасности, учитывающего как технологические, так и организационные меры защиты. Полученные результаты имеют практическую ценность для специалистов по кибербезопасности, системных администраторов и разработчиков защитных решений, предоставляя им методическую основу для создания устойчивых к DDoS веб-инфраструктур. Работа также обозначает перспективные направления для дальнейших исследований в области интеллектуальных систем обнаружения и нейтрализации атак.

Ключевые слова:

DDoS-атаки, Веб-сервер, Cloudflare, Rate Limiting, Fail2Ban, Кибербезопасность, Защита сети, WAF, Автоматизированное предотвращение атак, Машинное обучение

Введение

В современном мире интернет-технологий стабильная работа вебсерверов критически важна для бизнеса, государственных учреждений и частных пользователей. Однако с ростом цифровых сервисов возрастает и число кибератак, среди которых DDoS-атаки (Distributed Denial of Service) остаются одной из наиболее серьезных угроз. Эти атаки направлены на исчерпание вычислительных и сетевых ресурсов сервера, что приводит к его замедлению или полной недоступности для пользователей.

Особенность современных DDoS-атак заключается не только в их количестве, но и в усложнении механизмов реализации. Если раньше злоумышленники использовали примитивные методы, основанные на отправке большого количества запросов, то сегодня атаки маскируются под легитимный трафик, применяются ботнет-сети, анонимные прокси и даже технологии искусственного интеллекта для обхода защитных механизмов. Это создаёт значительные вызовы для администраторов информационных систем и требует применения комплексных решений для защиты.

DDoS-атаки классифицируются по уровню воздействия:

Сетевые атаки (например, UDP Flood, SYN Flood) перегружают серверные соединения огромным числом пакетов, блокируя сетевые ресурсы.

Атаки на уровне приложений (например, HTTP Flood, Slowloris) имитируют реальную пользовательскую активность, усложняя их обнаружение традиционными методами.

Комбинированные атаки объединяют несколько техник, что делает их особенно сложными для предотвращения.

Актуальность данной проблемы обусловлена не только ростом атак, но и их экономическими последствиями. Компании несут убытки из-за простоя сервисов, теряют доверие пользователей и могут столкнуться с утечками данных. В связи с этим необходимо детально изучить современные подходы к защите веб-ресурсов, включая облачные решения (Cloudflare, Google Cloud Armor), локальные механизмы защиты (Rate Limiting, Fail2Ban) и перспективные технологии, использующие машинное обучение и поведенческий анализ трафика.

Настоящая работа посвящена исследованию этих методов, их сравнительному анализу и перспективам дальнейшего развития стратегий защиты от DDoS-атак. Объектом настоящего исследования являются веб-серверы, подвергающиеся DDoS-атакам, а предметом — методы и средства их защиты от распределённых атак отказа в обслуживании, включая локальные и облачные технологии фильтрации и анализа трафика.

1. Классификация DDoS-атак и их последствия

DDoS-атаки (Distributed Denial of Service) представляют собой одну из наиболее распространённых форм киберугроз, направленных на нарушение работы серверной инфраструктуры при помощи отправки огромного числа запросов. Основная цель данных атак — нарушить доступность сервисов за счёт перегрузки вычислительных ресурсов чрезмерным объёмом внешнего трафика. DDoS-атаки классифицируются в зависимости от уровня сетевой модели OSI, на который они воздействуют [\[3, с. 251-252\]](#).

1.1. Атаки на уровень сети (L3/L4 OSI)

Сетевые атаки нацелены на перегрузку сетевого оборудования и исчерпание полосы пропускания канала связи. Они отличаются высокой скоростью передачи пакетов и

большим объемом входящего трафика.

UDP Flood – злоумышленник отправляет большое количество UDP-пакетов на случайные порты целевого сервера, заставляя его обрабатывать каждый пакет и отправлять ICMP-ответы «Destination Unreachable». В результате перегружается процессорная мощность сервера и сетевой канал [\[1, с. 574\]](#).

ICMP Flood (Ping Flood) – атака, при которой злоумышленник отправляет огромное количество ICMP Echo-запросов (ping-запросов) с целью перегрузки сетевого стека сервера.

SYN Flood – атака эксплуатирует механизм установления TCP-соединений (трёхстороннее рукопожатие). Злоумышленник отправляет большое количество SYN-запросов, но не завершает установление соединения, что приводит к исчерпанию доступных соединений сервера и отказу в обслуживании для легитимных пользователей.

SMURF – это вид DDoS-атаки, основанный на отправке ICMP-запросов (аналогичных ping) на широковещательный адрес сети через маршрутизатор. В результате все устройства в сети отвечают на запрос, перегружая жертву трафиком [\[4, с. 430\]](#).

1.2. Атаки на уровень приложений (L7 OSI)

DDoS-атаки на уровне приложений нацелены на перегрузку серверных ресурсов, таких как процессорное время и оперативная память, путем отправки большого количества сложных запросов, имитирующих активность реальных пользователей. Эти атаки сложнее в обнаружении, так как трафик выглядит легитимным.

HTTP Flood – злоумышленник отправляет многочисленные HTTP-запросы на сервер, вынуждая его загружать веб-страницы, выполнять обработку динамического контента и обращаться к базе данных.

Slowloris-атака – это метод атаки, при котором злоумышленник удерживает множество HTTP-соединений открытыми, отправляя неполные запросы. Сервер, ожидая завершения этих запросов, выделяет ресурсы для каждого соединения, что приводит к их истощению и делает сервер недоступным для обычных пользователей [\[5, с. 3-4\]](#).

1.3. Комбинированные атаки (multi-vector attacks)

Комбинированные атаки представляют собой угрозу, при которой злоумышленники одновременно используют несколько методов воздействия, например, перегрузку сервера на сетевом уровне (UDP Flood) и атаки на уровень приложений (HTTP Flood). Такой подход усложняет обнаружение и повышает вероятность успешного нападения.

Эффективная защита от подобных атак требует многоуровневых стратегий, включающих фильтрацию трафика, облачные сервисы и алгоритмы машинного обучения для выявления аномалий в реальном времени.

2. Локальные методы защиты от DDoS-атак и способы их выявления

2.1. Способы выявления DDoS-атак

Способы выявления DDoS-атак можно разделить на две основные группы: пассивные и активные.

Пассивные методы основаны на наблюдении за сетевым трафиком и анализе его

параметров. Они не взаимодействуют напрямую с сетью или источниками атаки, а лишь отслеживают поток данных, выявляя аномалии, характерные для DDoS-атак. К таким методам относятся мониторинг сетевого трафика, анализ нагрузки на серверы, изучение логов серверов и сетевых устройств, а также системы обнаружения аномалий.

Активные методы предполагают прямое взаимодействие с сетью или потенциальными атакующими устройствами. В их рамках может выполняться отправка тестового трафика для проверки доступности сервера и анализа его ответов, а также передача управляющих сигналов для блокировки атаки. Однако такие подходы могут не только создавать дополнительную нагрузку на сеть, но и оказаться неэффективными в случае мощных DDoS-атак. К активным методам относятся, например, системы обнаружения внешних атак [\[2, с. 5\]](#).

Основные проблемы при обнаружении DDoS-атак:

1. Ложные срабатывания – система может ошибочно распознать атаку, что приводит к потере времени и ресурсов.
2. Сложность настройки – требует специальных знаний для эффективной работы.
- 3 . Высокие затраты ресурсов – анализ большого трафика требует мощного оборудования.
4. Трудности с новыми атаками – некоторые системы не распознают новые методы атак.
5. Постоянное обновление – необходимо адаптироваться к новым угрозам.
- 6 . Медленное реагирование – обнаружение атаки не всегда означает ее мгновенную блокировку [\[2, с. 7\]](#).

2.2. Ограничение частоты запросов (Rate Limiting)

Rate Limiting — это технология, предназначенная для контроля нагрузки на сервер. Она ограничивает число запросов, которые может отправить один клиент в течение определенного периода. Этот подход используется для предотвращения атак, например, HTTP Flood, когда злоумышленники пытаются перегрузить систему массовыми запросами.

Суть метода заключается в мониторинге количества запросов, поступающих с каждого IP-адреса. Если количество запросов превышает допустимый лимит, система либо блокирует их, либо замедляет обработку, чтобы избежать перегрузки сервера [\[8\]](#).

2.3. Блокировка подозрительных IP-адресов (Fail2Ban)

Fail2Ban — это утилита для анализа логов, которая автоматически обнаруживает подозрительные действия, такие как попытки несанкционированного доступа, и блокирует соответствующие IP-адреса. Этот инструмент активно применяется для защиты серверов от атак методом перебора (brute-force), а также для выявления признаков DDoS-активности.

Когда система фиксирует аномальное поведение, например, множество неудачных попыток входа или резкий всплеск запросов, Fail2Ban временно блокирует подозрительный IP-адрес с помощью инструментов вроде iptables или firewalld. Блокировка действует в течение определенного времени, после чего автоматически снимается. Такой подход особенно эффективен для противодействия низкоинтенсивным

атакам и сканированию на наличие уязвимостей [\[9\]](#).

2.4. Web Application Firewall (WAF)

WAF (например, ModSecurity) работает на уровне сервера, анализируя HTTP-запросы и блокируя потенциально опасный трафик. В отличие от облачных решений, локальный WAF функционирует непосредственно на сервере, обеспечивая защиту от атак на уровне приложений, таких как SQL-инъекции, межсайтовый скрипting (XSS) и распределенные атаки на отказ в обслуживании (DDoS).

ModSecurity предоставляет возможность настраивать пользовательские правила фильтрации, которые можно адаптировать под конкретные требования веб-приложения. Однако для поддержания высокой эффективности защиты необходимо регулярно обновлять конфигурацию и отслеживать новые угрозы, что требует постоянного внимания со стороны администраторов [\[10\]](#).

3. Облачные сервисы защиты от DDoS-атак

Облачные провайдеры предлагают мощные инструменты защиты от DDoS-атак за счет высокой пропускной способности и глобально распределенной инфраструктуры. Например, использование CDN (сети доставки контента) позволяет скрывать исходный IP-адрес сервера, что затрудняет атаки [\[2, с. 7-8\]](#).

3.1. Cloudflare

Cloudflare — это облачная платформа, предназначенная для защиты веб-ресурсов от DDoS-атак. Она обеспечивает фильтрацию входящего трафика и повышает доступность веб-сервисов. Cloudflare функционирует как промежуточный слой между пользователями и сервером, анализируя запросы и блокируя вредоносные действия до того, как они достигнут инфраструктуры клиента [\[11\]](#).

Cloudflare использует глобальную распределенную сеть для балансировки нагрузки и минимизации воздействия атак. Встроенные механизмы защиты, такие как Web Application Firewall (WAF) и анализ поведения трафика, позволяют автоматически выявлять и блокировать подозрительные запросы. Дополнительно система применяет Rate Limiting для ограничения частоты обращений, а также CAPTCHA и JavaScript Challenge для фильтрации автоматизированных ботов.

3.2. Google Cloud Armor

Google Cloud Armor — это облачная платформа, предназначенная для защиты веб-приложений и сервисов, размещенных в инфраструктуре Google Cloud, от DDoS-атак и вредоносного трафика. Решение интегрируется с балансировщиками нагрузки Google Cloud и выполняет фильтрацию на границе сети, что позволяет минимизировать нагрузку на конечные серверы.

Google Cloud Armor использует адаптивные методы защиты, такие как анализ поведения трафика, интеллектуальные правила блокировки и технологии машинного обучения для обнаружения аномальных паттернов запросов. Платформа включает предустановленные политики безопасности, которые защищают от распространенных атак на уровне приложений, например, HTTP Flood. Кроме того, пользователи могут создавать собственные правила фильтрации, основанные на геолокации, IP-адресах и других параметрах, что обеспечивает гибкость в борьбе с новыми угрозами [\[12\]](#).

4. Методология исследования

В данном исследовании была проведена экспериментальная оценка эффективности современных технологий защиты веб-серверов от DDoS-атак. Основной целью работы стало тестирование различных механизмов фильтрации трафика в условиях эмуляции атак различного типа. Для этого был разработан и развернут тестовый сервер на языке Go. Непосредственно для развертывания использовалась операционная система Ubuntu.

В ходе исследования были поставлены следующие задачи:

- Реализация тестового веб-сервера с функциями логирования и анализа запросов.
- Моделирование различных видов DDoS-атак, таких как SYN Flood, UDP Flood, HTTP Flood и DNS Query Flood.
- Последовательное подключение различных технологий защиты и тестирование их эффективности.
- Анализ полученных данных с целью сравнительной оценки защитных механизмов и степени опасности различных типов атак.

Для эмуляции атак использовался инструмент MHDDoS. Это обусловлено тем, что он разворачивается на любой инфраструктуре (написан на Python) и имеет удобную для open-source проекта документацию. В ходе тестирования сервер подвергался последовательным атакам с включением различных технологий защиты, таких как Rate Limiting, ModSecurity WAF, Google Cloud Armor, Cloudflare. Для каждой технологии фиксировалось общее количество обработанных запросов, доля заблокированных атак, задержка ответа сервера и потребление системных ресурсов.

5. Реализация эксперимента

5.1. Разработка тестового сервера

Для проведения эксперимента был разработан тестовый сервер на языке Go. Данный язык был выбран благодаря своей высокой производительности, встроенной поддержке параллелизма (через механизм goroutines) и низким требованиям к системным ресурсам. Это делает его особенно подходящим для моделирования сценариев, типичных для DDoS-атак.

Разработанный сервер имитирует поведение и нагрузку, характерные для реальных веб-приложений, и обеспечивает следующие ключевые функции:

- Обработка различных типов HTTP-запросов, включая легкие и ресурсоёмкие (например, запросы к условной «базе данных» и загрузку страницы с внешними ресурсами).
- Логирование входящих запросов, включая регистрацию уникальных IP-адресов, путей и признаков потенциальных атак.
- Сбор подробной статистики: общее количество запросов, количество ошибок, среднее время отклика, текущая производительность (RPS), загрузка процессора.
- Предоставление API-интерфейса для доступа к накопленным статистическим данным в формате JSON.
- Обработка UDP-пакетов, что позволяет дополнительно моделировать атаки на сетевом уровне.

уровне.

5.2. Подключение технологий защиты и проведение DDoS-атак

Тип атаки	Кол-во запросов / пакетов	Успешных ответов (%)	Ошибки 5xx (%)	Среднее время ответа (мс)	Загрузка CPU (%)
http	100 000	61,5	5,6	180	72
header	100 000	58,1	41,9	195	76
slowloris	45 000 TCPсоединений	21,4	78,6	>6000	89
udp	1 000 000 пакетов	—	—	—	82

Таблица 1. Эффективность DDoS атак на сервер без защиты

Атаки, ориентированные на высокочастотные HTTP-запросы и заголовочные переполнения, вызывают значительное увеличение количества ошибок уровня 5xx (до 37–40%). Среднее время ответа превышает 180 мс, что свидетельствует о деградации качества обслуживания. Нагрузка на CPU достигает 74–79%.

После ослабления параметров HTTP-сервера (удлинённый

ReadHeaderTimeout, увеличенные MaxHeaderBytes), эффективность Slowlorisатаки значительно возросла. Удержание большого количества TCP-соединений привело к резкому снижению доступности — доля ошибок достигла 81%, а среднее время ответа превысило 6 секунд. Загрузка CPU приблизилась к пиковым значениям (91%), а использование памяти выросло более чем в 2 раза по сравнению с HTTP Flood.

Rate Limiting

Тип атаки	Кол-во запросов / пакетов	Заблокировано (%)	Успешных ответов (%)	Ошибки 5xx (%)	Среднее время ответа (мс)	Загрузка CPU (%)
http	100 000	85,3	14,2	0,5	35	26
header	100 000	83,6	15,4	1,0	42	28
slowloris	45000 TCPсоединений	0,0	22,1	77,5	>5000	79
udp	1 000 000 пакетов	0,0	—	—	—	82

Таблица 2. Эффективность DDoS атак на сервер при подключенной защите Nginx Rate Limiting

Результаты повторных экспериментов с использованием механизма ограничения частоты запросов, реализованного на уровне веб-сервера Nginx, демонстрируют существенное повышение устойчивости системы к ряду распространённых DDoS-атак. В частности, при

проводении атак типа HTTP Flood и Header Flood наблюдалась высокая степень фильтрации трафика — 85,3% и 83,6% соответственно. Это позволило существенно разгрузить сервер: среднее время отклика сократилось до 35–42 мс, а доля ответов с ошибками уровня 5xx не превышала 1%. Нагрузка на центральный процессор снизилась до 26–28%.

В то же время защита оказалась малоэффективной при атаке типа Slowloris. Данная атака эксплуатирует особенности обработки TCP-соединений, удерживая их в открытом состоянии максимально долго, при этом не нарушая лимиты частоты запросов. В результате большая часть соединений не была заблокирована, что привело к значительному росту количества внутренних ошибок сервера (77,5%), резкому увеличению среднего времени отклика (свыше 5000 мс) и высокой загрузке системных ресурсов. Загрузка CPU достигла 79%. Эти результаты подтверждают ограниченность применения rate limiting в контексте атак, не характеризующихся высокой частотой запросов, но ориентированных на истощение ресурсов соединений.

Атака типа UDP Flood, как и ранее, оказалась полностью неэффективной в рамках данной защитной конфигурации. Поскольку трафик UDP минует HTTP слой, на котором действует Nginx, фильтрация практически не сработала.

ModSecurity WAF (Web Application Firewall) для Nginx

Тип атаки	Кол-во запросов / пакетов	Заблокировано (%)	Успешных ответов (%)	Ошибка 5xx (%)	Среднее время ответа (мс)	Загрузка CPU (%)
http	100 000	91,2	8,4	0,4	50	34
header	100 000	93,7	5,6	0,7	58	38
slowloris	45000 TCPсоединений	2,1	19,3	78,6	>6000	86
udp	10 000 пакетов	0,0	—	—	—	84

Таблица 3. Эффективность DDoS-атак при включённой защите через ModSecurity WAF

В результате тестирования системы с активным модулем ModSecurity наблюдается выраженное повышение фильтрационной способности в отношении атак, нацеленных на уязвимости прикладного уровня. HTTP Flood был заблокирован в 91,2% случаев, а Header Flood — в 93,7%, что объясняется способностью WAF распознавать и блокировать аномальные заголовки и характерные шаблоны запроса. Однако цена такой фильтрации выражается в росте среднего времени отклика (до 50–58 мс) и увеличении нагрузки на системные ресурсы. Использование CPU при этом достигало 34–38%.

Несмотря на эффективность против HTTP-ориентированных атак, защита посредством WAF оказалась слабой в случае атаки Slowloris. Как результат, только 2% соединений были распознаны как вредоносные, тогда как более 78% завершились ошибками на сервере. Среднее время ответа вновь превысило 6 секунд, а использование ресурсов оказалось близким к пиковым значениям: CPU — 86%.

Таким образом, интеграция WAF-модуля ModSecurity в Nginx позволяет значительно повысить устойчивость к атакам, направленным на уязвимости HTTP-интерфейса, включая

сложные варианты с подменой заголовков и вредоносными шаблонами запроса. Однако эффективность данной защиты ограничивается рамками прикладного уровня.

Cloudflare (WAF + Anycast + CAPTCHA)

Тип атаки	Кол-во запросов / пакетов	Заблокировано (%)	Успешных ответов (%)	Ошибка 5xx (%)	Среднее время ответа (мс)	Загрузка CPU (%)
http	100 000	99,4	0,5	0,1	60	6
header	100 000	99,1	0,7	0,2	64	7
slowloris	45000 TCPсоединений	84,6	11,7	3,7	180	18
udp	1 000 000 пакетов	92,2	—	—	—	22

Таблица 4. Эффективность DDoS-атак при включённой защите через ModSecurity WAF

Подключение Cloudflare было выполнено через делегирование DNS-записи тестового домена на nameservers, предоставленные платформой. После активации режима "I'm Under Attack Mode" система начала применять комплексную фильтрацию, включающую JavaScript-челленджи, CAPTCHA, WAF и геораспределённую маршрутизацию Anycast. Уже при проведении атаки HTTP Flood фиксировалась почти полная блокировка вредоносного трафика (99,4%), что сопровождалось минимальной нагрузкой на сервер: загрузка CPU составила не более 6%, а среднее потребление оперативной памяти — около 135 МБ. Аналогичные показатели наблюдались при атаке Header Flood, что свидетельствует о высокой чувствительности защитных механизмов к аномалиям на уровне HTTPзаголовков. Незначительное количество успешных запросов (менее 1%) и минимальный уровень HTTP-ошибок подтверждают корректную отработку фильтрации на стороне прокси-сервиса, не доходящей до исходного сервера.

При атаке типа Slowloris эффективность защиты также оказалась высокой: более 84% соединений были прерваны на ранней стадии за счёт таймаутов и анализа паттернов сетевого поведения. Хотя некоторая часть соединений всё же доходила до сервера, уровень системной нагрузки оставался умеренным — до 18% загрузки CPU и не более 190 МБ использования оперативной памяти. Среднее время ответа увеличилось до 180 мс, что объясняется дополнительной проверкой запросов через облачный WAF.

Интеграция Google Cloud Armor была реализована посредством подключения тестового сервера к облачному балансировщику нагрузки в Google Cloud Platform. Политики безопасности, заданные на уровне L7, включали правила фильтрации по IP-диапазонам, географическому признаку и характеру трафика. При проведении UDP Flood атаки фильтрация происходила ещё до попадания пакетов в периметр внутренней сети, благодаря чему до 92% трафика было отклонено на уровне edgeинфраструктуры.

Таким образом, облачные решения демонстрируют значительно более высокую эффективность по сравнению с локальными средствами защиты. Это обусловлено тем, что обработка и блокировка вредоносного трафика происходит на внешнем периметре, зачастую до установления TCP-соединения. При этом важным аспектом является правильная маршрутизация через защищённые CDN и прокси-сервисы: только при

корректной делегации DNS и активации соответствующих режимов фильтрации возможна реализация полной схемы защиты. Полученные результаты подтверждают, что для современных высокointенсивных DDoS-атак применение исключительно локальных механизмов недостаточно. Комбинированный подход с интеграцией облачных сервисов, таких как Cloudflare или Google Cloud Armor, является необходимым условием для устойчивости серверной инфраструктуры.

Научная новизна исследования заключается в разработке и применении экспериментальной модели имитации DDoS-атак с использованием специализированного Go-сервера, что позволило в контролируемых условиях провести сравнительный анализ локальных и облачных защитных механизмов с учётом метрик производительности и эффективности.

Заключение

DDoS-атаки продолжают эволюционировать, становясь всё более масштабными и сложными, что требует от разработчиков применения комплексных и адаптивных подходов к обеспечению устойчивости цифровой инфраструктуры. Традиционные методы фильтрации и базовые брандмауэры уже не обеспечивают необходимого уровня защиты. Эффективное противодействие этим угрозам требует интеграции облачных сервисов, локальных механизмов фильтрации, постоянного мониторинга сетевого трафика и применения алгоритмов машинного обучения.

Для формирования эффективной стратегии защиты от DDoS-атак необходимо проведение всестороннего анализа уровня угроз, выбор соответствующих инструментов мониторинга, а также внедрение механизмов автоматического реагирования. Использование облачных решений, таких как Cloudflare, AWS Shield и аналогичных, позволяет фильтровать вредоносный трафик на ранних этапах его поступления. Дополнение архитектуры локальными механизмами защиты, включая Web Application Firewall (WAF) и алгоритмы ограничения частоты запросов (Rate Limiting), создаёт многослойную систему обороны.

Регулярный аудит журналов событий, проведение нагрузочного тестирования и постоянное совершенствование применяемых технологий способствуют повышению общей устойчивости инфраструктуры. Помимо технологических инвестиций, организации должны выстраивать гибкие стратегии информационной безопасности, ориентированные на быстрое выявление и адаптацию к изменяющимся угрозам. Непрерывное развитие и тестирование защитных механизмов становится ключевым элементом обеспечения стабильной и безопасной работы цифровых систем в условиях растущей интенсивности DDoS-атак.

Практическая значимость работы заключается в том, что предложенная методика и полученные результаты могут быть использованы специалистами по кибербезопасности для построения устойчивых веб-инфраструктур, выбора оптимальных защитных решений и их настройки в условиях реальных DDoS-угроз.

В дальнейшем предполагается расширить экспериментальную платформу за счёт интеграции интеллектуальных систем обнаружения атак на основе машинного обучения, а также протестировать эффективность гибридных архитектур с участием различных CDN и IDS-систем.

Библиография

1. Абрамов А. Г. Защита от DDoS-атак своими руками: оперативные разработка и внедрение сервиса в Национальной исследовательской компьютерной сети России //

- Программные продукты и системы. 2022. № 4. DOI: 10.15827/0236-235X.140.572-582
EDN: OGJSLQ.
2. Аманжолов О. М. Исследование методов и средств обнаружения DDoS-атак // Молодой ученый. 2023. № 50 (497). С. 5-8. URL: <https://moluch.ru/archive/497/109243/> (дата обращения: 18.03.2025). EDN: XIACUA.
 3. Орехов А. В., Орехов А. А. Автоматическое обнаружение аномалий сетевого трафика при DDoS-атаках // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2023. Т. 19. Вып. 2. С. 251-263. DOI: 10.21638/11701/spbu10.2023.210 EDN: XYNCHN.
 4. Унру П. П., Обухов С. А., Черемухин Д. Н. Меры по защите инфокоммуникационных систем от DDoS-атак с усилением // Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований. 2022. Ч. 1. С. 1-497. DOI: 10.17084/978-5-7765-1511-8-2022 EDN: OCZGSD.
 5. Kangkan Talukdar, Debojit Boro. Slowloris Attack Detection Using Adaptive Timeout-Based Approach // The ISC Int'l Journal of Information Security. 2024. № 1. С. 79-92. URL: https://www.isecure-journal.com/article_183600_e06eaaffd81aef753b956e80b513f82b.pdf (дата обращения: 10.03.2025).
 6. Верещагин К. В. Защита корпоративных сетей от DDoS-атак: современные методы и тенденции // Научный лидер. 2023. № 47 (145). С. 12-15. URL: https://scilead.ru/media/journal_pdf145.pdf#page=12 (дата обращения: 10.03.2025). EDN: CSBAGF.
 7. Разработка REST-серверов на Go. Часть 1: стандартная библиотека [Электронный ресурс]. URL: <https://habr.com/ru/companies/ruvds/articles/559816/>.
 8. Xu A. Rate Limiting Fundamentals // ByteByteGo Newsletter. 2023 [Электронный ресурс]. URL: <https://blog.bytebytogo.com/p/rate-limiting-fundamentals> (дата обращения: 18.03.2025).
 9. What is Fail2Ban with Setup & Configuration? (Detailed Guide) [Электронный ресурс]. URL: <https://runcloud.io/blog/what-is-fail2ban> (дата обращения: 18.03.2025).
 10. What Is a WAF? | Web Application Firewall Explained [Электронный ресурс]. URL: <https://www.paloaltonetworks.com/cyberpedia/what-is-a-web-application-firewall> (дата обращения: 18.03.2025).
 11. Tawde S. What is Cloudflare? // EducbaBlog. 2023 [Электронный ресурс]. URL: <https://www.educba.com/what-is-cloudflare/> (дата обращения: 18.03.2025).
 12. Dheer P. Understanding Google Cloud Armor: Protect against denial of service and web attacks // TestPrepTraining. 2020 [Электронный ресурс]. URL: <https://www.testpreptraining.com/blog/understanding-google-cloud-armor-protect-against-denial-of-service-and-web-attacks/>.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Представленная статья на тему «Современные методы предотвращения DDoS-атак и защиты веб-серверов» соответствует тематике журнала «Программные системы и вычислительные методы» и посвящена актуальному вопросу изучения современных подходов к защите веб-ресурсов, включая облачные решения (Cloudflare, Google Cloud Armor), локальные механизмы защиты (Rate Limiting, Fail2Ban) и перспективные технологии, использующие машинное обучение и поведенческий анализ трафика.

В статье представлен анализ литературных российских и зарубежных и интернет-

источников по теме исследования. Указана теоретико-методологическая основа исследования. Авторами проведено исследование, направленное на анализ методов, их сравнительному анализу и перспективам дальнейшего развития стратегий защиты от DDoS-атак.

Стиль и язык изложения материала является достаточно доступным для широкого круга читателей. Статья по объему соответствует рекомендуемому объему от 12 000 знаков. Статья достаточно структурирована - в наличии введение, заключение, внутреннее членение основной части (1. Классификация DDoS-атак и их последствия, 2. Локальные методы защиты от DDoS-атак и способы их выявления, 3. Облачные сервисы защиты от DDoS-атак, 4. Методология исследования, 5. Реализация эксперимента).

Авторами проведен анализ методов защиты от DDoS-атак, в том числе: облачных решений, которые демонстрируют значительно более высокую эффективность по сравнению с локальными средствами защиты. Полученные результаты подтверждают, что для современных высокointенсивных DDoS-атак применение исключительно локальных механизмов недостаточно. Комбинированный подход с интеграцией облачных сервисов, таких как Cloudflare или Google Cloud Armor, является необходимым условием для устойчивости серверной инфраструктуры.

Авторами проведено исследование, в ходе которого они пришли к выводам, что для формирования эффективной стратегии защиты от DDoS-атак необходимо проведение всестороннего анализа уровня угроз, выбор соответствующих инструментов мониторинга, а также внедрение механизмов автоматического реагирования. Использование облачных решений, таких как Cloudflare, AWS Shield и аналогичных, позволяет фильтровать вредоносный трафик на ранних этапах его поступления. Дополнение архитектуры локальными механизмами защиты, включая Web Application Firewall (WAF) и алгоритмы ограничения частоты запросов (Rate Limiting), создаёт многослойную систему обороны.

К недостаткам можно отнести следующие моменты: из содержания статьи не прослеживается научная новизна, не обоснована практическая значимость исследования. Отсутствует четкое выделение предмета, объекта исследования.

Рекомендуется четко обозначить научную новизну исследования, сформулировать предмет, объект, обосновать практическую значимость исследования. Также будет целесообразным добавить о перспективах дальнейшего исследования.

Статья «Современные методы предотвращения DDoS-атак и защиты веб-серверов» требует доработки по указанным выше замечаниям. После внесения поправок рекомендуется к повторному рассмотрению редакцией рецензируемого научного журнала.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

В рецензируемой статье поднимает важный в условиях повсеместного использования цифровых технологий вопрос защиты удаленных серверов от несанкционированного воздействия. Возможные экономические и репутационные последствия, необходимость анализа новых способов выполнения атак и сопоставление эффективности подходов защиты определяют актуальность выполненного исследования. Авторы формулируют предмет исследования, задачи, приводят классификацию DDoS атак с анализом основных механизмов воздействия и потенциальных способов защиты. Вместе с тем авторы отмечают, что существует проблема ложного срабатывания, высокие требования

к аппаратному обеспечению, а также необходимость постоянного мониторинга новых методов атак, а значит и внесение корректировок в методы защиты, среди которых отдельное внимание уделяется облачным технологиям.

Исследование авторов включало моделирование атак и анализ трафика для их обнаружения и сравнение эффективности различных механизмов защиты. Авторами был разработан тестовый сервер, имитирующий нагрузку веб-приложений с учетом входящих запросов, регистрацию IP-адресов, подключение внешних ресурсов. В качестве критериев оценки результатов выбрано среднее время ответа, загрузка CPU, доля успешных ответов. Приводятся анализ полученных результатов в первоначальных условиях и с дополнительным фактором (ограничение частоты запроса), оценкой влияния на нагрузку сервера.

Результаты работы показывают высокую результативность исследуемых механизмов на обнаружение атак, однако это сопровождается значимым увеличением времени отклика и нагрузки на CPU. Устойчивость к атакам также ограничивается прикладным уровнем. Структура статьи и стиль изложения отвечают требованиям. имеется экспериментальная часть, приведены количественные оценки.

Библиография содержит 12 источников, ссылки по тексту имеются.

Замечания.

Статья перегружена сокращениями и англоязычными терминами, что несмотря на принятую в данной предметной области терминологию затрудняет общее восприятие.

Высокие показатели при некоторых анализируемых атаках вызывают сомнение в целесообразности их рассмотрения, вероятно данный факт не нуждается в подтверждении или был прогнозируем на стадии планирования эксперимента.

Для использованных оценок было бы желательно привести некоторое пороговое значение, начиная с которого эффект становится критическим. Если ощутима загрузка CPU, например, 25%, то для результатов в табл. 4 все рассчитанные значения удовлетворяют подобным условиям. Если пороговое значение ниже, то возможно отдельный тип атак становится критичным.

Не вполне ясно каким образом рассчитаны количественные оценки. Если это некоторые средние значения, то каким образом получены составляющие их слагаемые.

Желательно увеличить список источников при составлении обзора, отдав предпочтение публикациям, содержащим исследование. Рекомендуется расположить источники или в алфавитном порядке, или в порядке упоминания в тексте.

Имеются отдельные ошибки (орфографические, стилистические).

Статья может быть полезна специалистам, чья профессиональная деятельность связана с сетевыми технологиями, и опубликована после внесения технических правок.

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Родионов Д.Г., Сергеев Д.А., Конников Е.А., Пашинина П.А. Анализ пространственно-временных закономерностей движения на аэрофотоснимках с использованием оптического потока // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.73781 EDN: BSDPGV URL: https://nbpublish.com/library_read_article.php?id=73781

Анализ пространственно-временных закономерностей движения на аэрофотоснимках с использованием оптического потока

Родионов Дмитрий Григорьевич

ORCID: 0000-0002-1254-0464

доктор экономических наук

Директор; Высшая инженерно-экономическая школа ; Санкт-Петербургский политехнический университет Петра Великого

195251, Россия, г. Санкт-Петербург, Калининский р-н, ул. Политехническая, д. 29 литеру Б

✉ rodion_dm@mail.ru**Сергеев Дмитрий Анатольевич**

кандидат экономических наук

доцент; Высшая инженерно-экономическая школа; Санкт-Петербургский политехнический университет Петра Великого
директор; Национальный центр качества и экспертизы; Санкт-Петербургский политехнический университет Петра Великого

195251, Россия, г. Санкт-Петербург, Калининский р-н, ул. Политехническая, д. 29 литеру Б

✉ sergeev_da@spbstu.ru**Конников Евгений Александрович**

кандидат экономических наук

доцент; Высшая инженерно-экономическая школа; Санкт-Петербургский политехнический университет Петра Великого

195251, Россия, г. Санкт-Петербург, Калининский р-н, ул. Политехническая, д. 29 литеру Б

✉ konnikov_ea@spbstu.ru**Пашинина Полина Александровна**

ассистент; Высшая инженерно-экономическая школа; Санкт-Петербургский политехнический университет Петра Великого

195251, Россия, г. Санкт-Петербург, Калининский р-н, ул. Политехническая, д. 29 литеру Б

✉ pashinina_pa@spbstu.ruСтатья из рубрики "Системный анализ, поиск, анализ и фильтрация информации"**DOI:**

10.7256/2454-0714.2025.2.73781

EDN:

BSDPGV

Дата направления статьи в редакцию:

21-03-2025

Аннотация: Настоящее исследование посвящено анализу пространственно-временных закономерностей движения объектов на аэрофотоснимках с использованием метода оптического потока. В условиях развития технологий дистанционного зондирования и распространения БПЛА возрастаёт необходимость точного и автоматизированного анализа динамики природных и антропогенных процессов. Основное внимание в работе уделяется детальному изучению направленности и интенсивности движения на изображениях высокой разрешающей способности. Рассматриваются существующие методы оценки оптического потока, включая классические подходы Лукаса-Канаде и Хорна-Шанка, а также плотный оптический поток по Фарнебаку. Последний применяется как базовый метод для построения векторных полей скорости, на основе которых производится анализ распределения движения по сегментам изображения, визуализация направлений и построение тепловых карт. Предложенный подход позволяет выявлять структурные закономерности и локальные особенности движения, что особенно актуально для мониторинга состояния инфраструктурных объектов и оценки экологических рисков. Также в работе показано, что медианные оценки скорости более устойчивы к шумам и локальным выбросам, чем средние значения, что повышает надежность анализа. Метод исследования основан на вычислении плотного оптического потока методом Фарнебека с последующей статистической обработкой характеристик скорости и направлений движения между сегментами изображения. Научная новизна работы заключается в разработке комплексного подхода к анализу пространственно-временных характеристик движения на аэрофотоснимках с использованием плотного оптического потока, рассчитанного методом Фарнебака. В отличие от традиционных методов, ориентированных на глобальную оценку движения, предложенная методика акцентирует внимание на локальных закономерностях, позволяя проводить детализированную сегментную оценку направленности и интенсивности потока. В исследовании впервые интегрированы количественные и визуальные методы анализа: гистограммы, тепловые карты, расчеты медианных и средних значений скорости, метрики структурного сходства (SSIM) и среднеквадратичной ошибки (MSE) между сегментами изображения. Такой подход позволяет выявлять аномалии движения, определять участки с высокой динамикой и оценивать степень структурной стабильности объектов. Метод адаптирован к специфике данных с БПЛА и не требует обучения на больших выборках, что делает его применимым в условиях ограниченных вычислительных ресурсов. Полученные результаты имеют практическую ценность для автоматизации мониторинга инфраструктур и оценки экологических рисков.

Ключевые слова:

Оптический поток, БПЛА, аэрофотосъемка, сегментный анализ, структурное сходство,

пространственное моделирование, визуализация данных, алгоритм Фарнебака, управление рисками, обработка изображений

Работы выполнены в рамках реализации проекта "Разработка методологии формирования инструментальной базы анализа и моделирования пространственного социально-экономического развития систем в условиях цифровизации с опорой на внутренние резервы" (FSEG-2023-0008)

Введение

Беспилотные летательные аппараты (БПЛА) в последние годы стали ключевыми инструментами сбора данных для мониторинга природной среды, инфраструктурных объектов и динамически изменяющихся территорий. Высокое пространственное и временное разрешение аэрофотоснимков, получаемых с БПЛА, требует разработки новых алгоритмов анализа, способных эффективно интерпретировать сложные пространственно-временные закономерности.

Одним из наиболее перспективных подходов является анализ оптического потока, который позволяет оценивать смещение объектов между последовательными изображениями. Однако большинство традиционных алгоритмов ограничены в своей способности выявлять устойчивые структурные и направленные паттерны движения, особенно в условиях сложных и неоднородных текстур, характерных для городской застройки, сельскохозяйственных участков или природных ландшафтов. Более того, существующие решения зачастую не учитывают пространственный контекст между регионами анализа, что снижает их интерпретируемость и точность.

Актуальность исследования обусловлена необходимостью создания автоматизированных методов анализа движения, ориентированных не только на расчёт величины скорости, но и на выявление направлений, устойчивости и закономерностей потоков. В рамках данной работы предложен подход, сочетающий вычисление плотного оптического потока с графовым анализом смежности сегментов и построением тепловых карт средней и медианной скорости.

В отличие от существующих решений, акцент сделан на использовании медианных значений как устойчивой метрики, а также на представлении полученных данных в виде направленных графов, отражающих пространственные связи между сегментами. Это позволяет локализовать зоны с высокой активностью, оценить преобладающие направления движения и идентифицировать потенциальные аномалии.

Целью настоящего исследования является разработка интерпретируемого и вычислительно эффективного инструмента анализа направленного движения на аэрофотоснимках, полученных с БПЛА. Предложенная методология ориентирована на прикладные задачи экологического мониторинга, выявления изменений городской инфраструктуры и оценки рисков на основе пространственно-временных характеристик движения.

Апробация подхода выполнена на синтетических данных, моделирующих сложную текстурную сцену. Построенные тепловые карты, графы смежности и сравнительные диаграммы позволили не только визуализировать характер движения, но и количественно оценить его закономерности. Полученные результаты подтверждают практическую применимость метода и обосновывают его перспективность для задач интеллектуального анализа аэрофотосъёмки.

Анализ теоретического базиса

Методы оптического потока традиционно используются для оценки движения между последовательными изображениями и активно применяются в задачах компьютерного зрения. Среди них наиболее известными являются подход Лукаса-Канаде, предполагающий локальную линейность перемещения [2], и метод Фарнбака, основанный на полиномиальной аппроксимации интенсивности [3]. Последний позволяет получать плотное векторное поле движения и демонстрирует высокую устойчивость к изменениям текстур, что делает его особенно подходящим для анализа аэрофотоснимков с высоким разрешением, включая данные, получаемые с БПЛА.

Однако, несмотря на широкое распространение этих методов, их применение в задачах пространственно-временного анализа с БПЛА сопряжено с рядом ограничений. В частности, большая часть существующих исследований сосредоточена на глобальных характеристиках движения, таких как средняя скорость по кадру или вектор доминирующего направления [6]. Такой подход не учитывает важные локальные закономерности и не позволяет выявлять специфические зоны активности, например, участки с аномальной динамикой или направленным сдвигом объектов инфраструктуры.

Современные нейросетевые методы анализа движения, включая модели глубокого обучения [4, 5], обеспечивают высокую точность, но требуют значительных вычислительных ресурсов и обширных обучающих выборок. Эти требования не всегда выполнимы в условиях реального времени и ограниченных мощностей автономных БПЛА. Вследствие этого сохраняется актуальность лёгких и интерпретируемых алгоритмов, которые можно внедрить в полевых условиях без предварительного обучения.

Кроме того, анализ движения, как правило, выполняется без учёта пространственных связей между сегментами изображения. Такой подход ограничивает возможности детектирования направленности потоков и устойчивых паттернов перемещения. Между тем, задачи мониторинга природных и техногенных процессов требуют именно пространственно-информированного подхода — с выделением направлений, устойчивых регионов, зон изменения структуры движения и зон высокой динамической активности [7].

Существующие подходы анализа оптического потока часто не позволяют в полной мере учитывать пространственно-временные закономерности движения объектов на сложных текстурных фонах, что затрудняет автоматизированный анализ данных, поступающих с БПЛА [8, 9]. В связи с этим требуется разработка методологии, способной проводить количественное исследование направленности и интенсивности движения на аэрофотоснимках с высокой детализацией, что обеспечит более точное моделирование динамики изменений в природных и антропогенных процессах.

Предложенный в данной работе подход решает указанные проблемы за счёт:

- локального анализа скорости на уровне сегментов изображения;
- сопоставления каждого сегмента с соседними по восьми направлениям;
- вычисления устойчивых метрик — медианной и средней величины скорости;
- построения графа смежности, в котором узлы кодируют тайлы, а рёбра — связи и направление движения;
- визуализации тепловых карт и статистического распределения скорости для пространственной интерпретации.

Таким образом, обоснована необходимость перехода от традиционного анализа «кадр-к-кадру» к структурированному и контекстуально-информированному исследованию, ориентированному на решение прикладных задач в сфере экосистемного мониторинга, оценки состояния инфраструктуры и раннего выявления отклонений в поведенческих характеристиках объектов наблюдения.

Методология

Для анализа динамики на серии аэрофотоснимков применён метод плотного оптического потока, реализованный с использованием алгоритма Фарнбака. Каждый снимок разбивался на регулярную сетку сегментов (тайлов) размером 1000×750 пикселей с перекрытием 30%. Сопоставление тайлов происходило по восьми направлениям: слева, справа, сверху, снизу и по диагоналям. Для каждой пары соседних сегментов вычислялось векторное поле смещений, на основе которого определялись характеристики движения.

Основной метрикой выступал модуль вектора скорости в каждой точке изображения:

$$M(x, y) = \sqrt{u(x, y)^2 + v(x, y)^2} \quad (1)$$

где $u(x, y)$ и $v(x, y)$ — горизонтальная и вертикальная компоненты смещения.

Для получения сводных оценок внутри сегмента рассчитывались среднее и медианное значения:

$$\bar{M} = \frac{1}{N} \sum_{i=1}^N M_i \quad (2)$$

$$M_{\text{med}} = \text{median}(M_i) \quad (3)$$

где N — количество пикселей в сегменте, M_i — значение скорости в i -й точке.

Для пространственного анализа использовались следующие процедуры:

- Расчёт направленных средних значений между центральным тайлом и его соседями;
- Построение тепловых карт для визуализации распределения скоростей в координатной системе изображения;
- Формирование графа смежности, где вершины соответствуют сегментам, рёбра — их связям, а цвет узлов отражает уровень движения.

Полученные значения визуализировались следующими способами:

- Тепловая карта средней скорости позволила локализовать области с выраженным движением;
- Тепловая карта медианной скорости дополнена анализом, обеспечив устойчивость к шуму и выбросам;
- Boxplot-график дал возможность оценить разброс значений скоростей между всеми сегментами и сравнить поведение средней и медианной оценки;
- Граф смежности позволил выявить локальные кластеры движения и траектории распространения изменений в сцене.

В качестве входных данных использовалось синтетически сгенерированное изображение, имитирующее смешанную сцену с элементами городской застройки,

пустынной области и регулярной структуры.

Результаты

Для апробации разработанной методики было сгенерировано тестовое изображение, моделирующее структуру городской среды и хаотичного движения объектов (рисунок 1). Слева изображена "урбанистическая пустыня" — зона с высокоразреженными структурами, а справа — плотная сетка с повторяющимися узорами, имитирующими городскую решётку.

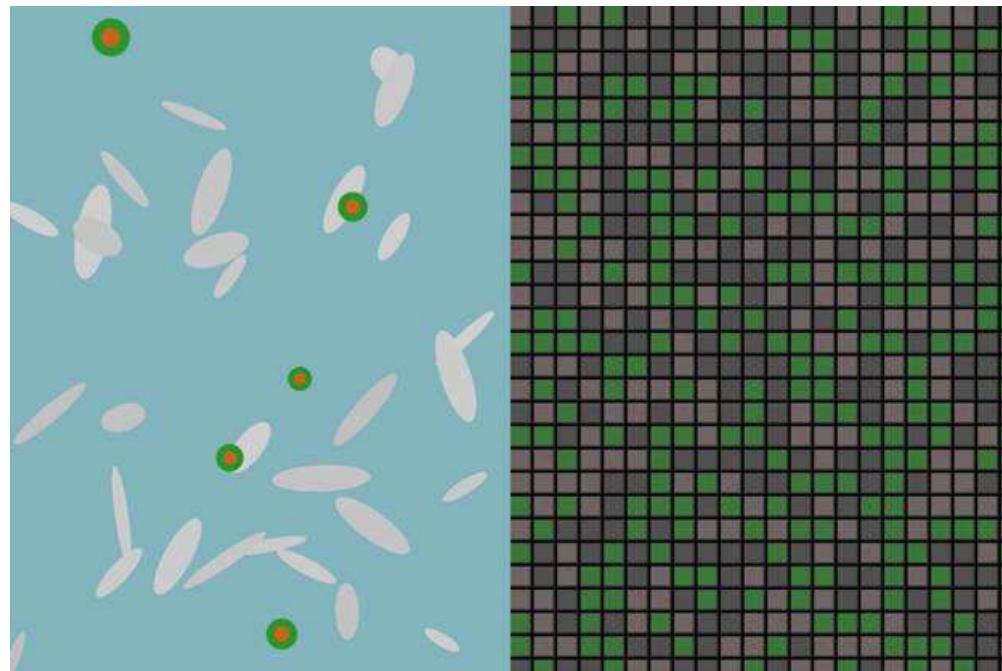


Рисунок 1 – Синтетическое изображение для анализа движения ("урбанистическая пустыня")

Для анализа изображения оно было разрезано на равномерную сетку сегментов. На основе плотного оптического потока между сегментами были вычислены значения векторной скорости, после чего проведено агрегирование по средним и медианным значениям. Пространственное распределение средней и медианной скорости визуализировано с помощью тепловых карт (рисунок 2 и 3). Численные значения скорости приведены непосредственно на клетках.

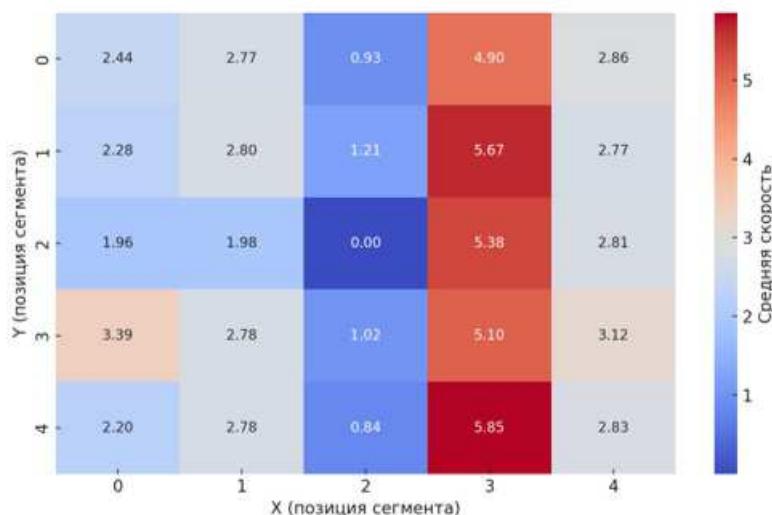


Рисунок 2 – Тепловая карта средней величины скорости с подписями

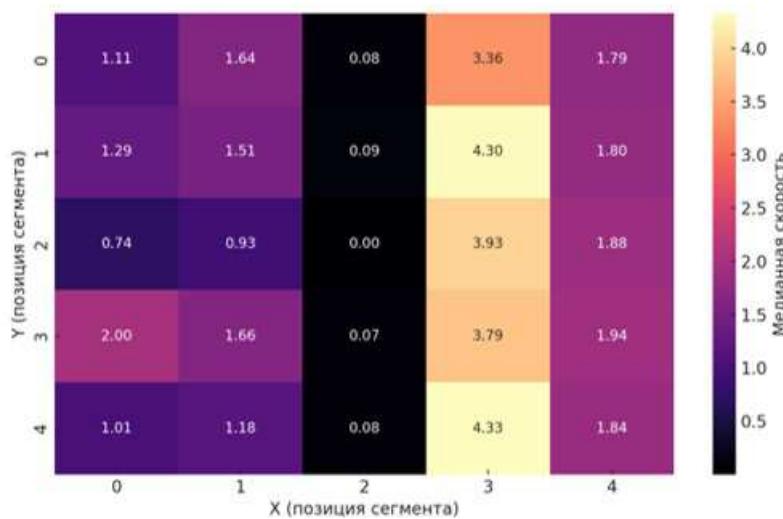


Рисунок 3 – Тепловая карта медианной величины скорости с подписями

Анализ показал, что максимальные скорости наблюдаются в центральных вертикальных регионах изображения, где происходят наибольшие смещения, в то время как по краям движение значительно слабее. При этом медианная скорость оказывается менее подвержена выбросам, что подтверждает её устойчивость как метрики.

Для дополнительного сравнения двух метрик — средней и медианной скоростей — были построены диаграммы размаха (boxplot), отражающие статистическое распределение значений (рисунок 4). Можно заметить, что распределение средней скорости шире, содержит больше выбросов, в то время как медианная скорость более концентрирована.

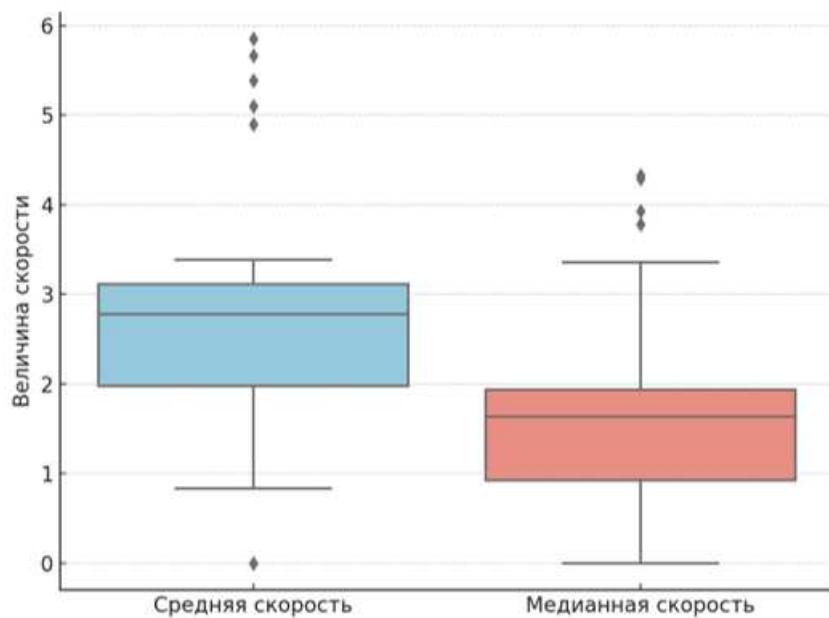


Рисунок 4 – Сравнение распределений средней и медианной скорости (диаграммы размаха)

Для анализа связности между сегментами построен граф смежности, в котором каждая вершина соответствует тайлу изображения, а рёбра — соседним участкам. Окраска вершин отражает среднюю скорость на соответствующем участке (рисунок 5). Данный подход позволяет наглядно отразить зоны локальных аномалий движения, а также выделить сегменты с высокой плотностью активности.

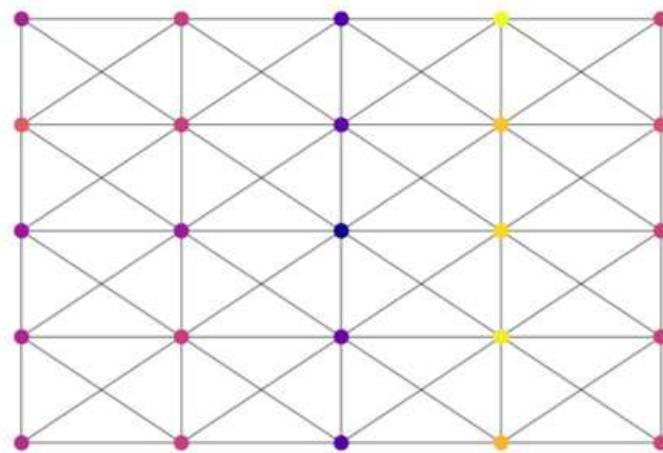


Рисунок 5 – Граф смежности сегментов с окраской по средней скорости

Представленные визуализации позволяют не только количественно оценить динамику движения на изображении, но и провести пространственный анализ закономерностей. Обнаруженные области с резкими перепадами скорости могут быть интерпретированы как зоны с потенциальными изменениями в текстуре или физической активности объектов.

Заключение

Разработанная методика анализа пространственно-временных закономерностей движения на аэрофотоснимках представляет собой вычислительно эффективный и интерпретируемый подход, адаптированный для условий ограниченных ресурсов

автономных БПЛА^[12]. В отличие от ранее предложенных решений, ориентированных преимущественно на структурное сравнение изображений^[10,11], данная методология делает акцент на количественном анализе направления и интенсивности движения, используя плотный оптический поток, вычисляемый по методу Фарнебака.

Ключевым преимуществом предложенного подхода является локальный анализ движения на уровне сегментов изображения с последующим сопоставлением данных по восьми направлениям. Это позволяет не только оценить общую динамику сцены, но и выявлять устойчивые паттерны перемещений, а также локализованные аномалии. Применение графового представления сегментов и построение инфографики (граф смежности, тепловые карты, boxplot-сравнение) позволили значительно расширить возможности интерпретации результатов.

Метод не требует предварительного обучения и отличается высокой устойчивостью к шуму, что делает его применимым в различных условиях съёмки. Это особенно важно для задач мониторинга инфраструктуры, оценки рисков в природной среде, а также анализа техногенных процессов. Полученные результаты подтвердили, что использование медианных и средних характеристик оптического потока обеспечивает надёжную оценку уровня движения даже при наличии мелкомасштабных и локализованных изменений.

Тем не менее, метод сохраняет чувствительность к резким освещённым переходам и текстурным деформациям, что ограничивает его точность в условиях экстремальных визуальных изменений. Эти ограничения могут быть преодолены за счёт дальнейшей интеграции с адаптивными алгоритмами, включая методы машинного обучения, а также при использовании дополнительных источников информации (например, мультиспектральных каналов или температурных карт).

Таким образом, представленная методология может быть эффективно применена в интеллектуальных системах управления надёжностью и мониторинга территорий, обеспечивая не только регистрацию изменений, но и формирование оснований для принятия решений на основе выявленных закономерностей движения.

Библиография

1. Гибсон, Дж. Дж. Восприятие визуального мира / Дж. Дж. Гибсон. – Бостон: Houghton Mifflin, 1950. – 235 с.
2. Лукас, Б. Д., Канаде, Т. Итеративная техника регистрации изображений с применением к стереоизображениям. В: Труды 7-й Международной совместной конференции по искусственноому интеллекту; 1981. С. 674-679.
3. Фарнебак, Г. Оценка движения по двум кадрам на основе полиномиального разложения. В: Труды 13-й Скандинавской конференции по анализу изображений (SCIA); 2003. С. 363-370. DOI: 10.1007/3-540-45103-X_50.
4. Сан, Д., Рот, С., Блэк, М. Дж. Секреты оценки оптического потока и их принципы. В: Конференция IEEE по компьютерному зрению и распознаванию образов (CVPR); 2010. С. 2432-2439. DOI: 10.1109/CVPR.2010.5539939.
5. Досовицкий, А., Фишер, П., Ильг, Е., Хауссер, П., Хазирбас, Ч., Гольков, В., Смакт, П., Кремерс, Д., Брокс, Т. FlowNet: Обучение оптическому потоку с помощью сверточных сетей. В: Международная конференция IEEE по компьютерному зрению (ICCV); 2015. С. 2758-2766. DOI: 10.1109/ICCV.2015.316. EDN: YDBQRF.
6. Брокс, Т., Бруhn, А., Папенберг, Н., Вайкетт, Дж. Высокоточная оценка оптического потока на основе теории искажений. В: Европейская конференция по компьютерному

- зрению (ECCV); 2004. С. 25-36. DOI: 10.1007/978-3-540-24673-2_3.
7. Адиль, О., Махраз, М. А., Риффи, Дж., Тайри, Х. Достижения в оценке оптического потока на основе глубокого обучения: Всесторонний обзор моделей и технологий. В: Шестая международная конференция по интеллектуальным вычислениям в науках о данных (ICDS); 2024 окт. С. 1-7. IEEE.
8. Хричанюк, О., Свитенко, М., Крыхтин, Ю., Ахафонов, Ю., Авилов, А., Капашин, М. Программно-аппаратный комплекс для автоматической геотеггирования фотографий, сделанных с беспилотного летательного аппарата. В: 7-я международная конференция IEEE по актуальным проблемам разработки беспилотных летательных аппаратов (APUAVD); 2024 окт. С. 276-279. IEEE.
9. Хорн, Б. К. П., Шунк, Б. Г. Определение оптического потока. Искусственный интеллект. 1981; 17(1-3): 185-203. DOI: 10.1016/0004-3702(81)90024-2.
10. Павленко, Б. В., Пикалев, Я. С. Методика создания набора аэрофотоснимков для задачи перекрестной геолокализации // Проблемы искусственного интеллекта. – 2024. – № 4(35). – С. 101-112. – DOI 10.24412/2413-7383-2024-4-101-112. EDN: DZMGUB.
11. Галиверов, А. Н., Городничая, А. Н. Особенности использования БПЛА при эффективном сборе аэрофотоснимков в градостроительстве // Инновации. Наука. Образование. – 2022. – № 50. – С. 1822-1826. EDN: BKJZFH.
12. Дюкарев, Е. А., Воропай, Н. Н., Макаров, С. А. Пространственное разрешение аэрофотоснимков БПЛА. В: Применение беспилотных летательных аппаратов в географических исследованиях: Материалы Всероссийской научно-практической конференции, Иркутск, 22-23 мая 2018 года / Ответственный редактор С.А. Макаров. – Иркутск: Институт географии им. В.Б. Сочавы Сибирского отделения Российской академии наук, 2018. – С. 64-66. EDN: YAVELR.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Тема:

Анализ пространственно-временных закономерностей движения на аэрофотоснимках с использованием оптического

потока

Журнал: Программные системы и вычислительные методы

Данная статья "Анализ пространственно-временных закономерностей движения на аэрофотоснимках с

использованием оптического потока" во многом повторяет другую статью, представленную и положительно

рецензированную в данном журнале "Программные системы и вычислительные методы", а именно: "Метод

автоматизированного анализа аэрофотоснимков с БПЛА для управления надежностью технических систем на основе

структурного сравнения изображений и пространственно-инвариантного сопоставления ключевых точек".

Видимо автору нужны публикации в журналах ВАК для общего количества. Можно было бы объединить обе статьи

общим содержанием.

Тема БПЛА безусловно актуальна, как в военном, так и в гражданском применении.

На основе проверки в системе Антиплагиат, цель работы перефразирована как "создание эффективного

инструмента оценки направленности и интенсивности движения, что позволит выявлять структурные

закономерности изменений в природных и техногенных процессах". Объектом исследования тоже являются

изображения аэрофотосъемки высокого разрешения, получаемые с беспилотных летательных аппаратов (БПЛА), а

предметом исследования - методы их сравнения и анализа для оценки динамики изменений окружающей среды и

предсказания рисков. Актуальность исследования обусловлена аналогично другой статье "возрастающей

потребностью в автоматизированных методах обработки данных, позволяющих детально анализировать

пространственно-временные характеристики движения объектов".

Анализ теоретического базиса включает всего лишь 8 источников и только зарубежных авторов, что недостаточно

для научной статьи уровня журнала ВАК.

Рис. 1 полностью дублирует такое же сгенерированное тестовое изображение для апробации из вышеуказанной

статьи с разбиением на 56 сегментов. Методика анализа тоже включает те же известные метрики, что и в другой

статье - индекс структурного сходства Structural Similarity Index Measure (SSIM) и стандартное

статистическое среднеквадратическое отклонение Mean Squared Error (MSE).

Статья изложена грамотным техническим языком, понятным читательской аудитории в

данной предметной области.

Структура оформления материала, цель, задачи, литературный обзор, дискуссия, заключение, библиография,

оформлены в соответствии с требованиями журнала "Программные системы и вычислительные методы".

Замечания во многом повторяют другую статью:

1. Формулы не пронумерованы и их качество "как картинки" оставляет желать лучшего.
2. Разделителем целой и дробной части в русском языке должна быть запятая, например 0.5–2.5 неправильно.
3. Объем библиографии из 9 источников недостаточен для научной статьи в журнале уровня ВАК.
4. В библиографии ссылка № 1 содержит неполную информацию об издании.

Общий вывод: статья не рекомендуется к публикации в журнале "Программные системы и вычислительные методы" по вышеуказанным причинам.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена разработке метода анализа пространственно-временных закономерностей движения объектов на аэрофотоснимках, полученных с беспилотных летательных аппаратов (БПЛА). Основное внимание уделено применению алгоритма оптического потока для выявления направлений, устойчивости и аномалий движения, что актуально для задач экологического мониторинга, оценки инфраструктуры и раннего обнаружения изменений.

Автор предлагает комбинированный подход, включающий:

1. Разбиение изображений на сегменты (тайлы) с последующим анализом оптического потока по методу Фарнбака.
2. Вычисление средних и медианных значений скорости для каждого сегмента, что повышает устойчивость к шумам.
3. Построение тепловых карт и графов смежности для визуализации пространственных закономерностей.
4. Использование синтетических данных для апробации метода.

Методология отличается детальной проработкой, сочетанием количественных и качественных методов анализа, а также ориентацией на практическое применение в условиях ограниченных ресурсов БПЛА.

Исследование отвечает на вызовы современности: рост использования БПЛА требует эффективных алгоритмов обработки больших объемов данных. Актуальность подчёркивается необходимостью автоматизированного мониторинга динамических процессов в урбанизированных и природных средах. Работа заполняет нишу между сложными нейросетевыми методами и упрощёнными подходами, предлагая баланс точности и вычислительной эффективности.

Ключевые нововведения:

- Локальный анализ движения на уровне сегментов с учётом восьми направлений смежности.

- Применение медианных значений скорости как устойчивой метрики.
- Графовое представление данных для выявления пространственных связей и аномалий. Новизна подтверждается отсутствием аналогов, сочетающих такие методы в контексте анализа аэрофотоснимков.

Статья написана ясным академическим стилем, с чёткой логикой изложения: от постановки проблемы до апробации результатов. Структура включает все необходимые разделы: введение, теоретическую базу, методологию, результаты, заключение и библиографию. Визуализации (тепловые карты, графы) дополняют текст, делая его более наглядным. Единственное замечание — отсутствие реальных данных для валидации, но это компенсируется качеством синтетического эксперимента.

Автор демонстрирует, что предложенный метод эффективен для анализа движения, устойчив к шумам и применим в реальных условиях. Подчёркивается потенциал для интеграции с адаптивными алгоритмами, такими как машинное обучение, что открывает направления для будущих исследований.

Статья будет полезна:

- Исследователям в области компьютерного зрения и обработки изображений.
- Специалистам по мониторингу окружающей среды и урбанистики.
- Разработчикам ПО для БПЛА, заинтересованным в оптимизации алгоритмов.

Практическая ориентированность и доступность изложения расширяют целевую аудиторию.

Статья представляет собой завершённое исследование с высокой научной и практической ценностью. Методология обоснована, результаты воспроизводимы, а выводы убедительны. Рекомендую статью к публикации в журнале «Программные системы и вычислительные методы» без доработок.

Программные системы и вычислительные методы*Правильная ссылка на статью:*

Родионов Д.Г., Сергеев Д.А., Конников Е.А., Попова С.Д. Метод анализа аэрофотоснимков с БПЛА на основе SSIM и MSE для оценки надежности технических систем // Программные системы и вычислительные методы. 2025. № 2. DOI: 10.7256/2454-0714.2025.2.73765 EDN: BSPENZ URL: https://nbpublish.com/library_read_article.php?id=73765

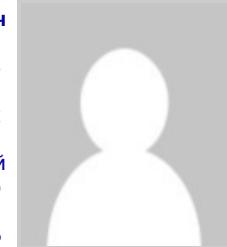
Метод анализа аэрофотоснимков с БПЛА на основе SSIM и MSE для оценки надежности технических систем**Родионов Дмитрий Григорьевич**

ORCID: 0000-0002-1254-0464

доктор экономических наук

Директор; Высшая инженерно-экономическая школа ; Санкт-Петербургский политехнический университет Петра Великого

195251, Россия, г. Санкт-Петербург, Калининский р-н, ул. Политехническая, д. 29 литеру Б



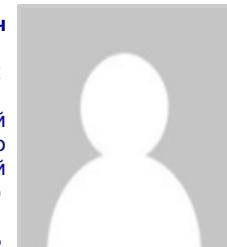
rodion_dm@mail.ru

Сергеев Дмитрий Анатольевич

кандидат экономических наук

доцент; Высшая инженерно-экономическая школа; Санкт-Петербургский политехнический университет Петра Великого
директор; Национальный центр качества и экспертизы; Санкт-Петербургский политехнический университет Петра Великого

195251, Россия, г. Санкт-Петербург, Калининский р-н, ул. Политехническая, д. 29 литеру Б



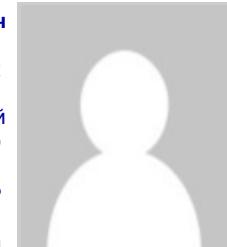
sergeev_da@spbstu.ru

Конников Евгений Александрович

кандидат экономических наук

доцент; Высшая инженерно-экономическая школа; Санкт-Петербургский политехнический университет Петра Великого

195251, Россия, г. Санкт-Петербург, Калининский р-н, ул. Политехническая, д. 29 литеру Б



konnikov_ea@spbstu.ru

Попова София Дмитриевна

ассистент; Высшая инженерно-экономическая школа; Санкт-Петербургский политехнический университет Петра Великого

195251, Россия, г. Санкт-Петербург, Калининский р-н, ул. Политехническая, д. 29 литеру Б



popova_s@spbstu.ru

[Статья из рубрики "Системный анализ, поиск, анализ и фильтрация информации"](#)**DOI:**

10.7256/2454-0714.2025.2.73765

EDN:

BSPENZ

Дата направления статьи в редакцию:

20-03-2025

Аннотация: В статье представлен способ автоматизированного анализа аэроснимков с беспилотных летательных аппаратов (БПЛА), ориентированный на повышение безотказности технических систем и отслеживание изменений в природных и техногенных процессах. Цель работы – создание алгоритма, обеспечивающего безошибочное обнаружение отклонений и прогнозирование угроз сбоев на основе обработки изображений. Методика подразумевает применение индекса структурного соответствия (SSIM) и средней квадратичной погрешности (MSE) для оценки пространственных вариаций между смежными сегментами съемок. Предложенный подход отличается высокой стабильностью к переменам освещенности, небольшими вычислительными издержками и возможностью встраивания в автономные комплексы БПЛА. Работа базируется на компьютерном моделировании и статистическом разборе точности выявления отклонений. Алгоритм был опробован на различных массивах аэроснимков с использованием методов машинного зрения, математической статистики для оценки результативности предлагаемого способа. Итоги – разработка и апробация алгоритма, построении тепловых карт SSIM и MSE, а также оценке точности и достоверности способа. Полученные сведения подтверждают его действенность в автоматизированном мониторинге инфраструктурных объектов и оценивании экологических рисков. Сфера использования созданного способа охватывает автоматизированный надзор за инженерными сооружениями, контроль состояния сельскохозяйственных земель, экологический контроль, разбор последствий стихийных бедствий. Способ может быть интегрирован в интеллектуальные системы управления безотказностью технических объектов. Таким образом, разработанный алгоритм позволяет существенно увеличить безошибочность обнаружения отклонений, минимизировать влияние внешних обстоятельств и автоматизировать процесс обработки аэроснимков. Его применение способствует повышению безотказности технических систем и снижению вероятности сбоев за счет заблаговременного выявления потенциальных угроз. Научная новизна: создание нового способа оценки пространственных вариаций на основе комбинации индекса структурного соответствия (SSIM) и средней квадратичной погрешности (MSE), что обеспечивает высокую безошибочность выявления отклонений. В отличие от традиционных способов анализа изображений, предложенный алгоритм характеризуется стабильностью к изменению условий съемки, его вычислительная эффективность позволяет применять его в режиме реального времени. Кроме того, способ интегрируется в системы автономного мониторинга, расширяя возможности интеллектуального разбора данных с БПЛА. Полученные результаты и предложенные решения могут быть использованы для совершенствования технологий автоматизированного контроля состояния объектов и анализа динамики природных процессов.

Ключевые слова:

БПЛА, Аэрофотоснимки, SSIM, MSE, Пространственно-инвариантное сопоставление, Мониторинг изменений, Надежность технических систем, автоматизированный анализ изображений, прогнозирование рисков отказов, интеллектуальные системы управления

Работы выполнены в рамках реализации проекта "Разработка методологии формирования инструментальной базы анализа и моделирования пространственного социально-экономического развития систем в условиях цифровизации с опорой на внутренние резервы" (FSEG-2023-0008)

Введение

Современные беспилотные летательные аппараты (БПЛА) играют ключевую роль в мониторинге природных и антропогенных процессов, обеспечивая сбор данных с высокой точностью и оперативностью. Их использование значительно расширяет возможности дистанционного зондирования, позволяя контролировать состояние инфраструктурных объектов, оценивать экологические риски и анализировать динамику изменений в окружающей среде. Однако автоматизированная обработка аэрофотоснимков, полученных с БПЛА, представляет собой сложную задачу, требующую разработки эффективных методов оценки пространственно-структурных характеристик изображений.

Одной из ключевых проблем является необходимость выявления и количественной оценки изменений между соседними участками съемки. Существующие подходы, такие как разностные методы и градиентный анализ, демонстрируют ограниченную устойчивость к вариативности условий съемки, а методы машинного обучения требуют значительных вычислительных ресурсов и объемов размеченных данных [1]. В связи с этим возникает потребность в разработке интерпретируемых, ресурсоэффективных алгоритмов, способных надежно анализировать пространственные изменения на изображениях, полученных с БПЛА.

В данной работе предложен метод анализа аэрофотоснимков, основанный на комбинации индекса структурного сходства (SSIM) и средней квадратичной ошибки (MSE). Такой подход позволяет количественно оценивать различия между соседними сегментами изображений, обеспечивая высокую точность обнаружения изменений. В отличие от традиционных статистических методов, SSIM учитывает пространственную структуру изображения, в то время как MSE позволяет дополнительно анализировать интенсивность пиксельных изменений.

Целью исследования является разработка алгоритма оценки изменений между соседними сегментами аэрофотоснимков, пригодного для интеграции в интеллектуальные системы управления надежностью технических систем. Основная задача — анализ пространственной корреляции изображений с целью выявления аномалий, предсказания отказов оборудования и оценки экологических рисков.

В статье изложены теоретические основы структурного анализа изображений, описана методология расчета SSIM и MSE для сегментированных аэрофотоснимков, представлены результаты апробации метода на тестовом изображении, а также проведён критический анализ предложенного подхода. Полученные результаты подтверждают эффективность метода в задачах автоматизированного мониторинга и выявления пространственных изменений, что делает его перспективным инструментом для интеграции в интеллектуальные системы анализа данных, получаемых с БПЛА.

Анализ теоретического базиса

Анализ последовательных аэрофотоснимков и их структурной согласованности является актуальной задачей в области дистанционного зондирования. Традиционные методы обработки изображений включают спектральный анализ, сегментацию и алгоритмы машинного обучения, однако их применение в условиях БПЛА сопряжено с рядом ограничений [2]. В частности, методы, основанные на спектральных характеристиках, чувствительны к изменениям освещения и атмосферных условий, что снижает их надежность при работе в динамично изменяющейся среде. Алгоритмы машинного обучения, включая сверточные нейронные сети (CNN), демонстрируют высокую точность в задачах классификации и сегментации [3], но требуют значительных вычислительных ресурсов и больших объемов размеченных данных, что ограничивает их применение на борту БПЛА.

Наиболее распространённые классические метрики оценки схожести изображений включают статистические и структурные методы. Одной из базовых статистических метрик является средняя квадратичная ошибка (MSE), измеряющая расхождения между интенсивностями пикселей двух изображений. Однако MSE не учитывает пространственные закономерности, и её значения могут быть высокими даже при визуально схожих структурах [4]. Альтернативная метрика — пиковое отношение сигнал/шум (PSNR) — также не отражает текстурные особенности изображений и, как следствие, не всегда применима для анализа изменений в природной и антропогенной среде [5].

Индекс структурного сходства (SSIM), напротив, представляет собой более информативную метрику, учитывающую локальные изменения яркости, контраста и текстуры [6]. SSIM особенно эффективен при анализе пространственной согласованности изображений, так как менее чувствителен к глобальным изменениям освещённости. Тем не менее, большинство существующих исследований сосредоточено на сравнении отдельных изображений, в то время как пространственные взаимосвязи между соседними сегментами часто остаются без должного внимания.

В задачах мониторинга с применением БПЛА важна не только оценка изменений в конкретных изображениях, но и их пространственная корреляция в динамике. Для этих целей традиционно применяются методы пространственного анализа, такие как разностные изображения (image differencing) и градиентные подходы, позволяющие выявлять границы изменений [7]. Однако такие методы уязвимы к шуму и колебаниям освещённости, что ограничивает их применение в условиях реального полевого мониторинга. Более совершенные алгоритмы, например, многомасштабный SSIM (MS-SSIM), позволяют учитывать особенности изображений на разных уровнях разрешения [8], однако требуют дополнительных вычислительных ресурсов и времени на обработку.

Одним из ключевых вызовов при использовании БПЛА для мониторинга инфраструктурных объектов и состояния окружающей среды является необходимость анализа согласованных изменений между соседними сегментами аэрофотоснимков. Это требует интеграции структурных метрик с методами пространственного моделирования данных. В большинстве существующих решений акцент делается либо на анализе отдельных изображений, либо на применении методов глубокого обучения, что требует предварительного обучения на обширных выборках. Однако в условиях автономной работы БПЛА необходим подход, способный оценивать изменения в реальном времени

без этапа обучения [\[9\]](#).

С учетом указанных ограничений возникает потребность в методе, одновременно учитываящем как структурное сходство, так и пространственные изменения между сегментами изображения. В предлагаемой методологии используется совместное применение SSIM и MSE для анализа различий между соседними участками аэрофотоснимков. Это обеспечивает интерпретируемость результатов, устойчивость к внешним факторам съёмки и возможность локализации аномальных зон, что особенно важно в контексте автоматизированного мониторинга с применением БПЛА.

В отличие от существующих решений, предложенный подход позволяет не только выявлять глобальные изменения, но и анализировать их пространственное распределение. Это критично для задач прогнозирования отказов инфраструктурных объектов, определения зон с повышенным экологическим риском и адаптивного управления техническими системами. Таким образом, предлагаемая методология восполняет существующий пробел в области анализа многомерных изображений с БПЛА, интегрируя структурные и пространственные подходы в единую аналитическую систему. Её внедрение в интеллектуальные платформы управления надёжностью позволит повысить точность оценки рисков, оптимизировать стратегии реагирования и обеспечить автоматизированный контроль за изменениями в природных и антропогенных процессах.

Методология

Пусть дано множество откорректированных аэрофотоснимков:

$$I = \{I_1, I_2, \dots, I_n\} \quad (1)$$

полученных в рамках систематического мониторинга территории. Каждый снимок I_i имеет пространственные координаты x_i, y_i и представляется как трёхканальное изображение в цветовом пространстве RGB:

$$I_i \in R^{H \times W \times 3} \quad (2)$$

где H и W — высота и ширина изображения соответственно.

Для оценки пространственной однородности изображений применяются две метрики: SSIM и MSE.

Для двух изображений G_1 и G_2 в оттенках серого, SSIM вычисляется по формуле:

$$G_1, G_2 = 2\mu G_1 \mu G_2 + C_{12} \sigma G_1 G_2 + C_2 \mu G_1 \mu G_2 \quad (3)$$

где:

- . $\mu G_1, \mu G_2$ — средние значения интенсивностей
- . $\sigma G_1, \sigma G_2$ — дисперсии,
- . C_{12} — ковариация между изображениями,
- . C_1, C_2 — постоянные, предотвращающие деление на ноль.

Средняя квадратичная ошибка рассчитывается как:

$$G_1, G_2 = \frac{1}{N} \sum_{i=1}^N (G_1 i - G_2 i)^2 \quad (4)$$

где N — общее количество пикселей, G_{1i} , G_{2i} — значения интенсивностей соответствующих пикселей.

Для каждого изображения I_i определяются восемь соседей:

$$N = \{ITL, IT, ITR, IL, IR, IBL, IB, IBR\} \quad (5)$$

где индексы обозначают направления: Top-Left, Top, Top-Right и т. д.

Метрики SSIM и MSE рассчитываются для каждой пары (I_i, I_j) , где $j \in N_i$.

Для направления d вводятся средние значения SSIM и MSE:

$$SSIM_d = \frac{1}{N_d} \sum_{i,j \in N_d} SSIM_{i,j} \quad (6)$$

$$MSE_d = \frac{1}{N_d} \sum_{i,j \in N_d} MSE_{i,j} \quad (7)$$

где N_d — множество пар изображений в направлении d .

Для повышения качества анализа проводится предварительная обработка:

- преобразование в LAB-пространство;
- адаптивное выравнивание гистограммы методом CLAHE:

$$L' = CLAHE(L) \quad (8)$$

где L — яркостный канал изображения;

- фильтрация шума методом fast non-local means:

$$I' = Denoise(I) \quad (9)$$

где $Denoise(\cdot)$ — функция шумоподавления.

Для отображения результатов строятся тепловые карты по координатам x, y :

$$HSSIM_{x,y} = SSIM_{x,y}, \quad HMSE_{x,y} = MSE_{x,y} \quad (10)$$

где $HSSIM$ и $HMSE$ — значения метрик в каждой точке покрытия.

Для каждого изображения также рассчитываются средние значения метрик по всем соседям:

$$Mavg_{ii} = \frac{1}{N_i} \sum_{j \in N_i} SSIM_{i,j} \quad (11)$$

$$Eavg_{ii} = \frac{1}{N_i} \sum_{j \in N_i} MSE_{i,j} \quad (12)$$

Результаты

Для апробации предложенного алгоритма было сгенерировано тестовое изображение, представленное на рисунке 1.

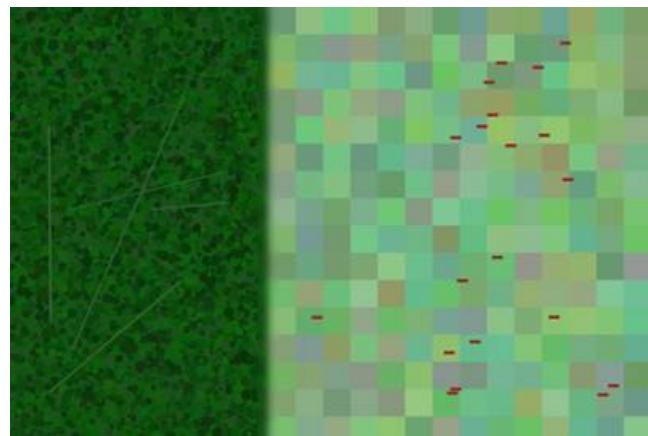


Рисунок 1 – Базовое изображение, используемое для анализа

Изображение имитирует естественные текстуры с варьирующимся уровнем плотности объектов и содержит искусственные линии, моделирующие возможные разрывы или аномалии в структуре поверхности. Оно было разделено на 56 равных сегментов, что позволило провести анализ структурной схожести и рассчитать пространственные метрики SSIM и MSE для каждой пары соседних сегментов.

После разбиения изображения и вычисления значений метрик для всех пар соседних сегментов была построена тепловая карта SSIM (рисунок 2).

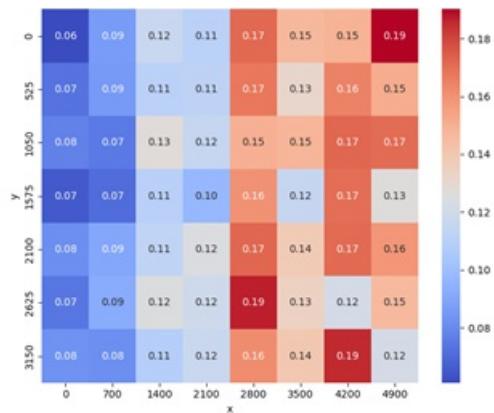


Рисунок 2 – Тепловая карта SSIM, отображающая уровни структурного сходства между соседними сегментами

В среднем значения SSIM находились в диапазоне от 0,06 до 0,19, что указывает на относительно низкую степень структурного сходства между соседними участками изображения.

На основе тепловой карты можно выделить следующие закономерности:

- В центральной и правой частях карты наблюдаются области с повышенными значениями SSIM (0,16–0,19), что свидетельствует о наличии структурно однородных сегментов;
- В левой части карты значения SSIM значительно ниже (0,06–0,09), что указывает на текстурные изменения или наличие аномалий;
- Локальные резкие изменения SSIM в пограничных областях могут быть обусловлены переходами между различными текстурами или присутствием искусственных объектов.

Для оценки интенсивности расхождений между сегментами дополнительно была построена тепловая карта MSE (рисунок 3).

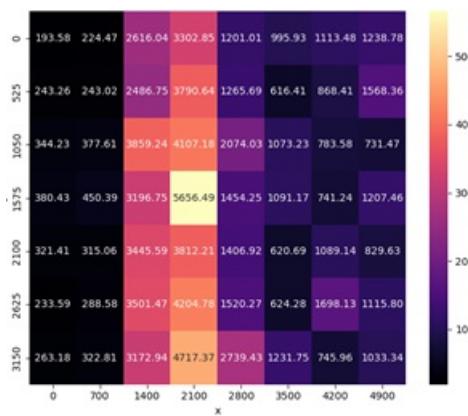


Рисунок 3 – Тепловая карта MSE, отображающая степень расхождения между сегментами

Значения MSE варьировались от 233,59 до 5656,49, демонстрируя высокую степень неоднородности между рядом соседних сегментов.

Основные наблюдения:

- Максимальное значение MSE ($\sim 5656,49$) зафиксировано в центральной части изображения, что может указывать на наличие значительных текстурных различий или искусственного разрыва;
- Локальные пики MSE наблюдаются также в других сегментах, структура которых подверглась существенным изменениям;
- В левой части изображения, где значения SSIM были минимальными, наблюдаются умеренные значения MSE (в пределах 200–4000), что может указывать на слабовыраженные, но существующие переходы между текстурами.

Анализ тепловых карт SSIM и MSE позволяет выявить устойчивые закономерности в распределении изменений:

- Высокие значения SSIM при низких значениях MSE соответствуют структурно однородным регионам;
- Низкие значения SSIM при высоких значениях MSE указывают на области с выраженными переходами и изменениями текстур;
- Локальные аномалии, сопровождающиеся резким снижением SSIM и увеличением MSE, вероятно, отражают наличие переходных зон или участков с резкой сменой плотности объектов, что может быть связано с моделируемыми разрывами в текстуре.

Дискуссия

Одним из ключевых преимуществ предложенного метода является его интерпретируемость и низкие вычислительные затраты. В отличие от методов машинного обучения, таких как сверточные нейронные сети [10], представленный подход не требует предварительного обучения на больших выборках данных. Это делает его особенно привлекательным для использования в автономных системах БПЛА, где вычислительные ресурсы ограничены. Вместе с тем следует отметить ограниченную адаптивность метода: в отличие от нейросетевых моделей, способных обучаться на специфических ландшафтах и условиях съёмки, разработанный алгоритм опирается на фиксированные метрики, что может снижать его точность при высокой вариативности освещённости, угла съёмки или сезонных изменений.

Дополнительным преимуществом является устойчивость к глобальным изменениям освещённости и яркости. Как показано в работах Wang et al. (2004) и Zhang et al.

(2017), метрика SSIM значительно лучше справляется с анализом текстурных изменений по сравнению с MSE, поскольку учитывает локальные различия в контрасте и структуре изображения [7, 8]. Это позволяет корректно выявлять неоднородности в инфраструктуре даже при существенных изменениях условий съёмки. Однако следует учитывать, что метод менее чувствителен к глобальным изменениям интенсивности, что может привести к пропуску значимых аномалий, таких как появление или исчезновение объектов, если они сопровождаются плавными текстурными переходами [10].

С точки зрения пространственного анализа, предложенный метод восполняет пробел в существующих подходах к оценке аэрофотоснимков. В традиционных методах мониторинга с БПЛА, таких как разностные изображения [7] или градиентный анализ, внимание сосредоточено на локальном сравнении изображений без учёта их пространственной структуры. Разработанный подход, напротив, позволяет системно анализировать взаимосвязи между сегментами изображения, что критически важно для отслеживания динамики изменений ландшафта [11]. Тем не менее, он также имеет ограничения: фиксированный шаг сегментации может приводить к потере информации на границах, особенно если объекты имеют сложную форму или пространственную дисперсию.

Анализируя точность и чувствительность метода, можно отметить, что комбинированное использование SSIM и MSE обеспечивает более полную картину различий между сегментами, чем применение каждой метрики в отдельности. В частности, SSIM более точно отражает структурные изменения, тогда как MSE позволяет количественно оценить расхождения на уровне пикселей. Однако в условиях мелкомасштабных изменений MSE может демонстрировать завышенные значения даже при сохранении общей структуры изображения [4]. В то же время SSIM, будучи более устойчивым к шуму, может не фиксировать незначительные, но критичные с точки зрения инфраструктурной диагностики отклонения. Это указывает на необходимость дальнейшей адаптации метода, например, за счёт введения порогов, зависящих от контекста задачи и особенностей анализируемых объектов.

Особенностью метода является его универсальность и широкая применимость. В отличие от специализированных алгоритмов, разрабатываемых под узкие задачи (например, в сельском или лесном хозяйстве), предложенный подход может быть использован для широкого круга приложений: от мониторинга состояния инженерной инфраструктуры до оценки экологических рисков [12]. Это особенно актуально в контексте создания интеллектуальных систем управления надёжностью технических объектов, в которых требуется универсальный аналитический модуль, способный интегрироваться с данными, получаемыми с различных сенсоров и платформ, включая БПЛА [13].

В то же время важно учитывать, что эффективность метода может снижаться при анализе изображений со сложной пространственной структурой изменений, таких как точечные разрушения, эрозионные процессы или постепенные трансформации поверхности. Эти сценарии требуют дополнительной адаптации алгоритма и, возможно, гибридизации с другими подходами (например, методами машинного обучения или фрактального анализа) для повышения чувствительности к мелким, но значимым изменениям.

Заключение

В данной работе предложена методология анализа аэрофотоснимков с БПЛА, основанная на комбинированном использовании метрик SSIM и MSE. Метод ориентирован

на автоматизированное выявление пространственно-структурных изменений и предназначен для повышения точности мониторинга природных и антропогенных процессов, контроля состояния инфраструктурных объектов и предсказания отказов технических систем.

Анализ существующих подходов показал, что традиционные статистические методы обладают ограниченной чувствительностью к текстурным особенностям, а нейросетевые решения требуют значительных ресурсов и обучающих выборок, что затрудняет их применение в автономных системах. В отличие от них, предложенный подход сочетает структурный анализ с пространственной интерпретацией, что позволяет эффективно выявлять локальные изменения при минимальных вычислительных затратах.

Результаты экспериментального тестирования на изображении, разделённом на 56 сегментов, подтвердили работоспособность метода. Построенные тепловые карты SSIM и MSE позволили точно локализовать участки с выраженными изменениями. Зоны с низкими значениями SSIM и высокими значениями MSE соответствовали аномалиям и переходам между различными текстурами, что подтверждает применимость метода для пространственного анализа.

Проведённый критический разбор выявил основные преимущества метода: интерпретируемость, устойчивость к изменению условий освещения, низкие вычислительные требования и способность учитывать пространственную согласованность. В то же время отмечены ограничения, связанные с фиксированными параметрами сегментации и снижением точности в условиях сложной морфологии изменений.

Предложенный подход может быть использован в интеллектуальных системах управления надёжностью технических объектов, интегрируемых с платформами БПЛА. Это открывает перспективы для предиктивной диагностики, экологического мониторинга и автоматизированного контроля за изменениями в окружающей среде.

В дальнейшем планируется развитие метода за счёт адаптивной настройки порогов SSIM и MSE, а также интеграция с алгоритмами машинного обучения для повышения точности и степени автоматизации анализа изображений.

Библиография

1. Бучаев М.А., Старченкова О.Д., Конников Е.А. Анализ и кластеризация недостоверной новостной информации как инструмент обеспечения информационной безопасности региона // Мягкие измерения и вычисления. 2024. № 84 (11-2). С. 37-51. DOI 10.36871/2618-9976.2024.11-2.004. EDN: PFWBDV.
2. Dai K., Ma C., Wang Z., Long Y., Li X., Feng S., Ye Y. Exploiting spatial-temporal dynamics for satellite image sequence prediction // IEEE Geoscience and Remote Sensing Letters. 2023. Т. 20. С. 1-5.
3. He X., Chen Y. Transferring CNN ensemble for hyperspectral image classification // IEEE Geoscience and Remote Sensing Letters. 2020. Т. 18. № 5. С. 876-880.
4. Veretelnikova E.L., Elantseva I.L. Selection of factor for root mean square minimum error criterion // 2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE). 2016. Т. 2. С. 221-223. IEEE. DOI 10.1109/APEIE.2016.7806454. EDN: WZWUYB.
5. Моттаева А.Б., Смирнова И.А., Конников Е.А., Шарафанова Е.Е. Нечетко-множественный подход к оценке трансрегиональной промышленной когерентности // Мягкие измерения и вычисления. 2024. № 76 (3). С. 35-44. DOI 10.36871/2618-

9976.2024.03.003. EDN: EGIQUM.

6. Raju K.N., Reddy K.S.P. Comparative study of Structural Similarity Index (SSIM) by using different edge detection approaches on live video frames for different color models // 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT). 2017. C. 932-937. IEEE.
7. Coppin P., Jonckheere I., Nackaerts K., Muys B., Lambin E. Digital change detection methods in ecosystem monitoring // International Journal of Remote Sensing. 2004. Т. 25. № 9. С. 1565-1596.
8. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. Image quality assessment: From error visibility to structural similarity // IEEE Transactions on Image Processing. 2004. Т. 13. № 4. С. 600-612.
9. Родионов Д. Г., Конников Е. А., Пашинина П. А., Шаныгин С. И. Тематическое моделирование информационной среды медиакомпаний: инструментальный комплекс LDA-TF-IDF // Мягкие измерения и вычисления. 2024. Т. 76, № 3. С. 72-84. DOI 10.36871/2618-9976.2024.03.006. EDN: COCJYG.
10. Dai K., Ma C., Wang Z., Long Y., Li X., Feng S., Ye Y. Exploiting spatial-temporal dynamics for satellite image sequence prediction // IEEE Geoscience and Remote Sensing Letters. 2023. Т. 20. С. 1-5.
11. Волков О. А., Дьякова Г. Н. Разработка математического аппарата интервальной оценки вероятностных показателей надежности технических систем с последовательно-параллельной структурой методом аппроксимации фидуциальных распределений в классе бета-распределений // Моделирование и ситуационное управление качеством сложных систем: Сборник докладов Научной сессии ГУАП, Санкт-Петербург, 08-12 апреля 2019 года. Санкт-Петербург: Санкт-Петербургский государственный университет аэрокосмического приборостроения, 2019. С. 27-34. EDN: IUCRKY.
12. Моттаева А.Б., Смирнова И.А., Конников Е.А., Шаныгин С.И. Построение алгоритмики анализа экономической результативности трансрегионального промышленного кластера в контексте цикла приращения уровня устойчивого развития с применением инструментария событийного моделирования // Мягкие измерения и вычисления. 2024. № 77(4). С. 28-38. DOI 10.36871/2618-9976.2024.04.004. EDN: EQCIDA.
13. Станин А.А., Митязов В.А., Сорокин В.И., Конников Е.А. Моделирование уровня несостоительности регионов на основе алгоритма оценки риска-предрасположенности // Мягкие измерения и вычисления. 2024. № 84 (11-1). С. 35-49. DOI 10.36871/2618-9976.2024.11.003. EDN: ZUAGEG.

Результаты процедуры рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Любая тематика по беспилотным летательным аппаратам (БПЛА), как в военном, так и в гражданском применении,

в настоящее время очень актуальна. Данная работа посвящена ключевой проблеме анализа изображений,

получаемых с аэрофотоснимков с высоким разрешением, позволяя контролировать состояние окружающей среды. Эта

сложная задача требует эффективных методов и больших вычислительных ресурсов, в

том числе искусственного

интеллекта и машинного обучения.

Цель работы в начале сформулирована не совсем корректно как "разработка алгоритма оценки изменений между

соседними сегментами аэрофотоснимков", а в заключении как "автоматизированное выявление пространственно-

структурных изменений". Это скорее средство достижения цели, а главная цель видится как "повышение

надежности мониторинга, качества прогнозирования отказов, выявления аномалий" или что то в этом роде.

Научная новизна состоит в предложенном методе сравнения и анализа аэрофотоснимков на основе известного

индекса структурного сходства Structural Similarity Index Measure (SSIM) и стандартных статистических

показателей среднеквадратического отклонения Mean Squared Error (MSE) или Mean Squared Deviation (MSD).

В статье приведен теоретический базис решаемой проблемы, сравнительный анализ существующих методов. Описана

предложенная методология анализа информации с БПЛА. Особое внимание уделено задачам прогнозирования

отказов, обнаружению зон с повышенным риском. Несомненным преимуществом работы является ее апробация на

тестовом сгенерированном изображении.

Статья изложена грамотным техническим языком, понятным читательской аудитории в данной предметной области.

Структура оформления материала, цель, задачи, литературный обзор, дискуссия, заключение, библиография,

оформлены в соответствии с требованиями журнала "Программные системы и вычислительные методы". Все

иллюстрации приемлемого качества по критерию читабельности надписей. Библиография содержит 13 ссылок, как

на российскую, так и на зарубежную научную литературу.

Критических замечаний нет, другие рекомендации:

1. Слишком длинное название статьи для понимания сути работы широкой аудиторией читателей, не заглядывая в

основной текст. Рекомендуется сформулировать покороче и более четко.

2. Аббревиатуры MSE, SSIM, БПЛА определены многократно.

3. В первом абзаце раздела "Методология" вместо математического обозначения пробел.

4. Формулы не пронумерованы и их качество "как картинки" оставляет желать лучшего.

5. Переменные x_{iy} , $G1_i$, $G2_i$, I_i , I_j , N_{ij} и т. д. без индексов.

6. Встречаются лишние пробелы перед формулами и рис. 2.

7. Разделителем целой и дробной части в русском языке должна быть запятая, например $MSE=5656.49$

неправильно.

8. В библиографии ссылки № 9 и № 11 содержат неполную информацию об издании.

Общий вывод: статья может быть принята к публикации в журнале "Программные системы и вычислительные методы"

с незначительными доработками.

Результаты процедуры повторного рецензирования статьи

В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена разработке метода автоматизированного анализа аэрофотоснимков, полученных с беспилотных летательных аппаратов (БПЛА), на основе комбинации метрик структурного сходства (SSIM) и средней квадратичной ошибки (MSE). Цель исследования — повышение точности выявления пространственных изменений для мониторинга технических систем, инфраструктурных объектов и экологических рисков. Автор предлагает инновационный подход, включающий предварительную обработку изображений (переход в LAB-пространство, шумоподавление), сегментацию и расчет SSIM и MSE для оценки различий между соседними участками. Метод позволяет количественно оценивать структурные и пиксельные изменения, обеспечивая высокую интерпретируемость результатов при низких вычислительных затратах. Экспериментальная часть демонстрирует применение метода на тестовом изображении, разделенном на 56 сегментов. Построенные тепловые карты SSIM и MSE подтвердили возможность точной локализации аномалий и зон с выраженным текстурными изменениями.

Ключевые преимущества метода:

- Устойчивость к изменениям освещенности и шумам;
- Отсутствие необходимости в предварительном обучении, в отличие от нейросетевых подходов;
- Возможность интеграции в автономные системы мониторинга на базе беспилотных летательных аппаратов.

Результаты исследования имеют практическую значимость для задач предиктивного обслуживания инфраструктуры, экологического контроля и управления надежностью технических систем. Перспективы дальнейшего развития включают адаптацию пороговых

значений метрик и гибридизацию с методами машинного обучения.

Исследование отвечает на вызовы современного мониторинга с использованием БПЛА, где традиционные методы (разностные изображения, машинное обучение) обладают ограничениями в точности и ресурсоемкости. Предложенный метод актуален для задач предиктивной диагностики инфраструктуры и экологического контроля.

Новизна работы заключается в комбинации SSIM и MSE для пространственного анализа сегментов, что обеспечивает интерпретируемость и низкие вычислительные затраты. Автор также предлагает адаптацию метода для реального времени, что отличает его от нейросетевых решений, требующих больших данных.

Статья хорошо структурирована: введение четко формулирует проблему, методология детализирована, результаты наглядно представлены, а обсуждение критически анализирует преимущества и ограничения. Стиль изложения научный, но доступный, с корректным использованием терминологии.

Автор делает обоснованные выводы о эффективности метода, подтвержденные экспериментальными данными. Отмечены перспективы дальнейшего развития, включая интеграцию с машинным обучением и адаптацию пороговых значений.

Статья будет полезна исследователям в области дистанционного зондирования, специалистам по обработке изображений и разработчикам систем мониторинга на основе БПЛА. Практическая значимость метода делает его интересным для промышленных и экологических приложений.

Статья представляет собой завершенное научное исследование с высокой степенью новизны и практической значимости. Методология проработана, результаты достоверны, а выводы обоснованы.

Рекомендация: принять к публикации без доработок.

Англоязычные метаданные

Dynamic RACH-Slot Allocation for Collision Minimization in NB-IoT Networks Based on Reinforcement Learning Algorithms

Shabrova Anna Sergeevna

Student; Department of Information Security; Bauman Moscow State Technical University

5, 2nd Baumanskaya str., building 4, Moscow, 105005, Russia

✉ shabrova.anna.2410@list.ru

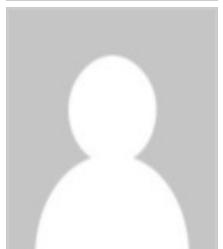


Knyazev Maxim Andreevich

Postgraduate student; Department of Information Security; MREA – Russian Technological University, Institute of Artificial Intelligence

78 Vernadsky Ave., Moscow, 119454, Russia

✉ maxiknyaz@mail.ru

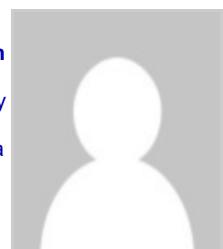


Kolesnikov Alexander Vladimirovich

Ph.D; Department of Information Security, Bauman Moscow State Technical University

5, 2nd Baumanskaya str., building 4, Basmanny district, Moscow, 105005, Russia

✉ avkolesnikov@bmstu.ru



Abstract. The subject of this research is the adaptive management of access to Random Access Channels (RACH) in Narrowband Internet of Things (NB-IoT) networks, which frequently face congestion due to high device density and limited channel capacity. The study focuses on the practical application of Reinforcement Learning algorithms, specifically Q-learning and Deep Q-Network (DQN), to address this issue. The authors thoroughly examine the problem of RACH overload and the resulting collisions that cause delays in data transmission and increased energy consumption in connected devices. The article analyzes the limitations and inefficiency of traditional static slot allocation methods and justifies the necessity of implementing a dynamic, learning-based approach capable of adapting to constantly changing network conditions. The research aims to significantly minimize collision rates, improve connection success rates, and reduce the overall energy consumption of NB-IoT devices. The research methodology involved the use of advanced machine learning methods, including Q-learning and DQN, together with simulation modeling conducted in the NS-3 environment, integrating a dedicated RL-agent for dynamic and intelligent RACH slot allocation. The main conclusions of the study highlight the demonstrated effectiveness of the adaptive RL-based approach for optimizing access to communication slots in NB-IoT networks. The scientific novelty lies in the development and integration of a specialized RL-agent capable of dynamically managing slot distribution based on real-time network conditions. As a result of implementing the proposed approach, the number of collisions was reduced by 74%, the number of successful connections increased by 16%, and the energy efficiency of the devices improved by 15% in comparison with traditional static methods. These results clearly demonstrate the practical applicability, and scalability of adaptive RL-based management techniques for enhancing both the performance and reliability of real-world NB-IoT networks.

Keywords: Internet of Things, IoT, RACH, reinforcement learning, NS-3, collisions, DQN, Q-learning, Reinforcement Learning, NB-IoT

References (transliterated)

1. Liu Y., Deng Y., Jiang N. [i dr.]. Analysis of Random Access in NB-IoT Networks With Three Coverage Enhancement Groups: A Stochastic Geometry Approach // IEEE Transactions on Wireless Communications. – 2021. – T. 20, № 1. – S. 549-563. DOI: 10.1109/twc.2020.3026331. EDN: WLVLVU.
2. Jia G., Zhu Y., Li Y., Zhu Z. Analysis of the Effect of the Reliability of the NB-IoT Network on the Intelligent System // Special Section on Innovation and Application of Internet of Things and Emerging Technologies in Smart Sensing. – 2019. – № 7. – S. 112809-112820.
3. Sahithya R., Pouria Z., Mohieddine E. S., Majid N. Evaluation, Modeling and Optimization of Coverage Enhancement Methods of NB-IoT // Electrical Engineering Department. – 2019. – № 1. – S. 1-17.
4. Chougrani H., Kisseleff S., Martins W. A., Chatzinotas S. NB-IoT Random Access for Nonterrestrial Networks: Preamble Detection and Uplink Synchronization // IEEE Internet of Things Journal. – 2022. – T. 9, № 16. – S. 14913-14927. DOI: 10.1109/jiot.2021.3123376. EDN: RVWLFC.
5. Agiwal M., Kumar M. M., Jin H. Power Efficient Random Access for Massive NB-IoT Connectivity // Sensors. – 2019. – № 19. – S. 1-24.
6. Jiang N., Deng Y., Nallanathan A. Deep Reinforcement Learning for Real-Time Optimization in NB-IoT Networks // School of Electronic Engineering and Computer Science. – 2018. – № 1. – S. 1-31.
7. Alcaraz J., Losilla F., Gonzalez-Castaño F.-J. Transmission Control in NB-IoT With Model-Based Reinforcement Learning // IEEE Access. – 2023. – № 11. – S. 57991-58005. DOI: 10.1109/access.2023.3284990. EDN: KTFAKS.
8. Anbazhagan S., Mugelan R.K. Next-Gen Resource Optimization in NB-IoT Networks: Harnessing Soft Actor-Critic Reinforcement Learning // Computer Networks. – 2024. – № 252. – S. 110670-110684. DOI: 10.1016/j.comnet.2024.110670. EDN: XHDOCK.
9. Shorin O.A., Aslanyan V.A. Podkhody k integratsii tekhnologii NB-IoT s set'yu 5G // Ekonomika i kachestvo sistem svyazi. – 2024. – № 3. – S. 56-62. EDN: CXRPPW.
10. Namiot D.E., Il'yushin E.A. Arkhitektura LLM agentov // International Journal of Open Information Technologies. – 2025. – T. 13, № 1. – S. 2307-8162.
11. Isaeva O.S. Postroenie tsifrovogo profilya ustroistv Interneta veshchei // Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii. – 2023. – T. 30, № 2. – S. 36-44. DOI: 10.25729/ESI.2023.30.2.004. EDN: EFPGJP.

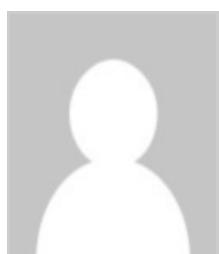
WebAssembly performance in the Node.js environment

Karpovich Vladimir Dmitrievich

Student; Faculty of Software Engineering and Computer Technology, ITMO National Research University

109369, Russia, Moscow, Marino district, Pererva str., 26 room 1, sq. 94

✉ mywinter4@yandex.ru



Gosudarev Ilya Borisovich

PhD in Pedagogy

Associate Professor; Megafacultet of Computer Technology and Management; ITMO National Research University

Kronverkskiy Prospekt, 49, Saint Petersburg, 190000, Russia

 goss@itmo.ru

Abstract. Modern runtime environments such as browsers, Node.js, and others provide developers with tools that go beyond traditional JavaScript. This study focuses on a modern approach to building web applications where components written in different programming languages can be executed and shared using WebAssembly. The subject of the research is the testing and analysis of performance benchmarks comparing JavaScript and WebAssembly modules in the Node.js runtime. The focus is on evaluating performance in computational tasks, memory interaction, data processing, and cross-language communication. The author thoroughly explores topics such as WebAssembly integration in applications, its advantages for resource-intensive tasks like image processing, and the objectivity, representativeness, and reproducibility of the tests. The work follows an applied, experimental approach. It includes performance comparisons between pure JavaScript and WebAssembly modules. Metrics like response time and system resource consumption were used to assess efficiency. The scientific novelty of this work lies in the development and theoretical grounding of testing approaches for web applications using WebAssembly. Unlike most existing studies focused on WebAssembly's performance and security in browser environments, this work emphasizes automated testing of WebAssembly modules outside the browser — a relatively unexplored area until now. A methodological approach is proposed for testing WebAssembly modules in Node.js, including principles for test structuring, integration with JavaScript components, and execution analysis. This approach takes into account the specifics of the server environment, where WebAssembly is increasingly used — particularly for high-load computational modules, cross-language logic, and secure isolated execution. The novelty also lies in defining criteria to evaluate whether certain application components are suitable for migration to WebAssembly in terms of testability, providing developers with a tool for making architectural decisions. The proposed ideas are backed by experimental results, including test case implementations for WebAssembly and JavaScript interaction scenarios.

Keywords: compilation and interpretation, Algorithmic optimization, experimental analysis, server-side processing, technology integration, images processing, heavy processing tasks, performance optimization, JavaScript, WebAssembly

References (transliterated)

1. Stefan Fredriksson. WebAssembly vs. its predecessors: A comparison of technologies // diva-portal.org. Jul. 25, 2020. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1460603&dswid=-4940> (data obrashcheniya 07.04.2025).
2. Marcus Alevärn. Server-side image processing in native code compared to client-side image processing in WebAssembly // diva-portal.org. Jul 27, 2021. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1587964&dswid=-8296> (data obrashcheniya 07.04.2025).
3. Repozitorii s iskhodnym kodom proekta. URL: <https://github.com/Winter4/image-processor-backend> (data obrashcheniya 07.04.2025).
4. Kirill Smelyakov. Efficiency of image convolution // ieee.org. Sep 8, 2019. URL: <https://doi.org/10.1109/CAOL46282.2019.9019450> (data obrashcheniya 07.04.2025).

5. Aravind Samy Shanmugam. Docker Container Reactive Scalability and Prediction of CPU Utilization Based on Proactive Modelling // ncirl.ie. Sep 13, 2017. URL: <https://norma.ncirl.ie/2884/1/aravindsamyshanmugam.pdf> (data obrashcheniya 07.04.2025).
6. Xinchen Xu, Aidong Xu. Research on Security Issues of Docker and Container Monitoring System in Edge Computing System // iopscience.iop.org. Sep 27, 2020. URL: <https://doi.org/10.1088/1742-6596/1673/1/012067> (data obrashcheniya 07.04.2025).
7. Matti Holopainen. Monitoring Container Environment with Prometheus and Grafana // theseus.fi. May 3, 2021. URL: https://www.theseus.fi/bitstream/handle/10024/497467/Holopainen_Matti.pdf (data obrashcheniya 07.04.2025).
8. Bahrami Sepide. Automated Performance Testing in Ephemeral Environments // thesis.unipd.it. Jul 9, 2024. URL: <https://thesis.unipd.it/handle/20.500.12608/66511> (data obrashcheniya 07.04.2025).
9. David Reis. Developing Docker and Docker-Compose Specifications: A Developers' Survey // ieee.org. Dec 22, 2021. URL: <https://doi.org/10.1109/ACCESS.2021.3137671> (data obrashcheniya 07.04.2025).
10. Adeel Ehsan. RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions // mdpi.com. Apr 26, 2022. URL: <https://doi.org/10.3390/app12094369> (data obrashcheniya 07.04.2025).
11. Veb-resurs, sbornik razlichnykh testovykh failov. URL: <https://examplefile.com> (data obrashcheniya 07.04.2025).
12. Mark van dek Wilk. Convolutional Gaussian Processes // proceedings.neurips.cc. 2017. URL: <https://proceedings.neurips.cc/paper/2017/hash/1c54985e4f95b7819ca0357c0cb9a09f-Abstract.html> (data obrashcheniya 07.04.2025).
13. Linus Hellberg. Performance evaluation of Web Workers API and OpenMP // diva-portal.org. Jul 6, 2022. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1681349&dswid=-9566> (data obrashcheniya 07.04.2025).
14. Philip Lassen. WebAssembly Backends for Futhark // futhark-lang.org. June 29, 2021. URL: <https://futhark-lang.org/student-projects/philip-msc-thesis.pdf> (data obrashcheniya 07.04.2025).

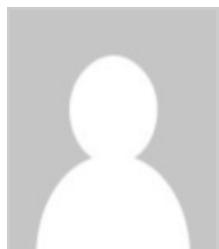
Analysis of DOM update methods in modern web frameworks: Virtual DOM and Incremental DOM

Bondarenko Olesya Sergeevna

Master's degree; Faculty of Software Engineering and Computer Technology, Federal State Autonomous Educational Institution of Higher Education 'ITMO National Research University'.

195299, Russia, St. Petersburg, Kalininsky district, Prospekt Prosvetleniya, 99

 rancerenly@gmail.com



Abstract. The article presents an analysis of modern methods for updating the Document Object Model (DOM) structure in popular client-side web frameworks such as Angular, React and Vue. The main focus is on comparing the concepts of Virtual DOM and Incremental DOM, which underlie the architectural solutions of the respective frameworks. The Virtual DOM used in React and Vue operates on a virtual tree, compares its versions in order to identify differences and minimize changes in the real DOM. This approach provides a relatively simple

implementation of the reactive interface, but comes with additional costs for computing and resource usage. In contrast, Angular uses an Incremental DOM, which does not create intermediate structures: changes are applied directly through the Change Detection mechanism. This approach allows to achieve high performance through point updates of DOM elements without the need for a virtual representation. The study uses a comparative analysis of architectural approaches to updating the DOM, based on the study of official documentation, practical experiments with code and visualization of rendering processes in Angular and React. The methodology includes a theoretical justification, a step-by-step analysis of the update mechanisms and an assessment of their impact on performance. The scientific novelty of the article lies in the systematic comparison of architectural approaches to updating the DOM in leading frameworks, with an emphasis on the implementation of the signal model in Angular version 17+. The impact of using signals on the abandonment of the Zone library is analyzed in detail.js and the formation of a more predictable, deterministic rendering model, as well as lower-level performance management capabilities. The article contains not only a theoretical description, but also practical examples that reveal the behavior of updates in real-world scenarios. The nuances of template compilation, the operation of the effect() and computed() functions are also considered. The comparison of Virtual DOM and Incremental DOM makes it possible to identify key differences, evaluate the applicability of approaches depending on the tasks and complexity of the project, and also suggest ways to optimize frontend architect

Keywords: diffing, rendering, signals, Vue, React, Angular, Incremental DOM, Virtual DOM, DOM, template compilation

References (transliterated)

1. Beteev K.Yu., Muratova G.V. Kontseptsiya virtual dom v biblioteke react.js // Inzhenernyi vestnik Dona. – 2022. – № 3. – S. 170-180. EDN: LHOOOS.
2. Incremental DOM [Elektronnyi resurs]. URL: <https://github.com/google/incremental-dom> (data obrashcheniya: 16.04.2025).
3. Introducing Incremental DOM [Elektronnyi resurs]. URL: <https://medium.com/google-developers/introducing-incremental-dom-e98f79ce2c5f> (data obrashcheniya: 12.04.2025).
4. Understanding Angular Ivy: Incremental DOM and Virtual DOM [Elektronnyi resurs]. URL: <https://blog.nrwl.io/understanding-angular-ivy-incremental-dom-and-virtual-dom-243be844bf36> (data obrashcheniya: 16.04.2025).
5. Razbiraemsya v Angular Ivy: Incremental DOM i Virtual DOM [Elektronnyi resurs]. URL: <https://habr.com/ru/articles/448048/> (data obrashcheniya: 03.03.2025).
6. Virtual DOM vs Incremental DOM in Angular [Elektronnyi resurs]. URL: <https://www.angularminds.com/blog/virtual-dom-vs-incremental-dom-in-angular> (data obrashcheniya: 17.02.2025).
7. Angular Dokumentatsiya [Elektronnyi resurs]. URL: <https://angular.dev/api/core/ChangeDetectionStrategy> (data obrashcheniya: 16.04.2025).
8. Dovzhenko M.I., Gotskaya I.B. Analiz sposobov realizatsii algoritma otslezhivaniya izmenenii v odnostranichnykh veb-prilozheniyakh // Al'manakh nauchnykh rabot molodykh uchenykh XLVII nauchnoi i uchebno-metodicheskoi konferentsii Universiteta ITMO. Tom 7. – 2018. – S. 123-126. EDN: YXNFSH.
9. Iskhodnyi kod primerov [Elektronnyi resurs]. URL:

<https://stackblitz.com/edit/6meb5pyu?file=src%2Fmain.ts> (data obrashcheniya: 16.04.2025).

10. Zone.js [Elektronnyi resurs]. URL:

<https://github.com/angular/angular/tree/main/packages/zone.js> (data obrashcheniya: 16.04.2025).

Analysis of the impact of prompt obfuscation on the effectiveness of language models in detecting prompt injections

Krohin Aleksei Sergeevich

Student; Moscow Institute of Electronics and Mathematics; National Research University 'Higher School of Economics'

13 Volgogradsky ave., Moscow, 109316, Russia

✉ askrokhin@edu.hse.ru



Gusev Maksim Mihailovich

Student; Moscow Institute of Electronics and Mathematics; National Research University 'Higher School of Economics'

115 k. 3 Volgogradsky ave., Moscow, 109117, Russia

✉ gusevmaxim04@mail.ru



Abstract. The article addresses the issue of prompt obfuscation as a means of circumventing protective mechanisms in large language models (LLMs) designed to detect prompt injections. Prompt injections represent a method of attack in which malicious actors manipulate input data to alter the model's behavior and cause it to perform undesirable or harmful actions. Obfuscation involves various methods of changing the structure and content of text, such as replacing words with synonyms, scrambling letters in words, inserting random characters, and others. The purpose of obfuscation is to complicate the analysis and classification of text in order to bypass filters and protective mechanisms built into language models. The study conducts an analysis of the effectiveness of various obfuscation methods in bypassing models trained for text classification tasks. Particular attention is paid to assessing the potential implications of obfuscation for security and data protection. The research utilizes different text obfuscation methods applied to prompts from the AdvBench dataset. The effectiveness of the methods is evaluated using three classifier models trained to detect prompt injections. The scientific novelty of the research lies in analyzing the impact of prompt obfuscation on the effectiveness of language models in detecting prompt injections. During the study, it was found that the application of complex obfuscation methods increases the proportion of requests classified as injections, highlighting the need for a thorough approach to testing the security of large language models. The conclusions of the research indicate the importance of balancing the complexity of the obfuscation method with its effectiveness in the context of attacks on models. Excessively complex obfuscation methods may increase the likelihood of injection detection, which requires further investigation to optimize approaches to ensuring the security of language models. The results underline the need for the continuous improvement of protective mechanisms and the development of new methods for detecting and preventing attacks on large language models.

Keywords: fuzzing, AI security, transformers, encoder, adversarial attacks, AI, jailbreak,

obfuscation, prompt injection, LLM

References (transliterated)

1. Liu Y. et al. Formalizing and benchmarking prompt injection attacks and defenses // 33rd USENIX Security Symposium (USENIX Security 24). – 2024. – S. 1831-1847.
2. Greshake K. et al. Not what you've signed up for: Compromising real-world lIIm-integrated applications with indirect prompt injection // Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security. – 2023. – S. 79-90.
3. Shi J. et al. Optimization-based prompt injection attack to lIIm-as-a-judge // Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. – 2024. – S. 660-674.
4. Sang X., Gu M., Chi H. Evaluating prompt injection safety in large language models using the promptbench dataset. – 2024.
5. Xu Z. et al. LLM Jailbreak Attack versus Defense Techniques--A Comprehensive Study // arXiv e-prints. – 2024. – S. arXiv: 2402.13457.
6. Hu K. et al. Efficient lIIm jailbreak via adaptive dense-to-sparse constrained optimization // Advances in Neural Information Processing Systems. – 2024. – T. 37. – S. 23224-23245.
7. Wei A., Haghtalab N., Steinhardt J. Jailbroken: How does lIIm safety training fail? // Advances in Neural Information Processing Systems. – 2023. – T. 36. – S. 80079-80110.
8. Li J. et al. Getting more juice out of the sft data: Reward learning from human demonstration improves sft for lIIm alignment // Advances in Neural Information Processing Systems. – 2024. – T. 37. – S. 124292-124318.
9. Kwon H., Pak W. Text-based prompt injection attack using mathematical functions in modern large language models // Electronics. – 2024. – T. 13. – №. 24. – S. 5008.
10. Steindl S. et al. Linguistic obfuscation attacks and large language model uncertainty // Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024). – 2024. – S. 35-40.
11. Kim M. et al. Protection of LLM Environment Using Prompt Security // 2024 15th International Conference on Information and Communication Technology Convergence (ICTC). – IEEE, 2024. – S. 1715-1719.
12. Wei Z., Liu Y., Erichson N. B. Emoji Attack: A Method for Misleading Judge LLMs in Safety Risk Detection // arXiv preprint arXiv:2411.01077. – 2024.
13. Rahman M. A. et al. Applying Pre-trained Multilingual BERT in Embeddings for Improved Malicious Prompt Injection Attacks Detection // 2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings). – IEEE, 2024. – S. 1-7.
14. Chen Q., Yamaguchi S., Yamamoto Y. LLM Abuse Prevention Tool Using GCG Jailbreak Attack Detection and DistilBERT-Based Ethics Judgment // Information. – 2025. – T. 16. – №. 3. – S. 204.
15. Aftan S., Shah H. A survey on bert and its applications // 2023 20th Learning and Technology Conference (L&T). – IEEE, 2023. – S. 161-166.
16. Chan C. F., Yip D. W., Esmradi A. Detection and defense against prominent attacks on preconditioned lIIm-integrated virtual assistants // 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). – IEEE, 2023. – S. 1-5.
17. Biarese D. AdvBench: a framework to evaluate adversarial attacks against fraud

- detection systems. – 2022.
18. Liu W. et al. DrBioRight 2.0: an LLM-powered bioinformatics chatbot for large-scale cancer functional proteomics analysis // Nature communications. – 2025. – Т. 16. – №. 1. – С. 2256. DOI: 10.1038/s41467-025-57430-4 EDN: JUMWJQ.
 19. Pannerselvam K. et al. Setfit: A robust approach for offensive content detection in tamil-english code-mixed conversations using sentence transfer fine-tuning // Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages. – 2024. – С. 35-42.
 20. Akpatso S. K. et al. Online News Sentiment Classification Using DistilBERT // Journal of Quantum Computing. – 2022. – Т. 4. – №. 1.
 21. Gritsai G. M., Khabutdinov I. A., Grabovoi A. V. Stackmore LLMs: effektivnoe obnaruzhenie mashinno-sgenerirovannykh tekstov s pomoshch'yu approksimatsii znachenii perpleksii // Doklady Rossiiskoi akademii nauk. Matematika, informatika, protsessy upravleniya. – 2024. – Т. 520. – №. 2. – С. 228-237. DOI: 10.31857/S2686954324700590 EDN: ASZIOX.
 22. Pape D. et al. Prompt obfuscation for large language models // arXiv preprint arXiv:2409.11026. – 2024.
 23. Evglevskaya N. V., Kazantsev A. A. Obespechenie bezopasnosti slozhnykh sistem s integratsiei bol'shikh yazykovykh modelei: analiz ugroz i metodov zashchity // Ekonomika i kachestvo sistem svyazi. – 2024. – №. 4 (34). – С. 129-144. EDN: CJEEAZ.
 24. Shang S. et al. Intentobfuscator: a jailbreaking method via confusing LLM with prompts // European Symposium on Research in Computer Security. – Cham : Springer Nature Switzerland, 2024. – С. 146-165.

Research on performance in modern client-side web-frameworks.

Ratushniak Evgenii Alekseevich

Student; Faculty of Software Engineering and Computer Technology, ITMO National Research University

197101, Russia, St. Petersburg, Petrogradsky district, Kronverksky ave., 49

✉ evgrat123@mail.ru



Abstract. The subject of the study is the comparative rendering performance of three modern frameworks — React, Angular, and Svelte — in typical scenarios of building and updating user interfaces in web applications. The object of the study is the frameworks themselves as complexes of technological solutions, including change detection mechanisms, virtual or compiled DOM structures, and accompanying optimizations. The author thoroughly examines aspects of the topic such as initial and subsequent rendering, element update and deletion operations, and working with linear and deeply nested data structures. Special attention is paid to the practical significance of choosing a framework for commercial products, where performance differences directly impact conversion, user experience, and the financial efficiency of the project. Key internal mechanisms are described — React's virtual DOM, Angular's change detector, and Svelte's compiled code — which determine their behavior in various load scenarios. The methodology is based on an automated benchmark: a unified set of test scenarios is executed by client applications on React, Angular, and Svelte, a reference JavaScript solution, and an Express JS orchestrator server; operation times are recorded using performance.now() in Chrome 126, with Time To First Paint as the performance criterion. The

novelty of the research lies in the comprehensive laboratory comparison of the three frameworks across four critically important scenarios (initial rendering, subsequent rendering, updating, and deleting elements) considering two types of data structures and referencing the current versions of 2025. The main conclusions of the study are as follows: Svelte provides the lowest TTFP and leads in deep hierarchy scenarios due to the compilation of DOM operations; React shows better results in re-rendering long lists, using an optimized diff algorithm and element keys; Angular ensures predictability and architectural integrity but increases TTFP by approximately 60% due to the change detector. There is no universal leader; a rational choice should rely on the analytical profile of the operations of a specific application, which is confirmed by the results of the presented experiment.

Keywords: web interfaces, performance, incremental DOM, virtual DOM, Core Web Vitals, Svelte, Angular, React, JavaScript frameworks, rendering

References (transliterated)

1. Morgan P. JavaScript DOM Manipulation Performance: Comparing Vanilla JavaScript and Leading JavaScript Front-end Frameworks. 2020.
2. Levlin M. DOM benchmark comparison of the front-end JavaScript frameworks React, Angular, Vue and Svelte. LUT University, 2020. S. 58.
3. Aggarwal S. Modern Web Development using ReactJS // International Journal of Recent Research Aspects. 2018. T. 5, № 1. S. 133-137.
4. Saks E. JavaScript frameworks: Angular vs React vs Vue. 2019.
5. Paakkanen J. Upcoming JavaScript Web Frameworks and Their Techniques. LUT University, 2022. 62 s.
6. Siahaan M., Kenidy R. Rendering performance comparison of React, Vue, Next and Nuxt // Jurnal Mantik. 2021. T. 5, № 3. S. 1851-1860.
7. Iseal Sheed. Performance Benchmarking Techniques for React Applications. 2025.
8. Iseal Sheed. Edge Computing and React: Enhancing Performance at the Edge. 2025.
9. Bialecki G., Pańczyk B. Performance analysis of Svelte and Angular applications // Journal of Computer Sciences Institute. 2021. T. 19. S. 139-143. DOI: 10.35784/jcsi.2633. EDN: OYEGVL.
10. Piastou M. Comprehensive performance and scalability assessment of frontend frameworks: React, Angular, and Vue.js // World Journal of Advanced Engineering Technology and Sciences. 2023. T. 9, № 2. S. 366-376. DOI: 10.30574/wjaets.2023.9.2.0153. EDN: JNSAZF.
11. Man'shina E. V., Ermolaeva A. A. Sravnitel'nyi analiz proizvoditel'nosti severnogo i klientskogo renderinga veb-prilozhenii, sozdannykh s pomoshch'yu freimvorka Angular // Nauchnoltekhnicheskie innovatsii i veb-tehnologii. 2022. № 2. S. 44-48.
12. Karić A., Durmić N. Comparison of JavaScript Frontend Frameworks - Angular, React, and Vue // International Journal of Innovative Science and Research Technology (IJISRT). 2024. S. 1383-1390. DOI: 10.38124/ijisrt/ijisrt.
13. Kaveri P. R. Framework-Agnostic JavaScript Component Libraries: Benefits, Implementation Strategies, and Commercialization Models // 2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN). 2024. S. 1441-1446. EDN IGADIU.
14. Kasenda R., Tenda J., Iman E., Manantung J., Moekari Z., Pantas M. The Role and Evolution of Frontend Developers in the Software Development Industry // Jurnal

- Syntax Admiration. 2024. T. 5. S. 5191-5196. DOI: 10.46799/jsa.v5i11.1852.
15. Akieva Z. M., Murzin D. A., El'taev S. I. Primenenie WebAssembly dlya povysheniya proizvoditel'nosti interaktivnykh veb-prilozhenii // Ekonomika i upravlenie: problemy, resheniya. 2024. T. 7, № 12. S. 168-174. DOI: 10.36871/ek.up.p.r. 2024.12.07.021. EDN: WJASJX.
 16. Dubaj S., Pańczyk B. Comparative of React and Svelte programming frameworks for creating SPA web applications // Journal of Computer Sciences Institute. 2022. T. 25. S. 345-349. DOI: 10.35784/jcsi.3020. EDN: AYUSVM.
 17. Chen K. Improving Front-end Performance through Modular Rendering and Adaptive Hydration (MRAH) in React Applications. 2025. DOI: 10.48550/arXiv.2504.03884.
 18. Kravtsov E. P. Razrabotka vysokoproizvoditel'nykh React-prilozhenii: metody i praktiki optimizatsii // European science. 2024. № 1 (69).
 19. Drogaitsev I. A., Tramova A. M. Frontend-freimvork Svelte kak al'ternativa populyarnym resheniyam v kontekste sozdaniya sistem podderzhki prinyatiya reshenii v sfere upravleniya organizatsiei s uchetom transformatsii ekonomiki Rossii // Izvestiya KBNTs RAN. 2024. № 4. DOI: 10.35330/1991-6639-2024-26-4-113-121. EDN: WRZTPH.
 20. Yarovaya E. V. Nestandardnye arkhitektura v napisanie veb prilozhenii // Stolypinskii vestnik. 2022.

Analysis of Microservices Granularity: Effectiveness of Architectural Approaches

Chikaleva Yuliya Sergeevna

Student; Megafacultet of Computer Technology and Management; ITMO National Research University

197101, Russia, St. Petersburg, Petrogradsky district, Kronverksky ave., 49

✉ y.chikaleva99@mail.ru



Abstract. Modern information systems require scalable architectures for processing big data and ensuring availability. Microservice architecture, based on decomposing applications into autonomous services focused on business functions, addresses these challenges. However, the optimal granularity of microservices impacts performance, scalability, and manageability. Suboptimal decomposition leads to anti-patterns, such as excessive fineness or cosmetic microservice architecture, complicating maintenance. The aim of the study is a comparative analysis of methods for determining the granularity of microservices to identify approaches that provide a balance of performance, flexibility, and manageability in high-load systems. The object of the study is the microservice architecture of high-load systems. The subject of the research is the comparison of granularity methods, including monolith, DDD, Data-Driven Approach, Monolith to Microservices, and their impact on the system. The study employs an experimental approach, including the implementation of a Task Manager application in four architectural configurations. Load testing was conducted using Apache JMeter under a load of 1000 users. Performance metrics (response time, throughput, CPU), availability, scalability, security, and consistency were collected via Prometheus and processed to calculate averages and standard deviations. The scientific novelty lies in the development of a methodology for comparative analysis of decomposition methods using unified metrics adapted for high-load systems, setting this study apart from works that focus on qualitative assessments. The results of the experiment showed that the monolithic architecture provides the minimum response time (0.76 s) and high throughput (282.5 requests/s) under a load of 1000 users, but

is limited in scalability. The Data-Driven Approach ensures data consistency, DDD is effective for complex domains, while Monolith to Microservices demonstrates low performance (response time 15.99 s) due to the overload of the authorization service. A limitation of the study is the use of a single host system (8 GB RAM), which may restrict the scalability of the experiment. The obtained data are applicable for designing architectures of high-load systems. It is recommended to optimize network calls in DDD (based on response time of 1.07 s), data access in Data-Driven (response time of 5.49 s), and to carefully plan decomposition for Monolith to Microservices to reduce the load on services (response time of 15.99 s).

Keywords: experimental analysis, data consistency, scalability, performance, Monolith to Microservices, Data-Driven Approach, Domain-Driven Design, granularity of microservices, microservice architecture, high-load systems

References (transliterated)

1. Vorsin V. A. Mikroservisnaya arkhitektura biznes-prilozhenii: perspektivy i problemy // GLOBUS. 2020. Vyp. 4 (50). S. 51-53. EDN: TYXBZZ.
2. Raj P., Vanga S., Chaudhary A. Microservices Security // CloudNative Computing. 2022. DOI: 10.1002/9781119814795.ch14.
3. Ziborev A. V. Antipatterny postroeniya mikroservisnykh prilozhenii v vysokonagrughennykh proektakh // UNIVERSUM: Tekhnicheskie nauki. 2023. Vyp. 11 (116). Noya. 2023. doi: 10.32743/UniTech.2023.116.11.16204. EDN: HBLBIO.
4. Newman S. Building Microservices. 2nd ed. Sebastopol, CA: O'Reilly Media, 2025.
5. Blinowski G. i dr. Monolithic vs. Microservice Architecture: Performance and Scalability Evaluation // IEEE Access. 2022. DOI: 10.1109/ACCESS.2022.3152803. EDN: JHKSCZ.
6. Richardson C. Microservice Architecture Pattern. 2020. Available at: <https://microservices.io/patterns/microservices.html> (accessed: Jan. 05, 2025).
7. Momil S., Qaisar A. Transition Strategies from Monolithic to Microservices Architectures: A Domain-Driven Approach and Case Study // VAWKUM Transactions on Computer Sciences. 2024. Vol. 12, No. 1. Pp. 94-110. DOI: 10.21015/vtcs.v12i1.1808. EDN: ABOYUL.
8. Radostev D. K., Nikitina E. Yu. Strategiya migratsii programmnogo koda iz monolitnoi arkhitektury v mikroservisy // Vestnik Permskogo Universiteta. 2021. Vyp. 2 (53). S. 65-68. doi: 10.17072/1993-0550-2021-2-65-68. EDN: DSNKOZ.
9. Jordanov J., Petrov P. Domain Driven Design Approaches in Cloud Native Service Architecture // TEM Journal. 2023. Vol. 12, Issue 4. S. 1985-1994. ISSN 2217-8309. DOI: 10.18421/TEM124-09. EDN: MMRBFA.
10. Vular H., Koyuncu M. Does Domain-Driven Design Lead to Finding the Optimal Modularity of a Microservice? // IEEE Access. 2021. Vol. 9. Pp. 27960-27971. DOI: 10.1109/ACCESS.2021.306089.
11. Hasselbring W., Steinacker G. Microservice Architectures for Scalability, Agility and Reliability in E-Commerce // 2017 IEEE International Conference on Software Architecture Workshops (ICSAW). Gothenburg: IEEE, 2017. P. 46-49. DOI: 10.1109/ICSAW.2017.11.
12. Schmidt R. A., Thiry M. Microservices Identification Strategies: A Review Focused on Model-Driven Engineering and Domain-Driven Design Approaches // 2020 15th Iberian Conference on Information Systems and Technologies (CISTI). Seville: IEEE, 2020. P. 1-6. DOI: 10.23919/CISTI49556.2020.9141150.
13. Christudas B. Practical Microservices Architectural Patterns. New York: Apress, 2019.

DOI: 10.1007/978-1-4842-4501-9.

14. Sriraman A., Wenisch T. F. µSuite: A Benchmark Suite for Microservices // 2018 IEEE International Symposium on Workload Characterization (IISWC). Raleigh: IEEE, 2018. P. 1-12. DOI: 10.1109/IISWC.2018.8573515.
15. Henning S., Hasselbring W. Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud // Journal of Systems and Software. 2024. Vol. 208. [Online]. Available: <https://doi.org/10.1016/j.jss.2023.111879>. EDN: RVNVGK.
16. Bjørndal N., & Pires de Araújo, etc. Benchmarks and performance metrics for assessing the migration to microservice-based architectures // The Journal of Object Technology. August 2021. DOI: 10.5381/jot.
17. Artamonov I. V. Pokazateli proizvoditel'nosti mikroservisnykh sistem // Vestnik NGIEI. 2018. Vyp. 8 (87). S. 24-33. EDN: XYTGKT.
18. Waseem M., Liang P., Shahin M., et al. Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective // Journal of Systems and Software. 2021. Vol. 182. Article 111061. DOI: 10.1016/j.jss.2021.111061. EDN: ATPBYU.
19. Ramu V. Performance Impact of Microservices Architecture // Rev. Contemp. Sci. Acad. Stud. 2023. Vol. 3. P. 1-6.
20. Hassan S., Bahsoon R., & Buyya R. Systematic scalability analysis for microservices granularity adaptation design decisions // Software: Practice and Experience. 2022. Vol. 52, No. 6. DOI: 10.1002/spe.3069. EDN: MOLQOU.

Intellectual infrastructure for automated control and interoperability of microservices in cloud environments

Rogov Dmitriy Vadimovich

student; Faculty of the Institute of Information Technology, Federal State Budgetary Educational Institution of Higher Education 'MREA- Russian Technological University'

123007, Russia, Moscow, Khoroshevsky district, Khoroshevskoe highway, 60

✉ 1664286@gmail.com



Alpatov Aleksei Nikolaevich

PhD in Technical Science

Associate Professor; Department of Instrumental and Applied Software; Federal State Budgetary Educational Institution of Higher Education 'MREA- Russian Technological University'

111033, Russia, Moscow, Lefortovo district, Tank passage, 4A, sq. 24

✉ aleksej01-91@mail.ru



Abstract. In the context of rapid growth in the scale and complexity of information systems, the questions of effective integration and support of microservices architectures are becoming increasingly relevant. One of the key challenges is ensuring the interoperability of software components, which implies the ability to reliably exchange data and share information between various services implemented using heterogeneous technologies, protocols, and data formats. In this work, the subject of research is the formalization and construction of an intelligent system ensuring the interoperability of microservice components within cloud infrastructure. A formalized approach is proposed, based on graph, categorical, and algebraic models, which allows for a strict description of data transmission routes, conditions for

interface compatibility, and the procedure for automated agreement on interaction formats. An operation for interface agreement is introduced, which identifies the need to use adapters and converters for the integration of various services. Special attention is paid to the task of building a universal interface through which any data streams can be routed, significantly simplifying the process of scaling and refining the microservice system. The developed system architecture encompasses the stages of creation, publication, and deployment of container microservices, automatic verification of data transmission routes, and dynamic management of service states based on load forecasting using artificial intelligence models. The application of the proposed methodology allows for a significant increase in the flexibility, reliability, and scalability of the infrastructure, reduction of operational costs, and automation of the processes of support and integration of new components.

The proposed solution is based on a formalized approach to ensuring the interoperability of microservice components within cloud infrastructure. A graph and categorical model is used as a foundation, allowing for a strict definition of data transmission routes and interface agreement procedures between various services. To unify interaction and enhance system flexibility, an interface agreement operation is introduced, as well as the capability for automated identification of the need for data adapters and converters. The developed intelligent load forecasting algorithm allows for dynamic management of component states and rapid adaptation of the infrastructure to changing operating conditions.

Keywords: Adapters, Deployment automation, Graph model, The categorical model, Interface negotiation, Containerization, Cloud infrastructure, Interoperability, Micro service architecture, Intelligent control

References (transliterated)

1. Paul A., Kelvin L., Brown K. Optimizing IT Growth: Strategies for Building and Scaling Robust Infrastructure Systems // Ladoke Akintola University of Technology. 2024. T. 17. URL: <https://www.researchgate.net/publication/377447014>.
2. Makarenko S. I., Oleinikov A. Ya., Chernitskaya T. E. Modeli interoperabel'nosti informatsionnykh sistem // Sistemy upravleniya, svyazi i bezopasnosti. 2019. № 4. DOI: 10.24411/2410-9916-2019-10408 EDN: NACKGD.
3. D'yakov O. A., Solyanov K. S. Formirovanie kreditnogo konveiera banka na osnove sistem biznes-analitiki // Strategii biznesa. 2016. № 7 (27). S. 7-12. EDN: WWJUUB.
4. ISO/IEC 2382:2015 Information technology. Vocabulary. – 2015. – Tekst: elektronnyi. URL: <https://www.iso.org/ru/standard/63598.html>
5. Mikhnevich S. Yu., Tezhar A. A. Evolyutsiya ponyatiya interoperabel'nosti otkrytykh informatsionnykh sistem // Tsifrovaya transformatsiya. 2023. T. 29. № 2. S. 60-66. URL: <https://doi.org/10.35596/1729-7648-2023-29-2-60-66>. EDN: RKSXQW.
6. Kovalev S. P. Teoretiko-kategoriyi podkhod k proektirovaniyu programmnykh sistem // Fundamental'naya i prikladnaya matematika. 2014. T. 19. № 3. S. 111-170.
7. Seiers E. Kh., Mill A. Docker na praktike / E. Kh. Seiers, A. Mill ; perevod s angliiskogo D. A. Belikov. Moskva : DMK Press, 2020. 516 s. ISBN 978-5-97060-772-5. Tekst : elektronnyi // Lan' : elektronno-bibliotechnaya sistema. URL: <https://e.lanbook.com/book/131719> (data obrashcheniya: 04.06.2025). Rezhim dostupa: dlya avtoriz. pol'zovatelei.
8. Markelov A. A. Vvedenie v tekhnologiyu kontejnerov i Kubernetes / A. A. Markelov. Moskva : DMK Press, 2019. 194 s. ISBN 978-5-97060-775-6. Tekst : elektronnyi // Lan' : elektronno-bibliotechnaya sistema. URL: <https://e.lanbook.com/book/131702> (data

- obrashcheniya: 04.06.2025). Rezhim dostupa: dlya avtoriz. pol'zovatelei.
9. Bontsanini M. Analiz sotsial'nykh media na Python. Izvlekaite i analiziruite dannye iz vsekh ugolkov sotsial'noi pautiny na Python / M. Bontsanini ; perevod s angliiskogo A. V. Logunova. Moskva : DMK Press, 2018. 288 s. ISBN 978-5-97060-574-5. Tekst : elektronnyi // Lan' : elektronno-bibliotechnaya sistema. URL: <https://e.lanbook.com/book/108129> (data obrashcheniya: 09.06.2025). Rezhim dostupa: dlya avtoriz. pol'zovatelei.
10. Gruzdev A. V. Predvaritel'naya podgotovka dannykh v Python / A. V. Gruzdev. Moskva : DMK Press, 2023. Tom 1 : Instrumenty i validatsiya. 2023. 816 s. ISBN 978-5-93700-156-6. Tekst : elektronnyi // Lan' : elektronno-bibliotechnaya sistema. URL: <https://e.lanbook.com/book/314945> (data obrashcheniya: 09.06.2025). Rezhim dostupa: dlya avtoriz. pol'zovatelei.

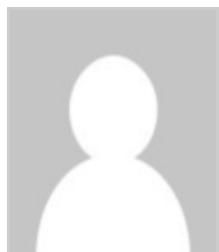
Areal data types in instrumental approach to programming

Dagaev Dmitry Viktorovich

General Director, SCADI LLC

115230, Russia, Moscow, Zeleny Prospekt, 5/12, building 3, room 1

 dvdagaev@oberon.org



Abstract. For a large amount of tasks classical structured programming approach is preferred to object-oriented one. These preferences are typical for deterministic world and in machine-representation-oriented systems. A modular Oberon programming language was oriented on such tasks. It demonstrate minimalistic way of reliability, which differs from vast majority of program systems maximizing amount of features supported. Usage of instrumental approach instead of OOP was proposed before for solving the problems of deterministic world. The data-code separation principle assumes that data lifecycle is independently controlled, and lifetime duration is longer then code lifetime. The areal data types proposed by author are aimed for implementation within instrumental approach. Areal data types provide orthogonal persistency and are integrated with codes, defined in types hierarchy for instruments. Areal data types are embodied in MultiOberon system compilers. Reference to address conversion methods are based on runtime system metadata. Areal types integration resulted in developing additional test in MultiOberon. MultiOberon restrictive semantics makes an opportunity to turn off pointer usage permissions and switch on areal types usage. Areal is fixed for specifically marked data type. Areal references are implemented as persistent ones in areal array. Due to such paradigm the problem of persistence reference during software restarts was solved. Novelty in work is using areal references which gains in index type and pointer type advantages. Such approach implements principles of generic programming without creating dependencies of types extensions and template specifications. An example of generic sorting algorithm is provided for areal types. A new data type differs with compactness and simplicity in comparison to dynamic structures. It demonstrates advantages for systems with complex technological objects data structures in relatively static bounds.

Keywords: data oriented programming, modularity, compiler, orthogonal persistency, semantic restrictions, Oberon, instrumental approach, areal types, informatika-21, metadata

References (transliterated)

1. Mansi Dhirajsinh Parmar, Sarthavi Parmar. Survey on Concept of Object-Oriented Programming. International Journal of Scientific Research in Computer Science Engineering and Information Technology. 2024. Vol. 10, No. 2. P. 427-431. DOI: 10.32628/CSEIT243647. EDN: AMCTWA.
2. West D. Object Thinking. Redmond: Microsoft Press, 2004.
3. Virt N. Algoritmy i struktury dannykh. M.: DMK-Press, 2016.
4. Dagaev D.V. Instrumental'nyi podkhod k programmirovaniyu v sisteme Mul'tiOberon // Programmnye sistemy i vychislitel'nye metody. 2024. № 1. S. 31-47. DOI: 10.7256/2454-0714.2024.1.69437 EDN: WVZVUU URL: https://nbpublish.com/library_read_article.php?id=69437
5. Dagaev D.V. Ogranichitel'naya semantika yazyka v sisteme Mul'tiOberon // Programmnye sistemy i vychislitel'nye metody. 2023. № 1. S. 26-41. DOI: 10.7256/2454-0714.2023.1.36217 EDN: IWIODR URL: https://nbpublish.com/library_read_article.php?id=36217
6. Rajive J., Ph.D. Data-Oriented Architecture: A Loosely-Coupled Real-Time SOA. Real-Time Innovations, Inc., 2007.
7. Hosking A., Nystrom N., Cutts Q., Brahnmath K. Optimizing the read and write barriers for orthogonal persistence // Advances in Persistent Object Systems / Morrison J., Atkinson M. (Eds.). San Francisco: Morgan Kaufmann, 1999. P. 11.
8. Lefort A. A Support for Persistent Memory in Java. Computer science. Institut Polytechnique de Paris, 2023. English. ffNNT : 2023IPPAS001.
9. Bläser L., Russo C., Greif G., Vandersmith R., Ibrahim J. Smarter Contract Upgrades with Orthogonal Persistence // Proceedings of the 16th ACM SIGPLAN International Workshop on Virtual Machines and Intermediate Languages (VMIL '24). Pasadena, CA, USA, 2024. P. 11.
10. Virt N., Gutknekht Yu. Razrabotka operatsionnoi sistemy i kompilyatora. Proekt Oberon. M.: DMK-Press, 2015.
11. Crelier R. OP2: A Portable Oberon Compiler. ETH, Departement Informatik, 1990. P. 32-34.
12. Leonenkov A.V. Samouchitel' UML. 2-e izd., pererab. i dop. SPb.: BKhV-Peterburg, 2006. P. 150.
13. Gamma E., Khelm R., Dzhonson R., Vlissides Dzh. Priemy ob"ektno-orientirovannogo proektirovaniya. Patterny proektirovaniya. SPb.: Piter, 2001.
14. Martin R. Chistaya arkhitektura. Iskusstvo razrabotki programmnogo obespecheniya. SPb.: Piter, 2022.
15. Martin R. Ideal'nyi programmist. Kak stat' professionalom razrabotki PO. SPb.: Piter, 2022.
16. Musser D.R., Stepanov A.A. Generic Programming // Proceedings of the International Symposium ISSAC'88 on Symbolic and Algebraic Computation. New York: Springer-Verlag, 1988. P. 13-25.
17. Roynard M. Generic programming in modern C++ for Image Processing. Signal and Image Processing. Sorbonne Université, 2022. English. ffNNT : 2022SORUS287ff.
18. Dagaev D.V. O razrabotke Oberon-sistemy s zadannymi svoistvami ergodichnosti // Trudy ISP RAN. 2020. T. 32, vyp. 6. S. 67-78. DOI: 10.15514/ISPRAS-2020-32(6)-5. EDN: HBGGIK.
19. Dagaev D.V. Ispolnyayushchaya mashina avtomatnykh programm // Nauchno-tehnicheskii vestnik informatsionnykh tekhnologii, mekhaniki i optiki. 2021. T. 21, №

4. S. 525-534. DOI: 10.17586/2226-1494-2021-21-4-525-534. EDN: MJZXQZ.
20. Lo S.K., Lu Q., Zhu L., Paik H.-Y., Xu X., Wang C. Architectural patterns for the design of federated learning systems // Journal of Systems and Software. 2022. Vol. 191. P. 357.
21. Ali H.M., Hamza M.Y., Rashid T.A. Exploring Polymorphism: Flexibility and Code Reusability in Object-Oriented Programming // Proceedings of the 4th International Conference on Recent Innovation in Engineering (ICRIE 2023). University of Duhok, College of Engineering, 13th-14th September 2023. Paper No. 33.
22. Khattak U.F., Hussein H.A. A review on graph representation for object-oriented programming // BIO Web of Conferences. 2024. Vol. 97. Article 00131. DOI: 10.1051/bioconf/20249700131. EDN: YIHEDW .

The use of neural networks for real-time big data analysis

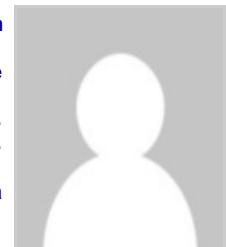
Makarov Igor Sergeevich

PhD in Technical Science

Head of the Department; Department of Software Engineering (PrI); Volga State University of Telecommunications and Informatics

23 Lva Tolstogo str., Samara, Samara region, 443010, Russia

✉ igor-psati@yandex.ru



Raikov Aleksandr Vyacheslavovich

Student, Department of Computer Science and Computer Engineering, Volga Region State University of Telecommunications and Informatics

23 Lva Tolstogo str., Samara, Samara region, 443010, Russia

✉ sraikov7@mail.ru

Kazantsev Andrei Alekseevich

Student; Department of Computer Science and Computer Engineering; Volga State University of Telecommunications and Informatics

23 Lva Tolstogo str., Samara, Samara region, 443010, Russia

✉ NuclearAndGoner@gmail.com

Nekhaev Maksim Vadimovich

Student; Department of Computer Science and Computer Engineering; Volga State University of Telecommunications and Informatics

23 Lva Tolstogo str., Samara, Samara region, 443010, Russia

✉ maks.popovich2014@yandex.ru



Romanov Mikhail Aleksandrovich

Student; Department of Computer Science and Computer Engineering; Volga State University of Telecommunications and Informatics

23 Lva Tolstogo str., Samara, Samara region, 443010, Russia

✉ gp.romanov@mail.ru



Abstract. The article is devoted to the exploration of the possibilities of using neural networks for real-time big data analysis in the field of information security. The relevance of the topic is due to the rapid growth of generated data volumes, the complexity of cyberattack

methods, and the necessity to develop new effective approaches to information protection. The work examines in detail the key tasks addressed using neural network technologies, including anomaly detection in network traffic, prevention of distributed DDoS attacks, classification of malware, and forecasting new cyber threats. Special attention is paid to the unique advantages of neural networks, such as the ability to process extremely large volumes of heterogeneous data, identify complex non-obvious attack patterns, continuously learn and adapt to rapidly changing conditions in the cyber environment. The study utilizes deep learning methods, including convolutional and recurrent neural networks, for big data analysis and cyber threat identification. Approaches for real-time data processing and model robustness assessment are applied. The conducted research demonstrates that modern neural network architectures possess significant potential for revolutionary transformation of information security systems. Key advantages include ultra-high speed of streaming data processing, the capability to detect previously unknown types of attacks through the identification of complex correlations, as well as the ability to predict threats based on historical data analysis. However, the research also revealed serious technological challenges: excessive demand for computational resources to train complex models, the "black box" problem in interpreting decisions, the vulnerability of the neural network models themselves to specialized attacks (adversarial attacks), and ethical aspects of automated decision-making in cybersecurity. Successful implementation cases are presented, including next-generation intrusion detection systems and malware analysis platforms. The authors see promising directions for further research in the development of energy-efficient neural network models, creation of explainable AI methods for security, and advancement of adaptive systems capable of evolving alongside cyber threats. The results obtained are valuable for cybersecurity specialists, developers of protective solutions, and researchers in the field of artificial intelligence.

Keywords: cyber attacks, DDoS, information security, threat forecasting, NLP, RNN, CNN, Big Data, neural networks, data

References (transliterated)

1. Vorontsov K.V. Mashinnoe obuchenie i iskusstvennye neironnye seti / K.V. Vorontsov. – M.: DMK Press, 2020. – 448 s. – ISBN 978-5-97060-799-1.
2. Gorban' A.N., Dunin-Barkovskii V.L. Neironnye seti: obuchenie, organizatsiya primenenie / A.N. Gorban', V.L. Dunin-Barkovskii. – M.: IPRZhR, 2018. – 292 s. – ISBN 978-5-93121-381-8.
3. Korneev V.V. Big Data v informatsionnoi bezopasnosti: analiz ugroz v real'nom vremeni // Prikladnaya informatika. – 2021. – № 4. – S. 45-58. – DOI: 10.25791/pfim.04.2021.1245.
4. Sokolov I.A., Petrov D.Yu. Primenenie LSTM-setei dlya obnaruzheniya DDoS-atak v potokovykh dannykh // Informatika i ee primeneniya. – 2022. – T. 16, № 3. – S. 72-83. – DOI: 10.14357/19922264220308.
5. Ivanov A.M., Kuznetsov S.P. Integratsiya Apache Kafka i neirosetevykh modelei dlya analiza kiberatak // Trudy mezhdunarodnoi konferentsii "Tsifrovaya transformatsiya-2023". – SPb.: Izd-vo Politekhnicheskogo universiteta, 2023. – S. 112-125.
6. Romanova O.L., Timofeev A.V. Etika iskusstvennogo intellekta v kontekste informatsionnoi bezopasnosti // Filosofiya i nauka. – 2021. – № 12. – S. 64-75. – DOI: 10.15372/PS20211206.
7. Babicheva M.V., Tret'yakov I.A. Primenenie metodov mashinnogo obucheniya dlya

- avtomatizirovannogo obnaruzheniya setevykh vtorzhenii // Vestnik Dagestanskogo gosudarstvennogo tekhnicheskogo universiteta. Tekhnicheskie nauki. – 2023. – T. 50, № 1. – S. 53-61. – DOI: 10.21822/2073-6185-2023-50-1-53-61. – EDN: MGBAGF.
8. Pozdnyak I.S., Makarov I.S. Modeli obnaruzheniya atak s ispol'zovaniem metodov mashinnogo obucheniya // Vestnik Rossiiskogo novogo universiteta. Seriya: Slozhnye sistemy: modeli, analiz i upravlenie. – 2024. – № 1. – S. 99-109. – DOI: 10.18137/RNU.V9187.24.01.P.99. – EDN: MNMSYZ.
9. Babicheva M.V., Tret'yakov I.A. Primenenie metodov mashinnogo obucheniya dlya avtomatizirovannogo obnaruzheniya setevykh vtorzhenii // Vestnik Dagestanskogo gosudarstvennogo tekhnicheskogo universiteta. Tekhnicheskie nauki. – 2023. – T. 50, № 1. – S. 53-61. – DOI: 10.21822/2073-6185-2023-50-1-53-61. – EDN: MGBAGF.
10. Kharrison M. Mashinnoe obuchenie: karmannyi spravochnik. Kratkoe rukovodstvo po metodam strukturirovannogo mashinnogo obucheniya na Python / Per. V.A. Kovalenko. – SPb.: Dialektika, 2020. – 320 s. – ISBN 978-5-907203-17-4.

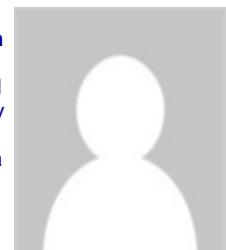
Implementation of Drag&Drop behavior in an Android application based on the gesture processing API.

Petrovsky Aleksandr Andreevich

Bachelor's degree in Software Engineering; Institute of Information Technology, MREA-Russian Technological University

78 Vernadsky Ave., Moscow, 119454, Russia

✉ petrovskiy.a.a@edu.mirea.ru



Rysin Mikhail Leonidovich

PhD in Pedagogy

Associate Professor; Department of Mathematical Support and Standardization of Information Technologies; MREA-Russian Technological University

78 Vernadsky Ave., Moscow, 119454, Russia

✉ rysin@mirea.ru



Abstract. The subject of the study is the organization of the movement of user interface objects (User Interface, UI) in Android applications. The focus of the research is the development of a software solution for implementing Drag&Drop behavior in mobile Android applications using the modern user interface framework Jetpack Compose.

The relevance of the presented work is due to the need to create flexible and intuitive mechanisms for user interaction with the interface of mobile Android applications.

The main results of the research include:

1. Development of a set of Composable functions for managing the Drag&Drop state of user interface objects.
2. Integration of the behavior of the "source" and "receiver" of the moving UI objects with the possibility of decoration.
3. Overcoming the limitations of the built-in tools of the Jetpack Compose framework.
4. Creation of a mechanism for handling user movement gestures.
5. Formation of a universal approach to implementing interactive interaction with interface elements.

The methodology is based on the application of the architectural pattern MVI (Model-View-Intent), which provides effective management of interface state, and the use of object-

oriented design patterns, in particular, the "decorator" pattern. Research methods include analyzing existing approaches to implementing Drag&Drop, designing a software solution, developing a prototype, and testing it within a mobile application. The scientific novelty of the research lies in the development of an innovative approach to organizing Drag&Drop interaction, which allows overcoming the limitations of built-in tools of the Jetpack Compose framework.

The proposed solution is characterized by:

- complete isolation of Drag&Drop components;
- possibility of decorating moving UI objects;
- flexible configuration of the behavior of the source and receiver of interface objects;
- absence of rigid connections between user interface components.

The practical significance of the work lies in the development of tools that can be successfully applied in various mobile software projects requiring complex user interactions. The conclusions of the research demonstrate the effectiveness of the proposed solution in overcoming the existing limitations of Jetpack Compose and open new opportunities for creating more dynamic and user-friendly interfaces in mobile applications.

Keywords: Model-View-Intent, Decorator pattern, MVI pattern, Jetpack Compose, gesture processing, user interface, Drag-and-Drop technology, Android, mobile application, Composable function

References (transliterated)

1. What is Drag and Drop? Drag and Drop explained [Elektronnyi resurs]. URL: <https://goodspeed.studio/glossary/what-is-drag-and-drop-drag-and-drop-explained> (data obrashcheniya: 14.03.2025).
2. Jetpack Compose [Elektronnyi resurs]. URL: <https://developer.android.com/compose> (data obrashcheniya: 18.03.2025).
3. Pochemu tak udobno ispol'zovat' pattern MVI v KMM [Elektronnyi resurs]. URL: <https://habr.com/ru/companies/kts/articles/729832/> (data obrashcheniya: 18.03.2025).
4. Jetpack Compose: Drag and Drop [Elektronnyi resurs]. URL: <https://developer.android.com/develop/ui/compose/touch-input/user-interactions/drag-and-drop> (data obrashcheniya: 20.03.2025).
5. Google IssueTracker: DragAndDropTarget [Elektronnyi resurs]. URL: <https://issuetracker.google.com/issues/324280271> (data obrashcheniya: 20.03.2025).
6. Martin R. Chistaya arkitektura. Iskusstvo razrabotki programmnogo obespecheniya. – SPb.: Piter, 2024. – 352 s.
7. Gamma E., Khelm R., Dzhonson R., Vlissides Dzh. Patterny ob"ektno-orientirovannogo programmirovaniya. – SPb.: Piter, 2022. – 448 s.
8. Jetpack Compose: CompositionLocal [Elektronnyi resurs]. URL: <https://developer.android.com/develop/ui/compose/compositionlocal> (data obrashcheniya: 05.03.2025).
9. Jetpack Compose: GraphicsLayer [Elektronnyi resurs]. URL: <https://developer.android.com/develop/ui/compose/graphics/draw/modifiers> (data obrashcheniya: 10.03.2025).
10. Kotlin: Scope-functions [Elektronnyi resurs]. URL: <https://kotlinlang.org/docs/lambdas.html#invoking-a-function-type-instance> (data obrashcheniya: 11.03.2025).
11. Jetpack Compose: Pointer Input [Elektronnyi resurs]. URL:

<https://developer.android.com/develop/ui/compose/touch-input/pointer-input> (data obrashcheniya: 15.03.2025).

Simulation modeling of the functional twin of the microclimate control system of an intelligent building.

Dushkin Roman Viktorovich 

Senior Lecturer; Department of Cybernetics; National Research Nuclear University MEPhI
General Director, OOO 'A-Ya expert'

31 Kashirskoe shosse, Mskvorechye-Saburovo district, Moscow, 115409, Russia

 roman.dushkin@gmail.com

Klimov Valentin Vyacheslavovich

PhD in Technical Science

Associate Professor; Institute of Intelligent Cybernetic Systems Institute; National Research Nuclear University
MEPhI

31 Kashirskoe highway, Mskvorechye-Saburovo district, Moscow, 115409, Russia

 vklimov@mephi.ru



Abstract. The presented work is dedicated to the development of an intelligent microclimate control system for buildings (an HVAC system). The research focuses on addressing the problem of insufficient adaptability of traditional approaches (PID controllers, knowledge-based systems) in the context of dynamically changing internal environmental parameters of a building. The main emphasis is on creating a hybrid method that combines the advantages of functional programming and artificial intelligence. The study examines issues of energy efficiency, accuracy in maintaining comfortable conditions for visitors of intelligent buildings, and the robustness of the HVAC system to external disturbances. A crucial task is to minimize operational costs while ensuring the safety and reliability of equipment operation. The presented research covers all stages of software development—from designing its architecture to practical testing. The core of the research is based on the approach of a functional twin implemented in Haskell. LSTM networks are used for forecasting, genetic algorithms for optimization, and the RETE algorithm for rule processing. Verification is conducted through simulation modeling, generating 1440 data points. The scientific novelty of the presented work lies in the application of a categorical-theoretic approach to model the functional twin, where each device (both sensors and actuators) is represented as a composition of pure functions. Results demonstrate a 14.7% reduction in energy consumption, an increase in the operational time within a comfortable range to 94.7%, and a threefold reduction in the switching frequency of the HVAC system modes. Practical significance is confirmed by a 15% decrease in operational costs and improved cyber resilience through the use of immutable data structures. The conclusions indicate that the combination of functional programming with a hybrid approach in artificial intelligence provides a balance of key system parameters. The proposed architecture can serve as a benchmark for integrating IoT and cyber-physical systems within the framework of Industry 4.0.

Keywords: HVAS System, Haskell, energy effectiveness, hybrid artificial intelligence, intellectual building, RETE Algorithm, microclimat control, simulations, functional twin, distributed computing

References (transliterated)

1. Komarov N.M., Zharov V.G. Kontseptsya pereustroistva upravleniya energoeffektivnost'yu intellektual'nogo zdaniya // Servis v Rossii i za rubezhom. 2013. № 7. S. 36-47. EDN: RAMRRV.
2. Sungatullin R.G. Energeticheskaya effektivnost' umnykh gorodov: ekonomicheskie aspekty vnedreniya novykh tekhnologii // Ekonomika i upravlenie: problemy, resheniya. 2025. T. 2. № 3. S. 97-105. DOI: 10.36871/ek.up.p.r.2025.03.02.012. EDN: HQVOQH.
3. Dushkin R.V. Funktsional'noe programmirovaniye na yazyke Haskell. M.: DMK-Press, 2007. 608 s.
4. Dushkin R.V., Mokhov A.I. Model' raspredelennykh vychislenii dlya organizatsii programmnoi sredy, obespechivayushchei upravlenie avtomatizirovannymi sistemami intellektual'nykh zdanii // Komp'yuternye issledovaniya i modelirovaniye. 2021. T. 13. № 3. S. 557-570. DOI: 10.20537/2076-7633-2021-13-3-557-570. EDN: GHZEVY.
5. Lipovacha M. Izuchai Haskell vo imya dobra! / per. s angl. D. Leushina, A. Sinityna, Ya. Arsanukaeva. M.: DMK Press, 2014. 490 s.
6. Mao Z., Wu L., Song L., Huang D. Data Preprocessing and Kalman Filter Performance Improvement Method in Integrated Navigation Algorithm // Chinese Control Conference (CCC). Guangzhou, China, 2019. P. 3416-3422. DOI: 10.23919/ChiCC.2019.8865567.
7. Nusrat I., Jang S.-B. A Comparison of Regularization Techniques in Deep Neural Networks // Symmetry. 2018. Vol. 10. No. 11. P. 648. DOI: 10.3390/sym10110648.
8. Xu Y., Wang R., Goswami N., Li T., Gao L., Qian D. Software Transactional Memory for GPU Architectures // Proceedings of Annual IEEE/ACM International Symposium on Code Generation and Optimization (CGO '14). New York: ACM, 2014. P. 1-10. DOI: 10.1145/2581122.2544139.
9. Bouktif S., Fiaz A., Ouni A., Serhani M.A. Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches // Energies. 2018. Vol. 11. No. 7. P. 1636. DOI: 10.3390/en11071636.
10. Karimi A., Mohajerani M., Alinasab N., Akhlaghinezhad F. Integrating Machine Learning and Genetic Algorithms to Optimize Building Energy and Thermal Efficiency Under Historical and Future Climate Scenarios // Sustainability. 2024. Vol. 16. No. 21. P. 9324. DOI: 10.3390/su16219324. EDN: PDGTHC.
11. Do S.W.S., Dubois M. Transaction-Based Core Reliability // 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS). New Orleans, 2020. P. 168-179. DOI: 10.1109/IPDPS47924.2020.00027.
12. Qi J., Kim Y., Chen C., Lu X., Wang J. Demand Response and Smart Buildings: A Survey of Control, Communication, and Cyber-Physical Security // ACM Trans. Cyber-Phys. Syst. 2017. Vol. 1. No. 4. Article 18. 25 p. DOI: 10.1145/3009972.
13. Alghamdi S.M., Ajour M.N., Abu-Hamdeh N.H., Karimipour A. Introducing a new PID controller to control the addition of PCM to the building with ventilation heat recovery installation to reduce the energy demand of the cooling system // Journal of Building Engineering. 2022. Vol. 56. 104766. DOI: 10.1016/j.jobe.2022.104766. EDN: YPRUFR.
14. Xue Y., Jiang J., Hong L. A LSTM based prediction model for nonlinear dynamical systems with chaotic itinerancy // Int. J. Dynam. Control. 2020. Vol. 8. P. 1117-1128. DOI: 10.1007/s40435-020-00673-4. EDN: CCFIWE.
15. Afram A., Janabi-Sharifi F. Effects of dead-band and set-point settings of on/off controllers on the energy consumption and equipment switching frequency of a residential HVAC system // Journal of Process Control. 2016. Vol. 47. P. 161-174. DOI: 10.1016/j.jprocont.2016.09.009.

16. Sagan J., Sobotka A. Analysis of Factors Affecting the Circularity of Building Materials // Materials. 2021. Vol. 14. No. 23. P. 7296. DOI: 10.3390/ma14237296. EDN: UYSVLB.
17. Mokhov A.B., Dushkin R.V. Funktsional'nyi podkhod k intellektualizatsii ob"ektov na osnove kompleksotekhniki // E-Management. 2020. T. 3. № 4. S. 13-25. DOI: 10.26425/2658-3445-2020-3-4-13-25. EDN: AKDRKN.
18. Valliappan N., Krook R., Russo A., Claessen K. Towards secure IoT programming in Haskell // Proceedings of the 13th ACM SIGPLAN International Symposium on Haskell (Haskell 2020). New York: ACM, 2020. P. 136-150. DOI: 10.1145/3406088.3409027.
19. Park D., Kim S., An Y., Jung J.-Y. LiReD: A Light-Weight Real-Time Fault Detection System for Edge Computing Using LSTM Recurrent Neural Networks // Sensors. 2018. Vol. 18. No. 7. P. 2110. DOI: 10.3390/s18072110.
20. Raiaan M.A.K., Mukta S.H., Fatema K., Fahad N.M., Sakib S., Mimet M.M.J. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges // IEEE Access. 2024. Vol. 12. P. 26839-26874. DOI: 10.1109/ACCESS.2024.3365742. EDN: LTUDIM.
21. Bilotta G., Genovese E., Citroni R., Cotroneo F., Meduri G.M., Barrile V. Integration of an Innovative Atmospheric Forecasting Simulator and Remote Sensing Data into a Geographical Information System in the Frame of Agriculture 4.0 Concept // AgriEngineering. 2023. Vol. 5. No. 3. P. 1280-1301. DOI: 10.3390/agriengineering5030081. EDN: ETBJWA.

Development of the PLAY VISION AI project for watching sports matches using artificial intelligence

Kovalev Sergei Vasil'evich 

PhD in Technical Science

Associate Professor; Department of Computer Engineering; I.N. Ulyanov Chuvash State University

428015, Russia, Republic of Chuvashia, Cheboksary, Moskovsky ave., 15, office B-309

 srgkov@gmail.com

Smirnova Tatiana Nikolaevna 

PhD in Physics and Mathematics

Associate Professor; Department of Mathematical and Hardware Support of Information Systems; I.N. Ulyanov Chuvash State University

428015, Russia, Republic of Chuvashia, Cheboksary, Moskovsky ave., 15, office B-304

 smirnova-tanechka@yandex.ru

Zverev Roman Evgen'evich 

Student; Faculty of Law; I.N. Ulyanov Chuvash State University

15 Moskovsky Ave., Cheboksary, Republic of Chuvashia, 428015, Russia

 roma_zverev2002@mail.ru

Rakov Ivan Vital'evich 

student; Faculty of Computer Science and Computer Engineering; I.N. Ulyanov Chuvash State University

428015, Russia, Republic of Chuvashia, Cheboksary, Moskovsky ave., 15, office B-309

 rakov-40@mail.ru

Abstract. With the development of digital technologies the sports industry is facing a growing

need for advanced analytical tools. In football the use of computer vision and machine learning technologies to analyze games is becoming not just a trend, but a necessity to maintain competitiveness.

The use of computer vision and machine learning in sports analytics allows to automatically extract meaningful data from video matches, which significantly increases the speed and accuracy of analysis compared to traditional methods. Such technologies can provide coaches with detailed reports on the movements, positioning and tactics of players in real time.

The goal is to create a system that will allow for a comprehensive analysis of football matches using the latest advances in artificial intelligence and computer vision.

The main method is a review and analysis of publications on the research topic; analysis of modern technologies that allow automatic processing of video data. The main methodology is the concept of developing the PLAY VISION AI project as a way to watch sports matches using artificial intelligence to evaluate the effectiveness of game strategies. The relevance of this work is due to the maximum modification of modern technical means to improve analytical capabilities in sports.

Main results: algorithms for calibration and correction of video distortions from matches have been developed; methods for detecting and tracking reference points and players have been developed; algorithms for comparing images with real coordinates on the field have been implemented; the developed methods have been integrated into a single system with an interface for end users. The developed system will provide coaches and analysts with tools to evaluate the effectiveness of game strategies and prepare for upcoming matches. It will also contribute to the further development of analysis technologies in sports.

Keywords: video system, player tracking, analysis of football matches, sports game analysis, the sports industry, machine learning, machine vision, artificial intelligence, sports technology, object detection

References (transliterated)

1. Kasisi D. Primenenie iskusstvennogo intellekta v sporte // In Situ. 2023. № 5. S. 30–33. EDN CHZPXZ.
2. Kovalev S. V., Kopysheva T. N., Mitrofanova T. V., Smirnova T. N. O priobretenii opyta nauchno-issledovatel'skoi raboty budushchimi IT-spetsialistami v usloviyakh studencheskogo kruzhka // Sovremennye naukoemkie tekhnologii. 2021. № 7. S. 117–122. DOI: 10.17513/snt.38762. EDN YCJEAI.
3. Kuznetsov A. Yu., Barsukova K. V. Big Data, gadzhety i sportivnaya analitika: kak ustroen rossiiskii sporttekh // Ispol'zovanie Big Data v ofitsial'noi statistike Using Big Data in official statistics : sb. mater. II Vseros. nauch.-issled. konf., Lipetsk, 29 iyunya 2022 goda. Lipetsk: Izd-vo LGTU, 2022. S. 162–165. EDN NHQSYP.
4. Nekotorye osobennosti izucheniya videoanaliza v sporte / R. F. Miftakhov, M. I. Galyautdinov, A. M. Situdikov, I. R. Fatkullov // Uchenye zapiski universiteta im. P.F. Lesgafta. 2019. № 12(178). S. 207–210. DOI: 10.34835/issn.2308-1961.2019.12.207–210. EDN BRW XMC.
5. O primenenii metoda IT-startapov v obuchenii budushchikh IT-spetsialistov / T. V. Mitrofanova, T. N. Smirnova, T. N. Kopysheva [i dr.] // Innovatsionnoe razvitiie professional'nogo obrazovaniya. 2021. № 2(30). S. 49–57. EDN LPIDCF.
6. Perechen' poruchenii po itogam konferentsii «Puteshestvie v mir iskusstvennogo intellekta» (utv. Prezidentom RF 29 yanvarya 2023 g. № Pr-172) [Elektronnyi resurs]. URL: <https://www.garant.ru/products/ipo/prime/doc/406165361/> (Data obrashcheniya:

09.11.2023).

7. Pokataev A. V., Testoedov V. A. Analitika v futbole: aktual'nye napravleniya, razvitiya, perspektivy v trenerskoi deyatel'nosti // Sovremennyi menedzhment v igrovых vidakh sporta, Moskva, 15 noyabrya 2017 goda / Materialy sovmestnoi konferentsii kafedry «Menedzhmenta i ekonomiki sporta im. V.V. Kuzina» i kafedry «Teorii i metodiki futbola» FGBOU VO «RGUFKSMiT». M. : Nauchnyi konsul'tant, 2017. S. 131–134. EDN YLDARF.
8. Polozov A. A., Mal'tseva N. A. Sushchestvuyushchie modeli sportivnoi analitiki i ikh web-servisy // Nauchnye i obrazovatel'nye osnovy v fizicheskoi kul'ture i sporte. 2022. T. 8, № 4. S. 60–77. DOI: 10.57006/2782-3245-2022-8-4-58-76. EDN IXDUCJ.
9. Primenenie onlain-kompilyatorov dlya obucheniya programmirovaniyu studentov IT-spetsial'nostei vuza / T. N. Kopysheva, T. N. Smirnova, T. V. Mitrofanova, M. V. Volik // Novye kompetentsii tsifrovoi real'nosti: teoriya i praktika ikh razvitiya u obuchayushchikhsya: sb. dokl. i nauch. st. II Vseros. nauch.-prakt. konf., Cheboksary, 21 marta 2021 goda. Cheboksary: Izd-vo Chuvash. un-ta, 2021. S. 168–173. EDN STREVV.
10. Chakhoyan T. A., Buchinchik P. V., Egorov A. R. Analitika sportivnykh dostizhenii futbol'nykh klubov i ikh igrokov pri pomoshchi instrumentov mashinnogo obucheniya // Vserossiiskaya Studencheskaya konferentsiya «Studencheskaya nauchnaya vesna», posvyashchennaya 85-letiyu Yu. A. Gagarina: sb. tezisov dokl., Moskva, 01–30 aprelya 2019 goda. M.: Izdatel'skii dom «Nauchnaya biblioteka», 2019. S. 315–316. EDN LLFMAN.
11. Rajasekaran, S. B. AI and Cybersecurity-How AI Augments Cybersecurity Posture of an Enterprise / S. B. Rajasekaran // International Journal of Intelligent Systems and Applications in Engineering. 2023. Vol. 11, No. 1. P. 179–182. EDN EZCRL

Modern methods of preventing DDoS attacks and protecting web servers

Kozyreva Nadezhda Ivanovna

PhD in Technical Science

Associate Professor; Department of Information Security, Volga State University of Telecommunications and Informatics

23 Lva Tolstogo str., Samara, Samara region, 443010, Russia

✉ n.kozyreva@psuti.ru

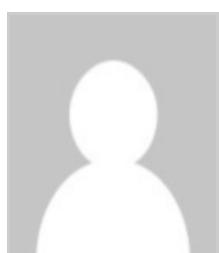


Muhtulov Mihail Olegovich

Student; Department of Computer Science and Computer Engineering; Volga Region State University of Telecommunications and Informatics

443125, Russia, Samara region, Samara, Industrial district, Moskovskoe shosse, 252, sq. 197

✉ mixa.1204@inbox.ru



Ershov Sergei Aleksandrovich

Student; Department of Computer Science and Computer Engineering; Volga Region State University of Telecommunications and Informatics

Ulyanovskaya str., 28, sq. 7, Kinel, Samara region, 446430, Russia

✉ vizionera8@gmail.com

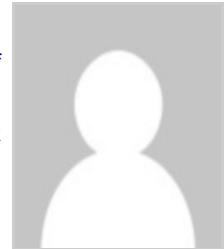


Novoseltseva Sofiya Vladimirovna

Student; Department of Computer Science and Computer Engineering; Volga Region State University of Telecommunications and Informatics

443093, Russia, Samara region, Samara, Zheleznodorozhny district, 31 Maurice Thorez St., 79 block

✉ sunny.tea.with.lilac@gmail.com



Akhmadullin Dinar Ayratovich

Student; Department of Computer Science and Computer Engineering; Volga Region State University of Telecommunications and Informatics

443101, Russia, Samara region, Samara, Kuibyshevsky district, Khasanovskaya str., 14, sq. 17

✉ dinarnevashno@yandex.ru



Abstract. The object of the study is modern methods and technologies for protecting web servers from distributed denial of service (DDoS) attacks. The subject of the research covers current strategies for preventing and mitigating DDoS threats, including a detailed classification of attacks by types and vectors of impact. Special attention is paid to the mechanisms of DDoS attack effects on information systems, addressing both the technical aspects of operational disruptions and their consequences for business processes. The field of study analyzes modern technological protection solutions: Anycast routing, rate limiting, behavioral analysis systems for network traffic, and CAPTCHA mechanisms. Additionally, the integration of innovative approaches with traditional cybersecurity tools—such as firewalls, intrusion prevention systems (IPS), and protective proxy servers—is explored. The relevance of the research is determined by the rapid digitization and exponential growth in the complexity of cyberattacks, making the issue of DDoS protection critically important for ensuring the resilience of web infrastructures. The methodology includes an analysis of DDoS attacks at the network, transport, and application levels, assessing their impact on IT systems. Modern protective technologies are examined, including anomaly detection systems, load balancing, ML traffic filtering, and cloud solutions. Special attention is given to the adaptability and scalability of protection. The scientific novelty of the work lies in a comprehensive analysis of the economic and technical aspects of countering DDoS threats, including an assessment of the cost and effectiveness of various solutions for businesses of different scales. The research offers practical recommendations for building multi-layered protection that combines innovative approaches (machine learning, cloud services) with proven methods (firewalls, IPS). An analysis of real cases demonstrates the effectiveness of adaptive strategies against modern complex attacks. The conclusions emphasize the need for a proactive approach to security that considers both technological and organizational protective measures. The results obtained have practical value for cybersecurity specialists, system administrators, and developers of protective solutions, providing them with a methodological basis for creating DDoS-resistant web infrastructures. The work also outlines promising directions for further research in the field of intelligent detection and neutralization systems for attacks.

Keywords: Machine learning, Automated attack prevention, WAF, Network protection, Cybersecurity, Fail2Ban, Rate Limiting, Cloudflare, Web server, DDoS attacks

References (transliterated)

1. Abramov A. G. Zashchita ot DDoS-atak svoimi rukami: operativnye razrabotka i vnedrenie servisa v Natsional'noi issledovatel'skoi komp'yuternoi seti Rossii // Programmnye produkty i sistemy. 2022. № 4. DOI: 10.15827/0236-235X.140.572-582

- EDN: OGJSQLQ.
2. Amanzholov O. M. Issledovanie metodov i sredstv obnaruzheniya DDoS-atak // Molodoi uchenyi. 2023. № 50 (497). S. 5-8. URL: <https://moluch.ru/archive/497/109243/> (data obrashcheniya: 18.03.2025). EDN: XIACUA.
 3. Orekhov A. V., Orekhov A. A. Avtomaticeskoe obnaruzhenie anomalii setevogo trafika pri DDoS-atakah // Vestnik Sankt-Peterburgskogo universiteta. Prikladnaya matematika. Informatika. Protsessy upravleniya. 2023. T. 19. Vyp. 2. S. 251-263. DOI: 10.21638/11701/spbu10.2023.210 EDN: XYNCXN.
 4. Unru P. P., Obukhov S. A., Cheremukhin D. N. Mery po zashchite infokommunikatsionnykh sistem ot DDoS-atak s usilением // Molodezh' i nauka: aktual'nye problemy fundamental'nykh i prikladnykh issledovanii. 2022. Ch. 1. S. 1-497. DOI: 10.17084/978-5-7765-1511-8-2022 EDN: OCZGSD.
 5. Kangkan Talukdar, Debojit Boro. Slowloris Attack Detection Using Adaptive Timeout-Based Approach // The ISC Int'l Journal of Information Security. 2024. № 1. S. 79-92. URL: https://www.isecure-journal.com/article_183600_e06eaaffd81aef753b956e80b513f82b.pdf (data obrashcheniya: 10.03.2025).
 6. Vereshchagin K. V. Zashchita korporativnykh setei ot DDoS-atak: sovremennye metody i tendentsii // Nauchnyi lider. 2023. № 47 (145). S. 12-15. URL: https://scilead.ru/media/journal_pdf145.pdf#page=12 (data obrashcheniya: 10.03.2025). EDN: CSBAGF.
 7. Razrabotka REST-serverov na Go. Chast' 1: standartnaya biblioteka [Elektronnyi resurs]. URL: <https://habr.com/ru/companies/ruvds/articles/559816/>.
 8. Xu A. Rate Limiting Fundamentals // ByteByteGo Newsletter. 2023 [Elektronnyi resurs]. URL: <https://blog.bytebytogo.com/p/rate-limiting-fundamentals> (data obrashcheniya: 18.03.2025).
 9. What is Fail2Ban with Setup & Configuration? (Detailed Guide) [Elektronnyi resurs]. URL: <https://runcloud.io/blog/what-is-fail2ban> (data obrashcheniya: 18.03.2025).
 10. What Is a WAF? | Web Application Firewall Explained [Elektronnyi resurs]. URL: <https://www.paloaltonetworks.com/cyberpedia/what-is-a-web-application-firewall> (data obrashcheniya: 18.03.2025).
 11. Tawde S. What is Cloudflare? // EducbaBlog. 2023 [Elektronnyi resurs]. URL: <https://www.educba.com/what-is-cloudflare/> (data obrashcheniya: 18.03.2025).
 12. Dheer P. Understanding Google Cloud Armor: Protect against denial of service and web attacks // TestPrepTraining. 2020 [Elektronnyi resurs]. URL: <https://www.testpreptraining.com/blog/understanding-google-cloud-armor-protect-against-denial-of-service-and-web-attacks/>.

Analysis of Spatiotemporal Motion Patterns in Aerial Images Using Optical Flow

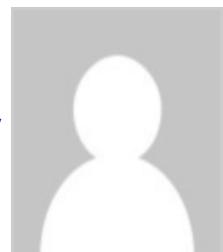
Rodionov Dmitrii

Doctor of Economics

Director; Higher School of Engineering and Economics; Peter the Great St. Petersburg Polytechnic University

195251, Russia, St. Petersburg, Kalininsky district, Politehnicheskaya str., 29 letter B

✉ rodion_dm@mail.ru



Sergeev Dmitrii Anatol'evich

PhD in Economics

Associate Professor; Higher School of Engineering and Economics; Peter the Great St. Petersburg Polytechnic University
 Director; National Center for Quality and Expertise; Peter the Great St. Petersburg Polytechnic University

195251, Russia, St. Petersburg, Kalininsky district, Politehnicheskaya str., 29 letter B

✉ sergeev_da@spbstu.ru

Konnikov Egenii Aleksandrovich

PhD in Economics

Associate Professor; Higher School of Engineering and Economics; Peter the Great St. Petersburg Polytechnic University

195251, Russia, St. Petersburg, Kalininsky district, Politehnicheskaya str., 29 letter B

✉ konnikov_ea@spbstu.ru

Pashinina Polina Aleksandrovna

Assistant Professor; Higher School of Engineering and Economics; Peter the Great St. Petersburg Polytechnic University

195251, Russia, St. Petersburg, Kalininsky district, Politehnicheskaya str., 29 letter B

✉ pashinina_pa@spbstu.ru

Abstract. This study focuses on the analysis of spatiotemporal motion patterns in aerial imagery using the optical flow method. With the advancement of remote sensing technologies and the widespread use of unmanned aerial vehicles (UAVs), the need for accurate and automated analysis of natural and anthropogenic dynamics is increasing. The work emphasizes the detailed examination of motion direction and intensity in high-resolution images. Existing methods of optical flow estimation are considered, including classical approaches such as the Lucas-Kanade and Horn-Schunck methods, as well as dense optical flow calculated using the Farnebäck method. The latter is applied as a core technique for constructing velocity vector fields, which serve as the basis for segment-wise motion distribution analysis, direction visualization, and heatmap generation. The proposed approach enables the identification of structural patterns and local motion features, which is particularly relevant for infrastructure monitoring and environmental risk assessment. It is also demonstrated that median motion estimates are more robust to noise and local outliers than mean values, thus improving analysis reliability. The research method is based on the calculation of dense optical flow using the Farnebäck algorithm, followed by statistical analysis of motion velocity characteristics and directional patterns across image segments. The scientific novelty of this study lies in the development of a comprehensive approach for analyzing spatiotemporal motion characteristics in aerial images using dense optical flow computed via the Farnebäck method. Unlike traditional techniques focused on global motion estimation, the proposed methodology emphasizes local patterns, enabling detailed segment-based evaluation of motion direction and intensity. For the first time, this study integrates both quantitative and visual analysis methods: histograms, heatmaps, calculations of median and mean velocities, and metrics such as structural similarity index (SSIM) and mean squared error (MSE) between image segments. This approach allows for detecting motion anomalies, identifying highly dynamic regions, and assessing structural stability. The method is tailored to UAV imagery and does not require large training datasets, making it suitable for low-resource environments. The results have practical relevance for automated infrastructure monitoring and environmental risk assessment.

Keywords: image processing, Farnebäck algorithm, risk management, data visualization, spatial modeling, structural similarity, segment analysis, aerial imaging, UAV, Optical flow

References (transliterated)

1. Gibson, Dzh. Dzh. Vospriyatie vizual'nogo mira / Dzh. Dzh. Gibson. – Boston: Khoughton Mifflin, 1950. – 235 s.
2. Lukas, B. D., Kanade, T. Iterativnaya tekhnika registratsii izobrazhenii s primeneniem k stereoizobrazheniyam. V: Trudy 7-i Mezhdunarodnoi sovmestnoi konferentsii po iskusstvennomu intellektu; 1981. S. 674-679.
3. Farnebak, G. Otsenka dvizheniya po dvum kadram na osnove polinomial'nogo razlozheniya. V: Trudy 13-i Skandinavskoi konferentsii po analizu izobrazhenii (SCIA); 2003. S. 363-370. DOI: 10.1007/3-540-45103-X_50.
4. San, D., Rot, S., Blek, M. Dzh. Sekrety otsenki opticheskogo potoka i ikh printsipy. V: Konferentsiya IEEE po komp'yuternomu zreniyu i raspoznavaniyu obrazov (CVPR); 2010. S. 2432-2439. DOI: 10.1109/CVPR.2010.5539939.
5. Dosovitskii, A., Fisher, P., Il'g, E., Khausser, P., Khazirbas, Ch., Gol'kov, V., Smakt, P., Kremers, D., Broks, T. FlowNet: Obuchenie opticheskому потоку с помошью сверточных сетей. V: Mezhdunarodnaya konferentsiya IEEE po komp'yuternomu zreniyu (ICCV); 2015. S. 2758-2766. DOI: 10.1109/ICCV.2015.316. EDN: YDBQRF.
6. Broks, T., Bruhn, A., Papenberg, N., Vaikett, Dzh. Vysokotochnaya otsenka opticheskogo potoka na osnove teorii iskazhenii. V: Evropeiskaya konferentsiya po komp'yuternomu zreniyu (ECCV); 2004. S. 25-36. DOI: 10.1007/978-3-540-24673-2_3.
7. Adil', O., Makhraz, M. A., Riffi, Dzh., Tairi, Kh. Dostizheniya v otsenke opticheskogo potoka na osnove glubokogo obucheniya: Vsestoronnii obzor modelei i tekhnologii. V: Shestaya mezhdunarodnaya konferentsiya po intellektual'nym vychisleniyam v naukakh o dannykh (ICDS); 2024 okt. S. 1-7. IEEE.
8. Khrichanyuk, O., Svitenco, M., Krykhtin, Yu., Akhafonov, Yu., Avilov, A., Kapashin, M. Programmno-apparatnyi kompleks dlya avtomaticheskoi geoteggirovaniya fotografii, sdelannykh s bespilotnogo letatel'nogo apparata. V: 7-ya mezhdunarodnaya konferentsiya IEEE po aktual'nym problemam razrabotki bespilotnykh letatel'nykh apparatov (APUAVD); 2024 okt. S. 276-279. IEEE.
9. Khorn, B. K. P., Shunk, B. G. Opredelenie opticheskogo potoka. Iskusstvennyi intellekt. 1981; 17(1-3): 185-203. DOI: 10.1016/0004-3702(81)90024-2.
10. Pavlenko, B. V., Pikalev, Ya. S. Metodika sozdaniya nabora aerofotosnimkov dlya zadachi perekrestnoi geolokalizatsii // Problemy iskusstvennogo intellekta. – 2024. – № 4(35). – S. 101-112. – DOI 10.24412/2413-7383-2024-4-101-112. EDN: DZMGUB.
11. Galiverov, A. N., Gorodnichaya, A. N. Osobennosti ispol'zovaniya BPLA pri effektivnom sbore aerofotosnimkov v gradostroitel'stve // Innovatsii. Nauka. Obrazovanie. – 2022. – № 50. – S. 1822-1826. EDN: BKJZFH.
12. Dyukarev, E. A., Voropai, N. N., Makarov, S. A. Prostranstvennoe razreshenie aerofotosnimkov BPLA. V: Primenie bespilotnykh letatel'nykh apparatov v geograficheskikh issledovaniyakh: Materialy Vserossiiskoi nauchno-prakticheskoi konferentsii, Irkutsk, 22-23 maya 2018 goda / Otvetstvennyi redaktor S.A. Makarov. – Irkutsk: Institut geografii im. V.B. Sochavy Sibirskogo otdeleniya Rossiiskoi akademii nauk, 2018. – S. 64-66. EDN: YAVELR.

Method of UAV Aerial Image Analysis Based on SSIM and MSE

for Assessing the Reliability of Technical Systems

Rodionov Dmitrii

Doctor of Economics

Director; Higher School of Engineering and Economics; Peter the Great St. Petersburg Polytechnic University

195251, Russia, St. Petersburg, Kalininsky district, Politehnicheskaya str., 29 letter B

 rodion_dm@mail.ru

Sergeev Dmitrii Anatol'evich

PhD in Economics

Associate Professor; Higher School of Engineering and Economics; Peter the Great St. Petersburg Polytechnic University

Director; National Center for Quality and Expertise; Peter the Great St. Petersburg Polytechnic University

195251, Russia, St. Petersburg, Kalininsky district, Politehnicheskaya str., 29 letter B

 sergeev_da@spbstu.ru

Konnikov Evgenii Aleksandrovich

PhD in Economics

Associate Professor; Higher School of Engineering and Economics; Peter the Great St. Petersburg Polytechnic University

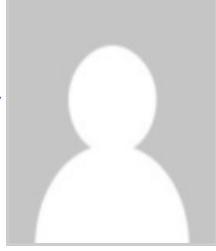
195251, Russia, St. Petersburg, Kalininsky district, Politehnicheskaya str., 29 letter B

 konnikov_ea@spbstu.ru

Popova Sofiya Dmitrievna

Assistant Professor; Higher School of Engineering and Economics; Peter the Great St. Petersburg Polytechnic University

195251, Russia, St. Petersburg, Kalininsky district, Politehnicheskaya str., 29 letter B

 popova_s@spbstu.ru

Abstract. This article presents an automated method for analyzing aerial images from unmanned aerial vehicles (UAVs), aimed at improving the reliability of technical systems and tracking changes in natural and anthropogenic processes. The objective of this work is to develop an algorithm that ensures accurate detection of anomalies and prediction of potential failure threats based on image processing. The methodology involves the application of the Structural Similarity Index (SSIM) and Mean Squared Error (MSE) for assessing spatial variations between adjacent segments of the imagery. The proposed approach is characterized by high stability to changes in illumination, low computational costs, and the possibility of integration into autonomous UAV systems. This work is based on computer modeling and statistical analysis of anomaly detection accuracy. The algorithm was tested on various datasets of aerial images using machine vision techniques and mathematical statistics to evaluate the effectiveness of the proposed method. The results include the development and validation of the algorithm, the construction of SSIM and MSE heatmaps, as well as the evaluation of the accuracy and reliability of the method. The obtained data confirm its effectiveness in automated monitoring of infrastructure facilities and the assessment of environmental risks. The scope of application of the developed method encompasses automated surveillance of engineering structures, monitoring the condition of agricultural lands, analyzing the consequences of natural disasters, and environmental control. The method can be integrated into intelligent control systems for the reliability of technical objects. In conclusion, the developed algorithm significantly enhances the accuracy of anomaly

detection, minimizes the influence of external factors, and automates the aerial image processing workflow. Its application contributes to improving the reliability of technical systems and reducing the probability of failures through the early identification of potential threats. Scientific Novelty: The scientific novelty lies in the development of a new method for assessing spatial variations based on a combination of the Structural Similarity Index (SSIM) and Mean Squared Error (MSE), which provides high accuracy in anomaly detection. In contrast to traditional image analysis methods, the proposed algorithm is characterized by robustness to changing imaging conditions, and its computational efficiency allows for real-time application. Furthermore, the method can be integrated into autonomous monitoring systems, expanding the capabilities of intelligent data analysis from UAVs. The obtained results and proposed solutions can be used to improve technologies for automated condition monitoring of objects and analysis of the dynamics of natural processes.

Keywords: failure risk prediction, automated image analysis, Technical system reliability, Change detection, Spatially invariant matching, MSE, SSIM, Aerial images, UAV, intelligent control systems

References (transliterated)

1. Buchaev M.A., Starchenkova O.D., Konnikov E.A. Analiz i klasterizatsiya nedostovernoi novostnoi informatsii kak instrument obespecheniya informatsionnoi bezopasnosti regiona // Myagkie izmereniya i vychisleniya. 2024. № 84 (11-2). S. 37-51. DOI 10.36871/2618-9976.2024.11-2.004. EDN: PFWBDV.
2. Dai K., Ma C., Wang Z., Long Y., Li X., Feng S., Ye Y. Exploiting spatial-temporal dynamics for satellite image sequence prediction // IEEE Geoscience and Remote Sensing Letters. 2023. T. 20. S. 1-5.
3. He X., Chen Y. Transferring CNN ensemble for hyperspectral image classification // IEEE Geoscience and Remote Sensing Letters. 2020. T. 18. № 5. S. 876-880.
4. Veretelnikova E.L., Elantseva I.L. Selection of factor for root mean square minimum error criterion // 2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE). 2016. T. 2. S. 221-223. IEEE. DOI 10.1109/APEIE.2016.7806454. EDN: WZWUYB.
5. Mottaeva A.B., Smirnova I.A., Konnikov E.A., Sharafanova E.E. Nechetko-mnozhestvennyi podkhod k otsenke transregional'noi promyshlennoi kogerentnosti // Myagkie izmereniya i vychisleniya. 2024. № 76 (3). S. 35-44. DOI 10.36871/2618-9976.2024.03.003. EDN: EGIQUM.
6. Raju K.N., Reddy K.S.P. Comparative study of Structural Similarity Index (SSIM) by using different edge detection approaches on live video frames for different color models // 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT). 2017. S. 932-937. IEEE.
7. Coppin P., Jonckheere I., Nackaerts K., Muys B., Lambin E. Digital change detection methods in ecosystem monitoring // International Journal of Remote Sensing. 2004. T. 25. № 9. S. 1565-1596.
8. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. Image quality assessment: From error visibility to structural similarity // IEEE Transactions on Image Processing. 2004. T. 13. № 4. S. 600-612.
9. Rodionov D. G., Konnikov E. A., Pashinina P. A., Shanygin S. I. Tematicheskoe modelirovanie informatsionnoi sredy mediakompanii: instrumental'nyi kompleks LDA-TF-IDF // Myagkie izmereniya i vychisleniya. 2024. T. 76, № 3. S. 72-84. DOI

10.36871/2618-9976.2024.03.006. EDN: COCJYG.

10. Dai K., Ma C., Wang Z., Long Y., Li X., Feng S., Ye Y. Exploiting spatial-temporal dynamics for satellite image sequence prediction // IEEE Geoscience and Remote Sensing Letters. 2023. T. 20. S. 1-5.
11. Volkov O. A., D'yakova G. N. Razrabotka matematicheskogo appara interval'noi otsenki veroyatnostnykh pokazatelei nadezhnosti tekhnicheskikh sistem s posledovatel'no-parallel'noi strukturoi metodom approksimatsii fidutsial'nykh raspredelenii v klasse beta-raspredelenii // Modelirovaniye i situatsionnoe upravlenie kachestvom slozhnykh sistem: Sbornik dokladov Nauchnoi sessii GUAP, Sankt-Peterburg, 08-12 aprelya 2019 goda. Sankt-Peterburg: Sankt-Peterburgskii gosudarstvennyi universitet aerokosmicheskogo priborostroeniya, 2019. S. 27-34. EDN: IUCRKY.
12. Mottaeva A.B., Smirnova I.A., Konnikov E.A., Shanygin S.I. Postroenie algoritmiki analiza ekonomiceskoi rezul'tativnosti transregional'nogo promyshlennogo klastera v kontekste tsikla priashcheniya urovnya ustochivogo razvitiya s primeneniem instrumentariya sobtyinogo modelirovaniya // Myagkie izmereniya i vychisleniya. 2024. № 77(4). S. 28-38. DOI 10.36871/2618-9976.2024.04.004. EDN: EQCIDA.
13. Stanin A.A., Mityazov V.A., Sorokin V.I., Konnikov E.A. Modelirovaniye urovnya nesostoyatel'nosti regionov na osnove algoritma otsenki risk-predraspolozhennosti // Myagkie izmereniya i vychisleniya. 2024. № 84 (11-1). S. 35-49. DOI 10.36871/2618-9976.2024.11.003. EDN: ZUAGEG.