

ТРУДЫ

**ИНСТИТУТА СИСТЕМНОГО
ПРОГРАММИРОВАНИЯ РАН**

**PROCEEDINGS OF THE INSTITUTE
FOR SYSTEM PROGRAMMING OF THE RAS**

ISSN Print 2079-8156
Том 37 Выпуск 3

ISSN Online 2220-6426
Volume 37 Issue 3

Институт системного
программирования
им. В.П. Иванникова РАН

Москва, 2025

ИСП **РАН**

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

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

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

Журнал издается по одному тому в год, шесть выпусков в каждом томе.

Поддерживается открытый доступ к содержанию издания, обеспечивая доступность результатов исследований для общественности и поддерживая глобальный обмен знаниями.

Труды ИСП РАН реферируются и/или индексируются в:

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

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

The journal is published one volume per year, six issues in each volume.

Open access to the journal content allows to provide public access to the research results and to support global exchange of knowledge. **Proceedings of ISP RAS** is abstracted and/or indexed in:



Редколлегия

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

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

Члены редколлегии

[Воронков Андрей Анатольевич](#), доктор физико-математических наук, профессор, Университет Манчестера (Манчестер, Великобритания)

[Вирбицкайте Ирина Бонавентуровна](#), профессор, доктор физико-математических наук, Институт систем информатики им. академика А.П. Ершова СО РАН (Новосибирск, Россия)

[Коннов Игорь Владимирович](#), кандидат физико-математических наук, Технический университет Вены (Вена, Австрия)

[Ластовенский Алексей Леонидович](#), доктор физико-математических наук, профессор, Университет Дублина (Дублин, Ирландия)

[Ломазова Ирина Александровна](#), доктор физико-математических наук, профессор, Национальный исследовательский университет «Высшая школа экономики» (Москва, Российская Федерация)

[Новиков Борис Асенович](#), доктор физико-математических наук, профессор, Санкт-Петербургский государственный университет (Санкт-Петербург, Россия)

[Петренко Александр Федорович](#), доктор наук, Исследовательский институт Монреаля (Монреаль, Канада)

[Черных Андрей](#), доктор физико-математических наук, профессор, Научно-исследовательский центр CICESE (Энсенада, Баха Калифорния, Мексика)

[Шустер Ассаф](#), доктор физико-математических наук, профессор, Технион — Израильский технологический институт Technion (Хайфа, Израиль)

Адрес: 109004, г. Москва, ул. А. Солженицына, дом 25.

Телефон: +7(495) 912-44-25

E-mail: info-isp@ispras.ru

Сайт: <http://www.ispras.ru/proceedings/>

Editorial Board

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

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

Editorial Members

[Igor Konnov](#), PhD (Phys.–Math.), Vienna University of Technology (Vienna, Austria)

[Alexey Lastovetsky](#), Dr. Sci. (Phys.–Math.), Professor, UCD School of Computer Science and Informatics (Dublin, Ireland)

[Irina A. Lomazova](#), Dr. Sci. (Phys.–Math.), Professor, National Research University Higher School of Economics (Moscow, Russian Federation)

[Boris A. Novikov](#), Dr. Sci. (Phys.–Math.), Professor, St. Petersburg University (St. Petersburg, Russian Federation)

[Alexandre F. Petrenko](#), PhD, Computer Research Institute of Montreal (Montreal, Canada)

[Assaf Schuster](#), Ph.D., Professor, Technion - Israel Institute of Technology (Haifa, Israel)

[Andrei Tchernykh](#), Dr. Sci., Professor, CICESE Research Centre (Ensenada, Baja California, Mexico).

[Irina B. Virbitskaite](#), Dr. Sci. (Phys.–Math.), The A.P. Ershov Institute of Informatics Systems, Siberian Branch of the RAS (Novosibirsk, Russian Federation)

[Andrew Voronkov](#), Dr. Sci. (Phys.–Math.), Professor, University of Manchester (Manchester, United Kingdom)

Address: 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Tel: +7(495) 912-44-25

E-mail: info-isp@ispras.ru

Web: <http://www.ispras.ru/en/proceedings>

С о д е р ж а н и е

Предисловие.....	7
Применение нейросетей для оценки разводимости при проектировании СБИС с учетом начальных параметров топологии. <i>Сайбодалов М.Х., Дашиев М.В., Карандашев Я.М., Желудков Н.В., Кочева Е.С.</i>	9
Контроль комбинационных схем по двум диагностическим признакам на основе логической коррекции сигналов с использованием свойств кодов Хэмминга. <i>Ефанов Д.В.</i>	19
Оптимизация конфиденциальных запросов к базе данных на облаке. <i>Варновский Н.П., Мартишин С.А., Храпченко М.В., Шокуров А.В.</i>	39
Моделирование сценариев деструктивного воздействия на целостность моделей машинного обучения. <i>Менисов А.Б., Ломако А.Г.</i>	59
Анализ и разработка методов очищения для защит метрик качества изображений. <i>Гуцин А.Е., Анциферова А.В., Ватолин Д.С.</i>	69
Нейросетевой метод для стабильного во времени матирования видеопоследовательностей с людьми. <i>Молодецких И.А., Ерофеев М.В., Москаленко А.В., Ватолин Д.С.</i>	85
Исследование специальных наборов модулей системы остаточных классов. <i>Луценко В.В., Кравцов М.Д., Горлачев Д.Е., Мирный Н.М.</i>	107
Макромоделирование компонентов защиты от электростатического разряда с применением программного обеспечения с открытым исходным кодом. <i>Кузнецов В.В., Андреев В.В., Ломакин С.А.</i>	121
Опыт отраслевого применения программного обеспечения с открытым исходным кодом при аэродинамическом проектировании и проверочном моделировании обтекания летательных аппаратов. <i>Павлевич А.Л.</i>	131
BERTScore для русского языка. <i>Бручес Е.П., Батунова Д.Т., Бондаренко И.Ю.</i>	147
Расширенная классификация текста с помощью DistilBERT с адаптацией LoRa: сравнительное исследование. <i>Абодо Элунду Б.Д., Цюаньюй В.</i>	159
SLAVA: бенчмарк социально-политического ландшафта и ценностного анализа. <i>Четвергов А.С., Шарафетдинов Р.С., Полукошко М.М., Ахметов В.А., Оружейникова Н.А., Аничков Е.С., Алексеевская И.С., Боловцов С.В., Голосов П.Е.</i>	171

Разработка системы оценки результатов тестирования когнитивной функции при введении эллипсисов в текстах планиметрических задач. <i>Найдёнова К.А., Булыкина Е.С., Мартирова Т.А.</i>	185
Глоттохронологическая классификация современных и наиболее ранних словарей самодийских языков на платформе ЛингвоДок. <i>Норманская Ю.В.</i>	195
Компьютерные экспериментальные методы изучения английского языка австралийских аборигенов. <i>Тимофеева Е.О.</i>	211
Распределение показателей двойственного числа существительных в селькупском языке (корпусно-ориентированный подход). <i>Ковылин С.В.</i>	225
Экстрактивная суммаризация с использованием скрытой структуры. <i>Тихонов П.А., Янина А.О., Малых В.А.</i>	237
Обзор методов контроля перегрузки с использованием машинного обучения. <i>Степанов И.А., Попов М.В., Гетьман А.И., Иконникова М.К., Белеванцев А.А.</i>	251
Методы динамической верификации промышленных средств защиты информации на основе формальных моделей управления доступом. <i>Петренко А.К., Десянин П.Н., Ефремов Д.В., Карнов А.А., Корныхин Е.В., Кулямин В.В., Хорошилов А.В.</i>	277
Поиск утечек памяти и ресурсов в статическом анализаторе Svace. <i>Малышев Н.Е., Бородин А.Е., Белеванцев А.А., Семенов В.А.</i>	291
Тестирование в жизненном цикле автоматизированных систем. <i>Позин Б.А.</i>	303
Оптимизации генерации иерархических уровней детализации для масштабных полигональных сцен <i>Шуткин В.Н., Морозкин Н.К., Семенов В.А., Тарлапан О.А.</i>	311
Обзор механизмов усиления защищенности операционных систем и пользовательских приложений. <i>Ефремов Д.В., Петренко А.К., Позин Б.А., Семенов В.А.</i>	325

Table of Contents

Preface	7
Application of Neural Networks for Routing Congestion Prediction in VLSI Design Using Initial Layout Parameters. <i>Saibodalov M.K., Dashiev M.V., Karandashev I.M., Zheludkov N.V., Kocheva E.S.</i>	9
Control of Combinational Circuits by Two Diagnostic Criteria Based on Boolean Signal Correction Using the Hamming Codes Properties. <i>Efanov D.V.</i>	19
Optimizing confidential database queries on the cloud. <i>Varnovskiy N.P., Martishin S.A., Khrapchenko M.V., Shokurov A.V.</i>	39
Modeling scenarios of destructive impact on the integrity of machine learning models. <i>Menisov A.B., Lomako A.G.</i>	59
Adversarial purification for no-reference image-quality metrics: applicability study and new methods. <i>Gushchin A.E., Antsiferova A.V., Vatolin D.S.</i>	69
Temporally coherent person matting trained on fake-motion dataset. <i>Molodetskikh I.A., Erofeev M.V., Moskalenko A.V., Vatolin D.S.</i>	85
Research of special sets of moduli of the residue number system. <i>Lutsenko V.V., Kravtsov M.D., Gorlachev D.E., Mirny N.M.</i>	107
Electrostatic discharge protection devices macromodelling using open-source tools. <i>Kuznetsov V.V., Andreev V.V., Lomakin S.A.</i>	121
Application of open-source software solving industrial tasks in aerodynamical design and external flow simulation of the aircrafts. <i>Pavlevich A.L.</i>	131
BERTScore for Russian. <i>Bruches E.P., Baturova D.T., Bondarenko I.Y.</i>	147
Enhanced Text Classification Using DistilBERT with Low-Rank Adaptation: A Comparative Study. <i>Abodo Eloundou B.D., Quanyu W.</i>	159
SLAVA: benchmark of Sociopolitical Landscape and Value Analysis. <i>Chetvergov A.S., Sharafetdinov R.S., Polukoshko M.M., Akhmetov V.A., Oruzheynikova N.A., Anichkov Ye.S., Alekseevskaya I.S., Bolovtsov S.V., Golosov P.E.</i>	171
Development of a system for evaluating cognitive function of introducing ellipses in the texts of planimetric problems. <i>Naidenova X.A., Bulykina E.S., Martirova T.A.</i>	185

Glottochronology classification of the modern and the earliest Samoyed dictionaries using LingvoDoc programs.
Normanskaja Ju.V......195

Computerised Experimental Methods of Studying Australian Aboriginal English.
Timofeeva E.O. 211

Distribution of the dual number markers of nouns in Selkup (corpus-oriented approach).
Kovylin S.V. 225

SumHiS: Extractive Summarization Exploiting Hidden Structure.
Tikhonov P.A., Ianina A.O., Malykh V.A. 237

Overview of congestion control methods using machine learning.
Stepanov I.A., Popov M.V., Getman A.I., Ikonnikova M.K., Belevantsev A.A. 251

Methods of dynamic verification of industrial information security tools based on formal access control models.
Petrenko A.K., Devyanin P.N., Efremov D.V., Karnov A.A., Kornyxhin E.V., Kuliamin V.V., Khoroshilov A.V. 277

Detecting Memory and Resource Leaks in the Svace static analyzer.
Malyshev N.E., Borodin A.E., Belevantsev A.A., Semenov V.A. 291

Testing in life cycle of automated systems.
Pozin. P.A. 303

Optimizations for hierarchical levels of detail generation on large-scale polygonal scenes.
Shutkin V. N., Morozkin N. K., Semenov V. A., Tarlapan O.A. 311

Overview of Hardening Mechanisms in Operating Systems and User Applications.
Efremov D.V., Petrenko A.K., Pozin B.A., Semenov V.A. 325

Предисловие

В третьем выпуске Трудов Института системного программирования представлены доклады участников Открытой конференции ИСП РАН (ISP RAS Open Conference - <https://www.isprasopen.ru/>), которая проходила в декабре 2024 года, а также новые статьи, которые поступили в редакцию в последнее время. Представленные доклады и статьи посвящены как традиционным технологиям системного программирования, так и новейшим подходам, связанным, в частности, с методами машинного обучения, нейросетевыми технологиями, большими языковыми моделями и другими методами искусственного интеллекта. Наряду с задачами системного программирования рассматриваются такие интересные приложения как разработка аппаратных средств защиты от электростатического разряда с применением программного обеспечения с открытым исходным кодом, проектирование интегральных схем на этапе размещения цифровых блоков с использованием нейросетевых моделей, разработка моделей деструктивного воздействия на целостность моделей машинного обучения на основе прогнозирования масштаба угроз и рисков при различных сценариях развития компьютерных атак, подходы к повышению производительности вычислительных систем на основе системы остаточных классов, исследования редких и исчезающих языков, изучение когнитивной функции, связанной с генерацией предложений в русском языке и другие темы на стыке лингвистики и информатики.

Preface

The third issue of the Trudy of the Institute for System Programming presents reports of the participants of the ISPRAS Open Conference (<https://www.isprasopen.ru/>), which was held in December 2024, as well as new articles that have been recently submitted to the editorial board. The presented reports and articles are devoted to both traditional technologies of system programming and the latest approaches related, in particular, to methods of machine learning, neural network technologies, large language models and other methods of artificial intelligence. Along with the tasks of system programming, such interesting applications as the development of hardware protection against electrostatic discharge using open source software, the design of integrated circuits at the stage of placing digital blocks using neural network models, the development of models of destructive impact on the integrity of machine learning models based on predicting the scale of threats and risks in various scenarios for the development of computer attacks, approaches to improving the performance of computing systems based on a system of residual classes, research into rare and endangered languages, the study of the cognitive function associated with the generation of sentences in the Russian language and other topics at the intersection of linguistics and computer science are considered.



Применение нейросетей для оценки разводимости при проектировании СБИС с учетом начальных параметров топологии

^{1,2} М.Х. Сайбодалов, ORCID: 0009-0003-0473-2161 <maratsaibodalov@gmail.com>

^{1,3} М.В. Дашиев, ORCID: 0009-0004-0309-7175 <dashiev@niisi.ras.ru>

^{1,2} Я.М. Карандашев, ORCID: 0000-0001-8483-072X <karandashev@niisi.ras.ru>

¹ Н.В. Желудков, ORCID: 0009-0009-0501-5686 <nvgel@cs.niisi.ras.ru>

¹ Е.С. Кочева, ORCID: 0009-0002-9587-0115 <kocheva@cs.niisi.ras.ru>

¹ НИЦ «Курчатовский институт», НИИСИ,

Россия, 117218, г. Москва, Нахимовский просп., 36, к.1.

² Российский Университет Дружбы Народов им. Патриса Лумумбы,

Россия, 117198, г. Москва, ул. Миклухо-Маклая, 6.

³ Московский физико-технический институт,

Россия, 141701, г. Долгопрудный, Институтский пер., 9.

Аннотация. В статье рассматривается задача оценки карты разводимости при проектировании интегральных схем на этапе размещения цифровых блоков с использованием нейросетевых моделей. Раннее прогнозирование разводимости позволит инженеру-проектировщику схем изменить план размещения, расположение макросов и портов ввода-вывода, чтобы предотвратить проблемы с маршрутизацией межсоединений на более поздних этапах, тем самым сократив количество запусков инструментов автоматизации электронного проектирования и общее время разработки схемы. В работе предлагается использовать начальные параметры топологии, которые не учитывались в предыдущих работах и позволяют моделям делать более точные прогнозы карты разводимости.

Ключевые слова: сверхбольшие интегральные схемы СБИС; карта разводимости; нейронные сети.

Для цитирования: Сайбодалов М.Х., Дашиев М.В., Карандашев Я.М., Желудков Н.В., Кочева Е.С. Применение нейросетей для оценки разводимости при проектировании СБИС с учетом начальных параметров топологии. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 9–18. DOI: 10.15514/ISPRAS-2025-37(3)-1.

Благодарности: Работа выполнена в рамках государственного задания НИЦ «Курчатовский институт» – НИИСИ по теме № FNEF-2024-0001 "Создание и реализация доверенных систем искусственного интеллекта, основанных на новых математических и алгоритмических методах, моделях быстрых вычислений, реализуемых на отечественных вычислительных системах" (1023032100070-3-1.2.1).

Application of Neural Networks for Routing Congestion Prediction in VLSI Design Using Initial Layout Parameters

^{1,2} M.K. Saibodalov, ORCID: 0009-0003-0473-2161 <maratsaibodalov@gmail.com>

^{1,3} M.V. Dashiev, ORCID: 0009-0004-0309-7175 <dashiev@niisi.ras.ru>

^{1,2} I.M. Karandashev, ORCID: 0000-0001-8483-072X <karandashev@niisi.ras.ru>

¹ N.V. Zheludkov, ORCID: 0009-0009-0501-5686 <nvgel@cs.niisi.ras.ru>

¹ E.S. Kocheva, ORCID: 0009-0002-9587-0115 <kocheva@cs.niisi.ras.ru>

¹ NRC «Kurchatov Institute» – SRISA,

bld. 1, 36, Nakhimovskiy pr., Moscow, 117218, Russia.

² The Patrice Lumumba Peoples' Friendship University of Russia,

6, Miklukho-Maklaya st., Moscow, 117198, Russia.

³ Moscow Institute of Physics and Technology,

9, Institutskiy per, Dolgoprudniy, 141701, Russia.

Abstract. This paper considers the problem of congestion map prediction at the pre-routing stage of VLSI layout design of digital blocks by applying neural network models. Early prediction of congestion will allow the VLSI design engineer to modify floorplan, macro placement and input-output port placement to prevent interconnect routing issues at later stages, thereby reducing the number of EDA tool runs and the overall circuit design runtime. In this work we propose the use of the initial layout parameters, which were not considered in previous works and allow for a more accurate congestion prediction.

Keywords: VLSI; congestion map; neural networks.

For citation: Saibodalov M.K., Dashiev M.V., Karandashev I.M., Zheludkov N.V., Kocheva E.S. Application of Neural Networks for Routing Congestion Prediction in VLSI Design Using Initial Layout Parameters. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 9-18 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-1.

Acknowledgements. The work was carried out within the framework of the state task of the NRC «Kurchatov Institute» – SRISA on the topic FNEF-2024-0001 «Development and deployment of trusted AI systems based on new mathematical and algorithmic approaches and fast computing models compatible with domestic computer hardware» (1023032100070-3-1.2.1).

1. Введение

В рамках топологического проектирования цифровых блоков СБИС (сверхбольших интегральных схем) важными критериями является быстрое достижение результата, соответствующего правилам проектирования DRC (Design Rule Check). Одним из ключевых этапов проектирования является детальная маршрутизация межсоединений в схеме. Это один из финальных и наиболее трудоемкие этапов, в результате которого могут возникать DRC-нарушения.

В зависимости от используемого технологического процесса может отличаться количество доступных для трассировки слоев металлизации и набор правил, согласно которым реализуется детальная маршрутизация. Для каждого слоя металлизации существует некая сетка, сформированная из вертикальных и горизонтальных треков, по которым должна осуществляться маршрутизация. Как правило, в процессе топологического проектирования, качество маршрутизации оценивается дважды: после глобальной маршрутизации, которая производится без жесткой привязки к конкретным трекам (после этапа размещения стандартных элементов), и после финальной детальной маршрутизации. Для данной оценки схема делится на некоторую сетку, высота ячейки которой кратна высоте стандартных ячеек. Для каждой ячейки оценивается отношение требуемых для разводки треков к имеющемуся при данном размещении количеству элементов. Соответственно, во избежание нарушений, количество требуемых треков не должно превышать доступное.

На ранних этапах проектирования существует целый ряд настроек, оказывающих непосредственное влияние на итоговую карту разводимости. На первом этапе – плане размещения, определяются: площадь схемы, от которой будет зависеть плотность расположения элементов (при слишком больших значениях плотности маршрутизация может оказаться вовсе невыполнимой); то, насколько близко расположены друг к другу макроблоки (при слишком плотном размещении могут возникать локальные зоны превышения допустимой разводимости); с какими настройками выполнено построение сетки земли/питания, в зависимости от чего определяется остаточное количество трексов, доступных для маршрутизации. Таким образом, в случае появления большого количества областей с превышением допустимой разводимости, необходимо варьирование параметров, определяемых в самом начале маршрута проектирования, что требует повторного прохождения всего цикла топологического проектирования, приводя к большим затратам по времени и вычислительным ресурсам.

В рамках данной работы мы хотим продемонстрировать применение машинного обучения для прогнозирования карты разводимости на основе начальных параметров топологии, минуя этап маршрутизации. Это позволит существенно сократить временные затраты и получить точную оценку карты разводимости для заданного нетлиста, который представляет собой список логических элементов и их межсоединений, описывающих работу определенной схемы на языке verilog. Мы использовали EDA-инструмент (Electronic design automation) с открытым исходным кодом OpenROAD [1], с помощью которого было произведено размещение ячеек, глобальная маршрутизация и извлечение фактической карты разводимости. На рис. 1 показано изображение карты разводимости, полученное с помощью OpenROAD. Более темные участки соответствуют областям с более низкими значениями разводимости.

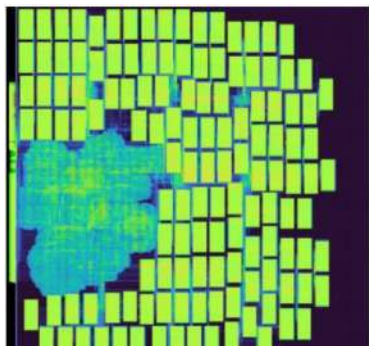


Рис. 1. Карта разводимости, полученная в OpenROAD.

Fig. 1. Congestion map in the OpenROAD EDA tool.

1.1. Смежные работы

Большинство работ по предсказанию разводимости связаны с графовыми нейронными сетями. Использование таких архитектур обусловлено тем, что нетлист содержит информацию о логических соединениях между ячейками. В более ранних работах [2] и [3] каждая ячейка представлялась как узел в графе, ее размер и количество соединений являлись признаками узла, а каждый net представлялся как ребро. Затем эти данные передавались в модели GAT (Graph Attention Network) [4] и GraphSAGE (sample and aggregate) [5] для получения предсказаний. В данных работах не использовалась информации о размещении ячеек, модели пытались предсказать значение разводимости на самом раннем из возможных этапов. В более современных работах [6] и [7] уже использовалась информация о размещении, так как она дает более подробную информацию о схеме, а также способствует улучшению конечного результата. В нашей более ранней работе [8] тоже использовались

графовые нейронные сети, но было предложено также использовать оценку RUDY (Rectangular Uniform Wire Density) [9] в качестве входной информации для модели. В данной работе мы предлагаем использовать алгоритмы компьютерного зрения для предсказания карты разводимости. Мы также будем использовать оценку, полученную с помощью алгоритма RUDY, но уже в формате изображения.

2. Данные

В качестве исходных данных мы используем 25 открытых блоков из набора данных openABC-D [10], которые представлены в виде RTL (Resistor-transistor level). Все эти схемы были запущены с различными параметрами и, таким образом, было получено 6700 образцов данных. Затем для каждого из этих образцов мы вычисляем оценку RUDY и карту плотности ячеек. На рис. 2 показаны примеры карты, полученной с помощью RUDY, а также карта плотности ячеек. Полученный набор данных делится на обучающую и валидационную выборку таким образом, чтобы 20 схем были в обучающей выборке и 5 в валидационной.

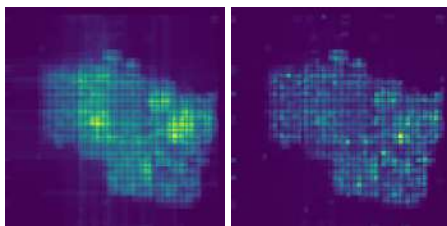


Рис. 2. Пример карты RUDY (слева) и карты плотности ячеек (справа)
Fig. 2. An example of RUDY map (on the l.h.s.) and cell density map (on the r.h.s.)

Основным нововведением данной работы является использование начальных параметров топологии, которые не учитывались в предыдущих работах по данной тематике. Параметры выбираются до стадий размещения ячеек и маршрутизации, и они имеют сильное влияние на итоговую карту разводимости. В данной работе мы используем 4 параметра:

- Начальная плотность заполнения (core utilization, CU) – считается как отношение площади ячеек в нетлисте к площади блока. Предполагается, что чем меньше значение CU, тем больше места для разводки и лучше разводимость. На рис. 3 показано влияние начальной плотности заполнения на итоговую карту разводимости.

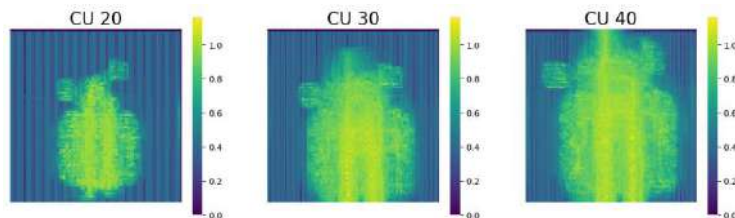


Рис. 3. Влияние начальной плотности заполнения на карту разводимости
(реальный размер у изображений разный, но поскольку работа идет со сверточными нейронными сетями, все изображения были уменьшены до одного размера).

Fig. 3. Core utilization influence on the final congestion map (in fact the size of the images differs, but since the convolutional neural networks are used, all the images were reduced to one size).

- Вид сетки земли-питания (power grid type) – варианты отличаются только разными расстояниями между шинами металлов. Вариант "0" предполагает самое маленькое расстояние, "4" - самое большое. Чем больше расстояние между шинами сетки, тем больше места остается под сигнальную разводку, тем лучше разводимость. То есть

для варианта "4" мы ожидаем увидеть лучшие значение. На рис. 4 показано влияние разных вариантов сетки земли-питания на итоговую карту разводимости.

- Верхний металл – доступный верхний металл для сигнальной разводки. Чем выше значение, тем лучше должна быть разводимость.
- Layer adjustment – параметр, запрещающий определенный процент трексов для разводки. Чем выше значения, тем больший процент трексов запрещен, однако, в некоторых ситуациях более высокие значения этого параметра помогают снизить пиковые значения разводимости.

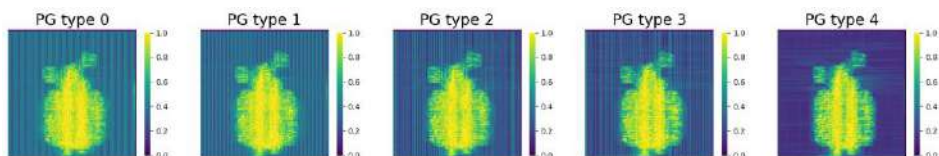


Рис. 4. Влияние вида сетки земли-питания на итоговую карту разводимости.

Fig. 4. Influence of power grid type on the final congestion map.

3. Предлагаемый подход

Основной архитектурой для данной задачи является U-Net [11]. На рис. 5 показана архитектура данной модели. Это одна из самых популярных архитектур сверточных нейронных сетей, изначально созданная для сегментации биомедицинских изображений. Несмотря на то, что этой архитектуре уже почти 10 лет, она по-прежнему показывает хорошие результаты во многих задачах компьютерного зрения и часто используется в качестве компонента более сложных архитектур. Популярность этой архитектуры заключается в ее простоте и эффективности. Эта сеть часто используется для различных задач "image-to-image", наша задача также относится к классу "image-to-image", поскольку мы хотим получить изображение карты разводимости, используя многоканальные изображения RUDY и плотности ячеек. В процессе разработки мы опробовали три разные модели: vanilla U-Net, Attention U-Net [12] и Pix2pix [13]. Pix2pix отличается от двух других моделей тем, что это генеративная состязательная сеть (GAN), которая состоит из генератора и дискриминатора. Как и в оригинальной статье, для генератора мы используем U-Net, а для дискриминатора - PatchGAN. Обученная модель использует только генератор.

Все модели принимают карту RUDY, карту плотности ячеек и начальные параметры топологии в качестве входных данных. Для RUDY и плотности ячеек мы используем функцию, которая представляет собой разделение всех значений карты RUDY и плотности ячеек на процентилях (0-50%, 50-75%, 75-90% и 90-100%) на 4 части (в нашем случае). Идея разделения заключается в том, чтобы помочь нашей модели, выделив области с более высокими и более низкими значениями RUDY и плотности ячеек, которые коррелируют с разводимостью. Количество частей, на которые мы делим исходное изображение, и процентиль, на который происходит разделение, являются гиперпараметрами. На рис. 6 показан пример разделения RUDY.

Поскольку архитектура модели U-Net предназначена для работы с изображениями и принимает двумерные матрицы в качестве входных данных, мы решили передать каждый параметр топологии в отдельном канале в виде двумерной матрицы, в которой каждая позиция заполнена одним и тем же значением (например, при использовании параметра начальной плотности заполнения (Core utilization) = 0,4 мы передаем матрицу ((0.4 0.4 ... 0.4), ..., (0.4 0.4 ... 0.4))). После разбиения на процентиля мы получаем 4 канала RUDY и 4 канала плотности ячеек, затем объединяем RUDY, плотность ячеек и параметры и передаем их на вход нашей модели. Финальный алгоритм показан на рис. 7.

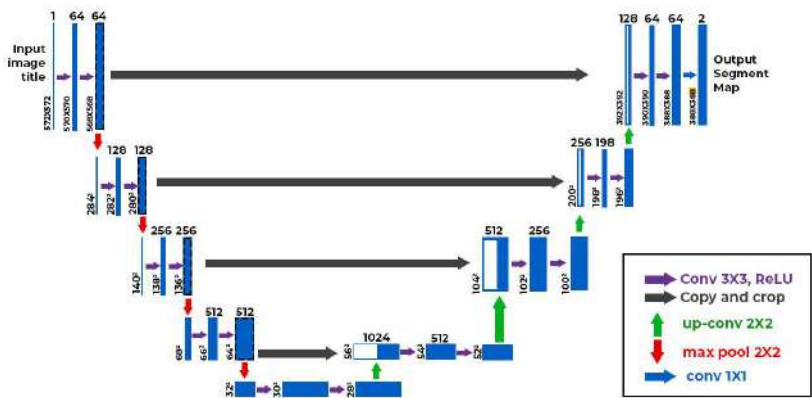


Рис. 5. Архитектура U-Net из оригинальной статьи [11].
Fig. 5. U-Net architecture from the original paper [11].

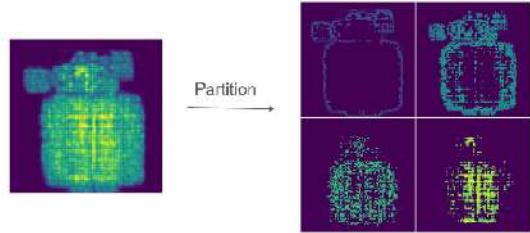


Рис. 6. Разделение RUDY – синий цвет означает меньшие значения, желтый – большие. Канал RUDY делится на 4 части по процентиям 0-50%, 50-75%, 75-90% и 90-100%.
Fig. 6. Partition of RUDY - blue means lower value, yellow means bigger. RUDY channel is divided into 4 parts by percentiles of 0-50%, 50-75%, 75-90% and 90-100%.

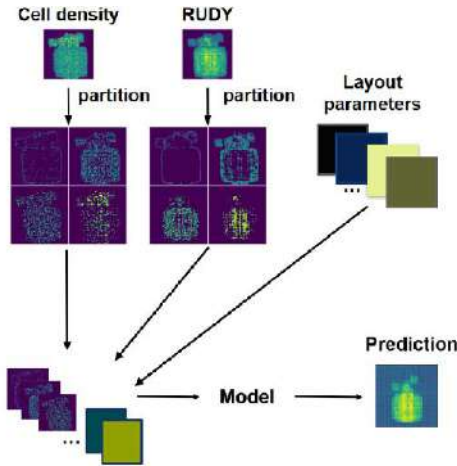


Рис. 7. Основные этапы нашего подхода.
Fig. 7. Main stages of our pipeline.

Так как мы работаем с изображениями, мы попытались использовать аугментацию данных, чтобы “расширить” набор данных. В задаче прогнозирования разводимости важно сохранить информацию обо всей структуре схемы, поэтому мы не можем использовать, например, добавление шума, поэтому были использованы повороты и зеркальное отражение, но это привело к ухудшению результатов. Мы связываем это с небольшим разнообразием схем, что

приводит к недостаточной обобщающей способности модели. Из-за этого аугментация данных только запутала модель.

Все модели обучались в течение 100 эпох, при этом карты RUDY и плотности ячеек были разделены с помощью разбиения на четыре части на основе процентилей: 50, 75 и 90 процентов. Предложенные алгоритмы обычно сходятся за 30 эпох, используется скорость обучения 2×10^{-4} и оптимизатор Adam с гиперпараметрами по умолчанию. Весь этот процесс был выполнен на графическом процессоре NVIDIA RTX 2070 Super.

3. Результаты

Наши результаты показаны на рис. 8. Как видно, модель правильно определяет многие участки с высокой разводимостью. Кроме того, наш подход не требует глобальной и детальной маршрутизации для прогнозирования разводимости по сравнению со стандартными подходами.

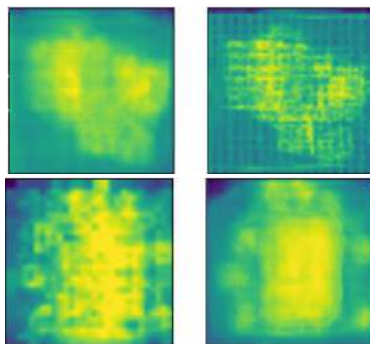


Рис. 8. Результаты модели Pix2pix. Слева предсказание модели, справа целевое изображение. Корреляции Пирсона и Кендалла для верхней пары 0.879 и 0.69, для нижней 0.875 и 0.673 соответственно.

Fig. 8. Pix2pix results, on the l.h.s is the generated image, on the r.h.s is target. Correlations on the top are 0.879 and 0.69, on the bottom 0.875 and 0.673 for Pearson and Kendall respectively.

Мы представляем новый подход к прогнозированию карты разводимости на основе начальных параметров топологии, которые сильно влияют на итоговую карту разводимости. Наш подход позволяет исключить трудоемкие этапы глобальной и детальной маршрутизации для прогнозирования разводимости. Хотя этот метод не предполагает полного отказа от инструментов EDA, поскольку используется информация о размещении, данный подход представляет собой переход к более эффективной методологии, основанной на данных. Эффективность моделей на различных схемах существенно различается и подчеркивает необходимость дальнейших исследований для повышения точности методов, основанных на нейронных сетях.

В целях подтверждения актуальности нашего подхода, мы проводим исследования по удалению компонентов модели. Основная цель этой статьи - продемонстрировать важность начальных параметров топологии в моделях прогнозирования разводимости. В табл. 1 можно увидеть результаты удаления различных компонентов из наших моделей. Значения корреляции Кендалла в таблице были рассчитаны путем усреднения по 5 различным итерациям обучения, а также представлены данные о стандартных отклонениях. Как вы можете видеть, начальные параметры топологии оказывают значительное влияние на результирующую корреляцию, их использование мы считаем очень важным нововведением. Кроме того, можно видеть, что использование разделения и карты плотности ячеек также улучшает результаты работы модели. В нашем случае использование разделения карт на 4 части приводит к лучшей корреляции, чем использование карты плотности ячеек, что

означает, что этот метод действительно помогает модели лучше прогнозировать разводимость. Все результаты, представленные в таблице, были получены без использования аугментации, поскольку обучение с аугментацией приводило к худшим результатам.

Основной проблемой является недостаток схем. Некоторые образцы значительно отличаются по значениям разводимости и количеству ячеек, что может отрицательно сказаться на результатах работы моделей. Несмотря на то, что наш набор данных содержит 6700 экземпляров данных, кажется, что 25 различных схем недостаточно для обучения модели с достаточными обобщающими способностями. Основным способом улучшения результатов является увеличение разнообразия схем. Это позволит использовать более глубокие модели, которые могут обеспечить лучшие результаты и обобщающую способность.

Табл. 1. Корреляции Пирсона и Кендалла.
Table 1. Kendall and Pearson correlations.

	UNet		Attention Unet		Pix2pix	
	Kendall	Pearson	Kendall	Pearson	Kendall	Pearson
w/o parameters	0.488 ± 0.015	0.708 ± 0.058	0.484 ± 0.016	0.652 ± 0.027	0.469 ± 0.024	0.643 ± 0.020
w/o partition	0.559 ± 0.046	0.815 ± 0.001	0.553 ± 0.007	0.809 ± 0.006	0.520 ± 0.056	0.713 ± 0.009
w/o density	0.575 ± 0.012	0.780 ± 0.014	0.572 ± 0.019	0.777 ± 0.013	0.572 ± 0.009	0.765 ± 0.025
ours	0.603 ± 0.010	0.810 ± 0.019	0.602 ± 0.008	0.812 ± 0.006	0.616 ± 0.007	0.834 ± 0.006

Список литературы / References

[1]. Ajayi et. al. OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain. Proc. GOMACTECH, 2019, pp. 1105–1110.

[2]. R. Kirby, S. Godil, R. Roy, and B. Catanzaro. CongestionNet: routing congestion prediction using deep graph neural networks. IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC), Cuzco, Peru, 2019, pp. 217–222.

[3]. A. Ghose, V. Zhang, Y. Zhang, D. Liu, W. Liu, and V. Coates. Generalizable cross-graph embedding for GNN-based congestion prediction. IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2021.

[4]. P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio. Graph attention networks. arXiv (Cornell University), 2018.

[5]. W. L. Hamilton, R. Ying, and J. Leskovec. Versatile multi-stage graph neural network for circuit representation. arXiv (Cornell University), 2017.

[6]. S. Yang, Z. Yang, D. Li, Y. Zhang, Z. Zhang, G. Song, and J. Hao. Inductive representation learning on large graphs. In Advances in Neural Information Processing Systems, 2022, vol. 35.

[7]. B. Wang, G. Shen, D. Liu, J. Hao, W. Liu, Y. Huang, H. Wu, Y. Lin, G. Chen, and P. Heng. LHNNet: Lattice hypergraph neural network for VLSI congestion prediction. arXiv (Cornell University), 2022.

[8]. M. Saibodalov, I. Karandashev, Z. Sokhova, E. Kocheva, and N. Zheludkov. Routing Congestion Prediction in VLSI Design Using Graph Neural Networks. DSPA 2024 26th International Conference on Digital Signal Processing and its Applications, Moscow, Russian Federation, 2024, pp. 1-4.

[9]. P. Spindler and F. M. Johannes. Fast and accurate routing demand estimation for efficient routability-driven placement. EDA Consortium Proceedings of the Conference on Design, Automation and Test in Europe, San Jose, CA, USA, 2007, pp. 1226–1231.

[10]. A. B. Chowdhury, B. Tan, R. Karri, and S. Garg. OpenABC-D: a large-scale dataset for machine learning guided integrated circuit synthesis. arXiv (Cornell University), 2021.

- [11]. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv (Cornell University), 2015.
- [12]. Ozan Oktay et al. Attention U-Net: Learning Where to Look for the Pancreas. arXiv (Cornell University), 2018.
- [13]. Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. arXiv (Cornell University), 2016.

Информация об авторах / Information about authors

Марат Халитович САЙБОДАЛОВ – аспирант РУДН, младший научный сотрудник НИЦ Курчатовский Институт – НИИСИ. Сфера научных интересов: графовые нейронные сети, автоматизация проектирования высокопроизводительных СБИС.

Marat Khalitovich SAIBODALOV – postgraduate student at RUDN University, junior researcher at SRISA. Research interests: graph neural networks, VLSI design automation.

Максим Вадимович ДАШИЕВ – инженер НИЦ Курчатовский Институт – НИИСИ, студент магистратуры МФТИ. Сфера научных интересов: разработка и применение методов машинного обучения и нейронных сетей для решения прикладных задач в различных областях.

Maxim Vadimovich DASHIEV – engineer at SRISA, master degree student at MIPT. Research interests: development of machine learning methods and neural networks to solve applied problems in various fields.

Яков Михайлович КАРАНДАШЕВ – кандидат физико-математических наук, ведущий научный сотрудник НИЦ Курчатовский Институт – НИИСИ, сфера научных интересов: компьютерное зрение, сегментация изображений.

Iakov Mikhailovich KARANDASHEV – Cand. Sci. (Phys.-Math.), leading researcher at SRISA. Research interests: computer vision, image segmentation.

Никита Владимирович ЖЕЛУДКОВ – младший научный сотрудник НИЦ Курчатовский Институт – НИИСИ, сфера научных интересов: автоматизация проектирования высокопроизводительных СБИС, применение методов машинного обучения для решения задач проектирования топологии интегральных схем, повышение сбоеустойчивости цифровых блоков СБИС.

Nikita Vladimirovich ZHELUDKOV – junior researcher at SRISA. Research interests: VLSI design automation, application of machine learning methods to solve integrated circuit topology design problems, increasing the stability of VLSI digital blocks.

Елизавета Сергеевна КОЧЕВА – инженер НИЦ Курчатовский Институт – НИИСИ, сфера научных интересов: топологическое проектирование СБИС, применение машинного обучения при проектировании цифровых блоков СБИС.

Elizaveta Sergeevna KOCHEVA – engineer at SRISA. Research interests: VLSI topological design, application of machine learning methods to solve integrated circuit topology design problems.



Контроль комбинационных схем по двум диагностическим признакам на основе логической коррекции сигналов с использованием свойств кодов Хэмминга

Д.В. Ефанов, ORCID: 0000-0002-4563-6411 <TrES-4b@yandex.ru>

Санкт-Петербургский политехнический университет Петра Великого,
Россия, 195251, г. Санкт-Петербург, ул. Политехническая, д. 29, литера Б.

Российский университет транспорта,
Россия, 127994, ГСП-4, г. Москва, ул. Образцова, д. 9, стр. 9.

Аннотация. Описывается метод синтеза самопроверяемых цифровых устройств с улучшенными показателями контролепригодности, основанный на синтезе схем встроенного контроля путем логической коррекции сигналов и применении кода Хэмминга (7, 4) с контролем вычислений по двум диагностическим признакам. В качестве признаков используется принадлежность кодовых слов коду (7, 4) и самодвойственность каждой функции, описывающей информационные и проверочные символы кода. Представлены «базовая» структура организации схемы встроенного контроля для семивыходного комбинационного устройства, в которой используются типовые блоки (кроме блока вычисления функций логической коррекции) и алгоритм синтеза блока вычисления функций логической коррекции, отвечающего условиям обеспечения самодвойственности формируемых сигналов и принадлежности кодовых слов коду (7, 4). Исследованы особенности применения «базовой» структуры организации схем встроенного контроля при контроле вычислений устройствами с числом выходов $n > 7$. С ростом n показатели сложности технической реализации типовых компонентов отдельных схем встроенного контроля уменьшаются в сравнении с традиционным методом дублирования, однако за счет роста сложности компаратора общие показатели сложности их технической реализации, наоборот, растут. Это приводит к уменьшению роста «запаса эффективности» по структурной избыточности предлагаемого метода с ростом n . Эффективность применения представленного метода по сравнению с дублированием может быть достигнута при существенном уменьшении показателей сложности отдельных блоков вычисления функций логической коррекции (с учетом возможностей по совместной оптимизации их структур). Предварительная оценка позволяет рекомендовать использование разработанного метода для частных случаев объектов диагностирования с небольшим числом выходов (не более 30). В каждом конкретном случае, однако, нужно проводить оценку эффективности по сравнению с дублированием. В сравнении с дублированием по показателю контролепригодности метод оказывается более выигрышным, поскольку дает возможность более просто обеспечивать формирование тестов для элементов схемы встроенного контроля, чем при использовании дублирования и позволяет достигать их формирования даже в тех случаях, когда это невозможно при дублировании. Предложенный метод синтеза самопроверяемых устройств может рассматриваться при проектировании высоконадежных цифровых систем на современной элементной базе.

Ключевые слова: самопроверяемая комбинационная схема; контроль по двум диагностическим признакам; коды Хэмминга с самодвойственными функциями, описывающими проверочные символы; логическая коррекция сигналов; структурная избыточность самопроверяемых схем.

Для цитирования: Ефанов Д.В. Контроль комбинационных схем по двум диагностическим признакам на основе логической коррекции сигналов с использованием свойств кодов Хэмминга. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 19–38. DOI: 10.15514/ISPRAS-2025-37(3)-2.

Control of Combinational Circuits by Two Diagnostic Criteria Based on Boolean Signal Correction Using the Hamming Codes Properties

D.V. Efanov, ORCID: 0000-0002-4563-6411<TrES-4b@yandex.ru>

*Peter the Great Saint Petersburg Polytechnic University,
litera B, 29, Politekhnikeskaya st., St. Petersburg, 195251, Russia.
Russian University of Transport,
building 9, 9, Obraztsova str. Moscow, GSP-4, 127994, Russia.*

Abstract. A method for synthesizing self-checking digital devices with improved testability indicators is described. The method is based on the concurrent error-detection circuit synthesis by signals Boolean correction and the Hamming code (7, 4) with control of calculations according to two diagnostic criteria. The attributes used are the belonging of code words to the code (7, 4) and the self-duality of each function describing the data and checking bits of the code. The concurrent error-detection circuit basic structure for a seven-output combinational device is giving. The structure uses standard blocks, apart from the Boolean correction function calculation block. An algorithm has been developed for the Boolean correction function calculation block synthesis that meets the conditions for ensuring the self-duality of the generated signals and the belonging of code words (7, 4) to code. The application features of the basic structure are studied. It is shown that as n increases, the technical implementation complexity of individual concurrent error-detection circuit standard components decreases in comparison with the traditional duplication method. However, due to the increase in the complexity of the comparator, the overall complexity indicators of their technical implementation are growing. This leads to a decrease in the growth of the “efficiency margin for structural redundancy” of the proposed method as n increases. Thus, the effectiveness of using the presented method compared to duplication can be achieved with a significant reduction in the complexity of individual Boolean correction function calculation blocks (considering the possibilities for joint optimization of their structures). A preliminary assessment allows to recommend the developed method for special cases of diagnostic objects with a small number of outputs (no more than 30). On a case-by-case basis, effectiveness versus duplication must be assessed. In comparison with duplication in terms of testability, the method turns out to be more advantageous, since it makes it possible to ensure the tests formation more easily for concurrent error-detection circuit elements than when using duplication and makes it possible to achieve their formation even in cases where this is impossible with duplication. The proposed method for synthesizing self-checking devices can be considered when designing highly reliable digital systems on a modern element base.

Keywords: self-checking combinational circuit; calculation control based on two diagnostic criteria; Hamming codes with self-dual complement describing check bits; Boolean signal correction; self-checking circuits structure redundancy.

For citation: Efanov D.V. Control of Combinational Circuits by Two Diagnostic Criteria Based on Boolean Signal Correction Using the Hamming Codes Properties. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025, pp. 19-38 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-2.

1. Введение

Применению классических кодов Хэмминга [1] при синтезе высоконадежных цифровых устройств и систем посвящено большое количество работ, например, работы [2-5].

Одной из особенностей классических кодов Хэмминга является то, что при числе разрядов в кодовых словах $n = m + k = 3 + 4l$, $l \in \mathbb{N}_0$, где m и k – число информационных и проверочных символов в кодовом слове, все их проверочные символы описываются самодвойственными булевыми функциями [6]. Это свойство позволяет использовать коды Хэмминга при организации контроля вычислений в цифровых устройствах по признаку самодвойственности сигналов в определенных контрольных точках. При этом одновременно можно учитывать и характеристики обнаружения кодами Хэмминга ошибок в кодовых словах [7, 8] и отдельно в информационных векторах, что актуально в тех приложениях, в которых информационные и проверочные символы формируются физически различными устройствами [9, 10]. Контроль вычислений в определенных точках логических схем по двум диагностическим признакам позволяет синтезировать самопроверяемые цифровые

устройства с высокими показателями контролепригодности [11].

Эффективным при синтезе самопроверяемых устройств является использование кодов, у которых проверочные символы описываются самодвойственными булевыми функциями. Это позволяет использовать логическую коррекцию сигналов (ЛКС) при синтезе СВК [12-15]. Использование ЛКС дает возможность сочетания двух диагностических признаков для их учета в схеме встроенного контроля (СВК). В [16] описано использование при синтезе СВК на основе ЛКС равновесных кодов «2 из 4» и показана принципиальная возможность применения в этих целях любых равновесных кодов « w из $2w$ », где w – вес кодового слова. В [17] в структуре СВК на основе ЛКС и применения равновесных кодов «2 из 4» добавлена схема сжатия сигналов от объекта диагностирования, которая может быть использована для уменьшения числа наблюдаемых выходов объекта диагностирования. Аналогично при синтезе СВК на основе ЛКС могут быть использованы и коды Хэмминга, проверочные символы которых описываются самодвойственными булевыми функциями.

Целью настоящей статьи является описание особенностей синтеза СВК на основе ЛКС и одного частного случая кодов Хэмминга – кода (7, 4), где первое число указывает на количество символов в кодовом слове, а второе – на число информационных среди них.

2. Контроль вычислений по двум диагностическим признакам

При организации СВК повсеместно применяют традиционную структуру. Принципы ее реализации основаны на дополнении информационного вектора, с которым отождествляют выходы контролируемого устройства, контрольным вектором таким образом, чтобы кодовое слово принадлежало заранее выбранному двоичному равномерному коду и, в том числе, коду Хэмминга [18, 19]. В этом случае информационные и проверочные символы кодовых слов вычисляются физически различными устройствами. Учитывая характеристики обнаружения ошибок в информационных символах при безошибочности проверочных символов, можно синтезировать полностью самопроверяемые структуры [20].

Недостатком традиционного подхода к синтезу СВК является то, что для выбранного равномерного кода при реализации СВК нет возможности гибкого влияния на характеристики конечного устройства. Это связано с однозначным соответствием всех информационных векторов, формируемых на каждой из входных комбинаций, контрольным векторам выбранного кода и, как следствие, с использованием только одной из форм представления булевых функций, описывающих блок вычисления функций логической коррекции, формирующий значения проверочных символов. Большую гибкость в части влияния на показатели эффективности СВК проектировщику самопроверяемого устройства дает подход, основанный на ЛКС [12, 13]. При этом, как отмечалось выше, можно не просто учитывать свойства обнаружения ошибок применяемыми при синтезе СВК кодами, но и особенности булевых функций, описывающих информационные и проверочные символы.

На рис. 1 изображена структура организации СВК на основе ЛКС с применением кодов Хэмминга (7, 4). Она подразумевает организацию контроля вычислений по двум диагностическим признакам: принадлежности формируемых в СВК кодовых слов коду (7, 4) и принадлежности каждой функции, описывающей информационные и проверочные символы, классу самодвойственных булевых функций. Естественно, это частный случай, при котором реализуется некоторая «базовая» структура для семивыходного исходного объекта диагностирования. Ее можно применять при синтезе СВК для многовыходных устройств с контролем вычислений по группам выходов.

Следует предварительно отметить, что структура, приведенная на рис. 1, отличается от исследованных ранее (см., например, [6]), тем, что организуется для исходных объектов с произвольными структурами (как самодвойственными, так и несамодвойственными) и подразумевает коррекцию выходных функций объекта диагностирования таким образом, чтобы на выходах блока коррекции сигналов формировались кодовые слова кода (7, 4). В

этом случае следует учитывать свойства обнаружения ошибок кодами Хэмминга во всех символах кодовых слов [7, 8], а не только в информационных, что актуально для традиционного подхода к синтезу СВК [9, 10].

СВК организована следующим образом. От объекта диагностирования $F(X)$ в СВК поступает семь сигналов с выходов $f_1(X)$, $f_2(X)$, ..., $f_7(X)$, где $\langle X \rangle = \langle x_t x_{t-1} \dots x_2 x_1 \rangle$ – наборы значений аргументов (входные комбинации). Они попадают в блок коррекции сигналов (БКС), образованный каскадом двухвходовых элементов XOR . Каждый сигнал от объекта диагностирования поступает на первые входы элементов преобразования в БКС. На вторые входы данных элементов поступают сигналы от блока вычисления функций логической коррекции $G(X)$, проектируемого таким образом, чтобы выходные сигналы с БКС $h_1(X)$, $h_2(X)$, ..., $h_7(X)$ позволяли на каждом входном воздействии на устройство с СВК формировать кодовые слова кода (7, 4). Такое преобразование возможно всегда и позволяет на каждой входной комбинации «настроить» формирование требуемого кодового слова на выходах БКС. При синтезе СВК по структуре, изображенной на рис. 1, как раз необходимо учитывать формирование определенных кодовых слов кода (7, 4).

Поскольку требуется контроль вычислений по двум диагностическим признакам, одним из которых является самодвойственность вычисляемых функций, для работы самопроверяемого устройства применяется импульсный режим. Он реализуется при использовании генератора прямоугольных импульсов G и каскада входных элементов XOR для устройства $F(X)$. В таком случае логический 0 и логическая 1 представляются в виде последовательностей прямоугольных импульсов: 0 – 0101...01, 1 – 1010...10. Комбинации же на входы системы подаются парами: прямое воздействие и воздействие, где все сигналы инвертированы (прямая и инверсная комбинации). Первое является рабочим, а второе – тестовым. Другими словами, если обозначить входную комбинацию соответствующим ей десятичным числом p , то ее парой будет число $2^t - 1 - p$, где $p \in \{0, 1, \dots, 2^t - 1\}$. При построении СВК учитывается

то, что на входы подается полное множество наборов входных комбинаций, которые разбиваются на 2^{t-1} пар, при поступлении каждой из которых на входы устройства требуется сформировать в СВК на выходе БКС такие кодовые слова кода (7, 4), которые будут отличаться во всех разрядах (также прямые и инверсные кодовые слова).

Для проверки корректности сигналов в процессе эксплуатации устройства используется специальный модуль контроля вычислений (МКВ). Он содержит подсхему контроля принадлежности формируемых кодовых слов коду (7, 4) и подсхему контроля самодвойственности каждой функции. Для контроля принадлежности кодовых слов коду (7, 4) устанавливаются каскад инверторов сигналов, которые соответствуют проверочным символам кода (сигналы $h_1(X)$, $h_2(X)$, $h_3(X)$), а также тестер, включающий в себя кодер $G(F)$, где $\langle F \rangle = \langle f_7(X) f_6(X) \dots f_1(X) \rangle$, и компаратор для трех пар сигналов – $3TRC1$, образованный двумя элементарными модулями сжатия парафазных сигналов TRC [21]. Для контроля вычислений по признаку самодвойственности устанавливается блок $7SDC1$ – тестер самодвойственности для семи самодвойственных сигналов. Он строится на основе схемы сжатия самодвойственных сигналов (см. рис. 4.5 и 4.8 в [22]) и одного тестера самодвойственности. Выходы обеих подсхем контроля объединяются на входах одного модуля TRC , что обеспечивает наблюдение одного контрольного сигнала.

Добавим здесь, что инверторы, установленные на верхних входах $3TRC1$, могут быть «опущены» на выходы $g_1(X)$, $g_2(X)$, $g_3(X)$ блока $G(X)$. То есть, он может быть сразу синтезирован таким образом, чтобы реализовывались инверсные значения тем, которые требуется записывать в разряды контрольных векторов кода (7, 4).

В структуре, данной на рис. 1, все элементы СВК, кроме блока $G(X)$, являются типовыми и имеют единственную реализацию в выбранном элементном базисе.

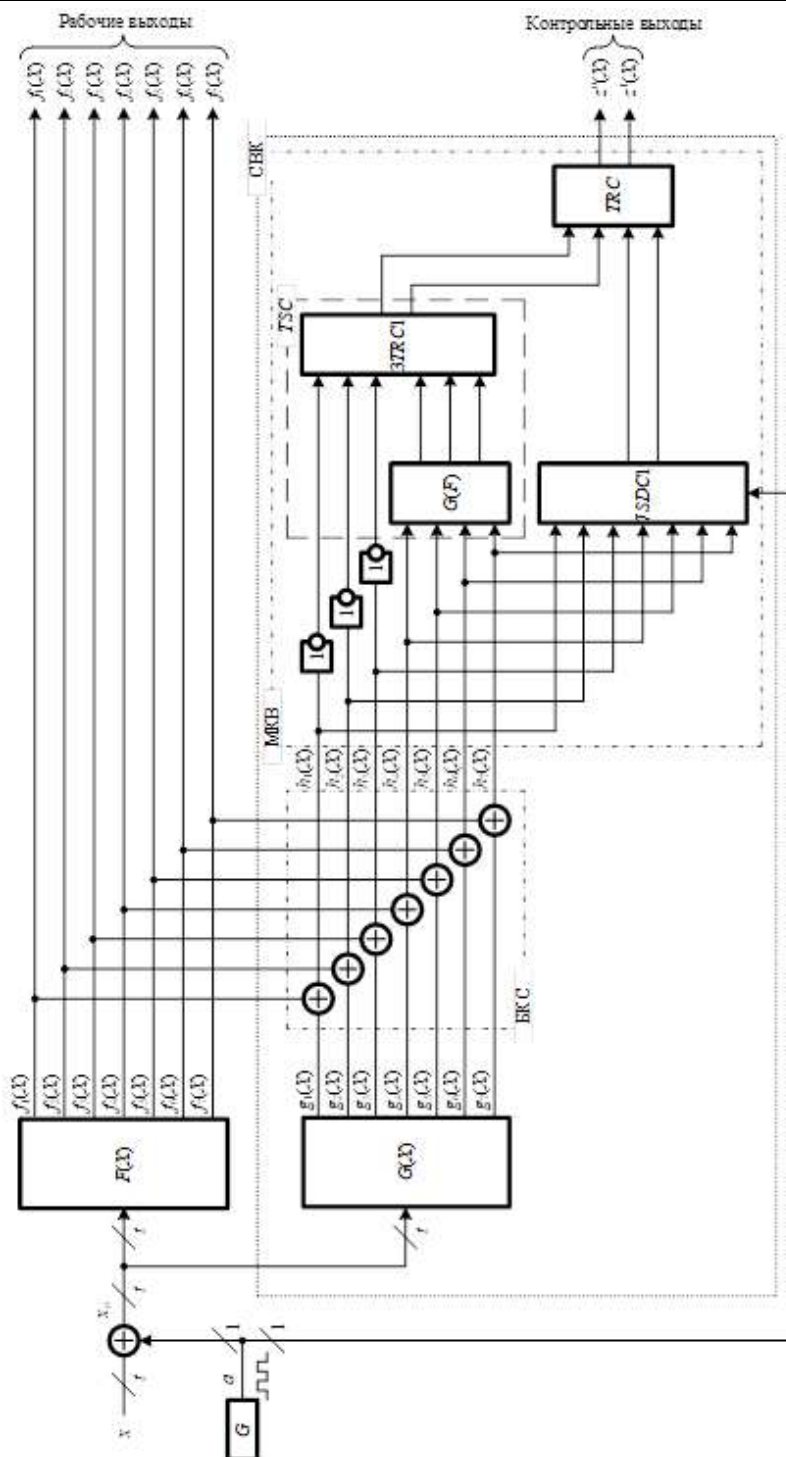


Рис. 1. Структурная схема самопроверяемого устройства с контролем вычислений по двум диагностическим признакам.

Fig. 1. Block diagram of a self-checking device with control of calculations based on two diagnostic features.

Коды Хэмминга строятся с использованием следующей проверочной матрицы [20]:

$$\begin{pmatrix} & & & h_4 & h_5 & h_6 & h_7 \\ h_3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ h_2 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ h_1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{1}$$

Проверочные символы кода (7, 4) описываются функциями:

$$\begin{cases} h_1 = h_4 \oplus h_5 \oplus h_7; \\ h_2 = h_4 \oplus h_6 \oplus h_7; \\ h_3 = h_5 \oplus h_6 \oplus h_7. \end{cases} \tag{2}$$

Другими словами, кодер $G(F)$ реализует систему (2).

Остальные элементы структуры, приведенной на рис. 1, за исключением блоков $G(F)$ и 7SDC1 (о них речь пойдет далее), имеют понятные структуры.

Введем в рассмотрение метрику, которая позволит сравнивать синтезируемые по предложенной структуре самопроверяемые устройства с устройствами, синтезированными иначе. Рассмотрим здесь один показатель – показатель структурной избыточности для самопроверяемого устройства. Для разрабатываемых методов синтеза самопроверяемых цифровых устройств принято производить оценку структурной избыточности в сравнении с типовым методом дублирования [23]. Здесь и далее будем использовать метрику библиотеки stdcell2_2.genlib и инструмент работы с логическими устройствами SIS [24, 25]. В рассматриваемой библиотеке функциональных элементов введен показатель сложности реализации в условных единицах, имеющий корреляцию с площадью, занимаемой устройством на кристалле. Например, в табл. 1 приведены значения показателя сложности реализации для простейших элементов. С использованием данных этой таблицы можно оценить сложность реализации структуры рис. 1.

Табл. 1. Показатели сложности реализации простейших элементов.

Table 1. Indicators of the complexity of implementing the simplest gates.

Функциональный элемент	Обозначение	Показатель сложности реализации, в усл. ед.
НЕ (NOR)	L_{NOR}	16
2И (2AND)	L_{2AND}	32
3И (3AND)	L_{3AND}	40
2ИЛИ (2OR)	L_{2OR}	32
3ИЛИ (3OR)	L_{3OR}	40
XOR	L_{XOR}	40

Здесь следует пояснить, какие именно рассматриваются далее устройства $G(F)$ и 7SDC1, так как их реализовать можно по-разному. Кодер $G(F)$ реализуется по системе функций (2) на пяти элементах XOR. Устройство же 7SDC1 включает в себя схему сжатия семи самодвойственных сигналов в один (такая схема реализуется из трех трехвходовых элементов XOR по структуре рис. 8 из [26]), а также тестер самодвойственного сигнала (см. рис. 2 из [6]). Теперь, зная состав МКБ, можно оценить показатель сложности его реализации в выбранной метрике библиотеки stdcell2_2.genlib.

Структура самопроверяемого устройства на основе дублирования приведена, например, в [18, 22]. Определим показатель сложности реализации неизменяемой части СВК на основе дублирования. В ней используется компаратор 7TRC1, синтезируемый на основе шести элементарных модулей TRC. Возьмем структуру TRC, состоящую из 6 элементов 2ИЛИ и 2И с показателем сложности реализации $L_{TRC} = 192$ усл. ед. Тогда сложность компаратора будет

оценена величиной $L_{7TRC1} = 1152$ усл. ед. Показатель сложности каскада из семи инверторов будет равен 112 усл. ед. Таким образом, сложность неизменяемой части структуры дублирования будет оцениваться величиной 1264 усл. ед.

Неизменяемая часть структуры на основе ЛКС с применением кода (7, 4) содержит в себе БКС и МКВ. Показатель сложности БКС, состоящего из семи элементов XOR, равен 280 усл. ед. Показатель сложности МКВ складывается из показателя сложности кодера $L_{G(F)} = 200$ усл. ед., $L_{3TRC1} = 384$ усл. ед., $L_{7SDC1} = 240$ усл. ед. (каждый трехвходовый XOR имеет показатель сложности, равный 80 усл. ед.), $L_{SDC} = 96$ усл. ед., показатель сложности трех инверторов – 48 усл. ед. и $L_{TRC} = 192$ усл. ед. Итого, показатель сложности реализации МКВ составляет 1160 усл. ед., а всей неизменяемой части СВК на основе ЛКС с применением кода (7, 4) – 1440 усл. ед.

Показатель сложности самопроверяемого устройства, синтезируемого на основе дублирования для семивыходного устройства, определяется по формуле:

$$L_D = 2L_{F(X)} + 7L_{NOT} + 6L_{TRC} = 2L_{F(X)} + 112 + 1152 = 2L_{F(X)} + 1264. \quad (3)$$

Показатель сложности самопроверяемого устройства на основе ЛКС с применением кода (7, 4) для семивыходного устройства вычисляют по формуле:

$$L_{BC} = L_{F(X)} + L_{G(X)} + L_{CCM} + 7L_{XOR} = L_{F(X)} + L_{G(X)} + 1160 + 280 = L_{F(X)} + L_{G(X)} + 1440. \quad (4)$$

Из сравнения (3) и (4) следует, что самопроверяемое устройство, синтезированное на основе ЛКС с применением кода (7, 4), будет эффективнее самопроверяемого устройства, синтезированного на основе дублирования, при условии:

$$\Delta = L_D - L_{BC} > 0, \quad (5)$$

$$\Delta = (2L_{F(X)} + 1264) - (L_{F(X)} + L_{G(X)} + 1440) = L_{F(X)} - L_{G(X)} - 176 > 0 \Leftrightarrow L_{F(X)} > L_{G(X)} + 176.$$

Таким образом, для «выигрыша» в показателе сложности технической реализации по сравнению с дублированием при синтезе СВК нужно получить такую структуру блока $G(X)$, для которой показатель сложности реализации:

$$L_{G(X)} < L_{F(X)} - 176. \quad (6)$$

За счет использования ЛКС это возможно. Покажем это далее на примере.

3. Принципы синтеза схем встроенного контроля

Синтез самопроверяемой СВК по структуре рис. 1 имеет ряд особенностей, которые продемонстрируем на примере. Рассмотрим объект диагностирования комбинационного типа, работа которого задается табл. 2 (см. графы $x_1 \dots x_4$ и $f_1(X) \dots f_7(X)$).

При синтезе СВК требуется обеспечить проверяемость блока вычисления функций логической коррекции $G(X)$, то есть неисправности его внутренней структуры должны вызывать искажения на выходах хотя бы на одной входной комбинации (или пары комбинаций, поскольку имеется подсхема контроля самодвойственности сигналов) [27]. Более того, в структуре СВК все элементы являются типовыми, кроме блока $G(X)$. Поэтому задачей синтеза является как раз получение логических выражений, описывающих данный блок. Для обеспечения самопроверяемости неизменной части СВК требуется тестировать каждый из элементов XOR в БКС и все компоненты МКВ. Наиболее сложной является задача тестирования компаратора в структуре тестера, что требует формирования полного множества контрольных векторов кода (7, 4). Для тестируемости элементов XOR также требуется подача полного множества комбинаций на их входы. Кроме обозначенных особенностей следует учесть еще одну, проявляющуюся именно в структурах с контролем вычислений по признаку самодвойственности сигналов: при синтезе блока $G(X)$ необходимо обеспечить формирование прямого и инверсного кодовых слов на противоположных

комбинациях. Синтез СВК по следующему алгоритму позволяет решить поставленную задачу (опишем его по шагам).

Табл. 2. Описание работы самопроверяемого устройства на линиях схемы.

Table 2. Description of the operation of a self-checking device on circuit lines.

$\mathbb{Z}_2$	x_4	x_3	x_2	x_1	$f_7(X)$	$f_6(X)$	$f_5(X)$	$f_4(X)$	$f_3(X)$	$f_2(X)$	$f_1(X)$	$h_7(X)$	$h_6(X)$	$h_5(X)$	$h_4(X)$	$h_3(X)$	$h_2(X)$	$h_1(X)$	$g_7(X)$	$g_6(X)$	$g_5(X)$	$g_4(X)$	$g_3(X)$	$g_2(X)$	$g_1(X)$
0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0
1	0	0	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1
2	0	0	1	0	1	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	0	0	1	1	0
3	0	0	1	1	1	1	0	1	1	1	0	0	0	0	1	0	1	1	1	1	1	0	0	1	0
4	0	1	0	0	0	0	1	1	1	0	1	0	0	1	0	1	0	1	0	0	0	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	1
6	0	1	1	0	1	0	0	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1	1	0	0
7	0	1	1	1	1	0	0	1	0	1	1	0	0	1	1	1	1	0	1	0	1	0	1	0	1
8	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
9	1	0	0	1	1	1	0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0
10	1	0	1	0	0	0	0	1	0	1	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0
11	1	0	1	1	1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	1	0	1	0
12	1	1	0	0	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	1	1	0	0	0
13	1	1	0	1	1	0	0	0	0	1	1	1	1	1	0	1	0	0	0	1	1	0	1	1	1
14	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0
15	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0	1

Шаг 1. Расширение таблицы описания устройства.

На данном этапе таблица истинности, описывающая логику работы устройства, расширяется, куда добавляются столбцы функций $h_1(X) \dots h_7(X)$ и $g_1(X) \dots g_7(X)$. Значения данных функций считаются не определенными на первом шаге алгоритма. Функции логической коррекции получаются по формуле: $g_i(X) = f_i(X) \oplus h_i(X), i = \overline{1, 7}$. Поэтому их значения формируются однозначно, исходя из заполняемых значений сигналов в столбцах $h_1(X) \dots h_7(X)$.

Шаг 2. Заполнение столбцов $h_1(X) \dots h_7(X)$.

На каждой входной комбинации векторы $\langle h_7(X) h_6(X) \dots h_2(X) h_1(X) \rangle$ должны быть доопределены так, чтобы было сформировано кодовое слово, принадлежащее коду (7, 4). В противном случае подсхема контроля принадлежности кодовых слов коду (7, 4) не будет работать корректно.

Среди столбцов $h_1(X) \dots h_7(X)$ имеются столбцы, соответствующие информационным и проверочным символам кода (7, 4). Будем выделять информационные и проверочные символы в информационный и контрольный векторы: информационный вектор $\langle h_7(X) h_6(X) h_5(X) h_4(X) \rangle$ и контрольный вектор $\langle h_3(X) h_2(X) h_1(X) \rangle$.

При заполнении столбцов $h_4(X), h_5(X), h_6(X), h_7(X)$ значения в столбцах $h_1(X), h_2(X), h_3(X)$ доопределяются однозначно. Другими словами, при заполнении столбцов, соответствующих информационным символам, оставшиеся столбцы, соответствующие проверочным символам, будут заполнены «автоматически».

Также необходимо на прямой и инверсной входных комбинациях формировать прямое и инверсное кодовые слова кода (7, 4). Поэтому достаточно заполнить столбцы $h_1(X) \dots h_7(X)$ только на первой половине таблицы – на входных комбинациях, соответствующих строкам с номерами $0, 1, \dots, 2^t - 1$. Оставшаяся часть таблицы заполняется «антисимметрично».

Заполнение можно осуществлять построчно, аналогично тому, как это сделано в [28] при синтезе СВК на основе ЛКС с применением кодов Боуза – Лина и преобразованием всех функций объекта диагностирования в БКС. При этом на каждом шаге заполнения следует проверять и запоминать формируемые проверочные комбинации для элементов СВК.

Будем рассматривать последовательно заполнение строк таблицы с номерами $0, 1, \dots, 2^t - 1$ и соответствующим образом доопределять значения сигналов на инверсных комбинациях, а также проверять формирование тестовых комбинаций. Для заполнения используем табл. 3, в которой приведено распределение всех информационных векторов кода $(7, 4)$ на группы, соответствующие контрольным векторам.

Табл. 3. Распределение информационных векторов кода $(7, 4)$ между контрольными векторами.

Table 3. Distribution of data vectors code $(7, 4)$ between check vectors.

Контрольные векторы							
000	001	010	011	100	101	110	111
Информационные векторы							
0000	1011	1010	0001	1001	0010	0011	1000
0111	1100	1101	0110	1110	0101	0100	1111

Будем доопределять значения сигналов следующим образом. На первой половине таблицы, на строках с номерами $0, 1, \dots, 2^t - 1$, сгенерируем по четыре раза произвольные комбинации с четырьмя контрольными векторами с весами $r=0$ и $r=2$. Тогда на инверсных им комбинациях будут сгенерированы инверсные кодовые слова кода $(7, 4)$, в которых веса контрольных векторов будут равны $r=1$ и $r=3$. Однозначно на всех входных комбинациях сгенерируются кодовые слова кода $(7, 4)$ и по четыре раза будет присутствовать каждый из контрольных векторов.

Шаг 3. Проверка формирования теста для элементов XOR.

Выполнив такое доопределение, проверим формирование тестовых комбинаций для элементов XOR [29]. Построчная проверка столбцов f_i и g_i при одинаковых $i = \overline{1, 7}$ говорит о том, что тесты для всех элементов XOR формируются (табл. 4).

Если же тест для какого-либо из элементов не формируется, то требуется поменять сигналы при доопределении для нескольких строк с учетом необходимости получения требуемой комбинации для конкретного элемента XOR. Это делается по аналогии с методом из [30]. Вообще, не для любого устройства можно достигнуть формирования тестов для всех элементов XOR (необходимо определенное количество 1 и 0 для каждой рабочей функции объекта диагностирования на всех используемых входных комбинациях). Однако если это возможно в принципе для заданного устройства, то путем изменения способа доопределения в рассматриваемом методе можно добиться формирования полных тестов для всех элементов XOR. Как следует из табл. 3, число способов доопределения довольно велико, что позволяет выбрать наилучший из них по требуемым показателям реализуемого самопроверяемого устройства.

Шаг 4. Синтез блока вычисления функций логической коррекции

Далее остается синтезировать блок $G(X)$ в выбранном элементном базисе. Данная задача является тривиальной, решается известными методами [31] и здесь не рассматривается. Используя SIS и библиотеку stdcell2_2.genlib, получаем следующие оценки сложности технической реализации.

Сложность блока $F(X)$ до оптимизации оценивается величиной 1248 усл. ед., а после оптимизации процедурой full_simplify – 1032 усл. ед.

Сложность блока $G(X)$ до оптимизации оценивается величиной 968 усл. ед., а после оптимизации процедурой full_simplify – 792 усл. ед.

Табл. 4. Тестовые комбинации для элементов преобразования в БКС.
Table 4. Check combinations for conversion elements in the signal correction block.

№	XOR ₇		XOR ₆		XOR ₅		XOR ₄		XOR ₃		XOR ₂		XOR ₁	
	$f_7(X)$	$g_7(X)$	$f_6(X)$	$g_6(X)$	$f_5(X)$	$g_5(X)$	$f_4(X)$	$g_4(X)$	$f_3(X)$	$g_3(X)$	$f_2(X)$	$g_2(X)$	$f_1(X)$	$g_1(X)$
0	0	0	0	0	1	1	1	1	0	0	1	1	0	0
1	0	0	1	1	1	1	0	0	0	0	1	1	1	1
2	1	1	0	0	0	0	0	1	1	1	1	0	1	0
3	1	1	1	1	0	0	1	0	1	1	1	0	0	1
4	0	0	0	0	1	0	1	1	1	0	0	0	1	0
5	0	0	0	0	1	0	0	0	0	1	0	0	0	1
6	1	1	0	0	0	1	0	1	0	1	1	0	0	0
7	1	1	0	0	0	1	1	0	0	1	1	0	1	1
8	1	0	1	0	1	1	0	0	0	0	0	0	1	0
9	1	0	1	0	0	0	1	1	0	0	0	0	1	0
10	0	1	0	1	0	0	1	0	0	0	1	0	0	0
11	1	0	0	1	0	0	0	1	0	0	0	1	0	0
12	1	0	0	1	0	1	0	0	1	0	0	0	0	0
13	1	0	0	1	0	1	0	0	0	1	1	1	1	1
14	0	1	1	0	1	0	1	0	0	1	1	0	1	0
15	1	0	0	1	1	0	1	0	1	0	0	1	0	1

Используя формулу (6), можно показать, что самопроверяемое устройство, синтезированное на основе ЛКС с применением кода (7, 4), эффективнее устройства, синтезированного на основе дублирования, поскольку неравенство выполняется как в случае использования не оптимизированных блоков $F(X)$ и $G(X)$, оптимизированных блоков $F(X)$ и $G(X)$, так и в случае использования не оптимизированного блока $F(X)$ и оптимизированного блока $G(X)$.

Таким образом, синтезировано самопроверяемое устройство с меньшей, чем при дублировании, структурной избыточностью. Более того, в структурах на основе дублирования оказывается довольно сложной процедура обеспечения формирования теста для компаратора и, вообще, его построить удастся не всегда [13]. Предложенный в настоящей статье метод позволяет добиться свойства самопроверяемости гораздо проще.

Существует большое число способов доопределения строк при рассматриваемом подходе. В структуре рис. 1 с числом разрядов t нужно доопределить сигналы на 2^{t-1} входной комбинации из полного их множества, поскольку на противоположных входных комбинациях требуется обеспечить появление противоположных значений всех функций h_i , $i = \overline{1, 7}$. Также, как отмечено выше, при синтезе СВК требуется обеспечить формирование всех контрольных векторов для кодовых слов кода (7, 4) в СВК. Половина этих контрольных векторов может быть сгенерирована на входных комбинациях из первой половины таблицы. Тогда нужно на первой половине таблицы истинности хотя бы по разу сгенерировать одно из двух кодовых слов кода (7, 4) с требуемым контрольным вектором (см. табл. 3). Так как алгоритм доопределения подразумевает генерацию равномерного количества кодовых слов с одинаковыми контрольными векторами, то на $\frac{2^t}{8} = 2^{t-3}$ входных комбинациях формируется

каждое из кодовых слов. Каждая строка из первой группы из 2^{t-3} строк может быть доопределена C_{16}^1 способами (одним из 16 кодовых слов). Однозначно сгенерируются инверсные кодовые слова на последней группе из 2^{t-3} строк. Так будут исключены из рассмотрения 2 контрольных вектора и потребуется сформировать оставшиеся 6 контрольных векторов. «Автоматически» можно исключить по два не использованных

кодовых слова с уже «использованными» контрольными векторами (см. табл. 3). Остается 12 из 16 кодовых слов. Каждая строка из второй группы из 2^{t-3} строк может быть доопределена C_{12}^1 способами (одним из оставшихся 12 кодовых слов). Аналогично рассмотренному случаю однозначно сгенерируются инверсные кодовые слова на предпоследней группе из 2^{t-3} строк. Из рассмотрения снова будут исключены еще два контрольных вектора. Остается 8 из 16 кодовых слов. Далее каждая строка из третьей группы из 2^{t-3} строк может быть доопределена C_8^1 способами (одним из оставшихся 8 кодовых слов). И, наконец, каждая строка из четвертой группы из 2^{t-3} строк может быть доопределена C_4^1 способами (одним из оставшихся 4 кодовых слов). После этих процедур таблица будет заполнена. Итого имеем следующее число способов доопределения при равномерном заполнении кодовыми словами кода (7, 4): $N = C_{16}^1 C_{12}^1 C_8^1 C_4^1 = 16 \cdot 12 \cdot 8 \cdot 4 = 6144$.

Так что произвольное доопределение группами по 2^{t-3} строк до одного кодового слова дает более 6 тыс. способов доопределения и, соответственно, синтеза СВК, что позволяет строить СВК с различными характеристиками и выбирать среди них наилучший по требуемым показателям.

4. Исследование возможностей применения разработанного метода при увеличении числа выходов объекта диагностирования

Введем величину ε , характеризующую разницу между показателем сложности технической реализации неизменяемой части СВК на основе дублирования и аналогичным показателем для СВК на основе ЛКС с применением кода (7, 4). Назовем ее «запасом эффективности по структурной избыточности».

Выше, на рис. 1, приведена базовая структура организации СВК на основе ЛКС с применением кода (7, 4) для устройства с семью выходами, а в формуле (5) величина $\varepsilon = -176$ усл. ед. библиотеки stdcell2_2.genlib. В общем случае же устройство может иметь произвольное число выходов n . Для каждого значения n будет свой запас устойчивости по структурной избыточности. Определим зависимости для различных случаев n .

Так как при построении СВК выходы устройства будут разбиваться на группы по 7 выходов в каждой, то для каждой такой группы будет обустраиваться своя СВК, выходы же нескольких СВК будут объединяться на входах самопроверяемых компараторов. Компаратор синтезируется на основе $q - 1$ элемента TRC , где q – число выделенных для контроля вычислений групп.

Введем в рассмотрение еще одну величину – ξ , характеризующую запас эффективности по структурной избыточности без учета компаратора. Это необходимо для исследования влияющих на структурную избыточность параметров.

Также рассмотрим семь типичных случаев по числу выходов n , характеризуемых величинами $r = n(\bmod 7)$: $r \in \{0, 1, \dots, 6\}$.

Найдем закономерности, проявляющиеся при увеличении числа n с учетом получаемых чисел $r = n(\bmod 7)$: $r \in \{0, 1, \dots, 6\}$. Будем наименьший неотрицательный вычет r приписывать к величинам ξ и ε в виде верхнего индекса: ξ^r и ε^r .

Рассмотрим случай $n(\bmod 7) = 0$. Это наилучший случай с позиции сравнения с дублированием, поскольку все выходы устройства присутствуют только в одной из контролируемых групп.

В общем случае сложность структуры на основе дублирования определяется величиной:

$$L_D = 2L_{F(x)} + nL_{NOT} + (n-1)L_{TRC} = 2L_{F(x)} + 16n + 192(n-1) = 2L_{F(x)} + 208n - 192. \quad (7)$$

Показатель сложности структуры на основе ЛКС и кода (7, 4) в общем случае имеет вид:

$$\begin{aligned}
 L_{BC} &= L_{F(X)} + L_{G(X)} + \left\lceil \frac{n}{7} \right\rceil L_{CCM} + \left\lceil \frac{n}{7} \right\rceil (7L_{XOR}) + \left(\left\lceil \frac{n}{7} \right\rceil - 1 \right) L_{TRC} = \\
 &= L_{F(X)} + L_{G(X)} + 1160 \cdot \left\lceil \frac{n}{7} \right\rceil + 280 \cdot \left\lceil \frac{n}{7} \right\rceil + 192 \cdot \left\lceil \frac{n}{7} \right\rceil - 192.
 \end{aligned} \tag{8}$$

Вычтем из (7) выражение (8):

$$\begin{aligned}
 \Delta &= L_D - L_{BC} = (2L_{F(X)} + 208n - 192) - \\
 &- \left(L_{F(X)} + L_{G(X)} + 1160 \cdot \left\lceil \frac{n}{7} \right\rceil + 280 \cdot \left\lceil \frac{n}{7} \right\rceil + 192 \cdot \left\lceil \frac{n}{7} \right\rceil - 192 \right) = \\
 &= L_{F(X)} - L_{G(X)} + 208n - 1632 \cdot \left\lceil \frac{n}{7} \right\rceil.
 \end{aligned} \tag{9}$$

Из формулы (9) следует, что

$$\xi = 208n - 1632 \cdot \left\lceil \frac{n}{7} \right\rceil. \tag{10}$$

Исключим в формуле (8) величину показателя сложности компаратора. Тогда можно выделить следующую величину

$$\varepsilon = 208n - 1440 \cdot \left\lceil \frac{n}{7} \right\rceil - 192. \tag{11}$$

Теперь перейдем к рассмотрению частных случаев вычисления величин (11) и (10).

Итак, обратим внимание на величины ξ^r и ε^r , для которых $r = n \pmod{7}$: $r \in \{0, 1, \dots, 6\}$.

Положим $j = \left\lceil \frac{n}{7} \right\rceil$. Имеем следующие формулы, описывающие все возможные случаи для ξ^r

и ε^r , представленные в табл. 5.

Из табл. 5 следует, что величины ε^r при увеличении j будут возрастать, достигнув сначала 0, а затем – увеличиваясь. Другими словами, с ростом j и, как следствие, n , запас эффективности по структурной избыточности без учета компаратора будет расти. При отсутствии в СВК на основе ЛКС с применением кода (7, 4) компаратора для отдельных подсхем контроля эффективности по показателям структурной избыточности в сравнении с дублированием достичь гораздо проще. Более того, запас эффективности по данному показателю с ростом n только растет. Однако значение величины ξ^r с ростом j будет всё больше и больше уменьшаться. Отсюда становится понятным, что общий запас эффективности по структурной избыточности с ростом j и, как следствие, n , будет уменьшаться. Весомое значение в общем уменьшении показателя эффективности по структурной избыточности как раз имеет компаратор.

В табл. 6 приведены расчеты для некоторых значений n , а на рис. 2 и 3 графически представлены зависимости величин ε^r и ξ^r для малых значений j , по которым прослеживаются закономерности изменения, в том числе, скорость их роста (уменьшения).

Из анализа данных табл. 6 становится очевидным то, что при больших значениях числа выходов у объекта диагностирования дублирование будет обладать существенным запасом эффективности по структурной избыточности по сравнению с рассматриваемым методом. Даже для случаев с $n = 7 \dots 27$ при самом «удачном» варианте мы видим отрицательный запас эффективности по структурной избыточности предлагаемого метода по сравнению с дублированием. Этот отрицательный запас «отыгрывается» при выборе структуры блока $G(X)$, но только в том случае, если реализация блока $F(X)$ не оказывается оптимальной или

квазиоптимальной для рассматриваемого количества булевых функций. В каждом отдельном случае синтеза блока $G(X)$ после оптимизации его структуры и определения показателей структурной избыточности потребуются сравнение с аналогичными показателями для стандартного метода дублирования. Ясно, что в ряде случаев предлагаемый метод не даст эффекта с позиции структурной избыточности по сравнению с дублированием.

Табл. 5. Формулы вычисления величин ξ^r и ε^r .
Table 5. Formulas for calculating the quantities ξ^r and ε^r .

r	ε^r , усл. ед.	ξ^r , усл. ед.
0	$\varepsilon^0 = 208 \cdot 7j - 1440j - 192 = 16j - 192$	$\xi^0 = 208 \cdot 7j - 1632j = -176j$
1	$\varepsilon^1 = 16(j+1) - 192 - 208 \cdot (7-1) = 16j - 1424$	$\xi^1 = -176(j+1) - 208 \cdot (7-1) = -176j - 1424$
2	$\varepsilon^2 = 16(j+1) - 192 - 208 \cdot (7-2) = 16j - 1216$	$\xi^2 = -176(j+1) - 208 \cdot (7-2) = -176j - 1216$
3	$\varepsilon^3 = 16(j+1) - 192 - 208 \cdot (7-3) = 16j - 1008$	$\xi^3 = -176(j+1) - 208 \cdot (7-3) = -176j - 1008$
4	$\varepsilon^4 = 16(j+1) - 192 - 208 \cdot (7-4) = 16j - 800$	$\xi^4 = -176(j+1) - 208 \cdot (7-4) = -176j - 800$
5	$\varepsilon^5 = 16(j+1) - 192 - 208 \cdot (7-5) = 16j - 592$	$\xi^5 = -176(j+1) - 208 \cdot (7-5) = -176j - 592$
6	$\varepsilon^6 = 16(j+1) - 192 - 208 \cdot (7-6) = 16j - 384$	$\xi^6 = -176(j+1) - 208 \cdot (7-6) = -176j - 384$

Оценочно можно рекомендовать использование разработанного метода для частных случаев объектов диагностирования с небольшим числом выходов (не более 30). При этом метод может оказаться эффективным и для объектов диагностирования с большим числом выходов, обладающих определенной сложностью реализуемых на них булевых функций, для которых блоки $G(X)$ будут реализовывать менее сложные булевы функции. Относительно вопроса сложности реализации булевых функций можно рекомендовать обзор [32].

5. Заключение

При синтезе самопроверяемых цифровых устройств может использоваться предложенная в настоящей работе «базовая» структура с контролем вычислений по двум диагностическим признакам. Данная структура позволяет синтезировать устройства, функционирующие в импульсном режиме, что дает возможность более частой смены сигналов и, соответственно, формирования большего числа проверочных комбинаций для элементов СВК. Это и результаты, например, работ [6, 11], говорят об улучшении показателей контролепригодности при синтезе СВК по предлагаемому методу в сравнении с традиционными подходами к их синтезу. Указанные обстоятельства подчеркивают основное преимущество представленного метода – возможности повышения показателей контролепригодности, что актуально в практических приложениях при построении систем критического применения, в которых смена входных состояний, как правило, происходит редко [33-35].

Предложенный в статье метод дает возможность синтеза самопроверяемых устройств с меньшей избыточностью, чем при использовании метода дублирования. Однако в каждом отдельном случае устройств $F(X)$ это необходимо проверять. Исследования показателей «эффективности запаса по структурной избыточности» для типовых компонентов СВК на основе предлагаемого метода показывают рост величины ε^r и снижение величины ξ^r с ростом числа выходов n объектов диагностирования. Существенное влияние на показатели общей структурной избыточности неизменяемой части СВК оказывают показатели сложности реализации компаратора. Неизменяемая часть СВК, реализуемая по структуре рис. 1, оказывается более сложной, чем при дублировании. Поэтому требуется существенно упростить блок вычисления функций логической коррекции для того, чтобы добиться «выигрыша»

Табл.6. Значения величин ε^r и ξ^r .
Table 6. Values of ε^r and ξ^r .

n	Неизменяемая часть при дублировании, усл. ед.	Неизменяемая часть при ЛКС без компаратора, усл. ед.	Компаратор, усл. ед.	r	ε^r , усл. ед.	ξ^r , усл. ед.
7	1264	1440	0	0	-176	-176
8	1472	2880	192	1	-1408	-1600
9	1680	2880	192	2	-1200	-1392
10	1888	2880	192	3	-992	-1184
11	2096	2880	192	4	-784	-976
12	2304	2880	192	5	-576	-768
13	2512	2880	192	6	-368	-560
14	2720	2880	192	0	-160	-352
15	2928	4320	384	1	-1392	-1776
16	3136	4320	384	2	-1184	-1568
17	3344	4320	384	3	-976	-1360
18	3552	4320	384	4	-768	-1152
19	3760	4320	384	5	-560	-944
20	3968	4320	384	6	-352	-736
21	4176	4320	384	0	-144	-528
22	4384	5760	576	1	-1376	-1952
23	4592	5760	576	2	-1168	-1744
24	4800	5760	576	3	-960	-1536
25	5008	5760	576	4	-752	-1328
26	5216	5760	576	5	-544	-1120
27	5424	5760	576	6	-336	-912
...	...	...	...	...	...	...
71	14576	15840	1920	0	-1264	-3184
72	14784	15840	1920	1	-1056	-2976
73	14992	15840	1920	2	-848	-2768
74	15200	15840	1920	3	-640	-2560
75	15408	15840	1920	4	-432	-2352
76	15616	15840	1920	5	-224	-2144
77	15824	15840	1920	6	-16	-1936
...	...	...	...	...	...	...
99	20400	21600	2688	0	-1200	-3888
100	20608	21600	2688	1	-992	-3680
101	20816	21600	2688	2	-784	-3472
102	21024	21600	2688	3	-576	-3264
103	21232	21600	2688	4	-368	-3056
104	21440	21600	2688	5	-160	-2848
105	21648	21600	2688	6	48	-2640

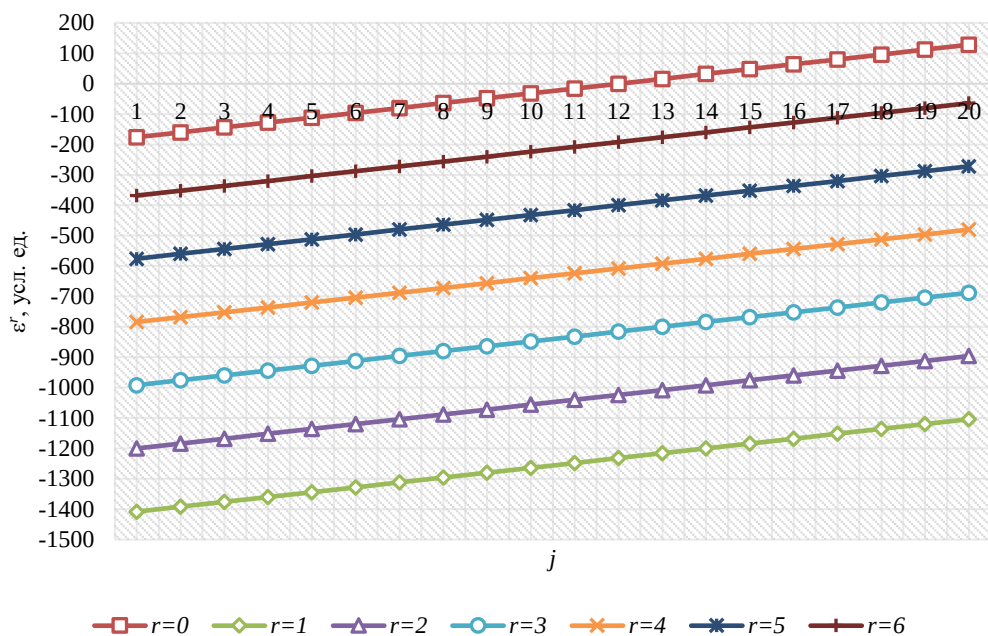


Рис. 2. Графики зависимостей величин ε^r от j .
Fig. 2. Graphs of ε^r versus j .

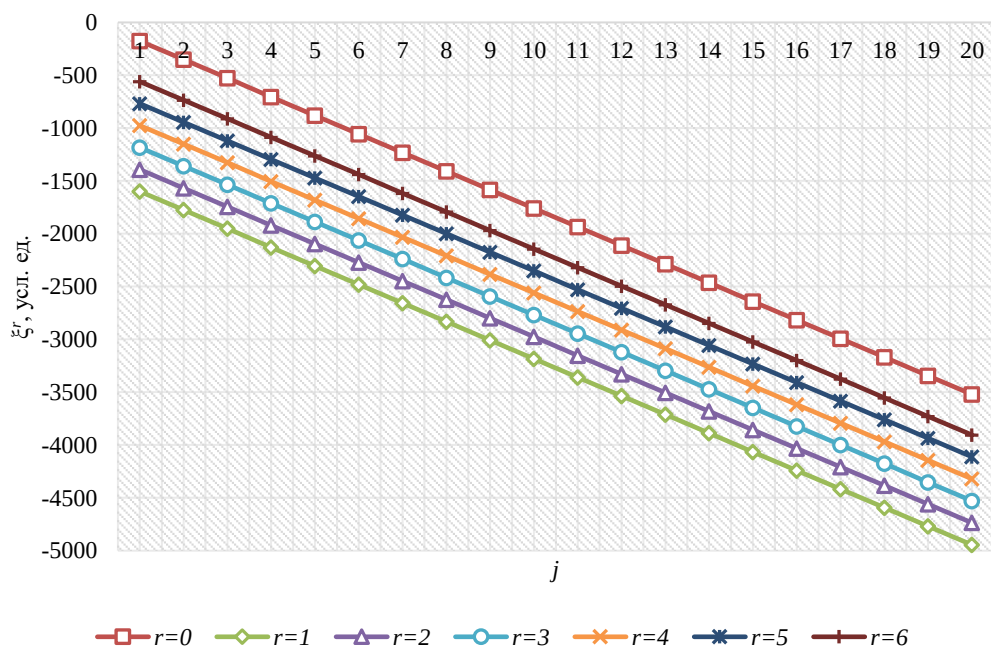


Рис. 3. Графики зависимостей величин ζ^r от j .
Fig. 3. Graphs of ζ^r versus j .

в структурной избыточности по сравнению с дублированием. Тем не менее, это возможно, как показано на простом примере.

Кроме того, следует обратить внимание на сам способ доопределения значений выходов блока вычисления функций логической коррекции, приводящий к следующим ограничениям применения метода. Во-первых, при использовании кода (7, 4) требуется формировать хотя бы единожды каждый контрольный вектор из восьми имеющихся. Таким образом, число аргументов должно быть $t \geq 3$, в противном случае данное условие не может быть выполнено. Во-вторых, так как требуется обеспечивать формирование довольно большого числа тестовых комбинаций для элементов *XOR*, для устройств с малым числом входов (при $t = 3, 4$) крайне сложно обеспечить условия самопроверяемости, поэтому применение структуры рис. 1 в данном случае имеет существенные ограничения. С ростом числа входов объекта диагностирования данная проблема нивелируется.

При использовании предложенной структуры будут обнаруживаться те неисправности, возникающие в блоке $F(X)$, которые будут вызывать обнаруживаемые кодом (7, 4) ошибки или нарушающие самодвойственность функций, формируемых на выходах БКС. Фактически для обнаружения любых ошибок, возникающих на выходах блока $F(X)$, требуется обеспечить отсутствие одновременных искажений значений всех функций, вычисляемых на выходах устройства, в каждой паре подаваемых входных комбинаций, или же исключить необнаруживаемые кодом Хэмминга ошибки. Это можно сделать всегда либо модификацией структуры блока $F(X)$, либо выделением групп из семи его выходов с использованием повторений выходов в группах по аналогии с методами, описанными в главе 6 [20]. С точки зрения моделей неисправностей могут рассматриваться, например, модель одиночной константной неисправности (*stuck-at fault*) внутренних логических элементов блока $F(X)$, модель мостиковой неисправности (*bridging fault*) или модели транзистора (устойчивых обрывов транзисторов *SOP* (*stuck-open*) и устойчивых замыканий транзисторов *SON* (*stuck-on*)) [36]. После выбора учитываемой модели неисправности уже осуществляется анализ возникающих на выходах блока $F(X)$ ошибок в значениях выходных функций при подаче заданного множества входных комбинаций.

Резюмируя, отметим, что представленный в статье метод синтеза СВК по двум диагностическим признакам интересен для практического применения. Он требует более серьезной экспериментальной проверки, в том числе, на тестовых схемах [37], а также моделирования для определения особенностей обнаруживаемых сочетаний искажений на выходах объектов диагностирования. Интерес представляет рассмотрение аналогичных «базовых» структур с контролем вычислений по двум диагностическим признакам для кодов Хэмминга при числе разрядов в кодовых словах $n = m + k = 3 + 4l$, $l \in \mathbb{N}_0$, кодеры которых являются самодвойственными цифровыми устройствами [6].

Проектировщикам высоконадежных и безопасных систем рекомендуется обратить внимание на результаты, полученные в настоящем исследовании.

Список литературы / References

- [1]. Hamming R.W. Error Detecting and Correcting Codes. *Bell System Technical Journal*, 1950, 29 (2), pp. 147-160, DOI: 10.1002/j.1538-7305.1950.tb00463.x.
- [2]. Sridhar K.P., Agalya R., Narmatha D., Vignesh B., Saravanan S. Test Data Compression Using Hamming Encoder and Decoder for System on Chip (SOC) Testing. 2014 International Conference on Circuits, Power and Computing Technologies (ICCPCT-2014), 20-21 March 2014, Nagercoil, India, doi: 10.1109/ICCPCT.2014.7054876.
- [3]. Mehta U.S., Parmar H. Improvement in Error Resilience for Compressed VLSI Test Data Using Hamming Code Based Technique. 2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA), 08-10 January 2015, Bengaluru, India, doi: 10.1109/VLSI-SATA.2015.7050494.

- [4]. Tshagharyan G., Harutyunyan G., Shoukourian S., Zorian Y. Experimental Study on Hamming and Hsiao Codes in the Context of Embedded Applications. Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS'2017), Novi Sad, Serbia, September 29 – October 2, 2017, pp. 25-28, doi: 10.1109/EWDTS.2017.8110065.
- [5]. Stempkovsky A.L., Zhukova T.D., Telpukhov D.V., Gurov S.I. CICADA: A New Tool to Design Circuits with Correction and Detection Abilities. International Siberian Conference on Control and Communications (SIBCON), 13-15 May 2021, Kazan, Russia pp. 1-5, doi 10.1109/SIBCON50419.2021.9438900.
- [6]. Ефанов Д.В., Погодина Т.С. Исследование свойств самодвойственных комбинационных устройств с контролем вычислений на основе кодов Хэмминга. Информатика и автоматизация, 2023, Том 22, №2, С. 349-392, DOI: 10.15622/ia.22.2.5. Efanov D.V., Pogodina T.S. Issledovanie svojstv samodvojstvennyh kombinacionnyh ustrojstv s kontrol'em vychislenij na osnove kodov Hemminga. Informatika i avtomatizaciya, 2023, tom 22, issue 2, pp. 349-392 (in Russian). DOI: 10.15622/ia.22.2.5.
- [7]. Ефанов Д.В. Особые свойства кодов Хэмминга, проявляющиеся при синтезе самопроверяемых цифровых устройств. Известия высших учебных заведений. Приборостроение, 2023, Т. 66, №2, С. 85-99, DOI: 10.17586/0021-3454-2023-66-2-85-99. Efanov D.V. Special Properties of Hamming Codes that Appear when Synthesizing Self-Checking Digital Devices. Izvestiya vysshih uchebnyh zavedenij. Priborostroenie, 2023, vol. 66, issue 2, pp. 85-99 (in Russian). DOI: 10.17586/0021-3454-2023-66-2-85-99.
- [8]. Ефанов Д.В., Зуева М.В., Пашуков А.В. Исследование характеристик обнаружения ошибок кодами Хэмминга, учет которых целесообразен при синтезе самопроверяемых устройств автоматики. Автоматика на транспорте, 2023, Том 9, №3, С. 283-297, DOI: 10.20295/2412-9186-2023-9-03-283-297. Efanov D.V., Zueva M.V., Pashukov A.V. Study of the Characteristics of Error Detection with Hamming Codes, the Consideration of Which is Appropriate for the Synthesis of Automatic Devices with Fault Detection. Avtomatika na Transporte, 2023, vol. 9, issue 3, pp. 283-297 (in Russian). DOI: 10.20295/2412-9186-2023-9-03-283-297.
- [9]. Ефанов Д.В. Предельные свойства кода Хэмминга в схемах функционального диагностирования. Информатика и системы управления, 2011, №3, С. 70-79. Efanov D.V. The Hamming Code's Limit Properties in Functional Control Scheme. Informatika i sistemy upravleniya, 2011, issue 3, pp. 70-79 (in Russian).
- [10]. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Особенности применения кодов Хэмминга при организации самопроверяемых схем встроенного контроля. Известия высших учебных заведений. Приборостроение, 2018, Том 61, №1, С. 47-59, DOI: 10.17586/0021-3454-2018-61-1-47-59. Sapozhnikov V.V., Sapozhnikov Vl.V., Efanov D.V. Features of Hamming Codes Application in Self-Checking Test Circuit Organization. Izvestiya vysshih uchebnyh zavedenij. Priborostroenie, 2018, vol. 61, issue 1, pp. 47-59 (in Russian). DOI: 10.17586/0021-3454-2018-61-1-47-59.
- [11]. Ефанов Д.В., Погодина Т.С. Самодвойственные цифровые устройства с контролем вычислений по кодам Сяо. Вестник Томского государственного университета. Управление, вычислительная техника и информатика, 2023, №63, С. 118-136, DOI: 10.17223/19988605/63/14. Efanov D.V., Pogodina T.S. Self-Dual Digital Devices with Calculations Testing by Hsiao Codes. Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika, 2023, issue 63, pp. 118-136 (in Russian). DOI: 10.17223/19988605/63/14.
- [12]. Гессель М., Морозов А.В., Сапожников В.В., Сапожников Вл.В. Логическое дополнение – новый метод контроля комбинационных схем. Автоматика и телемеханика, 2003, №1, С. 167-176. Goessel M., Morozov A.V., Sapozhnikov V.V., Sapozhnikov Vl.V. Logic Complement, a New Method of Checking the Combinational Circuits. Avtomatika i Telemekhanika, 2003, issue 1, pp. 167-176. (in Russian).
- [13]. Гессель М., Морозов А.В., Сапожников В.В., Сапожников Вл.В. Контроль комбинационных схем методом логического дополнения. Автоматика и телемеханика, 2005, №8, С. 161-172. Goessel M., Morozov A.V., Sapozhnikov V.V., Sapozhnikov Vl.V. Checking Combinational Circuits by the Method of Logic Complement. Avtomatika i Telemekhanika, 2005, issue 8, pp. 161-172. (in Russian).
- [14]. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1, Dordrecht: Springer Science+Business Media B.V., 2008. 184 p.
- [15]. Das D.K., Roy S.S., Dmitriev A., Morozov A., Gössel M. Constraint Don't Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes. Proceedings of the 10th International Workshops on Boolean Problems, Freiberg, Germany, September, 2012, pp. 33-40.

- [16]. Efanov D., Sapozhnikov V., Sapozhnikov V.I., Osadchy G., Pivovarov D. Self-Dual Complement Method up to Constant-Weight Codes for Arrangement of Combinational Logical Circuits Concurrent Error-Detection Systems. Proceedings of 17th IEEE East-West Design & Test Symposium (EWDTS'2019), Batumi, Georgia, September 13-16, 2019, pp. 136-143, doi: 10.1109/EWDTS.2019.8884398.
- [17]. Efanov D.V., Pivovarov D.V. The Hybrid Structure of a Self-Dual Built-In Control Circuit for Combinational Devices with Pre-Compression of Signals and Checking of Calculations by Two Diagnostic Parameters. Proceedings of 19th IEEE East-West Design & Test Symposium (EWDTS'2021), Batumi, Georgia, September 10-13, 2021, pp. 200-206, doi: 10.1109/EWDTS52692.2021.9581019.
- [18]. Согомонян Е.С., Слабаков Е.В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с. Sogomonyan E.S., Slabakov E.V. The Self-Checking Devices and Fault-Tolerant Systems. M.: Radio and Communications. 208 p. (in Russian).
- [19]. Nicolaidis M., Zorian Y. On-Line Testing for VLSI – A Compendium of Approaches. Journal of Electronic Testing: Theory and Applications, 1998, №12, pp. 7-20, DOI: 10.1023/A:1008244815697.
- [20]. Сапожников В.В., Сапожников В.В., Ефанов Д.В. Коды Хэмминга в системах функционального контроля логических устройств, СПб.: Наука, 2018, 151 с. Sapozhnikov V.V., Sapozhnikov V.I.V., Efanov D.V. Hamming Codes in Concurrent Error Detection Systems of Logic Devices. – St. Petersburg: Nauka, 2018, 151 p. (in Russian).
- [21]. Carter W.C., Duke K.A., Schneider P.R. Self-Checking Error Checker for Two-Rail Coded Data. United States Patent Office, filed July 25, 1968, ser. No. 747533, patented Jan. 26, 1971, N. Y., 10 p.
- [22]. Сапожников В.В., Сапожников В.В., Гессель М. Самодвойственные дискретные устройства, СПб: Энергоатомиздат (Санкт-Петербургское отделение), 2001. 331 с. Sapozhnikov V.V., Sapozhnikov V.I.V., Göessel M. Self-Dual Digital Devices. St. Petersburg: Energoatomizdat (St. Petersburg branch), 331 p. (in Russian).
- [23]. Goessel M., Graf S. Error Detection Circuits, London: McGraw-Hill, 1994. 261 p.
- [24]. SIS: A System for Sequential Circuit Synthesis / E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, A. Sangiovanni-Vincentelli. Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 4 May 1992, 45 p.
- [25]. Sentovich E.M., Singh K.J., Moon C., Savoj H., Brayton R.K., Sangiovanni-Vincentelli A. Sequential Circuit Design Using Synthesis and Optimization. Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors, 11-14 October 1992, Cambridge, MA, USA, pp. 328-333, doi: 10.1109/ICCD.1992.276282.
- [26]. Dmitriev A., Saposhnikov V., Saposhnikov V., Goessel M. New Self-Dual Circuits for Error Detection and Testing // VLSI Design, 2000, Vol. 11, Issue 1, pp. 1-21, doi: 10.1155/2000/84720.
- [27]. Гессель М., Мошанин В.И., Сапожников В.В., Сапожников В.В. Обнаружение неисправностей в самопроверяемых комбинационных схемах с использованием свойств самодвойственных функций. Автоматика и телемеханика, 1997, №12, С. 193-200. Göessel M., Moshanin V.I., Sapozhnikov V.V., Sapozhnikov V.I.V. Fault Detection in Self-Test Combination Circuits Using the Properties of Self-Dual Functions. Avtomatika i Telemekhanika, 1997, issue 12, pp. 193-200 (in Russian).
- [28]. Ефанов Д.В. Синтез самопроверяемых комбинационных устройств на основе метода логической коррекции сигналов с применением кодов Боуза – Лина. Информационные технологии, 2023, Том 29, №10, С. 503-511, DOI: 10.17587/it.29.503-511. Efanov D.V. Self-Checking Combinational Devices Synthesis Based on the Boolean Signal Correction Method Using Bose-Lin Codes. Information Technologies, 2023, Vol. 29, issue 10, pp. 503-511 (in Russian). DOI: 10.17587/it.29.503-511.
- [29]. Аксенова Г.П. Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю 2. Автоматика и телемеханика, 1979, №9, С. 126-135. Necessary and Sufficient Akseanova G.P. Conditions for Design of Completely Checkable Modulo 2 Convolution Circuits. Avtomatika i Telemekhanika, 1979, issue 9, pp. 126-135 (in Russian).
- [30]. Сапожников В.В., Сапожников В.В., Ефанов Д.В. Метод функционального контроля комбинационных логических устройств на основе кода «2 из 4». Известия высших учебных заведений. Приборостроение, 2016, Том 59, №7, С. 524-533, DOI: 10.17586/0021-3454-2016-59-7-524-533. Sapozhnikov V.V., Sapozhnikov V.I.V., Efanov D.V. Method of Operation Control Over Combinatory Logic Device Based on 2-out-of-4 Code. Izvestiya vysshih uchebnyh zavedenij. Priborostroenie, 2016, vol. 59, issue 7, pp. 524-533 (in Russian). DOI: 10.17586/0021-3454-2016-59-7-524-533.
- [31]. Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д. Логические основы проектирования дискретных устройств, М.: Физматлит, 2007. 592 с. Zakrevsky A.D., Pottosin Yu.V.,

- Cheremisinova L.D. Logical Foundations of Designing Discrete Devices, M.: Fizmatlit, 2007. 592 p. (in Russian).
- [32]. Коршунов А.Д. Сложность вычислений булевых функций. Успехи математических наук, 2012, Том 67, №1 (403), С. 97-168, DOI: 10.4213/rm9459. Korshunov A.D. Computational complexity of Boolean functions. Uspekhi matematicheskikh nauk, 2012, vol. 67, issue 1 (403), pp. 97-168 (in Russian). DOI: 10.4213/rm9459.
- [33]. Drozd A., Kharchenko V., Antoshchuk S., Sulima J., Drozd M. Checkability of the Digital Components in Safety-Critical Systems: Problems and Solutions. Proceedings of 9th IEEE East-West Design & Test Symposium (EWDTS'2011), Sevastopol, Ukraine, 2011, pp. 411- 416, doi: 10.1109/EWDTS.2011.6116606.
- [34]. Дрозд А.В., Харченко В.С., Антошук С.Г., Дрозд Ю.В., Дрозд М.А., Сулима Ю.Ю. Рабочее диагностирование безопасных информационно-управляющих систем. Под ред. А.В. Дрозда и В.С. Харченко, Харьков: Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2012, 614 с. Drozd A.V., Kharchenko V.S., Antoshchuk S.G., Drozd Yu.V., Drozd M.A., Sulima Yu.Yu. (2012) Working Diagnostics of Safe Information and Control Systems. Edited by A.V. Drozd and V.S. Kharchenko. Khar'kov: N.E. Zhukovsky National Aerospace University «KAI». 614 p. (in Russian).
- [35]. Сапожников Вл.В. Синтез систем управления движением поездов на железнодорожных станциях с исключением опасных отказов, М.: Наука, 2021. 229 с. Sapozhnikov Vl.V. Synthesis of Train Traffic Control Systems at Railway Stations with the Exception of Dangerous Failures. M. Nauka. 229 p. (in Russian).
- [36]. Багхдади А.А.А., Хаханов В.И., Литвинова Е.И. Методы анализа и диагностирования цифровых устройств (аналитический обзор) // Автоматизированные системы управления и приборы автоматики, 2014, № 166, с. 59-74. Baghdadi A.A.A., Hahanov V.I., Litvinova E.I. Metody analiza i diagnostirovaniya cifrovyyh ustrojstv (analiticheskij obzor) // Avtomatizirovannye sistemy upravleniya i pribory avtomatiki, 2014, № 166, s. 59-74 (in Russian).
- [37]. Collection of Digital Design Benchmarks: <https://ddd.fit.cvut.cz/www/prj/Benchmarks/>

Информация об авторах / Information about authors

Дмитрий Викторович ЕФАНОВ – доктор технических наук, профессор, действительный член Международной Академии транспорта, член Института инженеров электротехники и электроники, профессор Высшей школы транспорта Института машиностроения, материалов и транспорта Санкт-Петербургского политехнического университета Петра Великого (СПбПУ Петра Великого), профессор кафедры «Автоматика, телемеханика и связь на железнодорожном транспорте» Российского университета транспорта (МИИТ). Сфера научных интересов: дискретная математика; железнодорожная автоматика и системы управления; интеллектуальные транспортные системы и технологии; методы непрерывного мониторинга транспортных систем, систем автоматического управления, сложных инженерных конструкций и сооружений; надежность, безопасность и техническая диагностика дискретных систем; синтез отказоустойчивых и безопасных систем управления; синтез самопроверяемых схем встроенного контроля устройств автоматики.

Dmitry Viktorovich EFANOV – Dr. Sci. (Tech.), Professor, Full-member of the International Transport Academy, IEEE member, Professor of Transport Higher School of Mechanical Engineering, Material and Transport Institute at Peter the Great Saint Petersburg Polytechnic University, Professor of Automation, Remote Control and Communication on Railway Transport Department, Russian University of Transport. Research interests: discrete mathematics; railway automation and control systems; intelligent transport systems and technologies; methods of health monitoring of transportation systems, automatic control systems, complex engineering structures and structures; reliability, safety and technical diagnostics of discrete systems; synthesis of fault-tolerant and safe control systems; synthesis of self-checking circuits built-in control of automation devices.



Оптимизация конфиденциальных запросов к базе данных на облаке

¹ Н.П. Варновский, ORCID: 0000-0002-2363-0254 <otd13isp@gmail.com>

² С.А. Мартишин, ORCID: 0000-0001-5437-4049 <mart@ispras.ru>

² М.В. Храпченко, ORCID: 0000-0002-5147-5132 <khrap@ispras.ru>

² А.В. Шокуров, ORCID: 0000-0002-6801-7728 <shok@ispras.ru>

¹ *Институт проблем информационной безопасности МГУ имени М.В. Ломоносова.
119192, г. Москва, Мичуринский проспект, 1, офис 10.*

² *Институт системного программирования РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.*

Аннотация. Авторами рассматривается задача PIR (Private Information Retrieval) обеспечения безопасных запросов к базе данных, размещенной на облаке при наличии активного противника, который не вмешивается в выполнение протокола, но может производить атаку с известными открытыми запросами. Для представления номера бита i в виде числа, все цифры которого различны, в предложенных алгоритмах применяется система счисления по основанию l с числом разрядов d . Использование перестановок цифр номера запрашиваемого бита в качестве секретных ключей шифрования уменьшает вероятность угадывания номера бита. Хранящиеся на облаке биты исходной базы данных группируются в виде массивов, что позволяет снизить коммуникационную сложность. Используется датчик псевдослучайных чисел для замены значения бита в зависимости от запрашиваемого клиентом номера бита i . Это позволяет в случае сговора пассивного противника, находящегося на облаке, и активного противника вне облака, затруднить сопоставление значения бита конкретному номеру. Приведены оценки коммуникационной сложности, вероятности угадывания номера бита при однократной атаке с известным открытым запросом номера бита i и при атаке с неограниченным числом известных открытых запросов.

Ключевые слова: база данных; облачные вычисления; задача безопасных запросов PIR.

Для цитирования: Варновский Н.П., Мартишин С.А., Храпченко М.В., Шокуров А.В. Оптимизация конфиденциальных запросов к базе данных на облаке. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 39–58. DOI: 10.15514/ISPRAS-2025-37(3)-3.

Благодарности: Авторы выражают особую благодарность Лазареву Д.О. за ценные замечания в процессе работы над статьей.

Optimizing Confidential Database Queries on the Cloud

¹ N.P. Varnovskiy, ORCID: 0000-0002-2363-0254 <otd13isp@gmail.com>

² S.A. Martishin, ORCID: 0000-0001-5437-4049 <mart@ispras.ru>

² M.V. Khrapchenko, ORCID: 0000-0002-5147-5132 <khrap@ispras.ru>

² A.V. Shokurov, ORCID: 0000-0002-6801-7728 <shok@ispras.ru>

¹ Information Security Institute of Moscow State Lomonosov University,
Office 10, 1, Michurinskii prospect, 119192, Moscow, RUSSIA.

² Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Abstract. The authors consider the PIR (Private Information Retrieval) problem to ensure secure requests to a database hosted on the cloud in the presence of an active adversary who does not interfere with the PIR protocol, but can carry out an attack with known open requests. To represent the bit number i as a number, all digits of which are different, the proposed algorithms use a base l number system with the number of digits d . The permutations of the digits of the requested bit number which are regarded as secret encryption keys were used. To reduce the communication complexity the bits of the source database stored on the cloud are grouped as arrays. A pseudorandom number sensor is used to replace the bit value depending on the number i requested by the client. This makes it difficult to match the bit value to a specific number in the case of collusion between a passive adversary located on the cloud and an active adversary outside the cloud. The communication complexity and the probability of guessing the bit number in a single attack with a known open query for bit number i , as well as in an attack with an unlimited number of known open queries, are estimated.

Keywords: database; cloud computing; PIR.

For citation: Varnovskiy N.P., Martishin S.A., Khrapchenko M.V., Shokurov A.V. Optimizing confidential database queries on the cloud. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 39-58 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-3.

Acknowledgements. The authors are especially grateful to D.O. Lazarev for valuable comments during the work on the article.

1. Введение

В настоящее время при использовании облачных сервисов для хранения и обработки информации наиболее актуальным является вопрос работы с конфиденциальными данными. В работах [1,2] авторы исследовали задачу организации конфиденциальных запросов к базе данных (PIR – Private Information Retrieval) в случае размещения копий базы данных на облаке и наличия активного противника, работающего по протоколу, но имеющему возможность производить атаку с известными открытыми запросами. В данной статье под облаком будем понимать сеть удаленных серверов, администрирование которых осуществляется только их владельцами и не доступно для других.

Классическая постановка задачи PIR была сформулирована в 1995 году Шором, Голдрайхом, Кушелевицем и Суданом [3]: имеется база данных – бинарная строка $X = (x_1, \dots, x_n)$ длины n , хранящаяся на сервере; клиент хочет получить один бит информации x_i из базы данных X так, чтобы никто, кроме клиента, обращающегося с запросом, не смог определить, для какого индекса i был запрошен бит.

Чтобы PIR-протокол можно было бы применять на практике, он должен удовлетворять условиям:

корректности: для любого номера клиент может сформировать такой запрос, что после завершения выполнения протокола клиент получит данные, по которым он может правильно вычислить значение бита;

конфиденциальности: в результате выполнения протокола PIR противник не получает никакой информации о номере запрашиваемого бита из базы данных;

лаконичности: коммуникационная сложность протокола (общее количество бит, которыми обмениваются участники протокола за время его работы) должна быть существенно меньше размера базы данных.

Было показано, что при размещении базы данных на единственном сервере, который полностью контролируется противником, теоретико-информационное условие конфиденциальности корректного протокола PIR может быть выполнено только в том случае, когда клиент запрашивает базу данных целиком. Однако при этом коммуникационная сложность протокола будет не меньше размера базы данных.

Для уменьшения коммуникационной сложности в работах [3, 4] была предложена модель, в которой несколько одинаковых копий строки $X = (x_1, \dots, x_n)$ размещалось на разных серверах. Предполагалось, что клиент имеет связь с каждым из этих серверов, но сами серверы не имеют связи друг с другом. Также предполагалось, что противник может наблюдать любой из серверов, на которых размещена база данных, причем, возможно, разные серверы во время выполнения разных сеансов протокола. Было доказано, что для моделей с k копиями базы данных возможно построение корректных конфиденциальных лаконичных протоколов PIR.

Дальнейшие исследования в этом направлении были сосредоточены на изучении методов улучшения коммуникационной сложности [5, 6]. Кроме того, в работах [7-10] были исследованы возможности построения и оценки коммуникационной сложности (или лаконичности) схемы PIR на одном сервере при помощи различных техник, в том числе гомоморфного шифрования.

Однако, если рассматривать задачу построения PIR в предположении о том, что копии базы данных хранятся на облаке, то очевидно, в облачных информационных системах нельзя обеспечить изолированность нескольких копий базы данных друг от друга. Также нельзя исключить обмен информацией между копиями базы данных и наличие противника. При этом ни владелец данных (пользователь), ни клиент (имеющий официальный доступ для получения информации из базы данных) не могут контролировать облако.

Для облачных вычислений исследовались различные применения протокола PIR на практике. Например, в [11] рассматривалась реализация простой схемы PIR с несколькими серверами, аналогично построению систем RAID (Redundant Array of Inexpensive Disks) с избыточными массивами недорогих дисков. При таком подходе каждый сервер хранит только часть базы данных, и серверы могут быть опрошены параллельно. Предполагается, что не все серверы вступают в сговор, поскольку серверы могут работать на разных платформах и/или под управлением различных облачных провайдеров.

Также большое внимание было уделено исследованию различных методов, повышающих эффективность применения на практике PIR (CPIR – computational PIR). Например, в [12] рассматривались два метода: эффективное использование центрального процессора сервера и кодирование данных PBC (probabilistic batch codes – вероятностные пакетные коды), который используется для создания схемы PIR, предназначенной для обработки множества запросов. На основе этих методов была разработана библиотека CPIR – SealPIR, которая объединяет наиболее эффективный в вычислительном отношении протокол CPIR и предлагаемый метод сжатия запросов, что уменьшает коммуникационные затраты.

Из практических реализаций также стоит отметить MuchPIR – поиск частной информации с использованием гомоморфного шифрования, реализованный как расширение C/C++ Aggregate для Postgres [13]. Также полезным на практике может оказаться использование Piano – односерверной схемы PIR с сублинейным временем [14]. В отличие от классической схемы PIR, которая не выполняет никакой предварительной обработки базы данных, и сервер просто хранит исходную копию базы данных DB, односерверная схема PIR под названием Piano (сокращение от «Private Information Access NOW»), выполняет предварительную обработку запросов в сети Интернет. Такая схема может использоваться для сохранения

конфиденциальной информации при извлечении данных из блокчейна или создания безопасного DNS сервера.

Существует также достаточно большое количество статей, посвященных использованию в алгоритмах схемы PIR и гомоморфного шифрования, оценки вычислительных затрат и затрат на коммуникацию. Однако основным отличием исследуемой ниже модели является изучение ее характеристик с точки зрения математики распределённых систем при наличии противника, который не только пассивно наблюдает, но и может производить атаку с известными открытыми запросами.

Как и в работах [1, 2] рассматривается модель, включающая в себя облако, состоящее из нескольких серверов, на каждом из которых хранится копия одной и той же базы данных, дилера, пользователей, клиентов, противника. Будем предполагать, что серверы соединены посредством незащищенных каналов связи друг с другом и дилером. Облачные серверы и незащищенные каналы связи доступны для стороннего наблюдателя. Дилер находится вне облака, противнику недоступен. Предполагается, что на облаке имеется противник, который не только пассивно наблюдает, но производит атаку с известными открытыми запросами.

В отличие от работ [1, 2], где основной функцией дилера являлось шифрование и дешифрование информации при работе с облаком, в данной статье предлагается вместо шифрования использовать перестановки цифр числа как одностороннюю функцию. Применяется новая система счисления для представления номера бита и далее производятся перестановки цифр в новой системе счисления для запрашиваемого номера бита. Этот подход будет использоваться при формировании запроса к облаку, обработке ответа облака и получения значения искомого бита.

Для сокрытия при запросе искомого номера бита, вместо одного запрашиваемого номера бита дилер будет запрашивать у облака множество битов. Запрашиваемые множества массивов битов строятся независимо друг от друга. По одному такому множеству противник не может сделать вывод о другом множестве и сделать вывод об алгоритме их построения и запрашиваемом номере бита.

Предлагается использовать датчик псевдослучайных чисел для замены значения бита на противоположное при хранении номера бита на облаке. Перестановка цифр в преобразованном новом номере бита затрудняет противнику поиск рядом стоящих запрашиваемых битов.

Получены оценки вероятности угадывания номера бита при однократной атаке с известным открытым запросом номера бита i и при атаке с неограниченным числом известных открытых запросов при наличии противника на облаке. Приведена оценка коммуникационной сложности алгоритма.

2. Модель вычислений, основные определения и постановка задачи

2.1 Состав модели

Модель вычислений, для которой в данной статье исследуется облачный вариант задачи PIR. Модель включает в себя:

Облако. Состоит из нескольких удаленных серверов, администрирование которых осуществляется только их владельцами и не доступно для других. На облаке хранятся копии одной и той же базы данных. Копии на серверах имеют разные перестановки номеров массивов битов. Серверы соединены посредством незащищенных каналов связи друг с другом и дилером.

Пользователь. Хранит данные на облаке. Предполагается, что на облаке хранится k копий баз данных. Для загрузки данных пользователь обращается к дилеру по каналу связи, использующему стандартный криптографический протокол для обмена зашифрованными данными.

Клиенты. Запрашивают некоторую информацию из базы данных. Обращаются для выполнения запроса к дилеру. С дилером соединены каналами связи, использующими стандартный криптографический протокол для обмена зашифрованными данными. Клиенты не доверяют облаку и доверяют только дилеру.

Дилер. Находится вне облака, противнику недоступен. Представляет собой доверенный сервер, который в том числе осуществляет аутентификацию пользователей и клиентов. Канал связи с облаком является незащищенным. Объем памяти дилера для постоянного хранения данных пренебрежимо мал по сравнению с n . Получив от клиента данные для размещения на облаке, дилер выполняет перестановку уникальную для каждой копии базы данных, размещает данные на облаке. При запросе клиента дилер возвращает клиенту значение бита. В дальнейшем для простоты будем считать, что база данных загружается одним пользователем. Предполагается, что после загрузки база данных не может быть изменена. Либо может быть изменена полностью. Также будем рассматривать случай запроса одного бита.

2.2 Основные определения

Противник. Не вмешивается в выполнение криптографического протокола. Имеет доступ к базе данных на облаке, к каждой ее копии, к каналам связи на облаке. Может создавать фальшивых клиентов, работающих по протоколу. То есть противник не только пассивно наблюдает, но и сам может производить атаку с известными открытыми запросами. Противник может через дилера отправить запрос на облако. При этом противник знает алгоритм формирования запроса к облаку, но не знает конкретных параметров. Противник также знает алгоритм формирования ответа клиенту дилером на основании ответа, полученного дилером от облака.

Угроза: угадывание исходного номера бита, запрашиваемого клиентом в базе данных.

Атака:

- Атака пассивного противника на облаке заключается в наблюдении за запросом от дилера и ответом облака дилеру. Эта информация используется для сбора статистики и анализа.
- Атака активного противника (атака с известными открытыми запросами) заключается в создании фальшивых клиентов и управление ими. При помощи фальшивых клиентов активный противник может сформировать произвольное число запросов, что позволит в коалиции с пассивным противником на облаке собрать и проанализировать информацию для всей базы данных.

2.3 Постановка задачи и основные обозначения

В отличие от классической постановки задачи PIR будем рассматривать базу данных (бинарная строка) $X = (x_0, \dots, x_{n-1})$ длины n , где элементы X нумеруются с нуля. Это делает более удобной работу с различными системами счисления.

Клиент хочет получить один бит информации x_i с номером i из базы данных X так, чтобы противник не узнал ничего о том, для какого индекса i был запрошен бит.

Будем размещать на облаке k копий баз данных. Пусть $k = 2^d$, где $d \geq 2$. На практике $4 \leq k \leq 32$. В дальнейшем будем предполагать, что на k наложены эти ограничения.

Без потери общности предполагается, что $n = l^d$, то есть, $d = \log_2 k$ и $l = \sqrt[d]{n}$. Заметим, что поскольку l - основание системы счисления, то $l \geq 2$ и d подбирается так, чтобы $l \ll n$. Пусть $L_p = \frac{n}{l}$.

Так как $l = \sqrt[d]{n}$, округлим l до целого числа в большую сторону.

Пусть x – элемент группы Z_2 , обозначим через h ($h \in \{1, \dots, k\}$) – номер копии базы данных.

3. Представление номера бита в l -ичной системе счисления

Представим номер бита как d разрядов в l -ичной системе счисления. Выбор l и d основывается на известной модели [1, 2], где номер i бита x_i представлен в l -ичной записи: $i = a_0^{(i)} + a_1^{(i)}l + \dots + a_{d-1}^{(i)}l^{d-1}$. Цифры l -ичной записи представляют собой элементы кортежа $(a_0^{(i)}, \dots, a_{d-1}^{(i)})$ длины d , то есть на j -м месте кортежа стоит l -ичная цифра.

Эти d элементов кортежа можно интерпретировать как точку с целочисленными координатами в гиперкубе размерности d и длиной стороны l , где $l = \sqrt[d]{n}$. Такое множество точек куба размерности d можно рассматривать как множество слов длины d с алфавитом $(0, \dots, l-1)$.

4. Загрузка данных на облако

4.1 Инициализация массива X^a

Напомним, что $n = l^d$ и $L_p = \frac{n}{l}$.

Выберем целые числа $d_u \geq 2$ и $l_b \geq 2$ такие, чтобы $l_b^{d_u} \leq L_p$ и обозначим $L_p^- = l_b^{d_u}$ ($L_p^- \leq L_p$).

Пусть $l^* = \left\lceil \frac{n}{L_p^-} \right\rceil$, где $(l^* \geq l)$.

По аналогии с алгоритмом, изложенном в [2], для каждого из l^* интервалов применим алгоритм преобразования содержащихся в нем слов в слова без повторяющихся букв (цифр). В результате каждому номеру бита в l_b -ичной системе счисления будет соответствовать номер бита в $(l_b + d_u - 1)$ -ичной системе счисления. То есть номер бита можно интерпретировать как точку в гиперкубе размерности d_u со стороной $l_u = (l_b + d_u - 1)$. Количество битов в каждом интервале увеличится и станет равным L_p^+ ($L_p^+ = l_u^{d_u}$ и $L_p^+ > L_p^-$). Пусть $n_+ = l^* \cdot L_p^+$.

Ниже будет показано, что коммуникационная сложность зависит от l^* . Для того, чтобы коммуникационная сложность не возрастала, нужно подобрать значения d_u и l_u так, чтобы минимизировать разность $L_p - L_p^-$. Конкретный выбор значений d_u и l_u , который не только минимизирует разность $L_p - L_p^-$, но и уменьшает количество вычислительных операций, производимых дилером, будет описан ниже.

Рассмотрим интервал, содержащий L_p^+ элементов. Каждый номер бита, попавший в этот интервал, в l_u -ичной записи состоит из элементов кортежа длины d_u . Для представления номера бита используются неповторяющиеся цифры.

Рассмотрим номера из одного интервала. В разрядах этих номеров стоят различные цифры из одного множества различных цифр. Заметим, что из множества d_u различных цифр можно получить $d_u!$ перестановок цифр.

Обозначим через $n_u^- = \frac{L_p^+}{d_u!}$ – количество множеств различных цифр мощностью d_u . Как было показано в [2] все преобразованные номера, содержащие одинаковые цифры, лежат в одном интервале мощностью L_p^+ .

Значения битов для номеров с совпадающим набором цифр, можно хранить в виде массива. Отсортируем перестановки (например, по возрастанию) цифр каждого номера. Каждая перестановка цифр будет иметь номер в соответствующем интервале. Этот номер ставится в соответствие массиву битов для номеров, имеющих одинаковые цифры.

Тогда значения битов, соответствующих данному номеру, будут располагаться в массиве битов в соответствии с сортировкой цифр номера.

Будем в дальнейшем хранить одну перестановку, соответствующую минимальному номеру.

Таким образом, вместо номеров с одинаковым набором различных цифр будем хранить один отсортированный набор цифр, что в $d_u!$ раз меньше количества номеров. Массив битов будет содержать $d_u!$ элементов.

Каждый элемент массива битов преобразуется с использованием датчика псевдослучайных чисел $PRNG1(db, i)$ в зависимости от запрашиваемого клиентом номера i . Датчик псевдослучайных чисел возвращает 0 или 1. Если возвращен 0, то значение бита не меняется, а если 1, то значение бита меняется на противоположное. Это необходимо, чтобы в случае сговора, пассивного противника, находящегося на облаке и видящего базу данных, и активного противника вне облака, нельзя было сделать вывод о соответствии значения бита конкретному номеру.

Пусть $\hat{n} = l^* \cdot n_u^-$.

Пусть $n_a = l_a^{d_a}$, где l_a и d_a целые числа. Выберем l_a и d_a такие, чтобы n_a было максимально близким к $\hat{n}$. Тогда $\hat{n} \leq n_a$.

Обозначим массив битов, соответствующий группе номеров, состоящих из одинаковых цифр, через $x_{i_k}^a$. Обозначим через n_c количество битов, содержащихся во всех n_a массивах, тогда $n_c = n_a \cdot d_u!$.

Преобразуем номера массивов битов в l_a -ичную систему счисления.

На облако будет загружаться последовательность массивов битов $X^a = (x_0^a, \dots, x_{n_a-1}^a)$. Нумерация массива X^a начинается с нуля. Количество элементов массива битов $x_{i_k}^a$ равно $d_u!$. Каждому массиву битов $x_{i_k}^a$ соответствует $d_u!$ номеров битов.

Так как последовательность массивов битов X^a упорядочена, не нужно хранить номера массивов битов для доступа к ним.

4.2 Построение матрицы перестановок

При запросе клиентом номера бита i находится номер интервала, в котором будет содержаться i и номер массива битов в интервале. На основе этих двух номеров формируется номер массива в последовательности X^a .

Номер запрашиваемого клиентом массива битов в l_a -ичной системе счисления можно представить как целочисленную точку гиперкуба размерности d_a с длиной стороны l_a . При этом j -ая ($j \in \{1, \dots, d_a\}$) координата точки гиперкуба является j -ым разрядом представления числа в l_a -ичной системе счисления.

Изменим номера массивов битов путем изменения цифр в разрядах номеров в l_a -ичном представлении. Цифры в каждом разряде изменяются при помощи перестановок цифр от 0 до $l_a - 1$. Для каждого разряда вычисляется своя перестановка цифр. Поскольку перестановки являются случайными, то для разных разрядов возможно, что перестановки совпадают.

Выберем копию базы данных $h \in \{1, \dots, k\}$. Пусть β_{hj} - случайная перестановка из l_a чисел от 0 до $l_a - 1$, где $j \in \{1, \dots, d_a\}$ - номер разряда числа в номере массива битов в l_a -ичной системе счисления. Представим перестановку чисел в виде таблицы:

$$\begin{pmatrix} 0 & \dots & l_a - 1 \\ \beta_{hj}(0) & \dots & \beta_{hj}(l_a - 1) \end{pmatrix}.$$

Тогда $\beta_h = \{\beta_{h1}, \dots, \beta_{hd_a}\}$ - множество «поразрядных» перестановок числа в l_a -ичной системе счисления.

Построим матрицу M_β размера $k \times d_a$, где k - число копий базы данных, а d_a целое число, такое, что $n_a = l_a^{d_a}$. Элементом β_{ij} матрицы M_β является случайная перестановка целых чисел от 0 до $l_a - 1$. Таким образом, для каждой копии базы данных будет построено d_a случайных перестановок.

Для генерации перестановок цифр номера бита l_a -ичной системе счисления используется датчик псевдослучайных чисел $PRNG2(db, m)$. На вход $PRNG2(db, m)$ подается уникальное

значение db для конкретной базы данных и номер копии базы данных m . Датчик псевдослучайных чисел при одинаковой инициализации генерирует одинаковую перестановку.

Поскольку матрица M_β требует для хранения значительного объема памяти, матрица M_β при необходимости восстанавливается при помощи датчика псевдослучайных чисел.

4.3 Отображение $i_a \xrightarrow{\beta_h} i_c$

Для запрашиваемого клиентом номера бита i найдем соответствующий номер массива i_a в X^a .

Номер массива i_a в X^a представляем в системе счисления по основанию l_a с числом разрядов d_a :

$$i_a = \sum_{j=1}^{d_a} a_j \cdot l_a^{j-1}, \text{ где } a_j \in \{0, \dots, l_a - 1\}.$$

Для копии базы данных с номером h выполним перестановки для разрядов числа i_a .

$$i_c = \sum_{j=1}^{d_a} \beta_{hj}(a_j) \cdot l_a^{j-1}.$$

Обозначим такое преобразование через $i_a \xrightarrow{\beta_h} i_c$, где $\beta_h = \{\beta_{h1}, \dots, \beta_{hd_a}\}$ – строка матрицы M_β .

Необходимые для работы алгоритма элементы матрицы M_β генерируются при помощи датчика псевдослучайных чисел.

4.4 Вспомогательные построения, необходимые для работы алгоритма запроса бита – алгоритм построения матрицы G

Построим матрицу G размера $n_a \times 2$.

$$G = \begin{pmatrix} 0 & x_0^a \\ \vdots & \vdots \\ n_a - 1 & x_{n_a-1}^a \end{pmatrix}$$

Первый столбец этой матрицы – последовательность n_a целых чисел $[0, n_a - 1]$. Второй столбец матрицы G – элементы $X^a = (x_0^a, \dots, x_{n_a-1}^a)$. Далее на основании матрицы G будут построены матрицы G_h , которые будут загружены на облако.

Алгоритм построения матрицы G

Вход: X^a .

Выход: Матрица G .

Шаг 1. В первый столбец матрицы G заносится последовательность n_a целых чисел $[0, n_a - 1]$.

Шаг 2. Во второй столбец матрицы G заносятся элементы $X^a = (x_0^a, \dots, x_{n_a-1}^a)$ ■.

4.5 Формирование матриц G_h для каждой копии базы данных из матрицы G

Алгоритм формирования матриц G_h

Вход: матрица G .

Выход: матрицы G_h для всех $h \in \{1, \dots, k\}$.

Шаг 1. Выберем копию базы данных h .

Шаг 2. Скопируем матрицу G во вспомогательную матрицу G_{aux} .

Шаг 3. Для каждого элемента i_a , где $i_a = 0, \dots, n_a - 1$ первого столбца матрицы G_{aux} выполним преобразование $i_a \xrightarrow{\beta_h} i_c$.

Шаг 4. Отсортируем строки матрицы G_{aux} по первому столбцу. Получим матрицу G_h .

Шаг 5. Выполним Шаги 1-4 для всех $h \in \{1, \dots, k\}$ ■.

В результате работы алгоритма получим G_h ($h \in \{1, \dots, k\}$) матриц для каждой копии базы данных.

4.6 Загрузка данных на облако

Алгоритм загрузки данных на облако аналогичен алгоритму, приведенному в [1, 2].

На вход алгоритма подается бинарный массив $X = (x_1, \dots, x_n)$. Для удобства будем предполагать, что нумерация элементов начинается с нуля $X = (x_0, \dots, x_{n-1})$. В результате работы алгоритма на облако загружаются k преобразованных при помощи перестановки копий массивов битов баз данных.

Алгоритм загрузки данных на облако

Вход: X .

Выход: загрузка на облако 2-го столбца матрицы G_h , где $h \in \{1, \dots, k\}$.

Шаг 1. Дилер аутентифицирует пользователя. Пользователь передает дилеру массив значений битов $X = (x_0, \dots, x_{n-1})$. Дилер преобразует массив $X = (x_0, \dots, x_{n-1})$ в массив $X^a = (x_0^a, \dots, x_{n_a-1}^a)$.

Шаг 2. Дилер формирует матрицу $G = \|g_{i,j}\|$, где $i = 0, \dots, n_a - 1$, $j = 1, 2$.

Шаг 3. Дилер генерирует элементы матрицы M_β в процессе работы.

Шаг 4. Дилер создает матрицу G_h , где $h \in \{1, \dots, k\}$.

Шаг 5. Дилер загружает 2-ой столбец матрицы G_h на облако, где $h \in \{1, \dots, k\}$.

Шаг 6. Шаги 4-5 дилер выполняет в цикле для каждой копии базы данных. По окончании перебора k копий базы данных происходит выход из алгоритма ■.

Число битов, переданных каждой из k копий базы данных равно $n_a \cdot d_u!$.

Заметим, что поскольку в дальнейшем восстановление матрицы G не предполагается, матрицу G дилер не хранит.

Для выполнения перестановки дилеру требуется иметь объем памяти, сравнимый с объемом, необходимым для хранения одной копии базы данных. При этом предполагается, что дилер может работать с несколькими базами данных и нет необходимости их одновременного хранения. После загрузки базы данных на облако память дилера очищается для работы со следующей базой данных.

5. Выполнение дилером запроса клиента

Для сокрытия при запросе искомого номера бита, вместо одного запрашиваемого номера бита дилер будет запрашивать у облака множество битов мощностью $l^* \cdot d_u!$. Причем только один из этих битов был запрошен клиентом.

При запросе искомого номера бита вычисляется номер массива $x_{i_k}^a$, содержащего этот бит.

Для сокрытия при запросе искомого номера массива $x_{i_k}^a$, вместо одного запрашиваемого номера массива дилер будет запрашивать у облака множество номеров массивов мощностью l^* .

Каждый номер запрашиваемого массива битов принадлежит только одному из интервалов мощности n_u^- .

5.1 Использование множества номеров массивов битов в запросе для сокрытия номера бита

Пусть клиент интересуется элементом $x_i \in X$, где $0 \leq i < n$. Напомним, что $n_+ = l^* \cdot L_p^+$. Обозначим через $X^+ = (x_0^+, \dots, x_{n_+-1}^+)$.

Отрезок целых чисел от 0 до $n_+ - 1$ содержит l^* интервалов по L_p^+ элементов в каждом из них.

Номер бита i лежит ровно в одном из этих интервалов. Далее для массива в X^a , содержащего бит i , выберем по одному массиву битов в каждом из оставшихся $l^* - 1$ интервалов (алгоритм

выбора массивов описан в п. 5.2). Таким образом, для каждого массива битов выбираются $l^* - 1$ массивов битов, лежащих в различных интервалах и не попадающих в интервал, где лежит массив, содержащий число i .

Как было сказано выше, каждый интервал можно интерпретировать как дискретный гиперкуб размерности d_u с длиной стороны l_u .

Выберем интервал $m \in \{1, \dots, l^*\}$. Пусть σ_{mj} - случайная перестановка из l_u чисел от 0 до $l_u - 1$, где $j \in \{1, \dots, d_u\}$ - разряд числа в l_u -ичной системе счисления. Как и ранее, запишем перестановку чисел в виде:

$$\begin{pmatrix} 0 & \dots & l_u - 1 \\ \sigma_{mj}(0) & \dots & \sigma_{mj}(l_u - 1) \end{pmatrix}.$$

Тогда $\sigma_m = \{\sigma_{m1}, \dots, \sigma_{md_u}\}$ - множество «поразрядных» перестановок числа в l_u -ичной системе счисления d_u -разрядного числа.

Построим матрицу M_σ (по аналогии с матрицей M_β) размера $l^* \times d_u$, где l^* - число интервалов, а d_u целое число, такое, что $L_p^+ = l_u^{d_u}$. Элементом σ_{ij} матрицы M_σ является случайная перестановка целых чисел от 0 до $l_u - 1$. Таким образом, для каждого интервала на которые разбивается база данных будет построено d_u случайных перестановок.

Для генерации перестановок цифр номера бита матрицы M_σ используется датчик псевдослучайных чисел $PRNG3(db, m, j)$. На вход $PRNG3(db, m, j)$ подается уникальное значение db для конкретной базы данных, номер интервала m (всего l^* штук) и $j \in \{1, \dots, d_u\}$ - разряд числа в l_u -ичной системе счисления. Датчик псевдослучайных чисел при одинаковой инициализации генерирует одинаковую перестановку.

Поскольку матрица M_σ требует для хранения значительного объема памяти, элементы матрицы M_σ восстанавливаются при помощи датчика псевдослучайных чисел $PRNG3(db, m, j)$.

Поскольку отрезок целых чисел от 0 до $n_+ - 1$ разбивается на l^* интервалов по L_p^+ элементов в каждом интервале, то любое число из этого множества попадет в один из таких интервалов. Найдем номер интервала w , в который попало число i при разбиении отрезка целых чисел от 0 до $n - 1$ на l^* интервалов:

$$w = \left\lfloor \frac{i}{L_p^+} \right\rfloor + 1.$$

Заметим, что $w \in \{1, \dots, l^*\}$.

Приведем i по модулю L_p^- . Обозначим через $i^- = i \bmod L_p^-$ результат приведения. Тогда i^- является порядковым номером в интервале с номером w .

Представим порядковый номер i^- в интервале с номером w в l_u -ичной системе счисления: $i^+ = \sum_{j=1}^{d_u} a_j \cdot l_u^{j-1}$, где $a_j \in \{0, \dots, l_u - 1\}$.

Рассмотрим сумму $\sum_{j=1}^{d_u} (\sigma_{wj})^{-1}(a_j) \cdot l_u^{j-1}$, где $(\sigma_{wj})^{-1}(a_j)$ обратная перестановка, такая, что $\sigma_{wj}((\sigma_{wj})^{-1}(a_j)) = a_j$.

Обозначим через $D_j = (\sigma_{wj})^{-1}(a_j)$, величина D_j - секрет, известный только дилеру.

В каждом интервале m ($m \in [1, l^*]$) найдем число, такое, что $i^+(m) = \sum_{j=1}^{d_u} \sigma_{mj}(D_j) \cdot l_u^{j-1}$, где $j = 1, \dots, d_u$. Для всех интервалов таких чисел $i^+(m)$ будет l^* . Заметим, что для интервала w это будет число i^+ .

Обозначим через Set_i полученное по алгоритму 5.2 (Алгоритм построения множества Set_i дилером) множество из l^* чисел.

5.2 Построение множеств Set_i дилером

Алгоритм построения множества Set_i дилером

Вход: $i, l^*, d_u, l_u, d_a, l_a, L_p^-$.

Выход: множество Set_i .

$Set_i = \emptyset$

// найдем номер интервала

$w \leftarrow \left\lfloor \frac{i}{L_p^-} \right\rfloor + 1$

// найдем порядковый номер числа i

// в соответствующем интервале разбиения

$i^- = i \bmod L_p^-$

// найдем представление числа i^- в l_u -ичной записи

$i^+ = \sum_{j=1}^{d_u} a_j \cdot l_u^{j-1}, a_j \in \{0, \dots, l_u - 1\}$

// найдем координаты $D_j \in \{0, \dots, l_u - 1\}$

for $j \leftarrow 1$ to d_u do

$D_j = (\sigma_{wj})^{-1}(a_j)$

for $m \leftarrow 1$ to l^* do

$i^+(m) = \sum_{j=1}^{d_u} \sigma_{mj}(D_j) \cdot l_u^{j-1}$

// находим номер массива в X^a

// для номера элемента в X^+

$i_m = \text{номер}(x_{l_k}^a) \leftarrow L_p^+ \cdot m + i^+(m)$

$Set_i = Set_i \cup i_m$

Листинг 1. Алгоритм построения множества Set_i

5.3 Запрос клиентом бита с номером i . Предварительные замечания

Пусть клиент запрашивает бит с номером i .

После подтверждения полномочий клиента дилер разрешает клиенту отправить запрос.

Напомним, что физически в облаке хранятся 2-е столбцы матриц G_h , $h \in \{1, \dots, k\}$. Второй элемент в строке матрицы G_h представляет собой массив битов. Без потери общности (как было сказано выше) будем индексировать элементы базы данных с 0.

Также как и в [1, 2] для каждого элемента множества Set_i случайно и равномерно выбирается номер копии базы данных. Все элементы множества Set_i для которых выбрана копия базы данных h , обозначим через множества Set_i^h , где $h \in \{1, \dots, k\}$. Очевидно, что множества Set_i^h не пересекаются между собой, поскольку все элементы множества Set_i различны и каждому элементу ставится в соответствие ровно один номер h (номер копии базы данных). Объединение множеств Set_i^h содержит все элементы множества Set_i , то есть $Set_i = \cup Set_i^h$, где $h \in \{1, \dots, k\}$.

Пусть для запрашиваемого номера i в множестве Set_i была выбрана копия базы данных h .

Напомним, что элементами множества Set_i^h являются номера массивов битов множества Set_i для которых была выбрана копия базы данных h .

Для каждого элемента множества Set_i^h выполняется перестановка β^h , пусть $y = \beta^h(i_a)$. Полученные после перестановки элементы сортируются. Это множество обозначим через $Set_i^{\beta^h}$. Позиция i_{pos} для исходного запрашиваемого элемента $y = \beta^h(i_a)$ в множестве $Set_i^{\beta^h}$ запоминается.

Дилер на время выполнения запроса формирует вектор-строку s_{num} из четырех элементов $s_{num} = (s_1, s_2, s_3, s_4)$:

- Первый элемент вектора-строки s_{num} содержит номер бита i .
- Второй элемент вектора-строки s_{num} содержит номер элемента в возвращаемом массиве.
- Третий элемент вектора-строки s_{num} содержит номер копии h базы данных, в которую попал номер i_a .
- Четвертый элемент вектора-строки s_{num} содержит позицию i_{pos} номера u в множестве $Set_i^{\beta^h}$.

Вектор-строка s_{num} позволяет дилеру после получения ответа от облака сразу отбросить данные, кроме данных, необходимых для выполнения запроса клиента.

Вектор-строка s_{num} формируется у дилера и известна только дилеру. Вектор-строка s_{num} является секретной.

На k копий базы данных дилер отправляет в общей сложности l^* номеров.

После выполнения запроса дилер при помощи вектора-строки s_{num} получает значение искомого элемента и отправляет его клиенту.

Алгоритм подготовки запроса к облаку

Вход: $i, l^*, d_u, l_u, d_a, l_a, L_p^-$.

Выход: $Set_i^{\beta^h}, s_{num}$.

Шаг 1. Клиент обращается к дилеру и запрашивает значение бита с номером i .

Шаг 2. Дилер генерирует множество Set_i .

Шаг 3. Дилер на время выполнения запроса i -го бита резервирует память для вектора-строки s_{num} из четырех элементов.

Шаг 4. Дилер заносит номер бита в массиве i_a во второй элемент вектора-строки s_{num} .

Шаг 5. Дилер выполняет разбиение множества Set_i на непересекающиеся подмножества Set_i^h , где $h \in \{1, \dots, k\}$, путем случайного и равномерного выбора номера копии базы данных для каждого элемента множества Set_i .

Дилер заносит в третий элемент вектора-строки s_{num} номер копии базы данных s_3 , соответствующей искомому в множестве элементу i_a .

Шаг 6. Дилер выполняет перестановку β^h для элементов множества Set_i^h после чего полученные после перестановки новые элементы сортируются. Получаем множество $Set_i^{\beta^h}$. Этот шаг выполняется для всех $h \in \{1, \dots, k\}$.

Шаг 7. Дилер заносит в четвертый элемент вектора-строки s_{num} позицию i_{pos} номера $u = \beta^h(i_a)$ для множества $Set_i^{\beta^h}$.

5.4 Выполнение запроса к облаку и ответ облака

Алгоритм обмена информацией с облаком

Вход: $Set_i^{\beta^h} (h \in \{1, \dots, k\})$.

Выход: значения массивов битов в том же порядке, в котором передавались номера массивов битов множества $Set_i^{\beta^h}$ для каждой копии базы данных ($h \in \{1, \dots, k\}$).

Шаг 1. Для каждой копии базы данных дилер отправляет на облако множества $Set_i^{\beta^h}$.

Шаг 2. Для запрошенных номеров массивов битов облако возвращает дилеру значения массива битов, соответствующих второму столбцу матрицы G_h , где $h \in \{1, \dots, k\}$. Порядок возвращаемых массивов значений битов соответствует исходному порядку элементов множества $Set_i^{\beta^h}$.

5.5 Обработка ответа облака

Алгоритм обработки ответа облака

Вход: s_{num} , значения массивов битов в том же порядке, в котором передавались номера массивов битов множества $Set_i^{\beta^h}$ для каждой копии базы данных ($h \in \{1, \dots, k\}$).

Выход: значение бита i .

Шаг 1. Дилер при помощи элемента s_3 вектора-строки s_{num} выбирает возвращаемые значения из соответствующей базы данных h . Значения, возвращаемые из других баз данных, отбрасываются.

Шаг 2. Дилер при помощи элемента s_4 вектора-строки s_{num} выбирает возвращаемый массив битов из соответствующей базы данных h , стоящий на позиции i_{pos} . Остальные массивы значений битов отбрасываются.

Шаг 3. Дилер для выбранного массива значений битов при помощи элемента s_2 вектора-строки s_{num} получает значение бита. На основе датчика случайных чисел $PRNG1(db, i)$ дилер восстанавливает значение искомого бита.

Шаг 4. Дилер отправляет значение i бита клиенту ■.

6. Оценка требуемой памяти и сложности алгоритма

6.1 Объем информации, который необходимо хранить дилеру

На протяжении всего времени существования базы данных дилер хранит информацию для базы данных объема n :

- Значения $n, k, l^*, d_a, l_a, d_u, l_u, L_p^-$;
- Значение db для инициализации датчиков псевдослучайных чисел $PRNG1(db, i)$, $PRNG2(db, m)$ и $PRNG3(db, m, j)$;
- Вектор-строку s_{num} .

Объем хранимой дилером информации много меньше n .

6.2 Характеристики предложенной схемы

В данной схеме рассматривается активный противник, работающий по протоколу. Противник имеет доступ к базе данных на облаке, к каждой ее копии, к каналам связи на облаке, может создавать фальшивых клиентов, работающих по протоколу. То есть противник не только пассивно наблюдает, но и сам может производить запросы с помощью созданных фальшивых клиентов, то есть производит атаку с известными открытыми запросами. Противник не имеет доступа к дилеру.

6.3 Основные результаты

Под коммуникационной сложностью для предложенной схемы будем понимать общее количество пересылаемых битов, необходимых для обмена информацией между дилером и облаком. То есть сумму числа битов, отправляемых дилером на облако и числа битов, получаемых от облака дилером, необходимых для нахождения дилером значения бита, запрашиваемого у дилера. Пусть s_a – число битов для представления номера массива битов. Напомним, что l^* является мощностью множества Set_i .

Утверждение 1. Коммуникационная сложность схемы получения значения номера бита дилером без раскрытия его номера равна $l^* \cdot (s_a + d_u!)$ битов.

Доказательство. Для запроса значения бита дилер посылает копиям базы данных l^* номеров массивов битов, каждый номер такого массива имеет длину s_a битов. Облако отвечает l^* массивами битов, длина каждого массива $d_u!$ битов ■.

Число битов для представления номера массива битов меньше в $d_u!$ раз числа битов для представления номера бита. Таким образом, коммуникационная сложность не меняется по сравнению с алгоритмом передачи номера бита и возвращением бита.

Проанализируем вероятность угадывания противником номера запрашиваемого бита.

Если запросы выполнены к одному множеству, то они неразличимы. Таким образом, если противник совершает n или более запросов, он не узнает номер конкретного бита. Максимум, какую информацию противник может получить – биты, содержащиеся массивах с номерами множества Set_i , хотя был запрошен только один бит. Что будет означать, что противник не знает, а только угадывает, какой именно бит для Set_i был запрошен.

Утверждение 2. При однократной атаке с известным открытым запросом номера бита i и предположении о наличии пассивного противника на облаке вероятность угадывания противником номера бита не более $\frac{1}{l^* \cdot d_u!}$.

Доказательство. Множество Set_i состоит из l^* массивов битов и для любого бита $j \in Set_i$ выполняется $Set_i = Set_j$, если множество Set_i тоже содержит этот бит. То есть существует одинаковое множество для $l^* \cdot d_u!$ различных номеров битов. Таким образом вероятность угадывания номера бита из множества Set_i равна $\frac{1}{l^* \cdot d_u!}$, так как массивы с номерами множества Set_i содержат в общей сложности $l^* \cdot d_u!$ битов ■.

Сделав n или более запросов, противник получит информацию о числе реальных битов в каждом множестве Set_i . Число этих битов будет меньше или равно $l^* \cdot d_u!$. Противник понимает, что запрошенный клиентом номер бита находится среди истинных номеров битов для множества Set_i . И вероятность того, что клиент запрашивал конкретный номер реального бита для всех реальных номеров битов для множества Set_i одинаковая.

Для того, чтобы оценить вероятность угадывания конкретного номера бита из множества Set_i , необходимо посчитать среднее число истинных битов, попадающих в массивы битов с номерами из множества Set_i .

Пусть $l^+ = l^* \cdot d_u!$ и $n_c = n_a \cdot d_u!$.

Мощность множества X равна n ($n < n_c$). Из множества массивов битов X^a выбирается суммарно l^+ битов. В этом случае возникает вопрос: сколько элементов N из случайной выборки мощности l^+ в среднем соответствует элементам множества X ?

Пусть P_r – вероятность получения в выборке мощности l^+ ровно r элементов, соответствующих элементам множества X . Тогда P_r находится по формуле:

$$P_r = \frac{\binom{r}{n} \cdot \binom{l^+ - r}{n_c - n}}{\binom{l^+}{n_c}}.$$

Тогда среднее число истинных битов, попадающих в множество Set_i , равно:

$$N = \sum_{r=1}^{l^+} r \cdot P_r.$$

Утверждение 3. При атаке с неограниченным числом известных открытых запросов и предположении о наличии пассивного противника на облаке вероятность угадывания номера бита противником в среднем не более $\frac{1}{N}$.

Доказательство. Выполнив n запросов, противник определяет биты, входящие в массивы с номерами из множества Set_i . Если фиктивные клиенты будут выполнять любое количество запросов большее n , противник все равно не получит никакой дополнительной информации. Количество реальных бит, которые попали в каждую выборку мощности l^+ в среднем равно N . Тогда вероятность угадывания номера i в среднем будет не более $\frac{1}{N}$ ■.

Как было сказано ранее, для того чтобы коммуникационная сложность не возрастала, нужно подобрать значения d_u и l_u так, чтобы L_p^+ было максимально близким к L_p^- . Следовательно, $l^* \ll n$.

Теорема 1. Для предложенной схемы организации базы данных размера n :

- Коммуникационная сложность запроса равна $l^* \cdot (s_a + d_u!)$ битов.
- При однократной атаке с известным открытым запросом номера бита i и предположении о наличии пассивного противника на облаке вероятность угадывания противником номера бита не более $\frac{1}{l^* \cdot d_u!}$.
- При атаке с неограниченным числом известных открытых запросов и предположении о наличии пассивного противника на облаке вероятность угадывания номера бита противником в среднем не более $\frac{1}{N}$.

Доказательство. Из Утверждения 1 следует, что коммуникационная сложность запроса равна $l^* \cdot (s_a + d_u!)$ битов.

Из Утверждения 2 следует, что при однократном запросе номера бита вероятность угадывания номера равна $\frac{1}{l^* \cdot d_u!}$.

Из Утверждения 3 следует, что при любом числе запросов номеров битов фиктивными клиентами и наличии противника на облаке вероятность угадывания номера бита в среднем не более $\frac{1}{N}$ ■.

7. Задача дискретной оптимизации

Напомним, что $n = l^d$, $l = \sqrt[d]{n}$. По приведенным выше построениям d – размер гиперкуба, а l – длина его стороны. Для минимизации вычислений дилером необходимо минимизировать размер вычисляемых матриц M_β и M_σ . Для этого надо минимизировать размер строки каждой из матриц, то есть величины $d_a \cdot l_a$ ($l_a = \sqrt[d_a]{n_a}$) и $d_u \cdot l_u$ ($l_u = \sqrt[d_u]{L_p^+}$), где l_a и l_u – основания счисления, а d_a и d_u – количество разрядов числа для представления номера.

Рассмотрим функцию $f(x) = \sqrt[x]{n} \cdot x$ описывающую соотношение между длиной стороны гиперкуба и его размерностью. Исследуем функцию $f(x)$ для того, чтобы минимизировать размер генерируемой дилером информации. Заметим, что $x \geq 1$.

Найдем производную функции $f(x)$.

$$f'(x) = \left(n^{\frac{1}{x}} x\right)' = \left(n^{\frac{1}{x}}\right)' x + n^{\frac{1}{x}} x' = n^{\frac{1}{x}} \ln n \left(-\frac{1}{x^2}\right) x + n^{\frac{1}{x}} = n^{\frac{1}{x}} \left(1 - \frac{\ln n}{x}\right).$$

Производная равна нулю в точках экстремума. Найдем координату x точек экстремума.

Поскольку $n^{\frac{1}{x}} > 0$, приравняем к нулю второй сомножитель:

$$1 - \frac{\ln n}{x} = 0.$$

Тогда:

$$\begin{aligned} \frac{\ln n}{x} &= 1 \\ x &= \ln n \end{aligned}$$

Теперь выясним, является ли эта точка экстремума минимум или максимум.

$$\begin{aligned} f''(x) &= \left(n^{\frac{1}{x}} \left(1 - \frac{\ln n}{x}\right)\right)' = \left(n^{\frac{1}{x}}\right)' \cdot \left(1 - \frac{\ln n}{x}\right) + n^{\frac{1}{x}} \left(1 - \frac{\ln n}{x}\right)' \\ &= -n^{\frac{1}{x}} \cdot \frac{\ln n}{x^2} \left(1 - \frac{\ln n}{x}\right) + n^{\frac{1}{x}} \frac{\ln n}{x^2} = n^{\frac{1}{x}} \frac{\ln^2 n}{x^3} > 0 \end{aligned}$$

Таким образом, минимум функции $f(x) = \sqrt[x]{n} \cdot x$ находится в точке $x = \ln n$. Следовательно, минимальное количество элементов матриц M_β и M_σ , которое должен генерировать дилер, достигается при $x = \ln n$.

Найдем предел $\lim_{n \rightarrow \infty} \frac{\sqrt[x]{n}}{x}$. Произведем замену $x = \ln n$:

$$\lim_{n \rightarrow \infty} \frac{\ln n \sqrt{\ln n}}{\ln n}.$$

Произведем замену $n = e^t$. Тогда $t = \ln n$. Если $n \rightarrow \infty$, то и $t \rightarrow \infty$. Тогда:

$$\lim_{t \rightarrow \infty} \frac{(e^t)^{\frac{1}{t}}}{t} = \lim_{t \rightarrow \infty} \frac{e}{t} = 0$$

Так как предел $\lim_{n \rightarrow \infty} \frac{\sqrt[x]{n}}{x}$ равен 0, то это говорит о том, что число разрядов числа больше основания счисления.

Для того, чтобы n_a было близким к $\hat{n}$, надо уменьшать число разрядов (большее число) при фиксированном основании счисления (меньшее число). В этом случае можно точнее приблизить n_a ($n_a = l_a^{d_a}$) к $\hat{n}$ и количество хранимой на облаке информации будет минимальной.

Ранее был найден минимум функции $f(x) = \sqrt[x]{n} \cdot x$ для случая, когда переменная x является непрерывной. В общем случае $x = \ln n$ является вещественным числом. На практике основание счисления l и число разрядов d являются натуральными числами. Сформулируем задачу дискретной оптимизации, соответствующую поиску минимума функции $f(x)$:

$$\begin{cases} l \cdot d \rightarrow \min; \\ l^d \geq n; \\ l, d \geq 2. \end{cases}$$

В формуле $f(x) = \sqrt[x]{n} \cdot x$ в дискретном случае натуральное число l соответствует $\sqrt[x]{n}$, а натуральное число d соответствует x . Решение для $f(x)$ (абсолютный минимум) в действительных числах точнее целочисленного, поэтому $l^d \geq n$.

Так как $l \geq 2$ (и $d \geq 2$), то для d имеется оценка сверху $d \leq \lceil \log_2 n \rceil$.

На практике рассматривалась база данных размера 2^{40} бит и пусть по построению интервалов $l^* \approx 2^{10}$, на практике $d_u! = 24$, что требует 5 битов для записи числа.

Для этих параметров сначала найдем точное решение (абсолютный минимум) функции $f(x)$. Заметим, что дискретный минимум всегда больше абсолютного минимума. Поэтому далее будем искать целочисленное решение, максимально близкое к абсолютному минимуму.

Пусть $n_a \approx \frac{n}{(d_u - 1)!}$.

Минимизируем объем хранящейся на облаке информации и используемой дилером в процессе вычислений памяти. Рассмотрим следующее условие:

$$\frac{n_a}{\hat{n}} = \frac{l_a^{d_a}}{\hat{n}} = 1, \text{ тогда } d_a = \lceil \log_{l_a} \hat{n} \rceil.$$

Это условие соотношения количества элементов n_a гиперкуба к сумме количества элементов в каждом интервале. Чем ближе это соотношения к 1, тем меньше фиктивных массивов битов было добавлено.

Определим значение x для $\hat{n} = 2^{36}$.

$$x = \ln 2^{36} \approx 24,95, \text{ а } \log_2 2^{36} = 36$$

$$\sqrt[x]{n} = \sqrt{\ln 2^{36}} \approx 2,72.$$

Подберем целочисленные значения для l_a и d_a .

Пусть основание счисления $l_a = 2$, тогда:

$$\frac{n_a}{\hat{n}} = \frac{2^{d_a}}{2^{36}} = 1.$$

Следовательно, $d_a = 36$. Для $\hat{n} = 2^{36}$ нужно генерировать $36 \cdot 2 = 72$ цифры для вычисления нового номера при запросе к облаку.

Пусть основание счисления $l_a = 3$, тогда:

$$\frac{n_a}{\hat{n}} = \frac{3^{d_a}}{2^{36}} = 1$$

$$3^{d_a} = 2^{36}$$

$$d_a = \lceil \log_3 2^{36} \rceil = \lceil 22,71 \rceil = 23$$

Тогда:

$$\frac{n_a}{\hat{n}} = \frac{3^{23}}{2^{36}} \approx 1,37.$$

Количество разрядов для представления номера равно 23. В каждом разряде может находиться одна из трех цифр системы счисления.

Для $\hat{n} = 2^{36}$ нужно генерировать $23 \cdot 3 = 69$ цифр для вычисления нового номера при запросе к облаку.

Таким образом при $l_a = 2$ дилеру требуется генерировать 72 цифры, а при $l_a = 3$ дилеру требуется генерировать 69 цифр. Но при $l_a = 3$ на облаке придется хранить в 1.37 раза больше битов.

При генерации множества Set_i после получения нового номера в интервале память дилера освобождается.

Коммуникационная сложность предложенного алгоритма равна $l^* \cdot (s_a + d_u!)$. Заметим, что увеличение l^* приводит к увеличению коммуникационной сложности, а уменьшение увеличивает вероятность угадывания номера бита.

Число битов s_a для представления номера может быть найдено по формуле:

$$s_a = \left\lceil \log_2 \frac{\max(n, n_a)}{(d_u - 1)!} \right\rceil = \left\lceil \log_2 \frac{2^{40}}{2^4} \right\rceil = 36$$

Тогда коммуникационная сложность равна:

$$l^* \cdot (s_a + d_u!) = l^* \cdot (36 + 5) < 2^{10} \cdot 2^6 = 2^{16}.$$

Вероятность угадывания при однократном запросе с использованием перестановок:

$$\frac{1}{l^* \cdot d_u!} = \frac{1}{2^{10} \cdot 2^4}.$$

8. Заключение

Предложенный алгоритм обладает значительными преимуществами по сравнению с алгоритмами, описанными в [1, 2]:

- Ранее в алгоритмах использовалось шифрование номера бита. В предложенном алгоритме используются перестановки цифр номера бита в выбранной системе счисления, что на практике требует меньшего числа операций, чем шифрование и расшифрование номера бита.
- Также следует заметить, что в том случае, при загрузке БД на облако выполняется перестановка, то на практике биты, номера которых являются соседними числами, в исходной базе данных чаще всего не будут являться соседними при их размещении на облаке. Это затрудняет противнику поиск стоящих рядом запрашиваемых битов.
- Изменен алгоритм построения множества Set_i , который ранее использовал вектор $\mathbf{b}$, содержащий величину сдвига в интервале для каждого элемента любого множества Set_i . В предложенном алгоритме отсутствие вектора $\mathbf{b}$ не позволит противнику получить информацию о Set_j по Set_i для непересекающихся множеств Set_i и Set_j .

Заметим, что на практике при реализации вычислений легко организовать параллельные вычисления, что позволяет увеличить скорость выполнения алгоритма. Например, генерацию перестановок как для всей базы данных, так и интервалов, можно выполнять в разных потоках. Для каждого из l^* интервалов перестановки цифр числа при формировании множества Set_i могут выполняться параллельно.

Список литературы / References

- [1]. Мартишин С.А., Храпченко М.В., Шокуров А.В. Организация безопасного запроса к базе данных на облаке. Труды Института системного программирования РАН. Том 34, № 3, 2022г., с. 173-188, ISSN 2079-8156 (Print), ISSN 2220-6426 (Online).
- [2]. Варновский Н.П., Мартишин С.А., Храпченко М.В., Шокуров А.В. Организация конфиденциальных запросов к облаку. Труды Института системного программирования РАН. Том 35, выпуск 5, 2023, с. 37-54, ISSN 2220-6426 (Online), ISSN 2079-8156 (Print).
- [3]. Chor B., Goldreich O., Kushilevitz E., Sudan M. Private Information Retrieval, in IEEE Annual Symposium on Foundations of Computer Science, 1995, pp. 41–50.
- [4]. Chor B., Goldreich O., Kushilevitz E., Sudan M. Private Information Retrieval, Journal of the ACM, Vol. 45, No. 6, November 1998, pp. 965–982.
- [5]. Gasarch W. A survey on private information retrieval, Bulletin of the EATCS, 2004 pp. 72-107
- [6]. Yekhanin S. Locally Decodable Codes and Private Information Retrieval Schemes, Springer Heidelberg Dordrecht London New York, ISSN 1619-7100, ISBN 978-3-642-14357-1 e-ISBN 978-3-642-14358-8, DOI 10.1007/978-3-642-14358-8 2010, p.82
- [7]. Kushilevitz E., Ostrovsky R. Replication is not needed: Single database, computationally-private information retrieval (extended abstract). In Proc. of the 38th IEEE Sym. on Found. of Comp. Sci., pages 364 373, 1997.
- [8]. Kushilevitz E., Ostrovsky R. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In EUROCRYPT00, 2000.
- [9]. Ostrovsky R., Skeith III W. E. A Survey of Single-Database Private Information Retrieval: Techniques and Applications". Proceedings of the 10th International Conference on Practice and Theory in Public-Key Cryptography. Springer-Verlag. pp. 393–411.
- [10]. Aguilar-Melchor C., Barrier J., Fousse L. XPIR: Private Information Retrieval for Everyone, Proceedings on Privacy Enhancing Technologies 2016(2), pp. 155-174, DOI:10.1515/popets-2016-0010.
- [11]. Demmler D., Herzberg A., Schneider T. RAID-PIR: Practical multi-server PIR CCSW '14: Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security, November 2014 Pages 45–566 <https://doi.org/10.1145/2664168.2664181>, DOI:10.1145/2664168.2664181.
- [12]. Angel S., Chen H., Laine K., Setty S. PIR with compressed queries and amortized computation IEEE Symposium on Security and Privacy, S&P (Oakland) 2018, pp. 1-18.
- [13]. "MuchPIR Demo". 14 September 2021. URL: <https://github.com/ReverseControl/MuchPIR> (дата обращения: 22.08.2022).
- [14]. Zhou M., Park A., Shi E., Zheng W., Piano: Extremely Simple, Single-Server PIR with Sublinear Server Computation 2024 IEEE Symposium on Security and Privacy (SP) 2024, pp. 4296-4314 DOI Bookmark: 10.1109/SP54263.2024.00055

Информация об авторах / Information about authors

Николай Павлович ВАРНОВСКИЙ – старший научный сотрудник Института проблем информационной безопасности МГУ имени М.В.Ломоносова. Сфера научных интересов: математика и информационная безопасность, теория сложности.

Nikolay Pavlovich VARNOVSKIY – researcher of Mathematical Studies in Information Security Section of Information Security Institute of Moscow State Lomonosov University. His research interests are mathematics, Information Security and Cryptography, complexity theory.

Сергей Анатольевич МАРТИШИН – кандидат физико-математических наук, научный сотрудник отдела теоретической информатики Института системного программирования им. В.П. Иванникова РАН. Его научные интересы включают параллельные алгоритмы, базы данных, облачные вычисления.

Sergey Anatolievich MARTISHIN – Cand. Sci. (Phys.-Math.), researcher of the Department of Theoretical Computer Science of Ivannikov Institute for System Programming of the RAS. His research interests include parallel algorithms, databases, cloud computing.

Марина Валерьевна ХРАПЧЕНКО – научный сотрудник отдела теоретической информатики Института системного программирования им. В.П. Иванникова РАН. Её научные интересы включают параллельные алгоритмы, базы данных, облачные вычисления.

Marina Valerievna KHRAPCHENKO – researcher of the Department of Theoretical Computer Science of Ivannikov Institute for System Programming of the RAS. Her research interests include parallel algorithms, databases, cloud computing.

Александр Владимирович ШОКУРОВ – кандидат физико-математических наук, доцент, заведующий отделом теоретической информатики Института системного программирования им. В.П. Иванникова РАН с 2019 года. Сфера научных интересов: алгебраические структуры в полях Галуа, базисы Гребнера, модулярная арифметика, нейрокомпьютерные технологии, цифровая обработка сигналов, криптографические методы защиты информации.

Alexander Vladimirovich SHOKUROV – Cand. Sci. (Phys.-Math.), Professor, Head of the Department of Theoretical Computer Science of Ivannikov Institute for System Programming of the RAS since 2019. Research interests: algebraic structures in the Galois fields, modular arithmetic, neurocomputer technologies, Grobner bases, digital signal processing, cryptographic methods for protecting information.

DOI: 10.15514/ISPRAS-2025-37(3)-4



Моделирование сценариев деструктивного воздействия на целостность моделей машинного обучения

А.Б. Менисов, ORCID: 0000-0002-9955-2694 <men.arty@yandex.ru>

А.Г. Ломако, ORCID: 0000-0002-1764-1942 <vka@mil.ru>

*Военно-космическая академия имени А.Ф. Можайского,
Россия, 197198, г. Санкт-Петербург, ул. Ждановская, д. 13.*

Аннотация. Статья посвящена разработке моделей деструктивного воздействия на целостность моделей машинного обучения на основе SIR-прогнозирования масштаба угроз и рисков при различных сценариях развития компьютерных атак. В статье представлена оригинальная модель угроз информационной безопасности техническим компонентам искусственного интеллекта в условиях разнородно массовых компьютерных атак, отображающая уязвимые места и способы возможных действий злоумышленников. Авторами разработана методология адаптации модернизированных SIR-моделей природных эпидемий для выявления подобия и аналогов в характере распространения деструктивных сбоев в системах ИИ, вызванных разнородно-массовыми и таргетированными воздействиями. Выявленные закономерности позволили оценить риски возможного ущерба целостности и разработать эффективные стратегии предотвращения и исправления искажений моделей машинного обучения.

Ключевые слова: искусственный интеллект; целостность моделей машинного обучения; доверие; информационная безопасность; диагностическое тестирование; тестовая среда.

Для цитирования: Менисов А.Б., Ломако А.Г. Моделирование сценариев деструктивного воздействия на целостность моделей машинного обучения. Труды ИСП РАН, том 37, вып.3, 2025 г., стр. 59–68. DOI: 10.15514/ISPRAS-2025-37(3)-4.

Modeling Scenarios of Destructive Impact on the Integrity of Machine Learning Models

A.B. Menisov, ORCID: 0000-0002-9955-2694 <men.arty@yandex.ru>

A.G. Lomako, ORCID: 0000-0002-1764-1942 <vka@mil.ru>

*Mozhaisky Military Space Academy,
13, Zhdanovskaya st., St. Petersburg, 197198, Russia.*

Abstract. The article is devoted to the development of models of destructive impact on the integrity of machine learning models based on SIR forecasting of the scale of threats and risks of losses under various scenarios of computer attacks. The article presents an original model of information security threats to technical components of artificial intelligence in the context of heterogeneous mass computer attacks, displaying vulnerabilities and methods of possible enemy actions. The authors have developed a methodology for adapting modernized SIR models of natural epidemics to identify similarities and analogues in the nature of the spread of destructive failures in AI systems caused by heterogeneous mass and targeted impacts. The identified patterns made it possible to assess the risks of possible damage to integrity and develop effective strategies for preventing and correcting distortions of machine learning models.

Keywords: artificial intelligence; integrity of machine learning models; trust; information security; diagnostic testing; test environment.

For citation: Menisov A.B., Lomako A.G. Modeling scenarios of destructive impact on the integrity of machine learning models. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025, pp. 59-68 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-4.

1. Введение

Несмотря на существенный рост производительности и качества решений прикладных задач с использованием технологий искусственного интеллекта (ИИ) [1], следует понимать, что это создает опасные риски потери работоспособности, обусловленные использованием открытых наборов данных, сторонних предобученных моделей и чужой инфраструктуры машинного обучения [2]. Необходимость построения системы защиты компонентам ИИ вызвана исключительной актуальностью этой проблематики в условиях современных вызовов и угроз [3]. Сбои и отказы систем ИИ могут быть вызваны новыми типами атак, когда злоумышленник получает доступ и манипулирует незащищёнными данными и моделями машинного обучения [4].

Кроме того, использование технологий ИИ предоставляет новые возможности злоумышленникам для реализации угроз информационной безопасности, таких как обучение нежелательному поведению (модификация модели машинного обучения путем искажения («отравления») данных) [5], хищение обучающих данных [6], введение модели в заблуждение (состязательные атаки) [7], подмена модели машинного обучения [8] и другие [9]. Опасность реализации таких угроз определяется тем, что алгоритмы обработки данных в машинном обучении представляют собой наборы обученных коэффициентов (большие массивы чисел). Поэтому несанкционированные изменения в структуре модели машинного обучения не могут быть выявлены стандартными средствами контроля и анализа программного обеспечения, которые сейчас применяются. При этом сами нежелательные изменения в структуре и поведении модели машинного обучения могут осуществляться различными способами, включая, например, манипуляции с обучающей выборкой, либо процессом обучения, что также невозможно проконтролировать стандартными средствами.

Таким образом, формируется потребность в разработке специальных программных средств защиты информации для борьбы с компьютерными атаками и снижением уязвимости алгоритмов, основанных на технологиях ИИ. Данные средства должны обеспечить необходимый базис для формирования новых подходов к тестированию алгоритмов ИИ, а

также позволить определить потенциальные угрозы методов машинного обучения и сформировать рекомендации по коррекции специального программного обеспечения. Систематическое применение подобного подхода позволит сформировать ландшафт потенциальных специфических угроз информационной безопасности, а также выработать лучшие практики по возможным контрмерам, позволяющим минимизировать риски при использовании алгоритмов, основанных на машинном обучении в практических приложениях.

На рис. 1 показаны потенциально-возможные сценарии злоумышленников на компоненты ИИ. Анализ фактографии реализации угроз на системы ИИ позволил определить основные негативные действия злоумышленников по намеренному вызову сбоев и отказов. Они представлены в сценариях компьютерных атак на системы, содержащие компоненты машинного обучения и демонстрируют расширение арсенала технологий деструктивного воздействия.

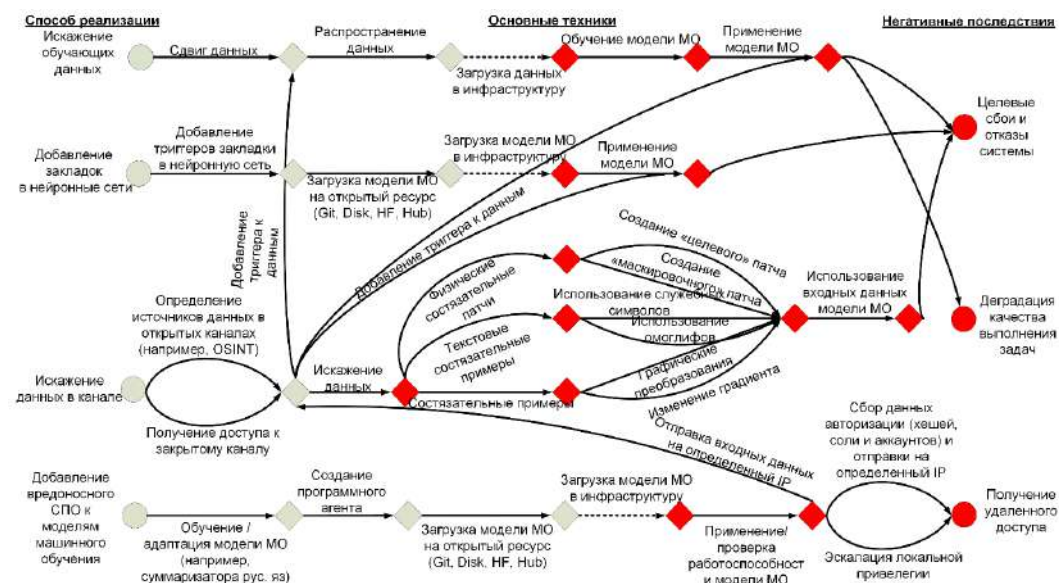


Рис. 1. Типизация сценариев деструктивных действий злоумышленников на системы, содержащие компоненты машинного обучения.

Fig. 1. Typification of scenarios of destructive actions of attackers on systems containing machine learning components.

Исследование путей обеспечения защищенности комплексов ИИ в условиях негативных сценариев, приводящих к нарушениям целостности данных и моделей машинного обучения, показало следующее:

- несовершенство методов обнаружения и нейтрализации для администрирования безопасности систем ИИ в условиях деструктивного влияния на качество решения прикладных задач, так как сравнительный анализ современных систем и мер обеспечения информационной безопасности систем ИИ показал не способность выявить признаки атак на компоненты ИИ;
- явное преимущество проактивной схемы обеспечения защиты в связи с преимуществами по сравнению с реактивной и постфактной;
- необходимости обеспечения своевременного выявления нетиповых аномалий функционирования систем ИИ с использованием идеологии обучаемого динамического детектирования.

Таким образом, существует необходимость разработки теоретических основ и специальных программных средств проактивной защиты информации для борьбы с компьютерными атаками и своевременной нейтрализацией уязвимостей алгоритмов, используемых в технологиях ИИ. Данные средства должны в итоге обеспечить необходимый и достаточный (в развитии) арсенал обеспечения защищенности систем ИИ в условиях угроз компьютерных атак на модели машинного обучения.

2. Формализованная постановка проблемы

Формализованная постановка проблемы исследования представлена следующим образом: пусть система защиты имеет исходные данные о предыдущих состояниях систем ИИ, а для изменения состояния можно использовать следующую модель, в которой выделены параметры, влияющие на функционирование систем ИИ:

$$\overline{f}_k = \overline{f}_k \left\{ s_{i_1}^a; s_{i_2}^a; s_{i_3}^p; s_{i_4}^p; g_{\omega_k}; t_k; d_j; l_s; z_g; r_k \right\}, \quad (1)$$

где k – номер конкретного комплекса, $k=1, \dots, K$; i – номер признака, $i=1, \dots, N$, т.е. результата действия злоумышленников; $s_{i_1}^a, s_{i_2}^a$ – данные наблюдения предыдущих состояний систем ИИ, $i_1=1, \dots, m_1, i_2=m_1+1, m_2$; $s_{i_3}^p, s_{i_4}^p$ – данные текущего контроля, $i_3=m_2+1, m_3, i_4=m_3+1, m_4=m$; $\sim$ – знак, указывающий на признаки, которые подвержены влиянию данной совокупности внешних условий; g_{ω_k} – совокупность внешних условий, $\omega_k=1, \dots, \Omega_k$; t_k – условная координата времени, показывающая полноту информации о k -м комплексе; d_j – класс атаки на систему ИИ, $j=1, \dots, N$; l_s – методы защиты, $s=1, \dots, S$; z_g – последующие состояния k -й системы, $g=1, \dots, G$; r_k – помехи, искажающие действительное состояние k -го комплекса.

Пусть известна модель защищаемого комплекса $\overline{f}_k$. Пусть l_s – защитные меры, которые могут системы ИИ $\overline{f}_k$ из состояния z_{g_i} перевести вероятность такого перехода. Обозначим через e_{s_j} меру результативности l_s -го мероприятия защиты.

Тогда задачу нахождения оптимальной совокупности защитных мер можно сформулировать следующим образом. Необходимо найти такую совокупность защитных мер l_s^* , чтобы их результативность была достаточной. В этом случае результативность защиты:

$$e_{s_j}(z_{g_j} \rightarrow z_{g'})_{l_s} = p(d_j)_{z_g} p(z_{g_j} \rightarrow z_{g'})_{l_s}, \quad (2)$$

где $p(d_j)_{z_g}$ – вероятность проведения компьютерных атак, а максимальное значение меры результативности, или защита систем ИИ для $(z_{g_j} \rightarrow z_{g'})_{l_s}$, достигается при

$$e_{s_j}^* = \min_{l_s} e_{s_j}(z_{g_j} \rightarrow z_{g'})_{l_s} \quad (3)$$

Это определение действует для тех пар $(z_{g_j} \rightarrow z_{g'})_{l_s}$, для которых справедливо утверждение о том, что состояние $z_{g'}$ лучше, чем z_{g_j} .

Общая результативность защитных мер e_j является аддитивной функцией, состоящей из e_{s_j} . Тогда для определения совокупности защитных воздействий можно сформулировать ограничение исследования: оптимальная совокупность защитных действий обладает тем свойством, что, каково бы не было первоначальное l_s при состоянии комплекса z_{g_j} , последующие защитные действия должны быть оптимальны относительно первоначального. Исходя из этого, искомую результативность защитных действий можно получить в следующем виде:

$$e_j = \max_{l_s} \sum |e_{s_j}(z_{g_j} \rightarrow z_{g'})_{l_s} + e_{s_j}(z_{g'} \rightarrow z_{g''})_{l_s}|. \quad (4)$$

Процесс обеспечения защищенности систем ИИ является сложным многоуровневым циклическим процессом, включающим в себя сбор и обработку данных различных систем, определение их состояния, выбор стратегии защиты и проведение защитных мер (рис. 2 и 3). В каждом цикле проводится оценка и коррекция управляющего воздействия. После того как установлены специфические особенности функционирования, можно переходить к управлению защиты системы ИИ.

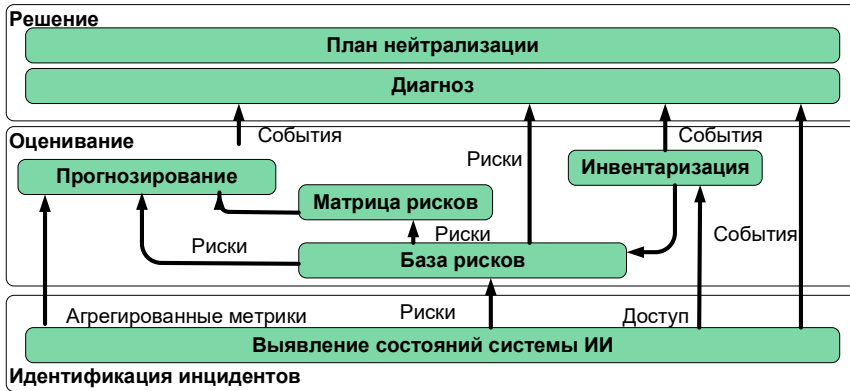


Рис. 2. Принципиальная схема обеспечения защиты систем ИИ.
Fig. 2. Schematic diagram of the protection of AI systems.

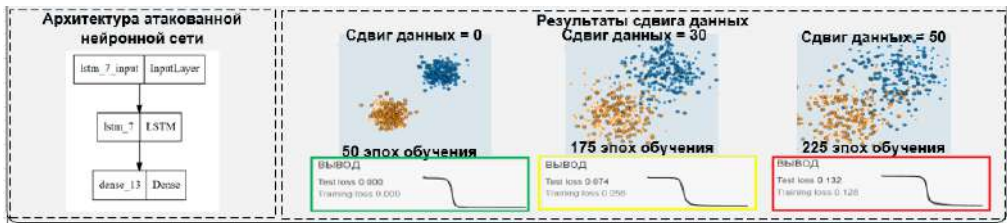


Рис. 3. Демонстрация результативности сдвига данных для 3-х слойной нейронной сети.
Fig. 3. Demonstration of the effectiveness of data shifting for a 3-layer neural network.

На каждом этапе защиты обрабатываются данные детектирования, потоков инцидентов, отслеживаемых за контролируемый период времени. При этом компонент прогнозирования определяет не только отдельные ситуации нарушений, но и всю возможную картину рисков реализации сценариев деградации систем ИИ.

Таким образом, выбор совокупности защитных мер начинается с выявления признаков нарушения функционирования (компьютерных атак) систем ИИ, последующей выработкой решений и нейтрализацией угроз целостности моделей машинного обучения.

3. Описание процесса моделирования

Для формирования семейства моделей оценивания опасностей деструктивного воздействия и потенциального ущерба на компоненты систем ИИ была выдвинута гипотеза о пригодности эпидемиологического моделирования ущерба штатному функционированию компонентов систем ИИ. Это связано с подобием структур и сходством параметров моделей машинного обучения и информационных систем (типы взаимосвязей представлены на рис. 4).

Результаты действий злоумышленника по деградации таких систем подобны ущербу в биологической популяции при распространении инфекционных заболеваний. Подобные модели уже использовались в области информационной безопасности, например, при моделировании распространения компьютерных вирусов и функционирование ботнетов.

Для нейронных сетей выявлена зависимость развития эпидемии от механизма распространения среди базовых элементов – нейронов.

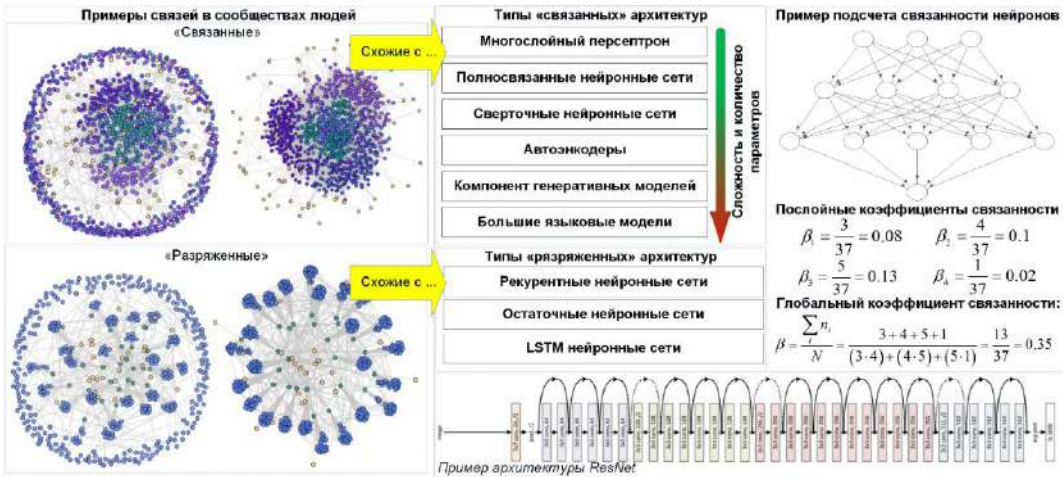


Рис. 4. Подобие структур и сходства параметров SIR-моделирования в нейронных сетях.

Fig. 4. Similarity of structures and similarities of parameters of SIR modeling in neural networks.

В связи с этим произведена адаптация SIR-модели:

$$\begin{aligned} \frac{dS_i}{dt} &= -\beta \rho S_i \sum_i A_{ij} I_j \\ \frac{dI_i}{dt} &= \beta \rho S_i \sum_i A_{ij} I_j - \gamma I_i, \\ \frac{dR_i}{dt} &= \gamma I_i \end{aligned} \tag{5}$$

где введены следующие параметры: β - коэффициент связанности нейронной сети; A_{ij} - матрица смежности, γ - коэффициент защищенности нейронной сети, ρ - степень инфицирования.

Решение системы дифференциальных уравнений следующее:

$$\frac{dS}{dt} = -\beta\rho S \frac{dR}{dt} \frac{1}{\gamma}, S = S_0 e^{-\frac{\beta\rho R}{\gamma}},$$

$$\frac{dI_i(t)}{dt} = \beta\rho(1-I_i) \sum_i A_{ij} x_j - \gamma I_i$$

которое формирует критерий опасности и выражается следующими условиями:

$$\begin{aligned} \beta\rho > \gamma, \quad R(\infty) &= \text{const} > 0 \\ \beta\rho < \gamma, \quad R(\infty) &\rightarrow 0 \end{aligned}$$

Интерпретацию параметров процесса деградации моделей машинного обучения начнем с описания базового элемента нейронных сетей. В основе всех нейронных сетей лежит базовый элемент – нейрон, который математически полностью аналогичен логистической регрессии (в случае если используется функция активации сигмоида). Это обусловлено тем, что данный элемент содержится во всех архитектурах нейронных сетей (рис. 5).

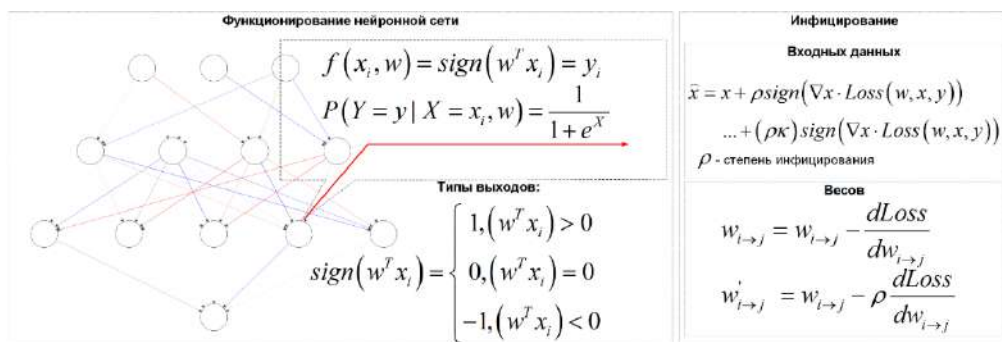


Рис. 5. Интерпретация процессов деградации моделей машинного обучения при деструктивных возмущениях.

Fig. 5. Interpretation of the degradation processes of machine learning models under destructive disturbances.

Параметрами любой нейронной сети являются - входные данные, веса, функция преобразования, функция активация. То есть, функционирование нейрона в обычных условиях происходит как

$f : X \rightarrow Y$, где $\{y_1, \dots, y_i\} \subset Y$ – пространство выходов, а $x = \{x_1, \dots, x_j\} \subset X$ – пространство входных данных.

Основная цель компьютерных атак на системы ИИ – нарушить заявленное качество ее функционирования небольшими изменениями входных данных и весов моделей машинного

обучения) так, чтобы модифицированные данные $\bar{X}$ вызовут сбои и ошибки $f(\hat{x}) \neq y$.

Дополнительно возникает необходимость маскировки таких воздействий, т. е. способность скрывать модифицированные данные.

Определим показатели эффективности атак на компоненты машинного обучения:

- метрика результативности, измеряющая несоответствие между обычными и модифицированными данными машинного обучения:

$$M_{res}(x_j, \hat{x}_j) = 1 - \frac{|f(x_j) \cap f(\hat{x}_j)|}{|f(x_j)|},$$

другими словами, метрика определена в интервале $[0, 1]$ и показывает, что при $M_{res} = 1$ все модифицированные данные привели к неверному результату, и наоборот, при $M_{res} = 0$ атака не достигла успеха;

- метрика скрытности, измеряющая возможность обнаружения модификации данных:

$$M_{hid}(y_j, \hat{x}_j) = \begin{cases} 0, & \text{если } f(\hat{x}_j) \approx y_j \\ 1, & \text{в другом случае} \end{cases} \quad (9)$$

4. Экспериментальное исследование

Обоснование адекватности разработанных моделей и механизмов прогнозирования рисков потерь качества функционирования нейронных сетей при нарушениях целостности моделей машинного обучения с различной степенью деградации продемонстрировано двумя экспериментами: 1) сдвигом данных на простой нейронной сети, 2) при состязательных атаках на языковые модели машинного обучения.

Зависимость параметров сдвига данных представлено на примере целенаправленного сдвига данных для трехслойной нейронной сети (как видно на рис. 3 функция ошибки увеличивается при увеличении сдвига данных).

Обобщение результатов экспериментов по проведению состязательных атак на языковые модели машинного обучения (лингвистические модели) представлены на рис. 6 и показали сходимость результатов моделирования и экспериментов по воздействию на модели машинного обучения, представленные в табл. 1.

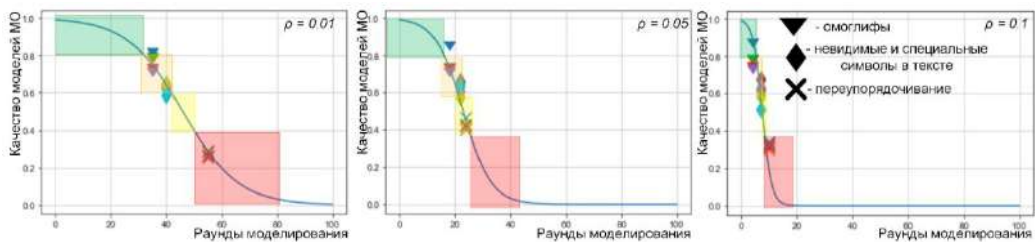


Рис. 6. Расчет уровня деградации для лингвистических моделей.
Fig. 6. Calculation of the degradation level for linguistic models.

Табл. 1. Лингвистические модели, отобранные для проведения эксперимента.
Table 1. Linguistic models selected for the experiment.

Название модели	Прикладная задача	Архитектура	Набор обучающих данных	Качество
IE-Net	Вопросно-ответные системы	Двоичные нейросети	SQuAD (Stanford QA Dataset)	F-мера=0,932
CB-NTR	Рубрикация текста	BERT	Reuters-21578	F-мера=0,907
ACE	Выявления поименованных существностей	LSTM, Transformer	CoNLL-2003	F-мера=0,946
AraBERTv1	Семантический поиск	BERT	Large-Scale Arabic Book Reviews	Точность=0,867
Bi-LSTM	Выявление фейков	Bi-LSTM	FakeNewsNet	Точность=0,822

Для расчета уровня деградации для различных значений параметра инфицирования моделей $\rho \in (0.01, 0.05, 0.1)$ было произведено 100 раундов моделирования. Уровни деградации имеют следующие значения: незначительный – до 0.8, умеренный – от 0.8 до 0.6, высокий – от 0.6 до 0.4 и критический – ниже 0.4.

Для формирования текстовых состязательных примеров использованы обучающие данные для каждой лингвистической модели. На каждую модель машинного обучения сформировано 100 состязательных примеров с тремя долями модификации текста. Результаты представлены в табл. 2 и показали сходимость результатов моделирования и экспериментов.

Разработанные модели противодействия для формирования риск-ориентированного администрирования безопасности комплексов интеллектуальной обработки данных с возможностями самоконтроля и самовосстановления применяются по следующей схеме: получение информации о моделях машинного обучения, определение начальных параметров модели, расчет ущерба и опасности. Комплексное применение дает возможность сформировать матрицу рисков для моделей машинного обучения. На рис. 7 более подробно рассмотрен отдельный сегмент матрицы рисков нейронной сети ResNet.

Табл. 2. Результаты модификации для текстовых состязательных примеров.

Table 2. Modification results for text adversarial examples.

Модель	Качество моделей машинного обучения после модификации								
	при доле омографов в тексте, %			при доле невидимых символов в тексте, %			при доле символов переупорядочивания, %		
	1	5	10	1	5	10	1	5	10
	1	5	10	1	5	10	1	5	10
IE-Net	0,894	0,645	0,342	0,844	0,676	0,289	0,932	0,876	0,765
CB-NTR	0,804	0,639	0,299	0,809	0,630	0,253	0,865	0,735	0,715
ACE	0,811	0,620	0,337	0,818	0,618	0,289	0,847	0,762	0,730
AraBERTv1	0,771	0,645	0,315	0,757	0,584	0,270	0,762	0,754	0,720
Bi-LSTM	0,754	0,589	0,323	0,735	0,515	0,264	0,745	0,734	0,701

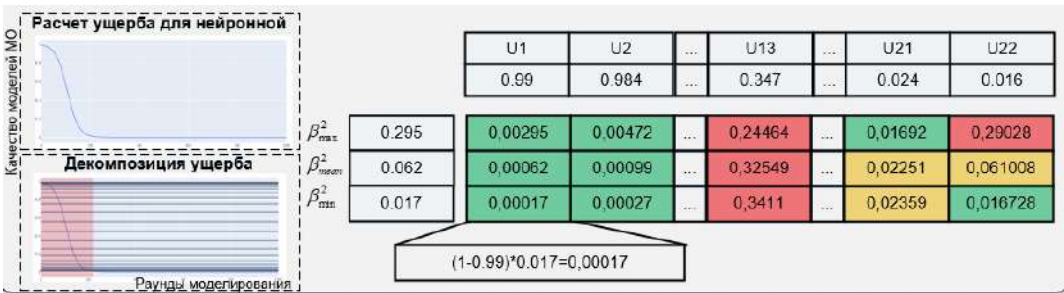


Рис. 7. Сегмент результатов моделирования рисков для нейронной сети типа ResNet.

Fig. 7. Segment of risk modeling results for a ResNet neural network.

Заключение

Таким образом, обоснована идеология адаптации модернизированных SIR-моделей природных эпидемий для выявления сходства в характере распространения деструктивных сбоев в системах ИИ, вызванных разнородно-массовыми и таргетированными воздействиями. Проведено обоснование пригодности семейства SIR-моделей для исследования процессов компьютерных атак на комплексы интеллектуальной обработки данных с моделями машинного обучения и разработки решений по противодействию. Выявленные закономерности позволили оценить риски возможного ущерба целостности и разработать эффективные стратегии предотвращения и

исправления искажений моделей машинного обучения. Разработанные математические модели пригодны для любой архитектуры нейронных сетей, а также для других парадигм машинного обучения. Модели позволяют сформировать обобщенную систему знаний для прогнозирования рисков нарушения целостности моделей машинного обучения в комплексах интеллектуальной обработки данных и уточнить требования к возможностям восстановления и самовосстановления.

Список литературы / References

- [1]. Qin Y. et al. Artificial intelligence and economic development: An evolutionary investigation and systematic review // *Journal of the Knowledge Economy*. – 2024. – vol. 15. – №. 1. – p. 1736-1770.
- [2]. Менисов А. Б., Ломако А. Г., Сабиров Т. Р. Метод тестирования лингвистических моделей машинного обучения текстовыми состязательными примерами // *Научно-технический вестник информационных технологий, механики и оптики*. – 2023. – т. 23. – №. 5. – с. 946-954.
- [3]. Paragianni A. et al. Frugal and Robust AI for Defence Advanced Intelligence // *Paradigms on Technology Development for Security Practitioners*. – Cham: Springer Nature Switzerland, 2024. – p. 427-437.
- [4]. Weng Y., Wu J. Leveraging artificial intelligence to enhance data security and combat cyber attacks // *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*. – 2024. – vol. 5. – №. 1. – p. 392-399.
- [5]. Nguyen T. T. et al. Manipulating recommender systems: A survey of poisoning attacks and countermeasures // *ACM Computing Surveys*. – 2024. – vol. 57. – №. 1. – p. 1-39.
- [6]. Rosenblatt M. et al. Data leakage inflates prediction performance in connectome-based machine learning models // *Nature Communications*. – 2024. – vol. 15. – №. 1. – p. 1829.
- [7]. Kim S. et al. Propile: Probing privacy leakage in large language models // *Advances in Neural Information Processing Systems*. – 2024. – vol. 36.
- [8]. Менисов, А. Б. Ландшафт угроз систем искусственного интеллекта: монография / А. Б. Менисов. — Москва: Ай Пи Ар Медиа, 2023. — 126 с.
- [9]. Костогрызов А. И., Нистратов А. А. Анализ угроз злоумышленной модификации модели машинного обучения для систем с искусственным интеллектом // *Вопросы кибербезопасности*. – 2023. – №. 5. – с. 9.

Информация об авторах / Information about authors

Артем Бакытжанович МЕНИСОВ – кандидат технических наук, старший преподаватель кафедры систем сбора и обработки информации Военно-космической академии имени А.Ф.Можайского. Сфера научных интересов: построение доверенных систем искусственного интеллекта, применение машинного обучения для задач обеспечения информационной безопасности.

Artem Bakytzhanovich MENISOV – Cand. Sci. (Tech.), Senior lecturer of the Department of Information Collection and Processing Systems at the Mozhaysky Military Space Academy. Research interests: building trusted artificial intelligence systems, using machine learning for information security tasks.

Александр Григорьевич ЛОМАКО – доктор технических наук, профессор, профессор кафедры систем сбора и обработки информации Военно-космической академии имени А.Ф.Можайского. Его научные интересы включают области теоретического и системного программирования, моделирования интеллектуального поведения кибернетических систем в приложении к задачам информационной безопасности.

Alexander Grigorievich LOMAKO – Dr. Sci. (Tech.), Professor, Professor of the Department of Information Collection and Processing Systems at the Mozhaysky Military Space Academy. His research interests include the areas of theoretical and system programming, modeling of intelligent behavior of cybernetic systems in application to information security problems.

DOI: 10.15514/ISPRAS-2025-37(3)-5



Анализ и разработка методов очищения для защит метрик качества изображений

А.Е. Гуцин, ORCID: 0000-0002-4055-7394 <alexanterg@gmail.com>

А.В. Анциферова, ORCID: 0000-0002-1272-5135 <aantsiferova@graphics.cs.msu.ru>

Д.С. Ватолин, ORCID: 0000-0002-8893-9340 <dmitriy@graphics.cs.msu.ru>

*Московский государственный университет имени М.В. Ломоносова,
Россия, 119991, Москва, Ленинские горы, д. 1.*

*Исследовательский центр доверенного искусственного интеллекта ИСП РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*

Аннотация. В последнее время начали исследовать область состязательных атак на метрики качества изображений, в то время как область защиты от них остается малоизученной. В данном исследовании мы стремимся охватить эту область и проверить возможность переноса защиты от атак с классификаторов изображений на методы оценки качества изображений. В этой работе мы применили несколько широко распространенных атак на модели оценки качества изображений и проверили успешность защиты от них. Методологии очистки охватывают различные техники предварительной обработки, включая геометрические преобразования, сжатие, очищение от шума и современные методы на основе нейронных сетей. Кроме того, мы рассматриваем проблему оценки эффективности методов защит, предлагая способы оценки визуального качества выходных данных и успешности нейтрализации атак. Мы тестируем защиту от атак на три метрики IQA – Linearity, MetaIQA и SPAQ.

Ключевые слова: состязательные защиты; состязательное очищение данных; оценка качества изображений.

Для цитирования: Гуцин А.Е., Анциферова А.В., Ватолин Д.С. Анализ и разработка методов очищения для защит метрик качества изображений. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 69– 84. DOI: 10.15514/ISPRAS-2025-37(3)-5.

Adversarial Purification for No-Reference Image-Quality Metrics: Applicability Study and New Methods

A.E. Gushchin, ORCID: 0000-0002-4055-7394 <alexanterg@gmail.com>
A.V. Antsiferova, ORCID: 0000-0002-1272-5135 <aantsiferova@graphics.cs.msu.ru>
D.S. Vatin, ORCID: 0000-0002-8893-9340 <dmitriy@graphics.cs.msu.ru>

Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia.
Research Centre for Trusted Artificial Intelligence of ISP RAS,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Abstract. Recently, the area of adversarial attacks on image quality metrics has begun to be explored, whereas the area of defences remains under-researched. In this study, we aim to cover that case and check the transferability of adversarial purification defences from image classifiers to IQA methods. In this paper, we apply several widespread attacks on IQA models and examine the success of the defences against them. The purification methodologies covered different preprocessing techniques, including geometrical transformations, compression, denoising, and modern neural network-based methods. Also, we address the challenge of assessing the efficacy of a defensive methodology by proposing ways to estimate output visual quality and the success of neutralizing attacks. We test defences against attacks on three IQA metrics – Linearity, MetaIQA and SPAQ.

Keywords: adversarial attacks; adversarial purification; image quality assessment.

For citation: Gushchin A.E., Antsiferova A.V., Vatin D.S., Adversarial purification for no-reference image-quality metrics: applicability study and new methods. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 69-84 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-5.

1. Введение

Метрики оценки качества изображений (IQA), основанные на машинном обучении, широко используются для разработки и оценки алгоритмов обработки изображений и видео. По сравнению с традиционными метриками, основанными на попиксельного или структурного сходства [1], метрики, основанные на машинном обучении, показывают более высокую корреляцию с субъективным качеством. Однако ряд исследований показал, что метрики качества изображений, основанные на машинном обучении, уязвимы к состязательным возмущениям, что потенциально может привести к многочисленным негативным последствиям. Во-первых, метрики IQA обычно используются как часть методологии оценки открытых бенчмарков для воспроизводимости. Участники могут использовать атаки на метрики с целью получения более высоких позиций в таблице лидеров. Обман бенчмарка - веская причина, когда результаты бенчмарка влияют на инвестиции в проект: например, разработка видеокodeка требует огромных ресурсов, и обман метрики упрощает победу в открытых сравнениях. В современных видеокodeках уже есть режимы оптимизации, ориентированные на метрику (например, в кодеке Google *libaom* есть опция `--tune-vmf`). Такая оптимизация может не повысить качество восприятия [2], поэтому использование нестабильных IQA-метрик в сетях доставки контента и потоковых сервисах может привести к ухудшению качества просмотра. Поскольку метрики IQA также используются для управления потоковым видео, увеличение показателя метрики может привести к увеличению битрейта после транскодирования, что негативно сказывается на канале трафика. Кроме того, метрики IQA используются в задачах компьютерного зрения, например, в медицинской сфере. Оптимизация, например, метода обработки магнитно-резонансной томографии с использованием неустойчивой метрики IQA внутри функции потерь модели может привести к некорректным результатам медицинских заключений.

Таким образом, область исследования устойчивости IQA к состязательным надбавкам начала развиваться недавно. Она не так хорошо изучена, как устойчивость методов классификации или сегментирования изображений. Для метрик качества изображений до сих пор проводились только эмпирические эксперименты с атаками противника, и никаких доказательных результатов получено не было. Среди существующих атак, предложенных для оценки устойчивости метрик, есть несколько методов, основанных на градиентной оптимизации ([3, 4, 5]), перцептивно-ориентированной маскировке [6, 7].

Механизмы защиты метрик качества изображений и видео не были тщательно изучены. Это объясняется двумя причинами: во-первых, область атак на метрики IQA находится на ранней стадии своего развития; во-вторых, постановка задачи защиты метрик IQA является более сложной, чем, например, классификация изображений. Семантика изображения остается неизменной после добавления к нему состязательной надбавки, поэтому истинная метка атакованного изображения по-прежнему известна. Однако его качество меняется, и метод защиты должен обеспечить правильную оценку исходного качества.

Насколько нам известно, только две метрики были позиционированы как более надежные версии оригинальных реализаций. Первая – VMAF [7] от Netflix, которая повышает устойчивость VMAF к улучшению изображения путем обрезания оценок базовых SVM-функций. R-LPIPS [9] и E-LPIPS [10] – это надежная версия метрики LPIPS, созданная с помощью дообучения и ансамблирования.

Нашей целью было изучить эмпирические средства защиты, полезные, когда атакующий не знает о механизме защиты, но мы также показали их эффективность против адаптивных атак. Наше исследование сосредоточено на изучении и улучшении устойчивости только метрик, не требующих исходного изображения (NR метрик) по нескольким причинам. Во-первых, эти метрики легче атаковать; они имеют больше практических применений из-за отсутствия доступа к исходному изображению (например, генерация изображений и видео, потоковое видео и т. д.). NR метрики могут быть интегрированы в функцию потерь в различных задачах компьютерного зрения. Наконец, приведенные выше примеры устойчивых метрик являются метриками, требующими исходное изображение (FR метрики); для NR метрик не было предложено устойчивых версий, и было проведено лишь несколько экспериментов.

Мы не рассматривали методы детектирования атак в этой работе, потому что они могут испытывать трудности с новыми или незаметными атаками. Методы очистки могут адаптироваться к более широкому спектру искажений и исправлять их. Также детектирование не предлагает способа правильной дальнейшей обработки атакованных изображений.

Korhonen и др. [6] показали, что состязательное дообучение помогает улучшить эмпирическую стойкость. Обрезка и изменение размеров изображений эффективно противостоят универсальным состязательным возмущениям в работе Shumitskaya и др [3]. На рис. 1 показан пример атаки на метрику Linearity и различные методы защиты от этой атаки. Надежные метрики необходимы для разработки современных методов обработки и сжатия изображений. Такие метрики приведут к разработке надежных бенчмарков и позволят исследователям использовать метрики в составе функции потерь для обучения методов обработки изображений. Наиболее популярными методами защиты нейронных сетей являются состязательное дообучение, очистка данных и детектирование атак. Данная статья посвящена эмпирическим механизмам защиты как первому шагу к улучшению устойчивости метрик. Несмотря на то, что состязательное дообучение может эффективно препятствовать атакам, оно также снижает корреляции метрик с субъективными оценками. В данной работе изучается применимость очистки изображений для защиты метрик IQA и предлагаются новые методы очистки.

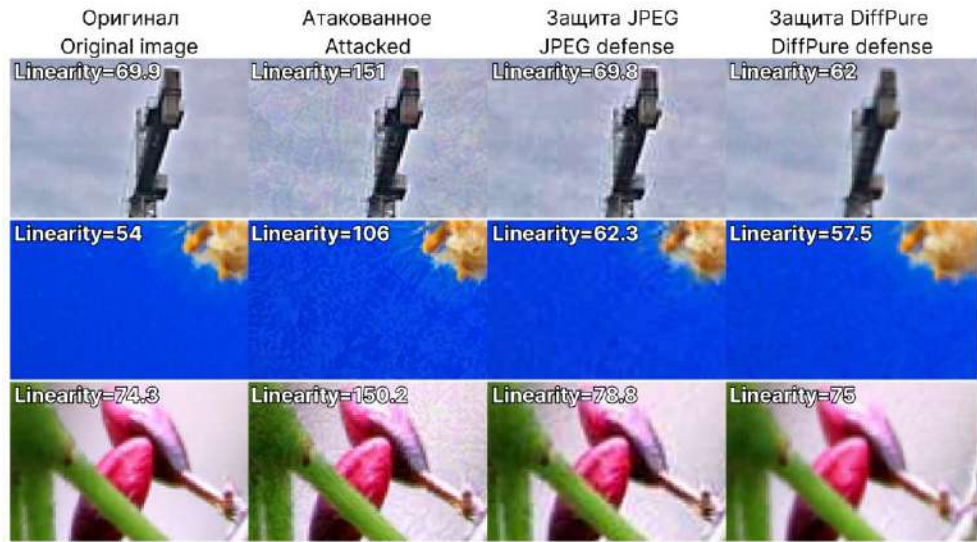


Рис. 1. Оригинальное изображение (первый столбец), атакованное изображение (второй столбец), пример работы двух методов защит – JPEG и DiffPure (третий и четвертый столбцы, соответственно).

Fig. 1. Original image (first column), image after the adversarial attack (AMI-FGSM, second columns), and two defence techniques applied to the adversarial image (third and fourth columns).

2. Обзор литературы

2.1 Состязательные атаки на метрики оценки качества

Szegedy и другие [11] показали, что нейронные сети уязвимы к специально сгенерированным входным данным, что приводит к неправильному выводу модели. Впоследствии это стало известно как состязательные атаки. Было предложено большое количество методов атаки, особенно для задачи классификации изображений. Большинство из этих атак [12, 13, 14, 15] являются аддитивными, поскольку к исходному изображению добавляется дополнительный шум, создаваемый противником. Эти методы решают оптимизационную задачу в пространстве пикселей с некоторыми ограничениями на величину возмущения в терминах нормы l_p . Поскольку было показано, что l_p неэффективна для аппроксимации визуального качества, были предложены состязательные атаки, сохраняющие качество восприятия, и методы преобразования цвета [16]. Несмотря на то, что большинство существующих атак создано для моделей компьютерного зрения, существуют атаки, специально разработанные для метрик IQA. AMI-FGSM [17] – это атака на метрики NR, которая является модификацией FGSM [13]. Shumitskaya и др. [3] предложили обучение универсального возмущения на наборе изображений. Ghildyal & Liu [18] обратили внимание на уязвимость FR метрик к атакам. Они адаптировали FGSM и PGD атаки для создания примеров атак на метрики IQA. Korhonen и др. [6] предложили итеративную атаку на NR метрики, которая улучшает визуальное качество атакованных изображений с помощью фильтра Собеля.

2.2 Состязательные защиты

Как и для методов атаки, для задачи классификации изображений было предложено множество способов защиты. Модификация процесса обучения или применения модели может повысить устойчивость к атакам. Одним из наиболее популярных методов защиты моделей от атак противника является состязательное дообучение [13, 15]. Оно предполагает

динамическое добавление в обучающий набор данных изображений, сгенерированных противником. Такой подход не только требует модификации процедуры обучения, но и увеличивает вычислительные затраты, так как на каждой итерации необходимо выполнить дополнительно один или даже несколько прямых и обратных проходов для генерации состязательных примеров. Кроме того, обучение на состязательных примерах приводит к снижению производительности на чистых входных данных. Аналогичный эффект от обучения с использованием состязательных примеров был продемонстрирован в работе Korhonen & You [6] для моделей IQA.

Другой тип защиты – предварительная обработка входных данных с помощью вспомогательных преобразований перед основной моделью, также называемая очисткой. Этот тип защиты требует модификации выводов модели. Graese и другие [19] предложили метод очистки, основанный на простых преобразованиях изображений, таких как обрезка и размытие. Guo и другие [20] предложили защищаться от атак противника путем усреднения предсказаний нескольких случайных участков изображений во время оценки. Этот подход направлен на изменение пространственного расположения неблагоприятных возмущений, которые не инвариантны к таким преобразованиям. В ряде работ [21, 22, 20] в качестве метода предотвращения атак противника предложено сжатие JPEG. Xu и др. [23] предложили подход, основанный на сжатии признаков внутри модели для уменьшения размеров входного изображения, чтобы снизить возможности злоумышленника по проведению атаки. Основная идея заключается в удалении ненужных признаков путем уменьшения битовой глубины цвета или применения пространственного сглаживания. Guo и другие изучали удаление отдельных пикселей с минимизацией общей вариации [24]. Этот подход предполагает выбор небольшого набора пикселей и восстановление изображения без состязательного шума. Meng и Chen в своей защите MagNet [25] использовали автокодировщик для перемещения состязательных примеров ближе к множеству чистых примеров. Однако этот метод предназначен для сценариев, в которых злоумышленник не может получить доступ к параметрам модели, и, следовательно, не подходит для атак на основе градиента, которые представляют собой значительную угрозу. С ростом популярности генеративных моделей они также стали использоваться для очистки от атак. В частности, Samangouei и другие [26] предложили использовать генеративные состязательные сети (GAN), которые удаляют состязательные возмущения из изображений. Недавно Nie и другие [27] предложили использовать все более популярные диффузионные модели для удаления неблагоприятных возмущений из изображений. Идея заключается в том, что в процессе прямой диффузии вредные возмущения должны смешиваться с шумом, а затем обратный процесс удаляет и шум, и возмущения, в результате чего получается чистое изображение.

Не все методы очистки классификаторов подходят для регрессионных моделей оценки качества, поскольку качество изображения не является решающим фактором в работе классификатора. В задаче IQA важно не только вернуть исходные оценки, но и не вносить существенных изменений в изображение. Для этих моделей было предложено всего несколько методов защиты. Korhonen и др. [6] использовали гауссово размытие и двустороннюю фильтрацию для защиты от атак. Авторы делают вывод, что методы предварительной обработки заслуживают детального изучения в будущем, что и представлено в нашей работе.

3. Методология

3.1 Постановка задачи

В нашей работе рассматриваются атаки, направленные на повышение прогнозируемых оценок качества. Эти атаки были предложены в последних работах и имеют большую практическую применимость [6, 3]. Атаки, направленные на снижение прогнозируемого

качества изображения, выглядят аналогично, за исключением знака вектора, обозначающего направление атаки. Таким образом, данный подход не нарушает общности исследования.

Для заданной NR метрики качества изображения f и входного изображения $x \in R^{H \times W \times 3}$, атака противника, направленная на увеличение прогнозируемой оценки качества, может быть математически описана следующим образом:

$$\max f(att(x)), \text{ такое что } dist(x, att(x)) \leq \epsilon \quad (1)$$

где $att: R^{H \times W \times 3} \rightarrow R^{H \times W \times 3}$ – алгоритм атаки противника, применяемый к x , $dist(\cdot)$ – некоторая метрика расстояния, а ϵ – ограничение на расстояние. Атака считается успешной, если она увеличивает предсказанную оценку, но не улучшает визуальное качество. Таким образом, изменения в изображении должны быть незаметны. Состязательная защита – это процесс, позволяющий уменьшить улучшение качества, вызванное атакой:

$$\min \left(\left| f(g(att(x))) - f(x) \right| + \lambda dist(x, g(att(x))) \right) \quad (2)$$

где $g: R^{H \times W \times 3} \rightarrow R^{H \times W \times 3}$ – метод защиты. Защита направлена на то, чтобы вернуть атакованное изображение к оригиналу без уменьшения корреляции с субъективным качеством.

3.2 Состязательные атаки

Мы выбрали 10 методов атак для оценки методов защит. Большинство из них первоначально были представлены как атаки на модели классификации изображений; таким образом, мы адаптировали их к задаче IQA, заменив функцию потерь, которая увеличивает оценку качества:

$$L(\theta, x) = 1 - \frac{f_{\theta}(x)}{\max(f_{\theta}) - \min(f_{\theta})} \quad (3)$$

(3) где $\max(f_{\theta})$ и $\min(f_{\theta})$ представляют собой наибольшую и наименьшую субъективные оценки соответственно, полученные метрикой на наборе данных NIPS 2017: Adversarial Learning Development Set [28].

Ниже мы опишем атаки, используемые в нашей работе. Первая группа атак – это метод быстрого знакового градиента (FGSM) и его вариации. FGSM [25] делает один шаг в направлении, противоположном градиенту, чтобы минимизировать объективную функцию (ур. 3). Его модификация I-FGSM [14] делает несколько итераций маленьких шагов в направлении градиента. MI-FGSM [29] дополнительно использует импульс при оптимизации. AMI-FGSM [30] был представлен как атака на модели IQA и представляет собой версию MI-FGSM, которая регулирует допустимую величину атаки путем ограничения качества восприятия, вычисляемого с помощью NR метрики NIQE [31].

В других выбранных методах используются различные техники, чтобы атака оставалась невидимой для человеческого глаза. Korhonen и др. [6] – метод генерации неблагоприятных изображений для оценки качества NR с использованием структурной информации для концентрации возмущений в текстурных областях. SSAH [12] определяет местоположение возмущений в высокочастотных компонентах, используя низкочастотные ограничения. MADC [32] – это метод сравнения метрик качества изображения. Он синтезирует пару изображений, чтобы максимизировать результат одной метрики IQA при неизменном значении другой метрики. Мы используем MADC в качестве атаки на метрику IQA, двигаясь в направлении метрики при фиксированных значениях MSE, как это было предложено в работе Antsiferova и др. [5]. предложили добавить FR метрику в качестве дополнительного элемента оптимизационной функции для создания атаки на метрику NR IQA. Мы использовали три версии этой атаки, где для сохранения визуального качества

использовались метрики SSIM [34], LPIPS [35] и DISTS [36]. Описанные выше атаки являются ограниченными, поскольку возмущения, добавляемые к исходному изображению, ограничены нормой l_p или другими метриками качества. Кроме того, мы используем неограниченную атаку на цветовой фильтр AdvCF [16], которая изменяет цвета изображения путем оптимизации простого цветового фильтра, широко используемого в популярных программах для редактирования фотографий.

3.3 Методы очищения данных

Мы исследуем несколько стандартных методов предварительной обработки для очистки изображений от аддитивного состязательного шума. Первый метод, который мы выбрали, – это техника изменения размера, которая меняет разрешение входных изображений на меньшее, а затем возвращает их к исходному разрешению. Мы провели эксперименты с различными режимами интерполяции. В качестве базового был выбран билинейный режим. Следуя примеру Guo и др. [20], мы случайным образом обрезали и изменяли размер изображений до исходного. Мы также рассмотрели защиту JPEG, которая широко используется для защиты классификации изображений. Кроме того, в наш список методов вошли пространственные преобразования, такие как переворот (зеркальное отображение изображения) и случайное вращение. По аналогии с экспериментами по IQA, проведенными Korhonen и др. [6], мы применили гауссово размытие и двустороннюю фильтрацию. Гауссовый фильтр сглаживает все изображение без учета деталей. Билатеральный фильтр, напротив, нелинейный, учитывает значения интенсивности соседних пикселей, тем самым сохраняя края при сглаживании. Кроме того, мы оценили медианный фильтр, который заменяет каждый пиксель медианой соседних пикселей. Также мы добавили композицию из двух методов: нерезкое маскирование и гауссово размытие. Суть метода заключается в следующем: сначала мы обесцвечиваем изображения и отфильтровываем высокочастотный шум, а затем повышаем резкость краев объектов. Таким образом, на первом этапе происходит фильтрация неблагоприятных возмущений, а на втором – восстановление качества очищенного изображения. Диффузионная модель DiffPure [27] показала свою эффективность в очистке состязательных изображений. Небольшое количество шума вносится в атакованное изображение с помощью прямой прохода модели. Процесс диффузии продолжается до тех пор, пока не будет достигнут оптимально рассчитанный временной шаг. Затем модель делает обратный проход чтобы восстановить очищенное изображение. Цель состоит в том, чтобы возмущения постепенно слились с шумом, в результате чего добавленный гауссовский шум станет доминирующим. В результате обратный процесс удаляет добавленный шум и неблагоприятные возмущения, что приводит к получению очищенного изображения.

Из-за эффекта размытия, создаваемого DiffPure, и с учетом того, что метрики качества снижают свои значения из-за размытия, мы реализовали две дополнительные модификации метода DiffPure для решения этой проблемы. Первый называется DiffPure+Edge и основан на смешивании атакованных и очищенных изображений. Мы выделяем края изображения с помощью фильтра Собеля, а затем заменяем краевые пиксели на краевые пиксели из атакованного изображения. Это позволяет повысить резкость изображения. Второй вариант (DiffPure+Unsharp) аналогичен методу Unsharp. Сначала к атакованному изображению применяется DiffPure для удаления всех нежелательных шумов, а затем нерезкое маскирование повышает резкость изображения, улучшая его визуальное качество.

Кроме того, мы исследуем популярные методы восстановления изображений, такие как MPRNet [37] и RealESRGAN [38], для очистки изображений от состязательных надбавок. MPRNet – это трехступенчатая сверточная нейронная сеть, предназначенная для решения трех задач восстановления изображений, таких как размытие, удаления эффекта дождя и удаления шума. Первые два этапа сети используют архитектуру кодировщик-декодировщик для извлечения контекстной информации на нескольких разрешениях. На последнем этапе изображение обрабатывается в исходном разрешении, чтобы сохранить мелкие детали.

Ключевыми особенностями MPRNet являются модули внимания, расположенные между этапами, и слияние признаков между этими этапами, которое обеспечивает эффективную передачу информации от ранних этапов к поздним. Real-ESRGAN [38] – это модель восстановления, основанная на генеративной состязательной сети, обученной на синтетических данных. Авторы использовали глубокую сеть, состоящую из нескольких остаточных линейных блоков, для выполнения задачи увеличения разрешения. Мы установили масштабный коэффициент 1, чтобы сохранить исходный размер изображения.

Ранее все средства защиты были разработаны для снижения уровня состязательной надбавки в изображениях, что делает их неэффективными против цветных атак, таких как AdvCF [16]. Мы предлагаем новый метод защиты, основанный на компактной полностью сверточной нейронной сети (FCN-фильтр). Наша модель состоит из трех сверточных слоев, которые применяют 64, 16 и 3 фильтра и сохраняют исходные размеры изображения. Набор данных для обучения - 200 изображений с NIPS 2017, атакованных AdvCf, из которых обучающая часть составляет 80 %, а валидационная – оставшиеся 20 %. Мы обучаем его с помощью оптимизатора Адама в течение 200 эпох с коэффициентом обучения $1e-3$ для восстановления оригинального изображения из атакованного, оптимизируя среднюю квадратичную ошибку (MSE). Мы сохраняем модель, показавшую наилучшие результаты на валидационном множестве по показателю SSIM между очищенными и оригинальными изображениями.

3.4 Набор атакованных данных

Мы создали набор данных с атакованными изображениями, чтобы проанализировать эффективность защиты от используемых атак. В качестве исходных изображений использовались изображения из набора данных NIPS 2017: Adversarial Learning Development Set [28], которые подвергались дальнейшим атакам. Этот набор данных ранее использовался для решения широкого круга задач компьютерного зрения, включая атаки на метрики IQA [39]. Атака на метод Linearity проводилась на каждом изображении с помощью всех методов атаки из раздела 3.2 с использованием параметров по умолчанию. Таким образом, из каждого чистого изображения мы получили 10 атакованных изображений. Примеры атакованных изображений представлены на рис. 2. Таблица всех параметров, которые использовались в атаках, представлена в Дополнительных материалах.

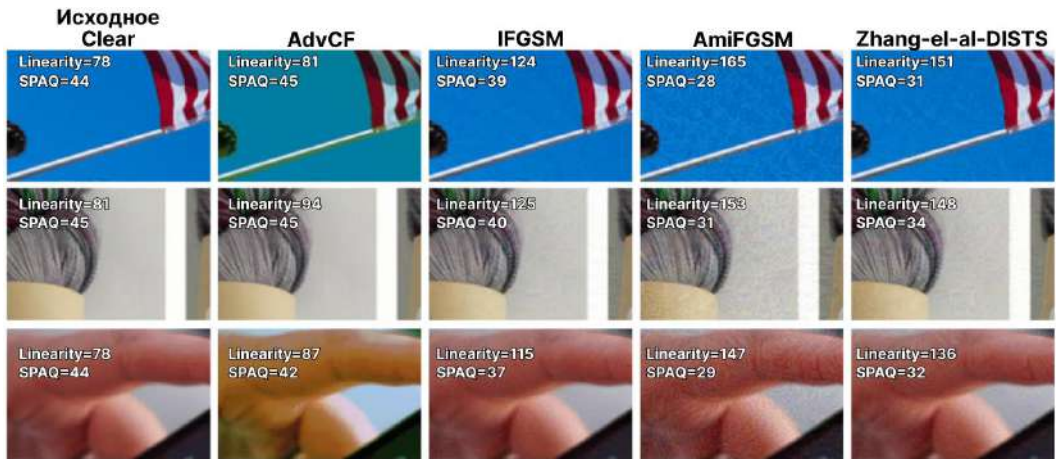


Рис. 2. Примеры атакованных изображений. Первая колонка показывает исходное изображение, следующие показывают атакованные изображения после AdvCF, IFGSM, AmiFGSM, Zhang атак, соответственно.

Fig. 2. Examples of attacks. The first column shows the original (clean) image, the following columns display the corresponding image after the attack.

В качестве атакуемых моделей для атак противника мы использовали три NR метрики качества изображения – Linearity [40], MetaIQA [41] и SPAQ [42]. Авторы Linearity создали собственную функцию потерь, которая сходится примерно в 10 раз быстрее, чем популярные функции потерь MAE и MSE. Linearity, MetaIQA и SPAQ были выбраны в качестве атакуемой метрики, поскольку они демонстрируют высокую производительность (корреляция с субъективным качеством и скоростью) и среднюю устойчивость к атакам противника:

- Корреляция с субъективными оценками и высокая скорость. В MSU No-Reference Video Quality Metrics Benchmark [43] показывает хорошую производительность среди всех метрик качества изображения без ссылок. Эти метрики также отличаются высокой скоростью вычисления по сравнению с аналогами [44].
- Средняя устойчивость. Для нашего исследования нам нужна была целевая метрика IQA со средней устойчивостью к атакам противника. Атаки на высокоустойчивые метрики приведут к получению изображений, которые будут сильно отличаться от чистых, или же атака будет малоэффективна для повышения значений метрики. Таким образом, практическая польза от таких атак незначительна. При использовании метрик с низкой устойчивостью атаки будут визуально незаметны. Такие атаки можно нейтрализовать, применив гауссово размытие с малым размером ядра. Согласно [39], метрики Linearity, MetaIQA и SPAQ демонстрируют среднюю устойчивость по сравнению с другими метриками, что делает ее подходящей целью для нашей работы.

3.5 Детали реализации

Мы использовали открытый исходный код метрик без дополнительного предварительного обучения и выбрали параметры по умолчанию. Гиперпараметры атак также были выставлены по умолчанию. Для обеспечения полной воспроизводимости результатов были использованы инструменты GITLAB CI/CD. Расчеты проводились на компьютере со следующими характеристиками: NVIDIA A100-PCIE-40GB, 32-ядерный процессор Intel Xeon Processor (Icelake) @ 2,90 ГГц. Все расчеты заняли в общей сложности около 50 часов работы GPU.

3.6 Метрики оценивания эффективности

Мы используем две группы метрик для оценки успешности защиты и качества восприятия очищенного изображения для сравнения методов защиты.

Метрики качества оценивают эффективность защиты при обработке изображения с точки зрения визуального качества. Мы выбрали метрики качества изображения PSNR и SSIM из-за их большей устойчивости к атакам противника по сравнению с метриками No-Reference. Кроме того, недифференцируемость этих метрик ставит перед злоумышленниками более сложную задачу. Чтобы объединить эти две метрики в одну оценку, мы складываем их с линейным коэффициентом:

$$Qualityscore = \frac{PSNR}{100} + \frac{SSIM}{2} \quad (4)$$

Эти коэффициенты были выбраны по следующим причинам: для SSIM максимальное значение равно 1. PSNR со значением более 50 означает, что сравниваемые изображения полностью неразличимы, хотя PSNR не имеет максимального значения.

Метрика прироста отражает эффективность защиты по показателям значения метрики. Мы рассчитываем относительный выигрыш для атакованных и очищенных (после последовательного применения атаки и защиты) изображений следующим образом:

$$Gainscore = \frac{|Clear-Distorted|}{Clear} \quad (5)$$

где Distorted – это либо атакованное, либо очищенное изображение, соответствующее чистому изображению. Мы также рассчитали коэффициент корреляции Спирмена (SROCC) между значениями метрики на чистых и очищенных изображениях.

4. Результаты

Мы выделяем три основные характеристики, которыми должны обладать средства защиты: нейтрализация эффектов атак противника, восстановление исходного качества изображения и сохранение корреляций значений атакуемой модели. Для оценки этих характеристик мы сравниваем методы очистки по показателям относительного выигрыша, метрикам PSNR/SSIM и коэффициенту корреляции Спирмена (SROCC) на наборе данных с атаки противника, описанном в разделе 3.4.

Общая эффективность. Табл. 1 показывает усредненные результаты для чистых и атакованных изображений вместе. По показателю качества выходного изображения лидируют методы Upscale, MPRNet и Unsharp. Тем не менее методы DiffPure и DiffPure+Unsharp демонстрируют сопоставимую производительность, отставая от лидера менее чем на 4% как по метрике PSNR, так и по метрике SSIM. Следующие методы лучше всего нейтрализуют последствия атаки DiffPure+Unsharp, DiffPure и изменение размера. Кроме того, DiffPure и DiffPure+Unsharp, наряду с билатеральной фильтрацией, лучше всего сохраняют корреляции между значениями метрик. Метод защиты, заключающийся только в переворачивании изображения по обеим осям, показывает наилучший показатель прироста. Таким образом, этот метод может использоваться для получения значения модели на исходном изображении, при этом по показателю качества он является наилучшим среди рассмотренных. Однако, любую атаку легко адаптировать против такой защиты. RealESRGAN демонстрирует относительно хороший показатель качества, но не справляется с последствиями атак противника и не сохраняет стабильность значений моделей в терминах SROCC. Метод хорошо работает в низкочастотных областях, но создает визуально неприятные артефакты в областях с высокой текстурой. Метод DiffPure+Edge получил один из лучших показателей качества, но показал низкий коэффициент SROCC и показатель прироста. Причиной этого может быть сильный акцент на краях при расчете значений моделей качества.

Подводя итог, можно сказать, что метод DiffPure является наиболее предпочтительным с точки зрения всех трех сравниваемых характеристик. Кроме того, добавление нерезкого маскирования после применения DiffPure несколько повышает их все. Основным недостатком этой техники является ее высокая вычислительная сложность, что приводит к увеличению времени выполнения. Табл. 1 также показывает среднюю скорость вычислений (FPS) для каждого метода.

Эффективность на чистых изображениях. При применении защиты обычно неизвестно заранее, производилась ли атака на это изображение, поэтому мы проверяем эффективность после применения всех защит и на чистых изображениях. Эксперименты показывают, что по показателю прироста простые преобразования, такие как поворот и кадрирование, показывают наилучшие результаты при применении к чистым изображениям. Фильтр FCN не сильно отстает, занимая третье место. Все три метода демонстрируют хороший коэффициент SROCC (выше 0,94), при этом практически не изменяя значения моделей с показателем прироста до 0,04. Однако по показателю качества поворот и обрезка находятся в самом низу рейтинга, в то время как фильтр FCN имеет высокое среднее значение SSIM, превышающее 0,9. По показателю качества лидерами являются четыре метода с очень похожими результатами – DiffPure+Edge, MPRNet, нерезкое маскирование и метод Upscale. Более подробно данные представлены в Табл. 2.

Качество изображений после применения защит. Рис. 3 (слева) иллюстрирует значения показателя прироста для изображений с различными минимальными значениями метрики SSIM. Он считался для всех очищенных изображений, для которых показатель SSIM между соответствующими исходными и атакованными изображениями не меньше порога (точки на оси X). Некоторые методы были исключены для лучшего визуального представления. Для большинства методов показатель прироста остается стабильным при любых значениях SSIM.

Некоторые методы, такие как Upscale, FCN-фильтр и DiffPure+Edge, ведут себя по-разному при слабых атаках (таких, после которых метрика SSIM имеет высокие значения).

Табл. 1. Усредненные показатели эффективности на атакованных изображениях.

Table 1. Results for attacked images.

Метод	Показатель качества ↑	Показатель прироста ↓	SROCC ↑	FPS ↑
Билатеральный фильтр	0.594	0.151	0.731	6.78
Зеркалирование	0.158	0.077	0.728	36.67
Размытие	0.625	0.138	0.471	16.65
JPEG	0.626	0.099	0.725	15.76
Медианный фильтр	0.615	0.118	0.671	16.56
MPRNet	0.653	0.179	0.611	6.33
Обрезка	0.227	0.247	0.626	29.92
Real-ESRGAN	0.636	0.360	0.562	10.15
Resize	0.626	0.099	0.725	63.31
Поворот	0.344	0.239	0.501	36.18
Нерезкое маскирование	0.649	0.133	0.513	15.96
Upscale	0.632	0.207	0.586	35.50
FCN (предложен)	0.591	0.409	0.619	35.56
DiffPure	0.628	0.093	0.706	3.08
+Edge (предложен)	0.660	0.195	0.480	2.98
+Unsharp (предложен)	0.632	0.062	0.750	2.90

Табл.2. Усредненные показатели эффективности на чистых изображениях.

Table 2. Results for clean images.

Метод	Показатель качества ↑	Показатель прироста ↓	PSNR ↑	SSIM ↑	SROCC ↑
Билатеральный фильтр	0.605	0.068	29.185	0.846	0.826
Зеркалирование	0.153	0.043	9.744	0.185	0.886
Размытие	0.651	0.235	30.433	0.923	0.752
JPEG	0.641	0.168	30.585	0.899	0.849
Медианный фильтр	0.635	0.080	29.867	0.896	0.875
MPRNet	0.667	0.070	31.147	0.932	0.854
Обрезка	0.242	0.022	13.655	0.314	0.972
Real-ESRGAN	0.639	0.073	28.250	0.925	0.812
Resize	0.641	0.168	30.585	0.899	0.849
Поворот	0.393	0.036	22.633	0.502	0.962
Нерезкое маскирование	0.663	0.206	31.476	0.932	0.772
Upscale	0.665	0.160	31.282	0.931	0.867
FCN (предложен)	0.606	0.038	24.535	0.906	0.976
DiffPure	0.637	0.149	30.147	0.898	0.955
+Edge (предложен)	0.682	0.074	32.170	0.933	0.958
+Unsharp (предложен)	0.643	0.115	30.584	0.904	0.918

Рис. 3 (справа) показывает значения показателя прироста, основанные на минимальном значении атакуемой модели. В крайней правой части графика показатель рассчитывался

только для сильных атак (то есть значение атакуемой модели велико). В крайнем левом случае он был рассчитан для всего набора данных. Мы видим, что показатель прироста монотонно изменяется по сравнению со значением модели после атаки. Три метода (DiffPure, фильтр Гаусса и нерезкое маскирование) выделяются на фоне остальных – их показатель прироста монотонно уменьшается, показывая, что они лучше справляются с последствиями сильных атак.

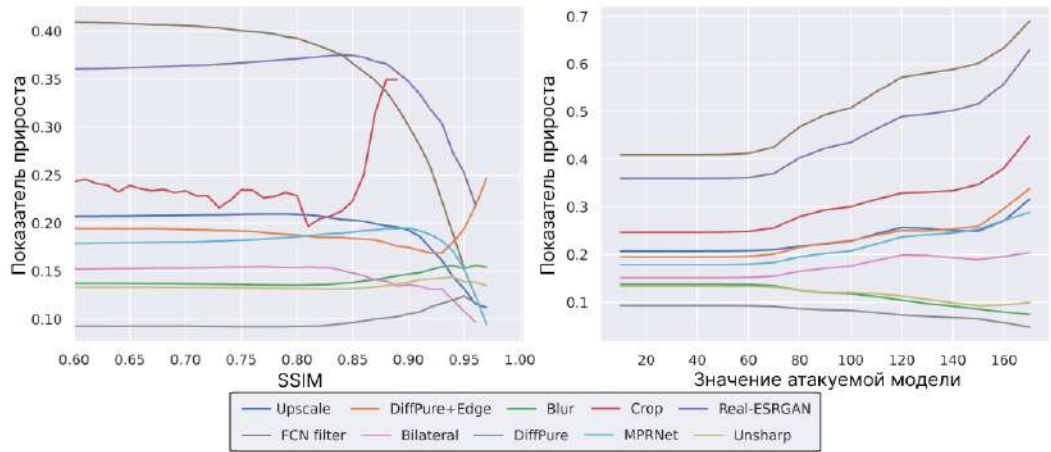


Рис. 3: Gain score в зависимости от минимальных значений SSIM (слева), значения модели после атаки (справа).

Рис. 3: Gain score as a function of minimum SSIM values (left), model's values after attack (right).

Субъективное оценивание. Мы провели субъективное исследование, оценив качество восприятия изображений, полученных с помощью методов защит на краудсорсинговой платформе Subjectify.us. Subjectify.us – это сервис для парных сравнений; он использует модель Брэдли-Терри для преобразования результатов парного голосования в оценку каждого изображения.

Для каждого набора из модели оценки качества, атаки, силы атаки и защиты было выбрано 10 случайных изображений, в результате чего получилось 10 (количество чистых изображений) * 14 (количество атак, включая изображения без атак) * 3 (количество параметров для каждой атаки) * 13 (количество методов защиты) = 5,460 изображений для оценки. Мы исключили 3 метода защиты с простыми геометрическими преобразованиями: поворот, зеркалирование и обрезка, поскольку их качество нельзя напрямую сравнивать с другими методами в субъективном исследовании.

Участникам исследования нужно было оценить пару изображений и указать какое из них имеет лучшее визуальное качество. Каждое изображение в паре имело одно исходное изображение, одну атаку и ее параметры. Различие между ними было только в примененной защите. Участники последовательно просматривали изображения из каждой пары в полноэкранном режиме. Каждый участник должен был сравнить 25 пар, две из которых имели вариант с более высоким качеством и служили проверочными вопросами. Все ответы тех, кто не смог правильно ответить на проверочные вопросы, были отброшены.

Чтобы обеспечить достоверность результатов, было получено не менее 10 голосов для каждой пары изображений.

В общей сложности мы собрали 72 425 ответов от 2 897 человек. Применяв модель Брэдли-Терри к таблице парных рангов, мы получили субъективные оценки для каждого изображения. Результаты субъективного сравнения можно найти на рис. 4.

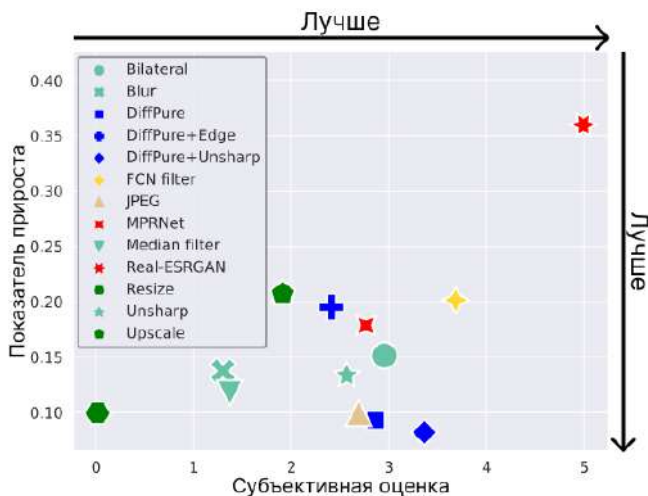


Рис. 4: Субъективные оценки качества и показатель прироста для различных защит, усредненные по всем атакам.

Рис. 4: Subjective scores and gain score for different defences averaged for all attacks.

5. Заключение

В данной работе мы исследовали эффективность методов очистки от состязательных атак в качестве защиты от атак на метрики оценки качества изображений. Мы провели обширное исследование, включающее 10 атак и 16 методов очистки, и опубликовали набор данных изображений, подвергшихся атакам.

Результаты показали, что даже простые и быстрые методы, такие как поворот или зеркалирование изображения, могут нивелировать эффекты атаки и сохранить визуальное качество очищенного изображения близким к исходному. Более сложные средства защиты могут сохранять и восстанавливать исходное качество: предложенные нами комбинации DiffPure с нерезким маскированием обеспечивают наивысший SROCC, сохраняя высокое качество очищенных изображений и Gain score. Эффективность протестированных защит схожа для различных итеративных градиентных атак, но отличается для неограниченных атак типа AdvCf. Такие атаки могут быть нейтрализованы путем обучения нейронной сети на состязательных примерах. В нашем исследовании мы предложили фильтр FCN, который эффективно защищает метрики оценки качества от атаки AdvCF.

Список литературы / References

- [1]. Duanmu, Z., Liu, W., Wang, Z., Wang, Z.: Quantifying visual image quality: A bayesian view. *Annual Review of Vision Science* 7, 437–464 (2021).
- [2]. Zvezdakova, A., Zvezdakov, S., Kulikov, D., Vatin, D.: Hacking vmaf with video color and contrast distortion. *arXiv preprint arXiv:1907.04807* (2019).
- [3]. Shumitskaya, E., Antsiferova, A., Vatin, D.: Universal perturbation attack on differentiable no-reference image- and video-quality metrics (2022).
- [4]. Shumitskaya, E., Antsiferova, A., Vatin, D.: Towards adversarial robustness verification of no-reference image- and video-quality metrics. *Computer Vision and Image Understanding* 240, 103913 (2024).
- [5]. Zhang, W., Li, D., Min, X., Zhai, G., Guo, G., Yang, X., Ma, K.: Perceptual attacks of no-reference image quality models with human-in-the-loop. *Advances in Neural Information Processing Systems* 35, 2916–2929 (2022).
- [6]. Korhonen, J., You, J.: Adversarial attacks against blind image quality assessment models. *Proceedings of the 2nd Workshop on Quality of Experience in Visual Multimedia Applications* (2022), <https://api.semanticscholar.org/CorpusID: 252546140> 16 F. Author et al.

- [7]. Luo, C., Lin, Q., Xie, W., Wu, B., Xie, J., Shen, L.: Frequency-driven imperceptible adversarial at-tack on semantic similarity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15315–15324 (June 2022).
- [8]. Li, Z.: On vmaf’s property in the presence of image enhancement operations (2021).
- [9]. Ghazanfari, S., Garg, S., Krishnamurthy, P., Khorrami, F., Araujo, A.: R-lpips: An adversarially ro-bust perceptual similarity metric. arXiv preprint arXiv:2307.15157 (2023).
- [10]. Kettunen, M., Härkönen, E., Lehtinen, J.: E-lpips: robust perceptual image similarity via random transformation ensembles. arXiv preprint arXiv:1906.03973 (2019).
- [11]. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. CoRR abs/1312.6199 (2013), <https://api.semanticscholar.org/CorpusID:604334> (дата обращения 12.09.2024).
- [12]. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks (2017).
- [13]. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2015).
- [14]. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world (2017).
- [15]. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks (2019).
- [16]. Zhao, Z., Liu, Z., Larson, M.: Adversarial color enhancement: Generating unrestricted adversarial images by optimizing a color filter (2020).
- [17]. Sang, Q., Zhang, H., Liu, L., Wu, X., Bovik, A.: On the generation of adversarial samples for image quality assessment. SSRN Electronic Journal (01 2022). <https://doi.org/10.2139/ssrn.4112969> (дата обращения 12.09.2024).
- [18]. Ghildyal, A., Liu, F.: Attacking perceptual similarity metrics (2023).
- [19]. Graese, A., Rozsa, A., Boulton, T.E.: Assessing threat of adversarial examples on deep neural networks (2016).
- [20]. Guo, C., Rana, M., Cisse, M., van der Maaten, L.: Countering adversarial images using input transformations (2018).
- [21]. Das, N., Shanbhogue, M., Chen, S.T., Hohman, F., Chen, L., Kounavis, M.E., Chau, D.H.: Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression (2017).
- [22]. Dziugaite, G.K., Ghahramani, Z., Roy, D.M.: A study of the effect of jpg compression on adversarial images (2016).
- [23]. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. CoRR abs/1704.01155 (2017), <http://arxiv.org/abs/1704.01155> (дата обращения 12.09.2024).
- [24]. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60, 259–268 (1992), <https://api.semanticscholar.org/CorpusID:13133466>.
- [25]. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples (2017).
- [26]. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-gan: Protecting classifiers against adversarial attacks using generative models (2018).
- [27]. Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., Anandkumar, A.: Diffusion models for adversarial purification (2022).
- [28]. NIPS 2017: Adversarial learning development set. <https://www.kaggle.com/datasets/google-brain/nips-2017-adversarial-learning-development-set> (2017) (дата обращения 12.09.2024).
- [29]. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9185–9193 (2018).
- [30]. On the generation of adversarial examples for image quality assessment. *Visual Computer* (2023). <https://doi.org/10.1007/s00371-023-03019-1> (дата обращения 12.09.2024).
- [31]. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a “completely blind” image quality analyzer. *IEEE Signal Processing Letters* 20, 209–212 (2013), <https://api.semanticscholar.org/CorpusID:16892725>.
- [32]. Wang, Z., Simoncelli, E.: Maximum differentiation (mad) competition: A methodology for comparing computational models of perceptual quantities. *Journal of vision* 8, 8.1–13 (02 2008). <https://doi.org/10.1167/8.12.8> (дата обращения 12.09.2024).
- [33]. Antsiferova, A., Abud, K., Gushchin, A., Shumitskaya, E., Lavrushkin, S., Vatolin, D.: Comparing the robustness of modern no-reference image- and video-quality metrics to adversarial attacks (2024).
- [34]. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4), 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861> (дата обращения 12.09.2024).

- [35]. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018).
- [36]. Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence* 44(5), 2567–2581 (2020).
- [37]. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: CVPR (2021) Adversarial purification for no-reference image-quality metrics 17.
- [38]. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind superresolution with pure synthetic data. In: International Conference on Computer Vision Workshops (ICCVW).
- [39]. Antsiferova, A., Abud, K., Gushchin, A., Shumitskaya, E., Lavrushkin, S., Vatin, D.: Comparing the robustness of modern no-reference image- and video-quality metrics to adversarial attacks (2024).
- [40]. Li, D., Jiang, T., Jiang, M.: Norm-in-norm loss with faster convergence and better performance for image quality assessment. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 789–797 (2020).
- [41]. Zhu, H., Li, L., Wu, J., Dong, W., Shi, G.: MetaIQA: Deep meta-learning for no-reference image quality assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14143–14152 (2020).
- [42]. Fang, Y., Zhu, H., Zeng, Y., Ma, K., Wang, Z.: Perceptual quality assessment of smartphone photography. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3677–3686 (2020).
- [43]. Antsiferova, A., Lavrushkin, S., Smirnov, M., Gushchin, A., Vatin, D., Kulikov, D.: Video compression dataset and benchmark of learning-based video-quality metrics. *Advances in Neural Information Processing Systems* 35, 13814–13825 (2022).
- [44]. https://videoprocessing.ai/benchmarks/video-quality-metrics_nrm.html (дата обращения 12.09.2024).

Информация об авторах / Information about authors

Александр Евгеньевич ГУЩИН получил степень магистра по прикладной математике и информатике в Московском государственном университете имени М. В. Ломоносова в 2024 году. В настоящее время учится в аспирантуре и ведет исследования в лаборатории компьютерной графики и мультимедиа, а также в Центре доверенного искусственного интеллекта ИСП РАН. В область его научных интересов входят методы оценивания качества видео и изображений, а также исследование устойчивости нейросетевых моделей и методов их защиты.

Aleksandr Evgenevich GUSHCHIN received his master degree in computer science from the Moscow State University in 2024. He is currently a postgraduate student at the MSU Graphics & Media Lab, and a researcher at the Research Centre for Trusted AI of ISP RAS. His research interests involve image and video processing, quality assessment, and machine learning. He is also a key contributor to the project analyzing video quality assessment methods, including their robustness to adversarial attacks.

Анастасия Всеволодовна АНЦИФЕРОВА аспирантка ВМК МГУ. Окончила магистратуру ВМК МГУ по специальности анализ больших данных в 2018 году. В настоящее время она является аспирантом и участником видеогруппы лаборатории графики и мультимедиа МГУ, ведет исследования в Центре доверенного искусственного интеллекта ИСП РАН. Сфера ее научных интересов включает анализ и оптимизацию видеокодеков, оценку субъективного качества стереоскопического видео. Анастасия является одним из организаторов проекта международного сравнения видеокодеков, ежегодно проводимого в МГУ, и проекта измерения качества 3D-видео.

Anastasia Vsevolodovna ANTSEFEROVA – received her master degree in computer science from Moscow State University in 2018. She is a postgraduate student at Moscow State University and a member of Video Group in MSU Graphics&Media Lab. She is also a researcher at the Research Centre for Trusted AI of ISP RAS. Her research interests involve video codecs analysis and optimization, stereoscopic video subjective quality assessment. She is one of the contributors to MSU Video Codec Comparison Project and to the 3D video quality measurement project.

Дмитрий Сергеевич ВАТОЛИН закончил ВМК МГУ в 1996 году, кандидат физико-математических наук, заведующий лабораторией компьютерной графики ВМК МГУ, исследователь Центра доверенного искусственного интеллекта ИСП РАН. Читает курсы по компьютерной графике и методам сжатия и обработки видео с 1997 года. Создатель популярных сайтов, посвященных алгоритмам

обработки и сжатия видео. Специализируется на исследованиях в области алгоритмов сжатия видео, современных методах измерения качества и обработке цифрового видео. Руководил проектами с компаниями Intel, Cisco, Real Networks, Samsung, Huawei, Broadcom и некоторыми другими. С 2008 года занимается измерением и исправлением артефактов стерео, в том числе организовал проект измерения качества стереофильмов.

Dmitriy Sergeevich VATOLIN – Cand. Sci. (Phys.-Math.) from Moscow State University, head of the MSU Graphics & Media Lab and MSU AI Institute Video Analysis Lab, and a researcher at the Research Centre for Trusted AI of ISP RAS. His research interests include compression methods, video processing, 3D video techniques (depth from motion, focus and other cues, video matting, background restoration, high-quality stereo generation), as well as video quality assessment and robustness of modern video quality metrics. He is a key cofounder of the 3D video quality measurement project, his most known project is the annual MSU Video Codecs Comparison, that includes up to 25 modern codecs compared subjectively and objectively in several nominations with detailed 20000+ charts.



Нейросетевой метод для стабильного во времени матирования видеопоследовательностей с людьми

^{1,2} И.А. Молодецких, ORCID: 0000-0002-8294-0770 <ivan.molodetskikh@graphics.cs.msu.ru>

¹ М.В. Ерофеев, ORCID: 0000-0001-7786-4358 <merofeev@graphics.cs.msu.ru>

¹ А.В. Москаленко, ORCID: 0000-0003-4965-0867 <andrey.moskalenko@graphics.cs.msu.ru>

^{1,2} Д.С. Ватолин, ORCID: 0000-0002-8893-9340 <dmitriy@graphics.cs.msu.ru>

¹ Московский государственный университет имени М.В. Ломоносова,
Россия, 119991, Москва, Ленинские горы, д. 1.

² Институт системного программирования им. В.П. Иванникова РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

Аннотация. Предлагается новый метод матирования видео с людьми на основе нейросетей, не требующий дополнительных входных данных, таких как тернарные карты. Разработанная нейросетевая архитектура достигает стабильности получаемых карт прозрачности во времени за счёт свёрточных LSTM-модулей в прямых соединениях U-Net в сочетании со сглаживанием карт сегментации на основе поблочной оценки движения. Также предлагается алгоритм генерации искусственного движения, генерирующий обучающие видеоклипы для сети матирования видео, используя фотографии с эталонными картами прозрачности и фоновые видео. К фотографиям и их картам прозрачности применяются случайные карты сдвигов, имитирующие движение в реальных видео. Затем производится композиция результата с фоновыми клипами. Искусственное движение позволяет обучать глубокие нейросети, работающие с видео, в отсутствие большого размеченного набора видеоданных, и предоставляет эталонный оптический поток переднего плана обучающих видео для использования в функциях потерь.

Ключевые слова: матирование видео; семантическое матирование людей; сглаживание во времени; глубокое обучение.

Для цитирования: Молодецких И.А., Ерофеев М.В., Москаленко А.В., Ватолин Д.С. Нейросетевой метод для стабильного во времени матирования видеопоследовательностей с людьми. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 85–106. DOI: 10.15514/ISPRAS-2025-37(3)-6.

Благодарности: Исследование поддержано грантом центрам искусственного интеллекта Аналитического центра при Правительстве РФ, соглашение № 000000D730321P5Q0002, и соглашением с ИСП РАН от 02.11.2021 № 70-2021-00142.

Temporally Coherent Person Matting Trained on Fake-Motion Dataset

^{1,2} I.A. Molodetskikh, ORCID: 0000-0002-8294-0770 <ivan.molodetskikh@graphics.cs.msu.ru>

¹ M.V. Erofeev, ORCID: 0000-0001-7786-4358 <merofeev@graphics.cs.msu.ru>

¹ A.V. Moskalenko, ORCID: 0000-0003-4965-0867 <andrey.moskalenko@graphics.cs.msu.ru>

^{1,2} D.S. Vatolin, ORCID: 0000-0002-8893-9340 <dmitriy@graphics.cs.msu.ru>

¹ Lomonosov Moscow State University,

GSP-1, Leninskie Gory, Moscow, 119991, Russia.

² Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Abstract. We propose a novel neural-network-based method to perform matting of videos depicting people that does not require additional user input such as trimaps. Our architecture achieves temporal stability of the resulting alpha mattes by using motion-estimation-based smoothing of image-segmentation algorithm outputs, combined with convolutional-LSTM modules on U-Net skip connections. We also propose a fake-motion algorithm that generates training clips for the video-matting network given photos with ground-truth alpha mattes and background videos. We apply random motion to photos and their mattes to simulate movement one would find in real videos and composite the result with the background clips. It lets us train a deep neural network operating on videos in an absence of a large annotated video dataset and provides ground-truth training-clip foreground optical flow for use in loss functions.

Keywords: video matting; semantic person matting; temporal smoothing; deep learning.

For citation: Molodetskikh I.A., Erofeev M.V., Moskalenko A.V., Vatolin D.S. Temporally coherent person matting trained on fake-motion dataset. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 85-106 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-6

Acknowledgements. This work was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement [agreement identifier 000000D730321P5Q0002] and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated November 2, 2021 [agreement number 70-2021-00142].

1. Введение

Матирование, или нахождение карты прозрачности объекта переднего плана, является одной из основных операций обработки изображений и видеопоследовательностей. Задача матирования – определить точные значения прозрачности пикселей объекта на переднем плане, а также цвета переднего и заднего плана. Результатом матирования является «вырезанное» изображение объекта, которое можно вставить на другой, произвольный фон при помощи композиции.

Помимо систем видеоконференций, замена фона, а значит определение прозрачности пикселей переднего плана, находит широкое применение в проведении прямых трансляций, в задачах дополненной реальности и в киноиндустрии в области специальных эффектов. С развитием технологий компьютерного зрения и компьютерной графики значительно возросло число сцен, для съёмки которых быстрее и выгоднее создать искусственный задний план при помощи компьютерного моделирования и прочих подобных техник, чем строить специальные павильоны и декорации. Для таких сцен актёры снимаются отдельно, а затем вставляются на искусственный фон при помощи композиции.

Наиболее надёжным и, до недавних пор, популярным способом выделения объектов переднего плана при съёмке является техника хромакея, или съёмки на заднем плане, окрашенном в определённый ключевой цвет, как правило, зелёный. Далее в отснятом таким образом видео пиксели цвета, близкого к ключевому, меняются на прозрачные с помощью

программного инструмента редактирования видео. Именно хромакей использовался для замены фона в ранних версиях системы Zoom.

Однако техника хромакея накладывает ряд ограничений. Самым существенным из них является необходимость покупки и установки зелёного фоновоего экрана. Для большинства пользователей систем видеоконференций это не рационально в финансовом плане и представляет значительные неудобства в связи с размерами экрана, который нужно разместить в комнате перед рабочим местом. Также необходимо помнить про ограничения на цвета одежды. Дополнительно усложняет использование хромакея необходимость в однородном освещении экрана.

Матирование помогает решить эти проблемы, поскольку не требует специального заднего плана и какого-либо оборудования со стороны пользователя. Однако, в связи с этим матирование – гораздо более сложная задача, чем выделение объекта при помощи хромакея, потому что в общем случае неизвестно, какие из объектов являются объектами переднего плана и должны быть выделены.

Для указания объектов в задаче матирования часто используются тернарные карты. На них каждый пиксель отмечен одним из трёх цветов: принадлежность к переднему плану, принадлежность к заднему плану или неопределённая принадлежность. Алгоритму матирования в таком случае нужно найти значения прозрачности у неопределённых пикселей. Примеры подобных алгоритмов описаны в работах [1-3]. Однако существуют другие ручные и автоматические способы указания принадлежности объектов переднему или заднему плану, например, мазки или нейросетевые признаки.

Конкретно для систем видеоконференций необходим полностью автоматический поиск объектов переднего плана, например, с использованием нейросетевого алгоритма. В этом случае в качестве первого шага используется нейросеть для сегментации изображений, определяющая пиксели, принадлежащие определённому классу, в данном случае «человек». Так алгоритм матирования будет автоматически выделять людей на видеопотоке без каких-либо дополнительных входных данных.

Дополнительной сложностью в задаче матировании видео является поддержание стабильности полученных карт прозрачности во времени. Даже при наличии дополнительных входных данных, пок кадровое применение большинства алгоритмов матирования изображений даёт карты прозрачности, которые заметно вибрируют и мерцают при преобразовании их обратно в видеоформат, что является неприятным и отвлекающим артефактом.

Особенно часто мерцание проявляется на мелких деталях, таких как кончики волос людей на переднем плане. Одним из способов решения проблемы стабильности во времени является матирование кадров пониженного разрешения и последующее повышение разрешения полученных карт прозрачности обратно до исходного. Таким образом, алгоритм матирования выполняет основную задачу выделения человека в кадре, а алгоритм повышения разрешения далее восстанавливает мелкие детали, основываясь на исходных кадрах видео, что позволяет уменьшить или избежать мерцания.

В данной работе для решения проблемы мерцания предлагается алгоритм темпорального сглаживания результатов работы метода сегментации, а также свёрточные LSTM-слои [4] в основном методе матирования.

При обучении нейросетевых алгоритмов, одним из основных факторов, влияющих на финальное качество, является количество и качество данных для обучения. На момент написания, публично доступные наборы видео с эталонными картами прозрачности людей сильно ограничены в размерах, и таких наборов немного. При этом, существует несколько больших наборов фотографий людей с эталонными картами прозрачности [5-6]. Использование данных наборов фотографий потенциально может сильно повысить качество и обобщаемость обучаемой нейросети матирования.

Поэтому, в данной работе также предлагается новый алгоритм генерации искусственного движения, с помощью которого из фотографии человека на переднем плане и видео заднего плана можно получить видео обучающий пример, путём плавного искажения переднего плана в течение нескольких кадров. В данной работе показано, что предложенное искусственное движение может быть использовано для обучения нейросети, которая впоследствии корректно работает на настоящих входных видео и выдаёт стабильные во времени карты прозрачности.

Основной вклад данной работы состоит в следующем:

- Предложен новый метод для матирования видео с людьми на переднем плане без дополнительных входных данных на основе глубокой нейросети с архитектурой U-Net, блоками LSTM и модулем внимания на прямых соединениях.
- Предложен новый алгоритм искусственного движения для генерации обучающих видеоклипов для нейросети, используя изображения с эталонными картами прозрачности и видео заднего плана.
- Предложен новый метод сглаживания во времени карт вероятности метода сегментации изображений, который значительно увеличивает их временную стабильность.

2. Обзор литературы

В данном разделе приведён обзор существующих подходов к задаче матирования изображений и видео, уделяя основное внимание полу- и полностью автоматическим семантическим методам.

2.1 Методы матирования изображений

В настоящее время активно развивается область матирования изображений – подзадача матирования видеопоследовательностей. Это в значительной степени связано со скачком в развитии нейросетей: матирование изображений – одна из задач, в которых нейросетевые подходы показали себя намного лучше классических. К тому же, алгоритмы матирования изображений менее ресурсозатратные по сравнению с алгоритмами матирования видео, что упрощает их интеграцию в системы видеоконференций, где они должны работать на пользовательских устройствах. Следующие работы наиболее релевантны разработанному методу; подробный обзор классических алгоритмов можно найти в работах [7-8].

Первым алгоритмом автоматического матирования фотографий людей без необходимости дополнительных входных данных был алгоритм, предложенный в [6]. Авторы создали набор данных, состоящий из 2000 портретов людей с эталонными картами прозрачности, и обучили нейросеть, предсказывающую тернарную карту по изображению. Затем изображение и тернарная карта подавались в классический алгоритм матирования для нахождения значений прозрачности в неизвестных областях. Также нейросети на вход подавалась «средняя» карта прозрачности по всем фотографиям из набора данных, выровненная относительно входной фотографии, что уменьшило число фоновых пикселей, ошибочно отмечаемых нейросетью как пиксели переднего плана.

Авторы [9] также предлагают решение задачи автоматического матирования портретов людей. Для этого используется архитектура из двух нейросетей: первая сеть предсказывает тернарную карту, а вторая сеть использует её в качестве дополнительных входных данных для предсказания карты прозрачности. Затем карта прозрачности комбинируется с тернарной картой для получения результирующей карты. Нейросети обучаются сначала по отдельности, а затем совместно. Кроме того, большим вкладом данной работы является набор из 34 425 портретов людей с эталонными картами прозрачности, опубликованный осенью 2019 года в открытом доступе [5].

В работе [10] предлагается новый нейросетевой блок для повышения разрешения карт признаков на этапе декодирования. Авторы заметили, что все используемые на текущий момент операции повышения разрешения (билинейная интерполяция, метод ближайшего соседа, транспонированная свёртка, повышение разрешения с использованием индексов с соответствующего слоя с этапа кодирования) можно обобщить на последний случай и предложили использовать обучаемый свёрточный слой для получения карты индексов. Использование подобных блоков для повышения разрешения привело к повышению качества работы ряда нейросетевых методов матирования изображений.

Метод, предложенный в [1] использует две нейросети. Первая, построенная по архитектуре кодировщик-декодировщик, выдаёт изначальное предсказание карты прозрачности, а вторая, состоящая из небольшого числа свёрточных слоёв, улучшает изначальное предсказание. За счёт отсутствия операций понижения и повышения разрешения, вторая сеть лучше обрабатывает мелкие детали изображения и, таким образом, увеличивает детализацию результирующей карты прозрачности.

Нейросетевой подход, предложенный в статье [11], ориентирован на быстрое выполнение на мобильных устройствах. Авторы использовали легковесную архитектуру с поканальными свёртками и квантование нейросети, что позволило им достичь скорости работы алгоритма, равной 30 кадрам в секунду, на современных мобильных устройствах. Данный подход ориентирован на автоматическое матирование портретов, и, соответственно, не принимает на вход тернарные карты. Вместо этого информация о принадлежности пикселей фону или переднему плану определяется самой нейросетью на основании обученной классификации людей в кадре. Основным ограничением данного подхода является низкое качество карт прозрачности, особенно для фотографий высокого разрешения.

Метод, предложенный авторами [12], может автоматически генерировать карты прозрачности для изображений с разными объектами переднего плана. В нём используется нейросеть из трёх частей: модули предсказания вероятности переднего и заднего плана и объединяющая сеть, которая выдаёт карту коэффициентов смешивания для вычисления итоговой карты прозрачности.

Авторы [13] предложили алгоритм, состоящий из нескольких нейронных сетей, что позволило им использовать как грубые, так и детальные эталонные карты прозрачности портретных фотографий. Грубые карты легко размечать, в то время как детальные карты размечать очень затратно по времени, но они сильно помогают в обучении хорошей нейросети для матирования. Сначала, нейросеть предсказания маски выдаёт грубую семантическую маску. Затем, нейросеть унификации качества увеличивает качество маски, чтобы сделать её похожей на детальные эталонные карты. Наконец, нейросеть улучшения матирования использует полученную маску как ориентир для генерации финальной карты прозрачности.

2.2 Методы матирования видеопоследовательностей

В отличие от задачи матирования изображений, задача матирования видеопоследовательностей исследуется менее активно. Наибольший интерес здесь представляют методы, принимающие на вход тернарную карту не на каждый кадр видеопоследовательности, а только на отдельные, ключевые кадры. Здесь всё ещё преобладают классические подходы, «протягивающие» карту прозрачности с ключевого кадра на последующие.

Метод [14] использует два ключевых кадра в начале и конце фрагмента видеопоследовательности, и обеспечивает стабильность полученной карты прозрачности во времени между этими двумя кадрами. Авторы предложили метод построения оптического потока между кадрами на основе алгоритма PatchMatch [15] и способ «протягивания» карт прозрачности с использованием построенного оптического потока. Отдельным плюсом

данного метода является возможность использовать произвольный базовый алгоритм матирования изображений для получения карт прозрачности на ключевых кадрах.

Авторы [16] предлагают классический подход, основанный на представлении пикселей в областях переднего и заднего плана произведением матрицы словаря и разреженной кодовой матрицы. Словарь обучается на входной видеопоследовательности с помощью оптимизационного процесса, принимающего во внимание стабильность во времени. К сожалению, в данном подходе требуются дополнительные входные данные в виде тернарных карт на каждый кадр входного видео, что ограничивает его применимость.

В работе [17] предлагается метод матирования портретных видео без использования дополнительных входных данных в виде тернарных карт или мазков. Однако, метод требует снятых вручную изображений-примеров фона видео, которые помогают нейросети обрабатывать сложные фоновые детали.

Для работы метода, предложенного в [18], от пользователя вместо тернарной карты требуется одна фотография заднего плана. Точное выравнивание обрабатываемого видео и фотографии заднего плана не требуется. В этом методе используется нейросеть для сегментации изображений. Одним из ограничений является динамический задний план, например, водопад, или сильное движение камеры при съёмке. В этих случаях алгоритм может ошибочно отметить какие-то элементы фона как передний план.

Авторы [19] предлагают метод сегментации объектов видео, использующий генерацию синтетических обучающих примеров при помощи случайных геометрических и цветовых преобразований как к композиции передних и задних планов, так и к передним и задним планам по отдельности. В следующей статье [20] авторы улучшили свой метод путём генерации коротких видеоклипов по похожему принципу.

2.3 Наборы данных

Существует несколько общедоступных наборов фотографий и видео людей с эталонными картами прозрачности.

- Набор данных AISegment.com [5], собранный авторами [9], содержащий 34 425 портретных фотографий людей с эталонными картами прозрачности.
- 2000 портретных фотографий с эталонными картами прозрачности, собранных авторами [6].
- Наборы данных PhotoMatte85 и VideoMatte240K [21], собранные авторами [22], которые содержат 85 портретных фотографий и 484 видеоклипов с людьми с эталонными картами прозрачности.

Эти наборы данных были сняты на зелёном фоне с помощью хромакея, что означает, что у них нет «родного» фона, и их можно использовать только для композиции с другим задним планом.

2.4 Выводы

Задача матирования изображений с использованием тернарных карт в последние годы активно исследуется. Это связано с возросшей популярностью нейросетевых алгоритмов для решения задач компьютерного зрения. Все наилучшие алгоритмы матирования в эталонном тесте <https://alphamatting.com> основаны на нейросетях.

Постепенно начинает исследоваться задача автоматического матирования без необходимости дополнительных входных данных в виде тернарных карт или мазков. В основном методы в данной области направлены на матирование портретов людей, что может быть связано с возрастающей популярностью мобильных приложений для наложения эффектов на фотографии.

Задача матирования видео исследуется менее активно. Это может быть связано с вычислительной сложностью в применении нейросетей к видеопоследовательностям и отсутствию достаточно больших наборов данных для обучения. Однако, в последнее время начинают появляться многообещающие подходы с использованием ограниченных входных данных, например, одного изображения фона, такие как [17-18].

Многие из подходов матирования изображений [23-25, 9-10] можно адаптировать для применения к задаче автоматического матирования видео. Также большую ценность для данной работы составляют наборы портретов людей с эталонными картами прозрачности [6, 9, 21], которые можно использовать для создания видеопоследовательностей для обучения нейросети.

3. Предложенный метод

Предложенный подход основан на свёрточной нейронной сети для предсказания карты прозрачности на основе карты вероятности, получаемой предобученным методом семантической сегментации изображений. Отсутствие полностью связанных слоёв в нейросети даёт возможность подавать на вход кадры произвольного размера. Схема архитектуры нейросетевого алгоритма представлена на рис. 1.

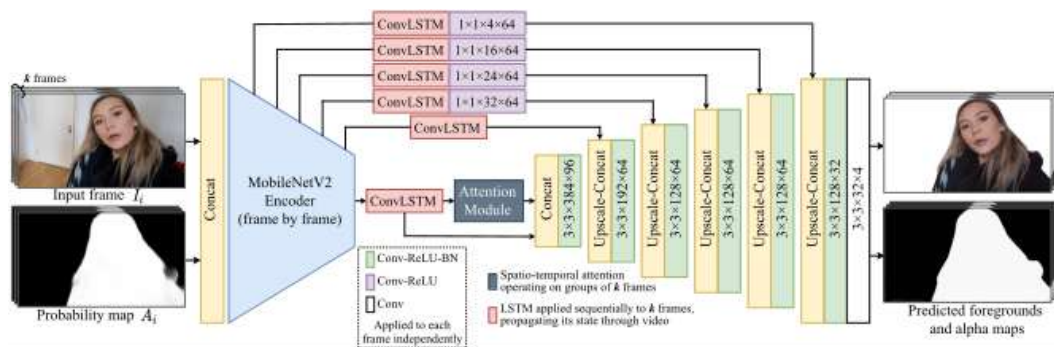


Рис. 1. Архитектура предложенной нейросети.

Fig. 1. Architecture of the proposed neural network.

Для учёта стабильности во времени в архитектуре используются модули долгосрочной кратковременной памяти (LSTM) [26], а именно их свёрточная адаптация [4]. Модули помещаются на пути прямых соединений между слоями кодировщика и декодировщика. Использование данных модулей позволяет нейросети передавать информацию о предсказываемых картах прозрачности вперёд во времени, что даёт возможность добавить в функцию потерь слагаемое, штрафующее за отсутствие стабильности.

На каждом шаге во времени s , нейросеть принимает на вход набор k последовательных кадров видео $\{I_i\}_{i=s}^{s+k}$, соответствующие карты вероятности $\{A_i\}_{i=s}^{s+k}$ и скрытое состояние LSTM с предыдущего кадра H_{s-1} . Результатом шага нейросети являются предсказанные карты прозрачности $\{\alpha_i\}_{i=s}^{s+k}$, пиксели переднего плана $\{FG_i\}_{i=s}^{s+k}$, а также новое скрытое состояние LSTM H_{s+k} .

Временная информация содержится в скрытом состоянии модулей LSTM. Это состояние инициализируется нулевыми значениями, и затем сохраняется между запусками нейросети на последовательных кадрах видео.

Предложенный подход может быть адаптирован для матирования других классов объектов путём использования соответствующего выходного класса объекта метода семантической сегментации изображений и переобучения предложенной нейросети на наборе данных матирования нужного класса объекта. Однако, фокус данной работы именно на матировании

людей, так как эта задача на текущий момент имеет много практических применений, и соответственно много наборов данных для обучения и экспериментальной оценки.

3.1 Архитектура нейросети

Нейросеть построена по модели кодировщик-декодировщик с прямыми соединениями между слоями кодировщика и декодировщика на одинаковой глубине. Подобный способ построения («U-Net») используется для решения большого количества различных задач компьютерного зрения. Прямые соединения помогают нейросети лучше восстанавливать мелкие детали изображений, теряющиеся при сжатии информации в размерность самого глубокого представления.

В качестве основы для кодировщика взята MobileNetV2 [27], так как данная архитектура хорошо себя показывает в задачах матирования видео [2][1]. Число каналов первого слоя сети увеличено с трёх до четырёх для передачи на вход карты вероятности. Для прямых соединений берутся выходы 2-го, 4-го, 7-го, 14-го и 18-го блоков сети в реализации TorchVision [28]. Прямые соединения проходят через свёрточные LSTM-модули, которые дают результирующим картам прозрачности стабильность во времени.

Кроме того, на самом глубоком прямом соединении используется модуль внимания, основанный на предложенном в работе [29]. Сначала, карта признаков размерностью в 320 каналов конкатенируется с тремя дополнительными каналами, содержащими для каждого пикселя его номер кадра, x -координату и y -координату. Эти дополнительные каналы часто называют картой индексов. Затем, два полносвязных слоя уменьшают число каналов «выпрямленной» карты признаков с 323 до 64 и генерируют карты запроса и ключа для вычисления внимания. Для карты значения также используется полносвязный слой, но здесь используется 320-канальная карта признаков сама по себе, без координат пикселей. Далее применяется масштабированная операция внимания с использованием скалярного произведения, и восстанавливается двумерная размерность результата.

В то время как модули LSTM обрабатывают кадры последовательно и накапливают информацию во времени, модуль внимания оперирует на группах по k кадров за раз, распространяя информацию внутри каждой группы.

3.2 Определение положения людей

Первым шагом в работе предложенного алгоритма является определение положения людей на входных кадрах видео – получение карты вероятности A_i . Для этого используется широко применяемая на практике нейросетевая архитектура для сегментации изображений DeepLabv3+ [30] с базовой архитектурой Xception [31], предобученная на наборе данных PASCAL VOC 2012 [32]. Конкретно, была взята реализация TorchVision [28].

Перед подачей в DeepLabv3+, входное изображение изменяется в размере до 520×520 пикселей при помощи билинейной интерполяции. Далее к выходной карте признаков применяется поканальная операция softmax. Результат – карта вероятностей принадлежности каждого пикселя изображения к каждому из 21 класса набора данных PASCAL VOC. Из этой карты выделяется слой, соответствующий классу «человек», и увеличивается или уменьшается до исходного размера входного кадра, используя билинейную интерполяцию. Таким образом, значение каждого пикселя в выходной карте данного шага можно рассматривать как вероятность того, что данный пиксель изображения принадлежит человеческой фигуре.

DeepLabv3+ является составной частью предложенного метода и используется как при обучении, так и при применении предложенной нейросети.

3.3 Генерация обучающих видеопоследовательностей

Для обучения глубокой нейронной сети для матирования видео требуется большой набор данных видео с эталонными картами прозрачности. К сожалению, наборы данных в открытом доступе ограничены в размерах. Однако, существуют большие наборы портретных фотографий людей с эталонными картами прозрачности. В этом разделе предлагается алгоритм для генерации обучающих видеопоследовательностей, используя фотографии переднего плана с эталонными картами прозрачности и фоновые видео. На рис. 2а представлен пример сгенерированной видеопоследовательности.

Для генерации обучающей последовательности длиной в N кадров, сначала выбирается случайное фоновое видео длиной N и два случайных портрета для переднего плана. Процедура искусственного движения, описанная ниже, обрабатывает портреты и выдаёт два видео переднего плана длиной N кадров, а также два соответствующих эталонных оптических потока. В половине случаев выходным видео является композиция обоих передних планов и фонового видео; в остальных случаях же на выход выдаётся одно из видео переднего плана как есть, а второе видео и фоновое видео отбрасываются. Наконец, в 5% случаев на выход выдаётся фоновое видео без каких-либо изменений, что сделано для улучшения качества обработки предложенной нейросетью видео без людей. В качестве последнего шага, к кадрам сгенерированного видео применяется сжатие алгоритмом JPEG со случайным параметром качества от 30% до 80%.

Использование двух видео переднего плана вместо одного улучшает качество работы нейросети матирования на видео, где на переднем плане присутствует несколько человек. На рис. 2а показан пример сгенерированного видео с двумя передними планами друг поверх друга. Одновременно с этим, 50%-я вероятность использования одного переднего плана как есть без композиции и JPEG-сжатие на последнем шаге предотвращает переобучение нейросети матирования на кадры, полученные искусственной композицией. Это происходит благодаря тому, что в половине случаев кадры вообще не являются результатом композиции, а в другой половине артефакты JPEG-сжатия не дают нейросети легко «найти» границы композиции.



Рис. 2. Примеры сгенерированных видеопоследовательностей: (а) итоговая обучающая последовательность; (б) последовательность, сгенерированная с помощью только основного компонента искусственного движения; (с) пример работы компонента сдвига за границу кадра.
Fig. 2. Examples of generated clips: (a) final training clip; (b) clip from the base fake-motion algorithm; (c) example of the fake-motion shift component's effect.

3.4 Искусственное движение

Процедура искусственного движения генерирует случайные карты оптического потока в трёх масштабах. Сначала, генерируется один вектор оптического потока путём выбора двух значений (x и y) из нормального распределения $\mathcal{N}(0,32)$. Этот вектор можно рассматривать как карту оптического потока размером 1×1 . Данная карта представляет собой сильное глобальное движение. Далее, генерируется карта оптического потока размерностью $\lfloor \frac{H}{128} \rfloor \times \lfloor \frac{W}{128} \rfloor$ из распределения $\mathcal{N}(0,16)$. Эта карта представляет собой более детальное движение. Наконец, генерируется карта размерностью $\lfloor \frac{H}{32} \rfloor \times \lfloor \frac{W}{32} \rfloor$ из распределения $\mathcal{N}(0,4)$, представляющая собой наиболее детальную компоненту движения.

Все сгенерированные карты увеличиваются до размера входного изображения ($H \times W$ пикселей) при помощи бикубической интерполяции, суммируются и разделяются N , желаемое число кадров генерируемой последовательности, для получения покадрового оптического потока. На рис. 2б показан пример клипа, сгенерированного описанным алгоритмом.

Для повышения устойчивости разрабатываемой нейросети к видео, где человек временно покидает границы кадра, к результирующему оптическому потоку с вероятностью $\frac{1}{3}$ добавляется компонента, сдвигающая человека на половину кадра и обратно на протяжении видео. Выбор направления для сдвига случаен с условием, что стороны, противоположной направлению сдвига, не касаются непрозрачные пиксели человека переднего плана. Это условие позволяет избежать артефактов повторения граничных пикселей. На рис. 2в показано влияние этой компоненты.

Также, так как в процедуре используется два передних плана, к оптическому потоку добавляется компонента, сдвигающая передний план на половину кадра на первом кадре. Этот шаг уменьшает вероятность полного перекрытия одного переднего плана другим при композиции.

Полученные покадровые карты оптического потока используются для смещения входных изображений и карт прозрачности, чтобы получить обучающую видеопоследовательность с необходимым количеством кадров. Оптический поток также используется в дальнейшем при вычислении функции потерь, описанных в следующем разделе.

3.5 Функции потерь

Для обучения предложенной нейросети используется четырёхкомпонентная функция потерь:

$$L = L_{\alpha} + L_G + L_L + \frac{1}{4} L_{FG}.$$

Первая компонента – попиксельное L_1 -расстояние между картами прозрачности α'_i , предсказанными нейросетью, и эталонными картами прозрачности α_i :

$$L_{\alpha} = \frac{\sum_{i=1}^N \sum_{j=1}^H \sum_{k=1}^W |\alpha'_{ijk} - \alpha_{ijk}|}{N \cdot H \cdot W},$$

где N – число кадров в обучающей видеопоследовательности, H и W – её высота и ширина в пикселях соответственно.

При обучении нейросетей для матирования применяется L_1 -расстояние, а не часто используемое в других задачах L_2 -расстояние, так как обучение с помощью L_2 -расстояния приводит к нейросети, предсказывающей сильно размытые карты прозрачности.

Вторая и третья компоненты отвечают за стабильность результирующих карт во времени. Они представляют из себя попиксельное L_1 -расстояние между картами прозрачности различных кадров из обучающей последовательности, сдвинутых в соответствии с исходным оптическим потоком, созданным в процессе генерации искусственного движения. Вторая

компонента «сравнивает» карту прозрачности первого кадра с картой прозрачности каждого последующего кадра, а третья компонента – карты прозрачности между последовательными кадрами:

$$L_G = \frac{\sum_{i=2}^N \sum_{j=1}^H \sum_{k=1}^W |\alpha'_{ijk} - \alpha'^{\rightarrow i}_{1jk}| V_{1jk}^{\rightarrow i}}{(N-1) \cdot H \cdot W}, L_L = \frac{\sum_{i=2}^N \sum_{j=1}^H \sum_{k=1}^W |\alpha'_{ijk} - \alpha'^{\rightarrow i}_{(i-1)jk}| V_{(i-1)jk}^{\rightarrow i}}{(N-1) \cdot H \cdot W}.$$

Здесь $\alpha'^{\rightarrow i}_l$ обозначает карту прозрачности, предсказанную для l -го кадра, сдвинутую к i -му кадру в соответствии с эталонным оптическим потоком, получаемым в ходе процедуры генерации искусственного движения, описанной в разделе 3.4. $V_l^{\rightarrow i}$ – маска пригодных для сравнения пикселей, равная 1 в точках, где, согласно оптическому потоку, пиксели i -го кадра соответствуют пикселям внутри l -го кадра и 0 в точках, где пиксели i -го кадра соответствуют пикселям за границей l -го кадра. Описанные карты проиллюстрированы на рис. 3.

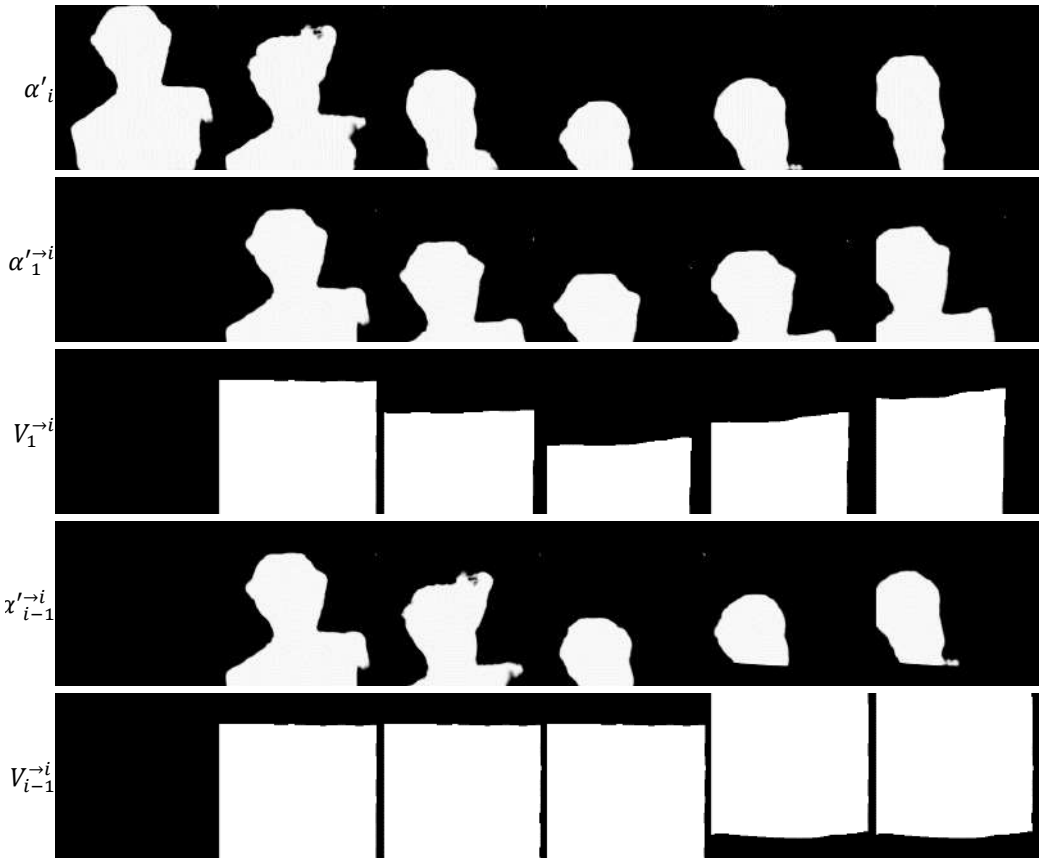


Рис. 3. Иллюстрация карт, использующихся в функции потерь.

Fig. 3. Loss function component map visualization.

Последний компонент, L_{FG} , – это попиксельное L_1 расстояние между предсказанным передним планом FG'_i в цветовом пространстве RGB и эталонным передним планом FG_i , умноженное на предсказанную карту прозрачности:

$$L_{FG} = \frac{\sum_{i=1}^N \sum_{j=1}^H \sum_{k=1}^W \alpha'_{ijk} \sum_{c=1}^3 |FG'_{ijkc} - FG_{ijkc}|}{N \cdot H \cdot W \cdot 3}.$$

3.6 Набор данных

Фоновые видео для процедуры генерации искусственных видео были взяты из набора DAVIS [33]. Конкретно, были выбраны 27 видео, не содержащие людей. Их средняя длина составила 80 кадров.

В качестве передних планов были взяты 40 074 портретных фотографий с эталонными картами прозрачности: 34 425 из набора [5], 2000 из набора [6] и 3649 изображений, размеченных в ходе данной работы. Из этих изображений, 4000 были отложены для валидации, а оставшиеся 36 074 были использованы для обучения.

При обучении нейросети использовались следующие аугментации. Для передних планов:

- случайный поворот кадра на $\pm 15\%$,
- случайная обрезка кадра вплоть до 20% исходного размера,
- случайное отражение относительно вертикальной оси,
- случайное изменение яркости и контраста.

Для фоновых видео:

- случайная обрезка кадра вплоть до 8% исходного размера,
- случайное отражение относительно вертикальной оси,
- случайное изменение яркости и контраста.

Для композиции передних планов и фоновых видео: сжатие JPEG со случайным параметром качества от 30% до 80%. Эти аугментации, вместе со случайными сдвигами, описанными в разделе 3.4, дают процедуру генерации обучающих видеопоследовательностей, схожую с описанной в работах [19-20]. Предлагаемый основной компонент искусственного движения работает совместно с этой процедурой и ещё больше улучшает качество обучающих примеров.

При обучении и валидации, каждые две фотографии переднего плана используются для генерации одной обучающей видеопоследовательности, как описано в разделе 3.3. Следовательно, в одной эпохе обучения содержалось 18 037 обучающих видео и 2000 валидационных видео.

Описанный набор данных использовался только для обучения и валидации нейросети. Итоговая экспериментальная оценка проводилась на отдельном наборе видеопоследовательностей, описанной в разделе 4.

3.7 Разметка эталонных карт прозрачности

На сервисе Яндекс.Толока [34] пользователям предлагалось скорректировать карты прозрачности, полученные при помощи автоматического метода, схожего с описанным в разделе 3.2. Карта признаков на выходе нейросети DeepLabv3+ подавалась на вход алгоритму разреза графа GrabCut [35] для получения пикселей, принадлежащих человеку на переднем плане фотографии. Затем, при помощи морфологического расширения и сужения, границы выделенной области помечались, как пиксели с неизвестным значением прозрачности, и полученная тернарная карта подавалась вместе с исходной фотографией на вход алгоритму матирования изображений [1] для создания карты прозрачности. От пользователей требовалось исправить ошибки в карте прозрачности, используя мазки трёх типов – передний план, задний план и полупрозрачная область. Первые два типа мазков передавались в алгоритм GrabCut в качестве вспомогательных данных, а мазок типа полупрозрачная область переносился на тернарную карту. Изображение пользовательского интерфейса можно увидеть на рис. 4.

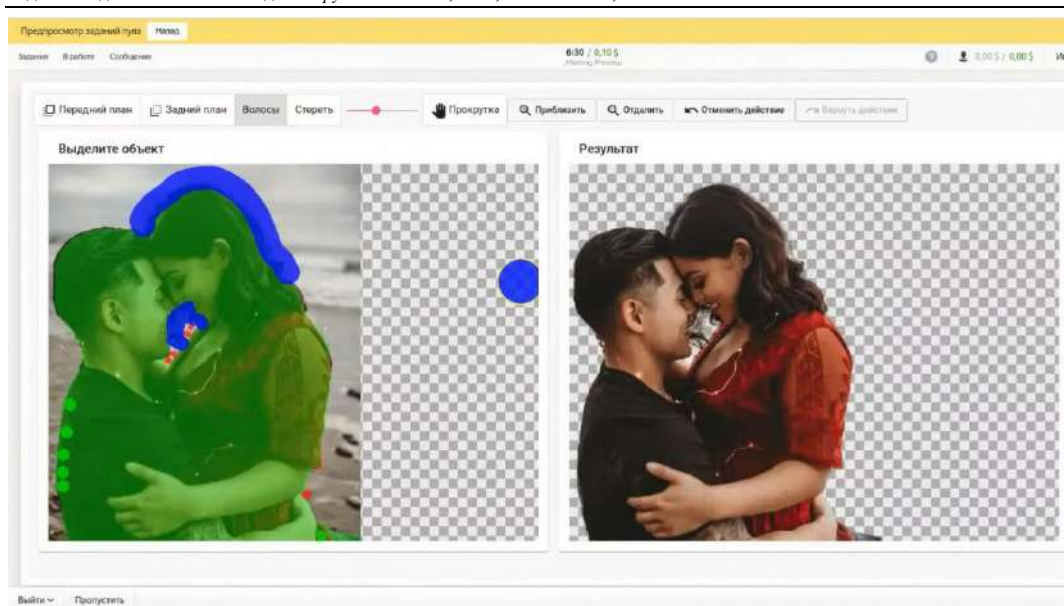


Рис. 4. Пользовательский интерфейс для разметки фотографий на сервисе Яндекс.Толока.
Fig. 4. User interface for photo alpha matte annotation on Yandex.Toloka.

3.8 Обучение нейросети

Нейросетевая модель была обучена на четырёх видеокартах Nvidia Tesla P100. В каждом обучающем наборе было по 8 видеоклипов длиной по шесть кадров разрешения 520×520 пикселей каждый. Для градиентного спуска использовался оптимизатор Adam [36] с темпом обучения, равным 0,001. После каждой эпохи темп обучения уменьшался в 0,7 раз. В процессе разработки, тестовые модели как правило сходилились через семь эпох, или около 9,5 часов обучения.

3.9 Применение нейросети

При применении нейросети используется только выход карты прозрачности, а выход цвета переднего плана игнорируется. Это сделано в связи с тем, что на нём часто содержатся цветовые артефакты. Тем не менее, наличие этого выход и соответствующей компоненты функция потерь улучшило результат обучения.

Применение обученной нейросети напрямую на длинных видеопоследовательностях даёт высокую стабильность во времени, однако ведёт к появлению артефактов: LSTM-слои «запоминают» положение людей на переднем плане и не могут достаточно быстро адаптироваться к движению человека. Для решения этой проблемы используется следующая схема применения нейросети. Обработка каждого кадра входного видео начинается со сброса состояния LSTM-слоёв. Затем нейросети на вход подаются 3 предшествующих кадра видео, после чего нейросеть обрабатывает текущий кадр видео.

Подобная схема применения с одной стороны не ведёт к значительной потере стабильности во времени результирующих карт прозрачности, а с другой стороны позволяет сети очень быстро адаптироваться к резким движениям человека и предотвращает накопление ошибки.

Во время применения нейросети на вход A_i передаются не исходная карта вероятности, полученная из алгоритма определения положения людей, а сглаженная карта вероятности, полученная алгоритмом, описанным в разделе 3.9. Это приводит к ещё большему повышению

стабильности во времени карт прозрачности на выходе нейросети, что подтверждается результатами проведённых объективных и субъективных оценок (раздел 4).

Передача сглаженных карт вероятности на вход нейросети во время обучения не дала видимого улучшения качества работы. Возможно, это связано с малой длиной обучающих примеров – сглаживание «не успевает» значительно изменить карты вероятности.

Скорость работы предложенного метода составляет около 1,7 кадров разрешения 1280×720 в секунду на машине с одной видеокартой Nvidia TITAN Xp и процессором Intel Xeon E5-2683 v3 с тактовой частотой 2 ГГц.

3.10 Алгоритм сглаживания карт вероятности

Покадровое применение алгоритма определения положения людей на видео даёт в результате сильно «дрожащую» последовательность масок. Для улучшения работы нейросети матирования, предлагается следующий алгоритм сглаживания во времени результатов работы сети определения положения людей при помощи векторов движения.

Пары последовательных кадров входной видеопоследовательности $\{I_i\}_{i=1}^N$ подаются в блочный алгоритм поиска векторов движения [37] в прямом и обратном порядке для получения прямых и обратных векторов движения. Векторы движения представляются в виде карты оптического потока: для каждого пикселя второго входного кадра известен вектор смещения в соответствующий ему пиксель из первого входного кадра, выраженный разницей координат по осям x и y . Таким образом, получаются две последовательности карт: $\{F_i\}_{i=2}^N$ и $\{B_i\}_{i=1}^{N-1}$ прямого и обратного оптического потока, соответственно.

Далее, вычисляется согласованность прямого и обратного потока способом, похожим на предложенный в работе [38].

1. Карты обратного оптического потока сдвигаются в соответствии с картами прямого оптического потока. Обозначим получившиеся карты как $\{B_{i-1}^{\rightarrow i}\}_{i=2}^N$.
2. Карты сдвинутого обратного оптического потока вычитаются попиксельно из карт прямого оптического потока: $D_i = F_i - B_{i-1}^{\rightarrow i}$.
3. В каждом пикселе полученных карт вычисляется L_2 -норма: $L_{ijk} = \sqrt{D_{ijkx}^2 + D_{ijk y}^2}$,
где $i = \overline{2, N}$, $j = \overline{1, H}$, $k = \overline{1, W}$, H – высота кадра в пикселях, W – ширина кадра в пикселях, x и y – компоненты векторов оптического потока.
4. Для получения карт согласованности от значений полученных карт попиксельно вычисляется экспонента: $C_i = e^{\left(\frac{-L_i}{100}\right)}$.

Карта согласованности принимает значения, близкие к единице, при близости прямых и соответствующих им обратных векторов движения (что говорит о правильности вектора движения в данном пикселе) и значения, близкие к нулю, в противоположном случае.

Сглаживание происходит по следующей процедуре. На входе – карты вероятности $\{p_i\}_{i=1}^N$, полученные из DeepLabv3+, как описано в разделе 3.2. Для карт вероятности p_i попиксельно вычисляются значения уверенности: $s_i = 4\left(p_i - \frac{1}{2}\right)$.

Финальные значения сглаженных карт вероятности $\{A_i\}_{i=1}^N$ вычисляются следующим образом:

$$A_1 = p_1,$$

$$A_i = s_i \cdot p_i + (1 - s_i)(C_i \cdot A_{i-1}^{\rightarrow i}),$$

где $i = \overline{2, N}$, а $A_{i-1}^{\rightarrow i}$ – карты вероятности A_{i-1} , сдвинутые в соответствии с картами оптического потока F_i .

Интуитивно, в пикселях, где вероятность присутствия человека близка к нулю или единице (нейросеть «уверена» в своём ответе), в качестве значения прозрачности берётся эта вероятность, а в пикселях, где вероятность близка к $\frac{1}{2}$ (нейросеть «не уверена»), в качестве значения прозрачности берётся скомпенсированное по векторам движения значение прозрачности предыдущего кадра, отмасштабированное в соответствии с согласованностью векторов.

4. Экспериментальная оценка

В процессе разработки различные модификации алгоритма оценивались как субъективно на выборке тестовых видео, так и с помощью объективных метрик. Сравнение проходило между предложенным нейросетевым методом и следующими методами.

- Deep Image Matting [1]: своя реализация, обучена на наборе данных Deep Image Matting.
- FBA Matting [2]: официальная реализация и модель, обученная на наборе данных Deep Image Matting.
- COSNet [39]: официальная реализация и модель, обученная на наборе данных DAVIS16.
- MMNet [11]: официальная реализация, обученная на наборе данных AISegment [5].
- Semantic Human Matting [9]: неофициальная реализация и модель, обученная на закрытом наборе данных разработчика.
- Late Fusion Matting [12]: официальная реализация и модель, обученная на наборе данных Deep Image Matting и закрытом наборе данных разработчика.
- CRGNN [40]: официальная реализация и модель, обученная на наборе данных Deep Image Matting и наборе данных разработчика.

Deep Image Matting и FBA Matting – это методы матирования изображений, занимающие первые места на эталонном сравнении <https://videomatting.com> [41], COSNet – это метод сегментации видео без учителя, занимающий высокое место на эталонном сравнении DAVIS16 [33], CRGNN – метод матирования видео, и MMNet, Semantic Human Matting и Late Fusion Matting – это методы семантического матирования фотографий с людьми. В сравнении также участвовали карты вероятности A_i , сложенные предложенным в разделе 3.9 алгоритмом, для чего они интерпретировались как итоговые карты прозрачности.

Тернарные карты, необходимые для работы алгоритмов [1-2, 40], генерировались при помощи процедуры, аналогичной описанной в разделе 3.2: к местоположению людей в кадре, полученному при помощи нейросети DeepLabv3+, применялось морфологическое расширение и сужение для пометки граничных областей как пикселей с неизвестным значением прозрачности.

Пример результатов предложенных и участвовавших в сравнении методов приведён на рис. 5.

4.1 Субъективное сравнение

Для субъективной оценки алгоритма был использован сервис проведения субъективных исследований <https://subjectify.us>. Пользователям показываются пары видеопоследовательностей, полученных путём композиции тестовых видео на статический фон, используя карты прозрачности, полученные сравниваемыми алгоритмами. Задача пользователей формулировалась в терминах сравнения алгоритмов замены фона на видео, и от них требовалось выбрать наилучший, по их мнению, вариант из двух. Интерфейс субъективного исследования со стороны пользователя приведён на рис. 6.



Рис. 5. Результаты участвовавших в сравнении методов на двух тестовых видео.
Fig. 5. Results of the evaluated methods on two of the test clips.

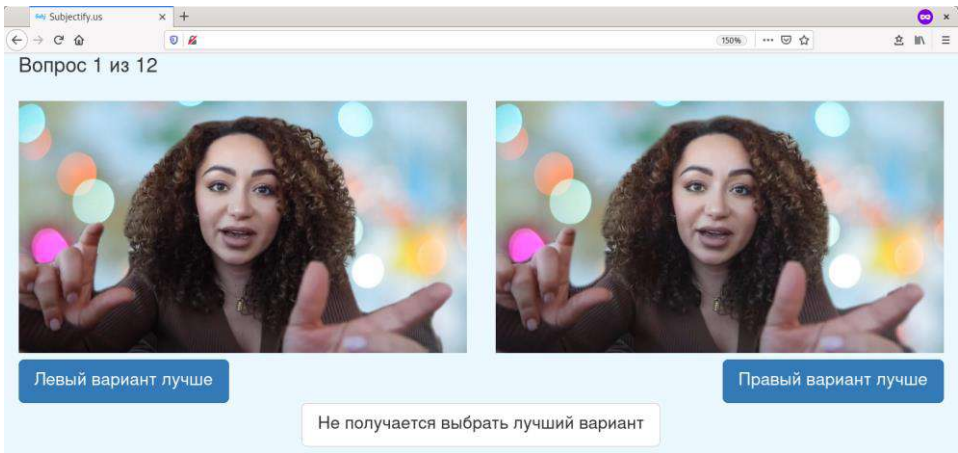


Рис. 6. Результаты субъективного исследования по всем тестовым видеопоследовательностям.
Чёрные метки погрешности показывают 95%-е доверительные интервалы.
Fig. 6. Subjective evaluation results on all test sequences. Error bars show 95% confidence intervals.

Для исследования была составлена выборка из 2 видеопоследовательностей (city и snow) с эталонного теста <https://videomattng.com>, попадающих в целевые входные данные для разрабатываемого алгоритма (видео с людьми, но без крупных полупрозрачных областей), 2 видео, снятые в процессе работы, и 55 видео с людьми средней длины в 371 кадр [42], вырезанных из различных видео любительской съёмки, найденных на видеохостинге YouTube.

В процессе исследования каждому пользователю необходимо было оценить 12 пар видеопоследовательностей. Среди них присутствовали две контрольных видеопоследовательности: результат пок кадрового применения алгоритма матирования изображения [1] и последовательность, полученная при помощи композиции с эталонной картой прозрачности. Если пользователь неправильно оценивал какую-либо из контрольных

пар, его ответы не учитывались в результатах. Для контрольных пар были использованы видеопоследовательности *city* и *snow*.

Всего таким образом было собрано 32 272 оценок пар видеопоследовательностей от пользователей, правильно оценивших контрольные пары. Парные оценки, данные пользователями, использовались для настройки параметров модели Брэдли-Терри [43]. Эта модель максимизирует субъективные оценки при условии данных парных оценок. Чем выше субъективная оценка, данная алгоритму настроенной моделью, тем выше вероятность предпочтения результата этого алгоритма в парном сравнении.

Общие результаты по всем видео приведены на рис. 7. Методы матирования изображений [9, 2, 1, 12] показали неплохой результат, хотя нестабильность во времени в результирующих картах прозрачности уменьшает предпочтение зрителей этим методам. Результаты COSNet [39] достаточно сильно мерцали, несмотря на то что это алгоритм матирования видео. MMNet [11] показал худший результат, скорее всего, потому что он не предназначен для больших изображений.

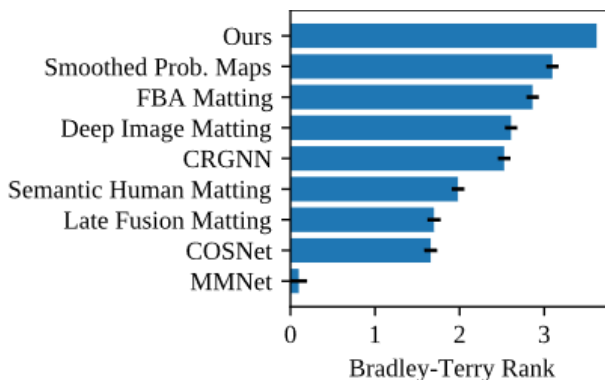


Рис. 6. Результаты субъективного исследования по всем тестовым видеопоследовательностям.

Чёрные метки погрешности показывают 95%-е доверительные интервалы.

Fig. 7. Subjective evaluation results on all test sequences. Error bars show 95% confidence intervals.

Сглаженные карты вероятности, интерпретированные как карты прозрачности, показали неожиданно хороший результат. Основной недостаток данного подхода – плохая обработка границ и «след», остающийся за движущимися объектами.

Предложенный в данной работе нейросетевой подход показал наилучший результат, и в большинстве сравнений был предпочтён другим алгоритмам.

4.2 Объективная оценка

Для объективной оценки алгоритма использовались метрики эталонного теста алгоритмов матирования видеопоследовательностей <https://videomattng.com> [41]. Они определяются следующим образом:

$$SSDA = \frac{\sum_{i=1}^N \sqrt{\sum_p^P (\alpha'_{i,p} - \alpha_{i,p})}}{N \cdot P},$$

$$dtSSD = \frac{\sum_{i=2}^N \sqrt{\sum_p^P ((\alpha'_{i,p} - \alpha'_{i-1,p}) - (\alpha_{i,p} - \alpha_{i-1,p}))^2}}{(N-1) \cdot P},$$

$$MESSDdt = \frac{\sum_{i=2}^N \sum_p^P |(\alpha'_{i-1,p} - \alpha_{i-1,p})^2 - (\alpha'^{i-1}_{i,p} - \alpha^{i-1}_{i,p})^2|}{(N-1) \cdot P},$$

- где:
- N – число кадров в видео,
 - $P = W \times H$ – число пикселей в одном кадре видео,
 - α_i – эталонная карта прозрачности i -го кадра,
 - α'_i – карта прозрачности i -го кадра на выходе оцениваемого алгоритма матирования,
 - $\alpha_i^{\rightarrow i-1}$ – карта прозрачности i -го кадра, сдвинутая к $i - 1$ -му кадру в соответствии с оптическим потоком.

Все три метрики принимают значения, большие либо равные нулю, и равны нулю в случае полного совпадения предсказанной и эталонной карты прозрачности. Чем ниже значение каждой метрики, тем ближе предсказанная карта прозрачности к эталонной. Выбор данных метрик для оценки мотивирован их использованием в эталонном сравнении и нескольких предыдущих работах, предлагающих методы матирования видео [40, 14].

Оценка проводилась на тех же двух видеопоследовательностях из эталонного теста <https://videomatting.com>, что использовались в субъективном сравнении: *city* и *snow*. Результаты представлены в табл. 1. Предложенный нейросетевой метод даёт либо наилучший результат, либо входит в четвёрку лучших методов вместе с [12, 2, 40]. Различие с субъективной оценкой скорее всего объясняется недостаточной чувствительностью существующих объективных методов оценки к нестабильности результатов во времени.

Объективная оценка также была проведена на первых 50 кадрах видео тестовой части набора VideoMatte240K [22] (всего пять видео). Так как видео из этого набора не содержат «фродного» фона, для оценки они были наложены на пять случайных фоновых видео без людей. Результаты представлены в табл. 2. Предложенный нейросетевой метод показывает второй или третий лучший результат после [12] и [2].

Табл. 1. Результаты объективной оценки на двух видеопоследовательностях <https://videomatting.com>. Лучший результат выделен жирным, второй подчёркнут, третий показан курсивом.
Table 1. Objective evaluation results on <http://videomatting.com> clips.
The best result is shown in bold, the second-best is underlined and the third-best is shown in italics.

Метод	SSDA	dtSSD	MESSDdt	SSDA	dtSSD	MESSDdt
	<i>city</i>			<i>snow</i>		
Предложенный нейросетевой метод	69,651	15,314	0,695	56,338	30,240	0,662
Сглаженные карты вероятности	91,577	<u>17,609</u>	<u>1,291</u>	65,772	35,133	1,264
FBA Matting	<u>57,700</u>	30,825	1,613	<u>27,113</u>	<u>20,881</u>	<u>0,423</u>
Deep Image Matting	97,506	47,107	3,258	59,648	41,463	2,128
CRGNN	76,456	35,591	2,354	34,735	27,183	1,244
Semantic Human Matting	108,393	53,086	5,696	71,844	43,689	2,595
Late Fusion Matting	44,766	31,621	4,152	24,602	19,484	0,341
COSNet	271,878	62,798	22,387	156,617	58,536	9,424
MMNet	154,656	62,439	13,580	347,065	143,696	58,429

5. Заключение

В данной работе предложен нейросетевой метод семантического матирования видео с людьми на переднем плане, не требующий дополнительных входных данных. Для генерации обучающих видеопоследовательностей для нейросети, используя портретные фотографии людей с эталонными картами прозрачности, предложена процедура создания искусственного движения. Также предложен алгоритм сглаживания карт вероятности, получаемых на выходе из алгоритмов сегментации изображений.

Табл. 2. Результаты объективной оценки на пяти тестовых видеопоследовательностях VideoMatte240K. Лучший результат выделен жирным, второй подчеркнут, третий показан курсивом.
Table 2. Objective evaluation results on five VideoMatte240K test clips.
The best result is shown in bold, the second-best is underlined and the third-best is shown in italics.

Метод	SSDA	dtSSD	MESSDdt
Предложенный нейросетевой метод	<i>84,710</i>	<i>46,792</i>	<u>2,080</u>
Сглаженные карты вероятности	117,253	53,836	4,452
FBA Matting	<u>62,679</u>	<u>40,996</u>	<i>2,901</i>
Deep Image Matting	165,094	100,595	15,039
CRGNN	140,095	84,434	13,900
Semantic Human Matting	166,325	113,648	22,300
Late Fusion Matting	29,146	25,459	0,845
COSNet	226,256	74,765	17,506
MMNet	445,550	156,778	75,171

Для оценки качества разработанного метода проведено субъективное сравнение с 7 существующими методами матирования и сегментации на 59 видеопоследовательностях, в рамках которого собрано 32 272 попарных оценок. Субъективное сравнение показало превосходство разработанного метода над аналогами. Дополнительно, было проведено объективное сравнение, в котором предложенный подход также оказался в числе лучших.

Также было проведено дополнительное субъективное сравнение 9 вариантов предложенного метода для анализа его избыточности, в рамках которого собрано 22 360 попарных оценок. Это сравнение показало положительный вклад компонентов предложенного метода, в том числе сильное положительное влияние на результат предложенных процедуры генерации искусственного движения и алгоритма сглаживания карт вероятности.

Список литературы / References

- [1]. Xu N., Price B., Cohen S., Huang T. Deep Image Matting // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
- [2]. Forte M., Pitié F. F. B, Alpha Matting // arXiv preprint arXiv:2003.07711. 2020.
- [3]. Goel A., Kumar M., Sudheendra P., Team V. I. IamAlpha: Instant and Adaptive Mobile Network for Alpha Matting. 2021.
- [4]. Shi X., Chen Z., Wang H., Yeung D.-Y., Wong W.-k., Woo W.-c. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting // Advances in Neural Information Processing Systems 28. Curran Associates, Inc., 2015. cc. 802–810.
- [5]. AISegment dataset [электронный ресурс]. 2019. URL: https://github.com/aisegmentcn/matting_human_datasets/tree/1829b5f722024d29b780993f06b45ea3f47ba777 (дата обращения: 12.04.2020).
- [6]. Shen X., Tao X., Gao H., Zhou C., Jia J. Deep automatic portrait matting // European Conference on Computer Vision. 2016. cc. 92–107.
- [7]. Wang J., Cohen M. F. Image and video matting: a survey. Now Publishers Inc, 2008.
- [8]. Li X., Li J., Lu H. A survey on natural image matting with closed-form solutions // IEEE Access. IEEE, 2019. т. 7. cc. 136658–136675.
- [9]. Chen Q., Ge T., Xu Y., Zhang Z., Yang X., Gai K. Semantic human matting // Proceedings of the 26th ACM international conference on Multimedia. 2018. cc. 618–626.
- [10]. Lu H., Dai Y., Shen C., Xu S. Indices Matter: Learning to Index for Deep Image Matting // The IEEE International Conference on Computer Vision (ICCV). 2019.
- [11]. Seo S., Choi S., Kersner M., Shin B., Yoon H., Byun H., Ha S. Towards Real-Time Automatic Portrait Matting on Mobile Devices // arXiv preprint arXiv:1904.03816. 2019.
- [12]. Zhang Y., Gong L., Fan L., Ren P., Huang Q., Bao H., Xu W. A Late Fusion CNN for Digital Matting // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.

- [13]. Liu J., Yao Y., Hou W., Cui M., Xie X., Zhang C., Hua X.-S. Boosting Semantic Human Matting With Coarse Annotations // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020.
- [14]. Backes M. H., Oliveira M. M. A PatchMatch-based Approach for Matte Propagation in Videos // Computer Graphics Forum. 2019. т. 38, № 7. cc. 651–662.
- [15]. Barnes C., Shechtman E., Goldman D. B., Finkelstein A. The Generalized PatchMatch Correspondence Algorithm // Computer Vision – ECCV 2010. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. cc. 29–43.
- [16]. Zou D., Chen X., Cao G., Wang X. Unsupervised video matting via sparse and low-rank representation // IEEE transactions on pattern analysis and machine intelligence. IEEE, 2019.
- [17]. Shen X., Wang R., Zhao H., Jia J. Automatic real-time background cut for portrait videos // arXiv preprint arXiv:1704.08812. 2017.
- [18]. Sengupta S., Jayaram V., Curless B., Seitz S. M., Kemelmacher-Shlizerman I. Background Matting: The World Is Your Green Screen // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020.
- [19]. Oh S. W., Lee J.-Y., Sunkavalli K., Kim S. J. Fast Video Object Segmentation by Reference-Guided Mask Propagation // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018.
- [20]. Oh S. W., Lee J.-Y., Xu N., Kim S. J. Video Object Segmentation Using Space-Time Memory Networks // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019. т. 0, № . cc. 9225–9234. DOI: 10.1109/ICCV.2019.00932.
- [21]. VideoMatte240K and PhotoMatte85 Datasets [электронный ресурс]. 2022. URL: <https://grail.cs.washington.edu/projects/background-matting-v2/#/datasets>.
- [22]. Lin S., Ryabtsev A., Sengupta S., Curless B. L., Seitz S. M., Kemelmacher-Shlizerman I. Real-Time High-Resolution Background Matting // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021. cc. 8762–8771.
- [23]. Cai S., Zhang X., Fan H., Huang H., Liu J., Liu J., Liu J., Wang J., Sun J. Disentangled Image Matting // The IEEE International Conference on Computer Vision (ICCV). 2019.
- [24]. Lutz S., Amphianitis K., Smolic A. Alphagan: Generative adversarial networks for natural image matting // arXiv preprint arXiv:1807.10088. 2018.
- [25]. Tang J., Aksoy Y., Oztireli C., Gross M., Aydin T. O. Learning-based sampling for natural image matting // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019. cc. 3055–3063.
- [26]. Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation. 1997. т. 9, № 8. cc. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- [27]. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018.
- [28]. TorchVision [электронный ресурс]. 2019. URL: <https://github.com/pytorch/vision/tree/a263704079d9d35db8b0966a65ec628d28998bce/torchvision>.
- [29]. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention is all you need // Advances in neural information processing systems. 2017. cc. 5998–6008.
- [30]. Chen L.-C., Zhu Y., Papandreou G., Schroff F., Adam H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation // Proceedings of the European Conference on Computer Vision (ECCV). 2018.
- [31]. Chollet F. Xception: Deep Learning With Depthwise Separable Convolutions // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
- [32]. Everingham M., Eslami S. A., Van Gool L., Williams C. K., Winn J., Zisserman A. The pascal visual object classes challenge: A retrospective // International journal of computer vision. Springer, 2015. т. 111, № 1. cc. 98–136.
- [33]. Perazzi F., Pont-Tuset J., McWilliams B., Van Gool L., Gross M., Sorkine-Hornung A. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation // Computer Vision and Pattern Recognition. 2016.
- [34]. Сервис Яндекс.Толока [электронный ресурс]. 2020. URL: <https://toloka.yandex.ru/>.
- [35]. Rother C., Kolmogorov V., Blake A. "GrabCut" interactive foreground extraction using iterated graph cuts // ACM transactions on graphics (TOG). ACM New York, NY, USA, 2004. т. 23, № 3. cc. 309-314.
- [36]. Kingma D. P., Ba J. Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. 2014.

- [37]. Simonyan K., Grishin S., Vatin D., Popov D. Video super-resolution using motion compensation and classification-aided fusion // *Proceedings of the 24th Spring Conference on Computer Graphics*. 2008. cc. 143–148. DOI: 10.1145/1921264.1921294.
- [38]. Hur J., Roth S. MirrorFlow: Exploiting Symmetries in Joint Optical Flow and Occlusion Estimation // *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.
- [39]. Lu X., Wang W., Ma C., Shen J., Shao L., Porikli F. See More, Know More: Unsupervised Video Object Segmentation With Co-Attention Siamese Networks // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
- [40]. Wang T., Liu S., Tian Y., Li K., Yang M.-H. Video Matting via Consistency-Regularized Graph Neural Networks // *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021. cc. 4902–4911.
- [41]. Erofeev M., Gitman Y., Vatin D., Fedorov A., Wang J. Perceptually Motivated Benchmark for Video Matting // *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2015. c. 99. DOI: 10.5244/C.29.99.
- [42]. Videos Used for Subjective Evaluation [электронный ресурс]. 2022. URL: <https://drive.google.com/drive/folders/1RL9zbn7MPHpyFT4ahW00fjmKINKdnyy7?usp=sharing>.
- [43]. Bradley R. A., Terry M. E. Rank analysis of incomplete block designs: I. The method of paired comparisons // *Biometrika*. JSTOR, 1952. т. 39, № 3/4. cc. 324–345.

Информация об авторах / Information about authors

Иван Андреевич МОЛОДЕЦКИХ получил степень магистра по прикладной математике и информатике в Московском государственном университете имени М. В. Ломоносова в 2020 году. В настоящее время является аспирантом в лаборатории компьютерной графики и мультимедиа. В область его научных интересов входят суперразрешение, семантическое матирование видео и машинное обучение. Иван курировал разработку бенчмарка методов суперразрешения для улучшения качества видео и был одним из организаторов соревнования по оценке качества суперразрешения видео на ECCV-AIM 2024.

Ivan Andreevich MOLODETSKIKH received his master degree in computer science from the Moscow State University in 2020. He is currently a postgraduate student at the MSU Graphics & Media Lab. His research interests include super-resolution, semantic video matting and machine learning. Ivan supervised the development of MSU video upscalers and SR+codec benchmarks and was one of the organizers of the ECCV-AIM 2024 video super-resolution quality assessment challenge.

Михаил Викторович ЕРОФЕЕВ получил степень специалиста по прикладной математике и информатике в Московском государственном университете имени М. В. Ломоносова в 2013 году. В 2017 защитил кандидатскую диссертацию по теме «Преобразование видеопоследовательностей, содержащих объекты с полупрозрачными границами, в стереоскопический формат». В настоящее время является научным сотрудником лаборатории компьютерной графики и мультимедиа факультета ВМК МГУ. В область его научных интересов входят матирование изображений и видео и машинное обучение. Михаил является активным участником проекта по объективному сравнению алгоритмов.

Mikhail Viktorovich EROFEEV – Cand. Sci. (Phys.-Math.). He graduated from the Computer Science department of Lomonosov Moscow State University, and currently he is a researcher at the MSU Graphics & Media Lab. His research interests include video and image matting, and machine learning. Mikhail is one of the major contributors to the video matting methods benchmark.

Андрей Викторович МОСКАЛЕНКО получил степень магистра по прикладной математике и информатике в Московском государственном университете имени М. В. Ломоносова в 2023 году. В настоящее время является студентом аспирантуры в лаборатории компьютерной графики и мультимедиа. В область его научных интересов входят моделирование визуального внимания, интерактивная сегментация, заполнение областей изображений и

оценка качества видео. Андрей является одним из ключевых разработчиков платформы анализа субъективного качества медиаконтента.

Andrey Viktorovich MOSKALENKO received his master degree in computer science from the Moscow State University in 2023. He is currently a postgraduate student at the MSU Graphics & Media Lab. His research interests involve visual attention modeling, interactive segmentation, image inpainting and video quality assessment. Andrey is one of the core contributors to the subjective media quality assessment platform.

Дмитрий Сергеевич ВАТОЛИН закончил ВМК МГУ в 1996 году, кандидат физико-математических наук, заведующий лабораторией компьютерной графики ВМК МГУ, исследователь Центра доверенного искусственного интеллекта ИСП РАН. Читает курсы по компьютерной графике и методам сжатия и обработки видео с 1997 года. Создатель популярных сайтов, посвященных алгоритмам обработки и сжатия видео. Специализируется на исследованиях в области алгоритмов сжатия видео, современных методах измерения качества и обработке цифрового видео. Руководил проектами с компаниями Intel, Cisco, Real Networks, Samsung, Huawei, Broadcom и некоторыми другими. С 2008 года занимается измерением и исправлением артефактов стерео, в том числе организовал проект измерения качества стереофильмов.

Dmitriy Sergeevich VATOLIN – Cand. Sci. (Phys.-Math.) from Moscow State University, head of the MSU Graphics & Media Lab and MSU AI Institute Video Analysis Lab, and a researcher at the Research Centre for Trusted AI of ISP RAS. His research interests include compression methods, video processing, 3D video techniques (depth from motion, focus and other cues, video matting, background restoration, high-quality stereo generation), as well as video quality assessment and robustness of modern video quality metrics. He is a key cofounder of the 3D video quality measurement project, his most known project is the annual MSU Video Codecs Comparison that includes up to 25 modern codecs compared subjectively and objectively in several nominations with detailed 20000+ charts.

DOI: 10.15514/ISPRAS-2025-37(3)-7



Исследование специальных наборов модулей системы остаточных классов

- ¹ В.В. Луценко, ORCID: 0000-0003-4648-8286 <vvlutcenko@ncfu.ru>
² М.Д. Кравцов, ORCID: 0009-0001-5098-5034 <micha_kravtsov_olymp@mail.ru>
¹ Д.Е. Горлачев, ORCID: 0009-0002-9703-1876 <dimagorlacev12@gmail.com>
¹ Н.М. Мирный, ORCID: 0009-0005-2476-494X <mirnyjn@gmail.com>

¹ Северо-Кавказский федеральный университет,
Россия, 355017, г. Ставрополь, ул. Пушкина, д. 1.

² Региональный центр «Сириус 26»,
Россия, 355017, г. Ставрополь, ул. Артема, д. 6.

Аннотация. В статье рассматриваются современные подходы к повышению производительности вычислительных систем на основе системы остаточных классов. Цель исследования – анализ специальных наборов модулей системы остаточных классов, которые позволяют проводить ключевые вычислительные операции, такие как сложение, обратное преобразование и определение знака, с минимальными затратами. Экспериментальные результаты показали, что базис $\{2^n - 1, 2^n, 2^n + 1\}$ оказался наиболее эффективным среди трех модульных наборов. Данный базис является перспективным для использования в высокопроизводительных вычислительных системах.

Ключевые слова: система остаточных классов; специальные наборы модулей; Китайская теорема об остатках; функция ядра Акушского.

Для цитирования: Луценко В.В., Кравцов М.Д., Горлачев Д.Е., Мирный Н.М. Исследование специальных наборов модулей системы остаточных классов. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 107–120. DOI: 10.15514/ISPRAS-2025-37(3)-7.

Благодарности: Работа выполнена при поддержке Российского научного фонда (проект № 22-71-10046).

Research of Special Sets of Moduli of the Residue Number System

¹ V.V. Lutsenko, ORCID: 0000-0003-4648-8286 <vvlutcenko@ncfu.ru>

² M.D. Kravtsov, ORCID: 0009-0001-5098-5034 <micha_kravtsov_olymp@mail.ru>

¹ D.E. Gorlancev, ORCID: 0009-0002-9703-1876 <dimagorlancev12@gmail.com >

¹ N.M. Mirny, ORCID: 0009-0005-2476-494X <mirnyjn@gmail.com >

¹ North-Caucasus Federal University, Stavropol,

1, Pushkin st., Stavropol, 355017, Russia

² Sirius 26 Regional Centre,

6, Artema st., Stavropol, 355017, Russia.

Abstract. The article examines modern approaches to enhancing the performance of computing systems based on the residue number system. The objective of the study is to analyze specific sets of residue number system moduli that allow for key computational operations, such as addition, reverse conversion, and sign determination, to be performed with minimal cost. Experimental results showed that the $\{2^n - 1, 2^n, 2^n + 1\}$ basis was the most efficient among the three moduli sets. This basis is promising for use in high-performance computing systems.

Keywords: residue number system; special sets of moduli; Chinese remainder theorem; Akushsky core functions; non-modular operations.

For citation: Lutsenko V.V., Kravtsov M.D., Gorlancev D.E., Mirny N.M. Research of special sets of moduli of the residue number system. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 107-120 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-7.

Acknowledgements. The research was supported by the Russian Science Foundation Grant No. 22-71-10046, <https://rscf.ru/en/project/22-71-10046/>.

1. Введение

Развитие алгоритмической базы вычислительных систем требует разработки и внедрения инновационных подходов к организации и выполнению вычислительных задач. Наряду с такими современными технологиями, как квантовые вычисления [1] и ИИ-ускорители [2], параллельная обработка данных остается одной из ключевых стратегий повышения производительности. В данной области основное внимание уделяется модулярным вычислительным структурам, основанным на системе остаточных чисел (СОК). В отличие от традиционных методов, СОК предлагает особые преимущества благодаря своей непозиционной природе и параллельным свойствам, вытекающим из теории чисел и абстрактной алгебры.

Известно, что СОК имеет ряд преимуществ перед обычными позиционными системами счисления при разработке и реализации высокопроизводительных вычислительных приложений, устройств и систем [3]. С момента своего появления в середине 1950-х годов и по сей день арифметика СОК привлекает постоянное внимание исследователей в следующих областях:

- 1) компьютерные технологии [3],
- 2) теоретико-числовые методы [4, 5],
- 3) блокчейн [6],
- 4) гомоморфное шифрование [7],
- 5) цифровая обработка сигналов и изображений [8, 9],
- 6) системы связи [10],
- 7) высоконадежные облачные среды [11, 12],
- 8) нейронные сети [13, 14].

Основным преимуществом СОК является возможность разложения чисел большой длины на набор остатков меньшей длины, которые обрабатываются параллельно в независимых модульных каналах. Параллелизм, присущий СОК, позволяет избежать переносов, получаемых при сложении, вычитании и умножении. Данные операции также называют модульными.

Тем не менее, существуют определенные трудности, связанные с применением СОК, которые требуют дополнительного внимания для расширения ее применимости в различных областях, например, в процессорах общего назначения [10]. Наиболее ресурсозатратными операциями в СОК являются операции обратного преобразования, определение знака, сравнения и деления чисел. Данные операции также называют немодульными.

Первоочередная задача при задании СОК – определить набор модулей системы или же, другими словами, базис системы. За более чем полувековой период существования СОК было предложено множество различных наборов модулей [15]. Для классических систем эффективным набор модулей считается набор в форме степеней двойки, который может значительно снизить сложность операций, описанных ранее. Правильный выбор модулей способствует снижению вычислительных затрат и более эффективной реализации алгоритмов на практике.

В данной статье рассмотрены наиболее известные наборы модулей СОК. В частности, уделяется внимание таким критериям, как время выполнения сложения в СОК, а также время обратного преобразования и определения знака числа. Исследование направлено на нахождение оптимальных наборов модулей для различных приложений, что открывает возможности для повышения эффективности современных вычислительных систем.

Статья имеет следующую структуру. В разделе 2 рассматриваются алгоритмы основных модульных и немодульных операций системы остаточных классов. В разделе 3 представлены наборы модулей специального вида. Затем, в разделе 4 проведен анализ их эффективности с использованием алгоритмов из раздела 2. В заключении суммируются полученные результаты.

2. Система остаточных классов

2.1 Основы арифметики в системе остаточных классов

Базисом системы остаточных классов называется множество модулей $\{p_1, p_2, \dots, p_n\}$, где каждый модуль $p_i \geq 2$ ($i = 1, 2, \dots, n$) и $\text{НОД}(p_i, p_j) = 1$. По умолчанию модули упорядочены по возрастанию, то есть $p_1 < p_2 < \dots < p_n$.

Произведение модулей $P = \prod_{i=1}^n p_i$ называется динамическим диапазоном системы остаточных классов.

Целое число $X \in [0, P)$ представимо в виде n -мерного вектора, составленного из наименьшего неотрицательного остатка от деления соответствующего числа на p_i :

$$X = (x_1, x_2, \dots, x_n),$$

где $x_i = X \bmod p_i$, что также обозначается как $x_i = |X|_{p_i}$.

Для введения отрицательных чисел необходимо разделить диапазон на два интервала. Пусть P – динамически диапазон системы. Тогда при введении отрицательных чисел число X удовлетворяет следующему соотношению:

$$\begin{cases} \frac{-P-1}{2} \leq X \leq \frac{P-1}{2}, \text{ если } P \text{ нечетное число,} \\ \frac{-P}{2} \leq X \leq \frac{P}{2} - 1, \text{ если } P \text{ четное число.} \end{cases} \quad (1)$$

Рассмотрим СОК с базисом $\{3, 4\}$. В этом базисе можно взаимно-однозначно представить числа из полуинтервала $[-6; 6)$, так как $P = 3 \cdot 4 = 12$. Если $X = (x_1, x_2, \dots, x_n)$, то отрицательное число $-X = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$, где $\bar{x}_i$ является дополнением x_i до модуля p_i . Для СОК $\{3, 4\}$ и числа $X = (1, 1)$ получим $-X = (3 - 1, 4 - 1) = (2, 3)$.

В табл. 1 представлены соответствия чисел из позиционной системы счисления и системы остаточных классов.

Табл. 1. Представление чисел для СОК с базисом $\{3, 4\}$.

Table 1. Number representation for RNS with basis $\{3, 4\}$.

$-6 \xrightarrow{\text{СОК}} (0, 2)$	$-5 \xrightarrow{\text{СОК}} (1, 3)$	$-4 \xrightarrow{\text{СОК}} (2, 0)$	$-3 \xrightarrow{\text{СОК}} (0, 1)$
$-2 \xrightarrow{\text{СОК}} (1, 2)$	$-1 \xrightarrow{\text{СОК}} (2, 3)$	$0 \xrightarrow{\text{СОК}} (0, 0)$	$1 \xrightarrow{\text{СОК}} (1, 1)$
$2 \xrightarrow{\text{СОК}} (2, 2)$	$3 \xrightarrow{\text{СОК}} (0, 3)$	$4 \xrightarrow{\text{СОК}} (1, 0)$	$5 \xrightarrow{\text{СОК}} (2, 1)$

Предположим, что два числа X и B представлены как $X = (x_1, x_2, \dots, x_n)$ и $B = (b_1, b_2, \dots, b_n)$ в СОК. Используя $\circ \in \{+, -, \times\}$, мы можем выразить арифметику в СОК следующим образом:

$$X \circ B = (r_1, r_2, \dots, r_n), \tag{2}$$

где

$$r_i = |x_i \circ b_i|_{p_i}.$$

В контексте модульных операций, если результат вычитания x_i из b_i отрицательный, то значение r_i определяется как:

$$r_i = p_i + (x_i - b_i).$$

Алгоритм 1 представляет метод выполнения модульных операций в СОК. В примере 1 представлено сложение двух чисел в СОК.

Пример 1 (Сложение в СОК). Сложим два числа $X = -2$ и $B = 3$ в базисе $\{3, 4\}$. Их представление в заданном базисе указано в таблице 1. Воспользуемся (2) для сложения:

$$X + B = (|1 + 0|_3, |2 + 3|_4) = (1, 1).$$

Далее рассмотрим процесс обратного преобразования числа из СОК в позиционную систему счисления.

Алгоритм 1. Выполнение модульных операций в СОК
Input: $\{p_1, p_2, \dots, p_n\}, (x_1, x_2, \dots, x_n), (b_1, b_2, \dots, b_n), \circ \in \{+, -, \times\}$
Output: $(r_1, r_2, \dots, r_n)$
1. for $i = 1, i \leq n, i++$ do:
1.2 $r_i = x_i \circ b_i _{p_i}$.
2. return $(r_1, r_2, \dots, r_n)$

Алгоритм 1. Метод выполнения модульных операций в СОК.
Algorithm 1. A method for performing modular operations in RNS.

2.2 Китайская теорема об остатках

Если число X задается в виде остатков $x_1, x_2, \dots, x_n$ от деления по модулям $p_1, p_2, \dots, p_n$, число X можно восстановить на основе Китайской теоремы об остатках (КТО) [9]:

$$X = \left| \sum_{i=1}^n P_i \cdot x_i \cdot |P_i^{-1}|_{p_i} \right|_P, \tag{3}$$

где $P_i = \frac{P}{p_i}$ и $|P_i^{-1}|_{p_i}$ представляет собой мультипликативную инверсию P_i по модулю p_i и удовлетворяет следующему соотношению $|P_i^{-1}|_{p_i} \cdot P_i|_{p_i} = 1$.

Алгоритм 2 предназначен для восстановления числа из СОК в позиционную систему счисления.

Алгоритм 2. Обратное преобразование числа с помощью КТО

Input: $\{p_1, p_2, \dots, p_n\}, (x_1, x_2, \dots, x_n), P, P_i, |P_i^{-1}|_{p_i}$ for $i = \overline{1, n}$

Output: X

1. $sums = 0$
2. **for** $i \leq n, i++$ **do**:
- 2.1 $sums += P_i \cdot x_i \cdot |P_i^{-1}|_{p_i}$
3. $X = |sums|_P$
4. **return** X

Алгоритм 2. Обратное преобразование числа с помощью КТО.

Algorithm 2. Inverse number conversion with CRT.

Рассмотрим пример 2 для иллюстрации восстановления числа с помощью КТО.

Пример 2 (Обратное преобразование числа с помощью КТО). Возьмем СОК с таким же набором модулей $\{3, 4\}$. Преобразуем число $X = (2, 1)$ в позиционную систему счисления. Для этого, найдем значения P_i :

$$P_1 = \frac{P}{p_1} = 4, P_2 = \frac{P}{p_2} = 3.$$

Затем вычислим мультипликативные инверсии $|P_i^{-1}|_{p_i}$:

$$|P_1^{-1}|_{p_1} = 1, |P_2^{-1}|_{p_2} = 3.$$

Имея эти значения, можно вычислить значение X , используя (3) получим:

$$X = |4 \cdot 1 \cdot 2 + 3 \cdot 3 \cdot 1|_{12} = 5.$$

Далее рассмотрим алгоритм определения знака числа в СОК.

2.2 Функция ядра Акушского

С целью снижения вычислительной сложности при определении знака числа в СОК за счет определения позиционных характеристик, в работе [16] была разработана новая функция, известная как функция ядра Акушского (ФЯА). Функция определяется следующим уравнением:

$$C(X) = \sum_{i=1}^n w_i \cdot \left\lfloor \frac{X}{p_i} \right\rfloor. \quad (4)$$

Функция ядра Акушского может эффективно применяться для определения знака [17], масштабирования [18], общего деления [19], обратного преобразования [20], коррекции ошибок [21].

Веса w_i из уравнения (4) представляют собой константы, определяемые выбором точки интерполяции, они определяют каждую конкретную основную функцию и могут варьироваться в зависимости от конкретной задачи. Более того, веса w_i могут быть в определенной степени произвольными. Алгоритм определения оптимальных весов для ФЯА представлен в [22].

Диапазон функции ядра вычисляется следующим образом:

$$C(P) = C_p = \sum_{i=1}^n w_i \cdot P_i. \quad (5)$$

Определим так называемые ортогональные базисы:

$$B_i = P_i \cdot |P_i^{-1}|_{p_i}, \quad (6)$$

Тогда функция ядра для ортогональных базисов будет определена как

$$C(B_i) = B_i \cdot \frac{C(P)}{P} - \frac{w_i}{p_i}. \quad (7)$$

Имея число в СОК, а также функции ядра от ортогональных базисов, функция ядра может быть определена следующим образом:

$$C(X) = \left| \sum_{i=1}^n x_i \cdot C(B_i) \right|_{C_p}. \quad (8)$$

Используя ФЯА, можно определить знак числа без преобразования числа в позиционную систему счисления. Основываясь на выражениях (1) и (3), получим:

$$\begin{cases} \begin{cases} C(X) < C\left(\frac{P-1}{2}\right) & \text{если } X \text{ положительное} \\ C(X) \geq C\left(\frac{P-1}{2}\right) & \text{если } X \text{ отрицательное} \end{cases} & \text{для нечетного } P, \\ \begin{cases} C(X) < C\left(\frac{P}{2}\right) & \text{если } X \text{ положительное} \\ C(X) \geq C\left(\frac{P}{2}\right) & \text{если } X \text{ отрицательное} \end{cases} & \text{для четного } P. \end{cases} \quad (9)$$

Алгоритм 3 предназначен для определения знака с использованием функции ядра.

Алгоритм 3. Определение знака числа с помощью функции ядра Акушского

Input: $\{p_1, p_2, \dots, p_n\}, (x_1, x_2, \dots, x_n), P, C\left(\left\lfloor \frac{P}{2} \right\rfloor\right), C(B_i)$ for $i = \overline{1, n}$

Output: S

1. $sum = 0$
 2. **for** $i \leq n, i++$ **do**:
 - 2.1 $sum += x_i \cdot C(B_i)$
 3. $C(X) = |sum|_{C_p}$
 4. **if** $C(X) < C\left(\left\lfloor \frac{P}{2} \right\rfloor\right)$:
 - 4.1 $S = 0$
 5. **else**:
 - 5.1 $S = 1$
 6. **return** S
-

Алгоритм 3. Определение знака числа с помощью функции ядра Акушского.

Algorithm 3. Determining the sign of a number using the Akushsky core function.

Рассмотрим пример определения знака числа.

Пример 3 (Определение знака числа с помощью функции ядра Акушского). Исследуем СОК с базисом $\{3, 4\}$ и весами $w_1 = 1, w_2 = 0$. Предрасчитанные параметры останутся такими же $P = 12, \frac{P}{2} = 6, P_1 = 4, P_2 = 3$ и $|P_1^{-1}|_{p_1} = 1, |P_2^{-1}|_{p_2} = 3$.

Во-первых, найдем B_i :

$$B_1 = P_1 \cdot |P_1^{-1}|_{p_1} = 4, B_2 = P_2 \cdot |P_2^{-1}|_{p_2} = 9.$$

Во-вторых, вычислим $C(P)$ и $C(\frac{P}{2})$:

$$C(P) = 1 \cdot 4 + 0 \cdot 3 = 4,$$

$$C\left(\frac{P}{2}\right) = 1 \cdot \left\lfloor \frac{6}{3} \right\rfloor + 0 \cdot \left\lfloor \frac{6}{4} \right\rfloor = 2.$$

Затем найдем $C(B_i)$ используя формулу (7):

$$C(B_1) = 4 \cdot \frac{4}{12} - \frac{1}{3} = 1, C(B_2) = 9 \cdot \frac{4}{12} + \frac{0}{4} = 3.$$

Имея эти значения, можем определить знак числа $X = (2, 3)$, используя формулу (15):

$$C(X) = |2 \cdot 1 + 3 \cdot 3|_4 = |11|_4 = 3.$$

Так как $3 > C(\frac{P}{2}) = 2$, тогда $X = (2, 3)$ это отрицательное число, что истинно, так как $X = -1 < 0$.

3. Специальные наборы модулей системы остаточных классов

Выбор набора модулей очень важен для достижения подходящей реализации СОК. Общие наборы модулей, содержащие произвольные числа, неэффективны для реализации, поэтому предпочтительны специальные наборы модулей СОК с популярными модулями вида $2^k - 1, 2^k, 2^k + 1$ [23]. В табл. 2 представлены наиболее распространённые наборы модулей специального вида и их характеристики.

Далее представлен сравнительный анализ наборов модулей из табл. 2. Для набора под номер 25 были подобраны параметры так, чтобы по количеству модулей было равно пяти.

4. Сравнительный анализ

Эффективность наборов модулей из табл. 2 была проверена для трех алгоритмов:

- Сложение чисел в СОК (алгоритм 1).
- Обратное преобразование числа с помощью КТО (алгоритм 2).
- Определение знака числа с помощью ФЯА (алгоритм 3).

Алгоритмы реализованы на языке Python. Эксперимент проводился с использованием операционной системы Windows 10 на компьютере с процессором AMD Ryzen 5 3600, оперативной памятью DDR4 16 ГБ 3200 МГц и SSD 512 ГБ.

Для каждого набора модулей (табл. 3) было произведено 10000 запусков по каждой из рассматриваемых операций и замерено среднее время исполнения алгоритмов. Средняя погрешность для операции модульного сложения составила 0,00015 секунд, для алгоритма обратного преобразования 0,0018 и для определения знака числа 0,0015. Результаты представлены в табл. 4.

Первым шагом приведем глобальный анализ результатов. А затем сравним специальные наборы в рамках количества модулей в одном наборе.

Табл.2. Наборы модулей специального вида.

Table 2. Sets of special type moduli.

Номер набора	Ссылка	Набор модулей	Год	Характеристика
1	[23]	$\{2^n - 1, 2^n, 2^n + 1\}$	1967	Удобное обратное преобразование
2	[24]	$\{2n - 1, 2n, 2n + 1\}$	1995	Неэффективный
3	[25]	$\{2^{2n} + 1, 2^n + 1, 2^n - 1\}$	1997	Удобное обратное преобразование
4	[26]	$\{2^n - 1, 2^n, 2^{n-1} - 1\}$	1998	Удобный для арифметики
5	[27]	$\{2^n - 1, 2^n, 2^{n+1} - 1\}$	1999	Удобный для арифметики
6	[28]	$\{2^n - 1, 2^n, 2^{2n+1} - 1\}$	2008	Удобный для арифметики
7	[29]	$\{2^n - 1, 2^n, 2^{2n} + 1\}$	2008	Удобное обратное преобразование
8	[30]	$\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$	2008	Гибкий, удобное обратное преобразование
9	[31]	$\{3^n - 2, 3^n - 1, 3^n\}$	2007	Удобный для арифметики
10	[32]	$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$	1999	Удобный для арифметики
11	[33]	$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$	2000	Удобный для арифметики
12	[34]	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 1\}$	2003	Удобное обратное преобразование
13	[35]	$\{2^n - 1, 2^n + 1, 2^n - 3, 2^n + 3\}$	2004	Сбалансированные модули
14	[36]	$\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$	2008	Большой динамический диапазон
15	[37]	$\{2^n - 1, 2^n + 1, 2^n, 2^{2n} + 1\}$	2009	Удобное обратное преобразование
16	[37]	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$	2009	Удобный для арифметики
17	[37]	$\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$	2009	Большой динамический диапазон, удобный для арифметических вычислений
18	[38]	$\{2^k, 2^n - 1, 2^n + 1, 2^{n+1} + 1\}$	2014	Удобный для арифметики
19	[38]	$\{2^k, 2^n - 1, 2^n + 1, 2^{n+1} - 1\}$	2014	Удобный для арифметики
20	[39]	$\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1\}$	2005	Удобное преобразование
21	[40]	$\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1, 2^{n+1} - 1\}$	2007	Сбалансированный, удобный для арифметики
22	[41]	$\{2^{n/2} - 1, 2^n, 2^{n/2} + 1, 2^n + 1, 2^{2n-1} - 1\}$	2009	Несбалансированный
23	[42]	$\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$	2012	Очень большой динамический диапазон, высокий параллелизм
24	[42]	$\{2^n - 1, 2^{n+\beta}, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$	2012	Гибкий, очень большой динамический диапазон
25	[43]	$\{2^{n+\beta}, 2^n - 1, 2^n + 1, 2^n - k_1, 2^n + k_1, \dots, 2^n - k_f, 2^n + k_f\}$	2018	Очень большой динамический диапазон, сбалансированные модули

Табл. 3. Наборы модулей, используемые для исследования (Table 3. Moduli sets used for the research).

Номер набора	Размер динамического диапазона, бит			
	8	16	24	32
1	{7, 8, 9}	{63, 64, 65}	{255, 256, 257}	{2047, 2048, 2049}
2	{7, 8, 9}	{43, 44, 45}	{257, 258, 259}	{1629, 1630, 1631}
3	{65, 9, 7}	{257, 17, 15}	{4097, 65, 63}	{65537, 257, 255}
4	{15, 16, 7}	{63, 64, 31}	{511, 512, 255}	{2047, 2048, 1023}
5	{7, 8, 15}	{63, 64, 127}	{255, 256, 511}	{2047, 2048, 4095}
6	{3, 4, 31}	{15, 16, 511}	{63, 64, 8191}	{255, 256, 131071}
7	{7, 8, 65}	{31, 32, 1025}	{127, 128, 16385}	{511, 512, 262145}
8	{16, 7, 9}	{128, 31, 33}	{1024, 127, 129}	{16385, 511, 513}
9	{7, 8, 9}	{79, 80, 81}	{727, 728, 729}	{2185, 2186, 2187}
10	{3, 4, 5, 9}	{15, 16, 17, 33}	{63, 64, 65, 129}	{255, 256, 257, 513}
11	{3, 4, 5, 7}	{15, 16, 17, 31}	{63, 64, 65, 127}	{255, 256, 257, 511}
12	{3, 4, 5, 17}	{15, 16, 17, 257}	{31, 32, 33, 1025}	{127, 128, 129, 16385}
13	{7, 9, 5, 11}	{31, 33, 29, 35}	{63, 65, 61, 67}	{255, 257, 253, 259}
14	{3, 5, 6, 29}	{7, 9, 10, 125}	{31, 33, 18, 2045}	{127, 129, 263, 2765}
15	{3, 5, 8, 9}	{15, 17, 16, 17}	{127, 129, 28, 29}	{2047, 2049, 44, 45}
16	{3, 4, 5, 31}	{15, 16, 17, 511}	{31, 32, 33, 2047}	{127, 128, 129, 32767}
17	{3, 5, 16, 31}	{7, 9, 64, 127}	{15, 17, 256, 511}	{63, 65, 4096, 8191}
18	{4, 3, 5, 9}	{8, 15, 17, 33}	{4, 127, 129, 257}	{16, 511, 513, 1025}
19	{2, 7, 9, 5}	{4, 31, 33, 17}	{2, 255, 257, 129}	{8, 1023, 1025, 513}
20	{7, 8, 9, 5, 13}	{31, 32, 33, 25, 41}	{127, 128, 129, 113, 145}	{511, 512, 513, 481, 545}
21	{3, 4, 5, 1, 7}	{15, 16, 17, 7, 31}	{31, 32, 33, 15, 63}	{127, 128, 129, 63, 255}
22	{3, 16, 5, 17, 127}	{7, 64, 9, 65, 2047}	{7, 64, 9, 65, 2047}	{15, 256, 17, 257, 32767}
23	{7, 8, 9, 5, 13, 15}	{31, 32, 33, 25, 41, 63}	{31, 32, 33, 25, 41, 65}	{127, 128, 129, 113, 145, 255}
24	{7, 8, 9, 5, 13, 5}	{7, 16, 9, 5, 13, 5}	{31, 32, 33, 25, 41, 17}	{31, 256, 33, 25, 41, 17}
25	{8, 7, 9, 5, 11}	{16, 15, 17, 13, 19}	{32, 31, 33, 29, 35}	{128, 127, 129, 125, 131}

Табл. 4. Результаты исследования, время в секундах (Table 4. Research results, time in seconds).

Номер набора	Размер динамического диапазона, бит											
	8			16			24			32		
	Сложение	КТО	ФЯА	Сложение	КТО	ФЯА	Сложение	КТО	ФЯА	Сложение	КТО	ФЯА
1	1,9E-03	0,2813	0,2837	1,8E-03	0,2758	0,2803	1,8E-03	0,2776	0,2830	1,8E-03	0,2807	0,8182
2	1,8E-03	0,2810	0,2878	1,8E-03	0,2760	0,2793	1,8E-03	0,2768	0,2812	1,9E-03	0,2793	0,8157
3	1,8E-03	0,2795	0,2839	1,8E-03	9,2758	0,2810	1,8E-03	0,2765	0,2793	1,8E-03	0,2801	0,8123
4	1,9E-03	0,2798	0,2826	1,8E-03	0,2759	0,2805	1,8E-03	0,2762	0,2800	1,9E-03	0,2794	0,2866
5	1,8E-03	0,2795	0,2827	1,8E-03	0,2752	0,2769	1,8E-03	0,2756	0,2798	1,9E-03	0,2803	0,8157
6	1,9E-03	0,2804	0,2843	1,8E-03	0,2766	0,2801	1,8E-03	0,2765	0,2805	1,8E-03	0,2814	0,8162
7	1,8E-03	0,2862	0,2929	1,9E-03	0,2854	0,2894	1,8E-03	0,2880	0,2893	1,9E-03	0,2865	0,8326
8	1,8E-03	0,2798	0,2832	1,8E-03	0,2767	0,2791	1,8E-03	0,2761	0,2796	1,8E-03	0,2782	0,8159
9	1,9E-03	0,2893	0,2983	1,8E-03	0,2846	0,2949	1,9E-03	0,2825	0,2862	1,9E-03	0,2829	0,8173
10	2,4E-03	0,3365	0,3388	2,3E-03	0,3300	0,3347	2,3E-03	0,3342	0,3335	2,4E-03	0,3349	1,043
11	2,3E-03	0,3377	0,3406	2,3E-03	0,3308	0,3352	2,3E-03	0,3305	0,3342	2,3E-03	0,3369	1,053
12	2,3E-03	0,3367	0,3389	2,3E-03	0,3309	0,3361	2,3E-03	0,3313	0,3350	2,3E-03	0,3379	1,049
13	2,3E-03	0,3350	0,3391	2,3E-03	0,3310	0,3339	2,3E-03	0,3316	0,3372	2,4E-03	0,3360	0,3423
14	2,4E-03	0,3355	0,3376	2,3E-03	0,3294	0,3352	2,3E-03	0,3310	0,3323	2,3E-03	0,3383	1,057
15	2,3E-03	0,3375	0,3373	2,3E-03	0,3331	0,3345	2,3E-03	0,3320	0,3347	2,3E-03	0,3350	1,045
16	2,3E-03	0,3362	0,3385	2,3E-03	0,3311	0,3357	2,3E-03	0,3317	0,3340	2,4E-03	0,3370	1,048
17	2,4E-03	0,3353	0,3393	2,2E-03	0,3303	0,3320	2,4E-03	0,3308	0,3352	2,3E-03	0,3388	1,046
18	2,3E-03	0,3369	0,3385	2,4E-03	0,3315	0,3335	2,3E-03	0,3324	0,3337	2,3E-03	0,3381	1,047
19	2,3E-03	0,3363	0,3396	2,3E-03	0,3309	0,3367	2,3E-03	0,3303	0,3326	2,4E-03	0,3404	1,057
20	2,8E-03	0,3941	0,3967	2,8E-03	0,3860	0,3905	2,8E-03	0,3908	1,265	2,8E-03	0,3939	1,281
21	2,9E-03	0,3938	0,3977	2,8E-03	0,3863	0,3900	2,8E-03	0,3867	0,3875	2,8E-03	0,3940	1,294
22	2,9E-03	0,3920	0,3957	2,8E-03	0,3894	0,3922	2,8E-03	0,3885	0,3891	2,8E-03	0,3920	1,281
23	3,3E-03	0,4487	0,4502	3,3E-03	0,4429	0,4432	3,3E-03	0,4451	0,4468	3,3E-03	0,4494	1,521
24	3,3E-03	0,4483	0,4522	3,3E-03	0,4413	0,4434	3,3E-03	0,4436	0,4437	3,4E-03	0,4492	1,504
25	2,8E-03	0,3929	0,3963	2,8E-03	0,3856	0,3907	2,8E-03	0,3879	0,3875	2,9E-03	0,3928	1,287

При выполнении операции модульного сложения для различных наборов данных были получены результаты, демонстрирующие высокую степень сходства между собой. Наиболее эффективными для сложения, оказались наборы модулей под номерами 1-9. Рассмотрим результаты сравнения для немодульных операций.

Результаты обратного преобразования с использованием КТО:

Среди 8, 16 и 24 битных, лучший результат показал набор модулей $\{2^n - 1, 2^n, 2^{n+1} - 1\}$, при этом:

- Среди 8 битных наборов он эффективнее набора $\{2^n - 1, 2^n, 2^{n-1} - 1\}$ на 0,1% и в среднем эффективнее остальных базисов на 17%.
- Среди 16 битных наборов он эффективнее набора $\{2^{2n} + 1, 2^n + 1, 2^n - 1\}$ на 0,2% и в среднем эффективнее остальных базисов на 17,1%.
- Среди 24 битных наборов он эффективнее набора $\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$ на 0,18% и в среднем эффективнее остальных наборов на 17,2%

Среди 32 битных лучший результат у набора $\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$, при этом он быстрее набора модулей $\{2^n - 1, 2^n, 2^n + 1\}$ на 0,32% и в среднем эффективнее остальных наборов на 17,5%.

Результаты в определении знака с использованием ФЯА:

- Для 8 битных базисов лучший результат у набора $\{2^n - 1, 2^n, 2^{n-1} - 1\}$, при этом он эффективнее набора $\{2^n - 1, 2^n, 2^{n+1} - 1\}$ на 0,03% и в среднем эффективнее остальных наборов на 17%.
- Среди 16 битных базисов лучший результат у набора $\{2^n - 1, 2^n, 2^{n+1} - 1\}$, при этом он лучше второго по оптимальности набора $\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$ на 0,79% и в среднем эффективнее остальных наборов на 17,5%.
- Среди 24 битных базисов лучший результат у набора $\{2^{2n} + 1, 2^n + 1, 2^n - 1\}$, при этом он эффективнее набора $\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$ на 0,1% и в среднем эффективнее показателей остальных наборов на 24,8%.
- Среди 32 битных базисов лучший результат у набора $\{2^n - 1, 2^n, 2^{n-1} - 1\}$, он эффективнее набора $\{2^n - 1, 2^n + 1, 2^n - 3, 2^n + 3\}$ на 19,43% и в среднем эффективнее остальных наборов на 71,9%.

Для оценки эффективности выполнения операций в различных разрядностях наборов данных были вычислены суммы временных затрат для каждого типа набора, в рамках каждой операции. Полученные результаты представлены в табл. 5, которая позволяет провести сравнительный анализ и сформировать итоговый рейтинг производительности.

Исходя из данных приведенных в табл. 5 можно сделать следующие выводы.

Для трёх модульных базисов:

- В сложении лучшие результаты показали наборы $\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$ и $\{2^{2n} + 1, 2^n + 1, 2^n - 1\}$, они в среднем быстрее других трех модульных наборов на 1.69%.
- В операции обратного преобразования лучший результат показал набор $\{2^n - 1, 2^n, 2^{n+1} - 1\}$, он быстрее других трех модульных наборов на 0.78%.
- В определении знака с лучшим результатом показал набор $\{2^n - 1, 2^n, 2^n + 1\}$, он быстрее остальных трех модульных наборов на 42.51%.

Для четырех модульных базисов:

- В сложении лучшие результаты показали наборы $\{2^n - 1, 2^n + 1, 2^n, 2^{2n} + 1\}$, $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 1\}$ и $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$, они в среднем быстрее других четырех модульных наборов на 0.87%.
- В операции обратного преобразования лучший результат показал набор $\{2^n - 1, 2^n + 1, 2^n - 3, 2^n + 3\}$, он быстрее других четырех модульных наборов на 0.19%.

- В определении знака с лучший результат показал набор $\{2^n - 1, 2^n + 1, 2^n - 3, 2^n + 3\}$, он быстрее других четырех модульных наборов на 46.87%.

Для пяти модульных базисов:

- В сложении лучшие результаты показали наборы $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1\}$, они в среднем быстрее других пяти модульных наборов на 6.54%.
- В операции обратного преобразования лучший результат показал набор $\{2^{n+\beta}, 2^n - 1, 2^n + 1, 2^n - k_1, 2^n + k_1, \dots, 2^n - k_f, 2^n + k_f\}$, он быстрее других пяти модульных наборов на 4.91%.
- В определении знака с лучший результат показал набор $\{2^{n/2} - 1, 2^n, 2^{n/2} + 1, 2^n + 1, 2^{2n-1} - 1\}$, он быстрее других пяти модульных наборов на 11.38%.

Табл.5. Рейтинг наборов модулей специального вида.

Table 5. Ranking of special type moduli sets.

Количество модулей	Место	Модули	Сумма времени, сек.		
			Сложение	КТО	ФЯА
3	1	$\{2^n - 1, 2^n, 2^n + 1\}$	0,0074	1,1113	1,1297
	2	$\{2^n - 1, 2^n, 2^{n+1} - 1\}$	0,0073	1,1106	1,6551
	3	$\{2^{2n} + 1, 2^n + 1, 2^n - 1\}$	0,0072	1,1119	1,6565
	4	$\{2^\alpha, 2^\beta - 1, 2^\beta + 1\}$	0,0072	1,1108	1,6578
	5	$\{2^n - 1, 2^n, 2^{2n+1} - 1\}$	0,0073	1,1149	1,6611
	6	$\{2n - 1, 2n, 2n + 1\}$	0,0073	1,1131	1,6640
	7	$\{2^n - 1, 2^n, 2^n + 1\}$	0,0073	1,1154	1,6652
	8	$\{3^n - 2, 3^n - 1, 3^n\}$	0,0075	1,1393	1,6967
	9	$\{2^n - 1, 2^n, 2^{2n} + 1\}$	0,0074	1,1461	1,7042
4	1	$\{2^n - 1, 2^n + 1, 2^n - 3, 2^n + 3\}$	0,0093	1,3336	1,3525
	2	$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$	0,0094	1,3356	2,0504
	3	$\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$	0,0093	1,3352	2,0528
	4	$\{2^n - 1, 2^n + 1, 2^n, 2^{2n} + 1\}$	0,0092	1,3376	2,0511
	5	$\{2^k, 2^n - 1, 2^n + 1, 2^{n+1} + 1\}$	0,0093	1,3389	2,0528
	6	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$	0,0093	1,3360	2,0560
	7	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 1\}$	0,0092	1,3368	2,0591
	8	$\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$	0,0093	1,3342	2,0616
	9	$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$	0,0092	1,3359	2,0625
	10	$\{2^k, 2^n - 1, 2^n + 1, 2^{n+1} - 1\}$	0,0093	1,3379	2,0662
5	1	$\{2^{n+\beta}, 2^n - 1, 2^n + 1, 2^n - k_1, 2^n + k_1, \dots, 2^n - k_f, 2^n + k_f\}$	0,0113	1,5619	2,4577
	2	$\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1, 2^{n+1} - 1\}$	0,0113	1,5592	2,4613
	3	$\{2^n - 1, 2^{n+\beta}, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$	0,0113	1,5608	2,4690
	4	$\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1, 2^{n+1} + 1\}$	0,0133	1,7824	2,8432
	5	$\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1\}$	0,0132	1,7861	2,8612
	6	$\{2^{n+\beta}, 2^n - 1, 2^n + 1, 2^n - k_1, 2^n + k_1, \dots, 2^n - k_f, 2^n + k_f\}$	0,0112	1,5648	3,3329

5. Заключение

В данной статье были исследованы известные наборы модулей СОК и их производительность по ключевым операциям: сложение, обратное преобразование и определение знака числа. Результаты показывают, что эффективность различных наборов может сильно меняться при

выполнении немодульных операций, что оказывает значительное влияние на производительность, особенно в случае более высоких разрядностей.

Экспериментальные результаты показали, что базис $\{2^n - 1, 2^n, 2^n + 1\}$ оказался наиболее эффективным среди трех модульных наборов. Это делает его оптимальным выбором для современных вычислительных систем. Дальнейшие исследования будут направлены на алгоритм генерации модулей, способный задавать набор очень большой размерности, но при этом сохранять эффективность.

Список литературы / References

- [1]. Horowitz M., Grumbling E. Quantum Computing: Progress and Prospects. 2019.
- [2]. Reuther A., Michaleas P., Jones M., Gadepally V., Samsi S., Kepner J. AI Accelerator Survey and Trends. In Proceedings of the 2021 IEEE High Performance Extreme Computing Conference (HPEC); IEEE, 2021; pp. 1–9.
- [3]. Mohan P.V.A. Residue Number Systems; Springer International Publishing: Cham, 2016, ISBN 978-3-319-41383-9.
- [4]. Амербаев В. М. Теоретические основы машинной арифметики. АлмаАта: Наука, 1976. 324 с. / Amerbayev V.M. Theoretical Foundations of Machine Arithmetic. Alma-Ata, Science 1976, 324 (in Russian).
- [5]. Krasnobayev V., Kuznetsov A., Yanko A., Kuznetsova T. The Analysis of the Methods of Data Diagnostic in a Residue Number System. Computer Modeling and Intelligent Systems 2020, 2608, 594–609, doi:10.32782/cmisis/2608-46.
- [6]. Guo Z., Gao Z., Mei H., Zhao M., Yang J. Design and Optimization for Storage Mechanism of the Public Blockchain Based on Redundant Residual Number System. IEEE Access 2019, vol. 7, pp. 98546–98554, doi:10.1109/ACCESS.2019.2930125.
- [7]. Su Y., Yang B.-L., Yang C., Zhao S.-Y. ReMCA: A Reconfigurable Multi-Core Architecture for Full RNS Variant of BFV Homomorphic Evaluation. IEEE Transactions on Circuits and Systems I: Regular Papers 2022, vol. 69, pp. 2857–2870, doi:10.1109/TCSI.2022.3163970.
- [8]. Cardarilli G.C., Nunzio L.D., Fazzolari R., Nannarelli A., Petricca M., Re M. Design Space Exploration Based Methodology for Residue Number System Digital Filters Implementation. IEEE Transactions on Emerging Topics in Computing 2022, vol. 10, pp. 186–198, doi:10.1109/ETC.2020.2997067.
- [9]. Chervyakov N.I., Molahosseini A.S., Lyakhov P.A., Babenko M.G., Deryabin M.A. Residue-to-Binary Conversion for General Moduli Sets Based on Approximate Chinese Remainder Theorem. International Journal of Computer Mathematics 2017, vol. 94, pp. 1833–1849, doi:10.1080/00207160.2016.1247439.
- [10]. Omondi A.R., Premkumar A.B. Residue Number Systems: Theory And Implementation; World Scientific, 2007, ISBN 978-1-908979-11-7.
- [11]. Chervyakov N., Babenko M., Tchernykh A., Kuchero N., Miranda-López V., Cortés-Mendoza J.M. AR-RRNS: Configurable Reliable Distributed Data Storage Systems for Internet of Things to Ensure Security. Future Generation Computer Systems 2019, vol. 92, pp. 1080–1092, doi:10.1016/j.future.2017.09.061.
- [12]. Lutsenko V., Zgonnikov M. Fault Tolerant System for Data Storage, Transmission and Processing in Fog Computing Using Artificial Neural Networks. In Current Problems of Applied Mathematics and Computer Systems. Lecture Notes in Networks and Systems. Springer Nature Switzerland: Cham, 2024, vol. 1044, pp. 199–212, ISBN 978-3-031-64009-4.
- [13]. Valueva M.V., Nagornov N.N., Lyakhov P.A., Valuev G.V., Chervyakov N.I. Application of the Residue Number System to Reduce Hardware Costs of the Convolutional Neural Network Implementation. Mathematics and Computers in Simulation 2020, vol. 177, pp. 232–243, doi:10.1016/j.matcom.2020.04.031.
- [14]. Roohi A., Taheri M., Angizi S., Fan D. RNSim: Efficient Deep Neural Network Accelerator Using Residue Number Systems. In Proceedings of the 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), November 2021, pp. 1–9.
- [15]. Molahosseini A.S., Navi K. Study of the Reverse Converters for the Large Dynamic Range Four-Moduli Sets; InTech Press, 2011.
- [16]. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. М., Советское радио, 1968, 440 с. / Akushsky I. Ya., Yuditsky D. I. Computer arithmetic in residual classes. Moscow, Soviet Radio, 1968, 440 p. (in Russian).

- [17]. Shiriaev E., Kuchero N., Babenko M., Nazarov A. Fast Operation of Determining the Sign of a Number in RNS Using the Akushsky Core Function. *Computation*, 2023, vol. 11, issue 124, doi:10.3390/computation11070124.
- [18]. Burgess N. Scaling an RNS Number Using the Core Function. In *Proceedings of the 16th IEEE Symposium on Computer Arithmetic*, 2003. *Proceedings.*; IEEE Comput. Soc: Santiago de Compostela, Spain, 2003, pp. 262–269.
- [19]. Луценко В.В., Бабенко М.Г., Черных А.Н., Лапина М.А. Оптимизация алгоритма деления чисел в системе остаточных классов на основе функции ядра Акушского. *Труды ИСП РАН*, том 35, вып. 5, 2023 г., стр. 157–168. / Lutsenko V.V., Babenko M.G., Tchernykh A.N., Lapina, M.A. Optimization of a Number Division Algorithm in the Residue Number System Based on the Akushsky Core Function. *Proceedings of the Institute for System Programming of the RAS*, 2023, vol. 35, issue 5, pp. 157–168. (in Russian), DOI: 10.15514/ISPRAS–2023–35(5)–11.
- [20]. Lutsenko V., Babenko M. High Speed Method of Conversion Numbers from Residue Number System to Positional Notation.
- [21]. Lutsenko V., Zgonnikov M. Investigation of Neural Network Methods for Error Detection and Correction in the Residue Number System. In *AISMA-2024: International Workshop on Advanced Information Security Management and Applications. Lecture Notes in Networks and Systems*; Springer Nature Switzerland: Cham, 2024, vol. 863, pp. 194–206 ISBN 978-3-031-72170-0.
- [22]. Shiriaev E., Kuchero N., Babenko M., Lutsenko V., Al-Galda S. Algorithm for Determining the Optimal Weights for the Akushsky Core Function with an Approximate Rank. *Applied Sciences*, 2023, vol. 13, 10495.
- [23]. Szabo N.S., Tanaka R.I. *Residue Arithmetic and Its Applications to Computer Technology*, 1967.
- [24]. Premkumar A.B. An RNS to Binary Converter in a Three Moduli Set with Common Factors. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 1995, vol. 42, pp. 298–301.
- [25]. Pourbigharaz F. A Signed-Digit Architecture for Residue to Binary Transformation. *IEEE Transactions on Computers*, 1997, vol. 46, pp. 1146–1150.
- [26]. Hiasat A.A., Abdel-Aty-Zohdy S.H. Residue-to-Binary Arithmetic Converter for the Moduli Set $2^k, 2^k - 1, 2^{k-1} - 1$. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 1998, 45, 204–209.
- [27]. Mathew J., Radhakrishnan D., Srikanthan T. Fast Residue-to-Binary Converter Architectures. In *Proceedings of the 42nd Midwest Symposium on Circuits and Systems (Cat. No. 99CH36356)*. IEEE, 1999; vol. 2, pp. 1090–1093.
- [28]. Molahosseini A.S., Navi K., Rafsanjani M.K. A New Residue to Binary Converter Based on Mixed-Radix Conversion. In *Proceedings of the 2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, 2008, pp. 1–6.
- [29]. Hariri A., Navi K., Rastegar R. A New High Dynamic Range Moduli Set with Efficient Reverse Converter. *Computers & mathematics with applications* 2008, vol. 55, pp. 660–668.
- [30]. Molahosseini A.S., Navi K., Hashemipour O., Jalali A. An Efficient Architecture for Designing Reverse Converters Based on a General Three-Moduli Set. *Journal of Systems Architecture*, 2008, vol. 54, pp. 929–934.
- [31]. Hosseinzad M., Navi K. A New Moduli Set for Residue Number System in Ternary Valued Logic. *Journal of Applied Sciences*, 2007, vol. 7, pp. 3729–3735.
- [32]. Bhardwaj M., Srikanthan T., Clarke C.T., Center M.D., Microelectronics S. Reverse Converter for the 4-Moduli Superset $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$. In *Proceedings of the Proceedings of the IEEE Conference on Computer Arithmetic*; Citeseer, 1999, vol. 14, pp. 168–175.
- [33]. Vinod A.P., Premkumar A.B. A memoryless reverse converter for the 4-moduli superset $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$. *J circuit syst comp*, 2000, vol. 10, pp. 85–99, doi:10.1142/S0218126600000044.
- [34]. Cao B., Chang C.-H., Srikanthan T. An Efficient Reverse Converter for the 4-Moduli Set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 1\}$ Based on the New Chinese Remainder Theorem. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 2003, vol. 50, pp. 1296–1303.
- [35]. Sheu M.-H., Lin S.-H., Chen C., Yang S.-W. An Efficient VLSI Design for a Residue to Binary Converter for General Balance Moduli $\{2^n - 1, 2^n + 1, 2^n - 3, 2^n + 3\}$. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2004, vol. 51, pp. 152–155.
- [36]. Zhang W., Siy P. An Efficient Design of Residue to Binary Converter for Four Moduli Set $(2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3)$ Based on New CRT II. *Information Sciences*, 2008, vol. 178, pp. 264–279.

- [37]. Molahosseini A.S., Navi K., Dadkhah C., Kavehei O., Timarchi S. Efficient Reverse Converter Designs for the New 4-Moduli Sets $\{2^n - 1, 2^n + 1, 2^n, 2^{2n} + 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n+1} - 1\}$ Based on New CRTs. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2009, vol. 57, pp. 823–835.
- [38]. Patronik P., Piestrak S.J. Design of Reverse Converters for General RNS Moduli Sets $\{2^k, 2^n - 1, 2^n + 1, 2^{n+1} - 1\}$ and $\{2^k, 2^n - 1, 2^n + 1, 2^{n+1} + 1\}$ (n Even). *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2014, vol. 61, pp. 1687–1700.
- [39]. Hiasat A.A. VLSI Implementation of New Arithmetic Residue to Binary Decoders. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2005, vol. 13, pp. 153–158.
- [40]. Cao B., Chang C.-H., Srikanthan T. A Residue-to-Binary Converter for a New Five-Moduli Set. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2007, vol. 54, pp. 1041–1049.
- [41]. Molahosseini A.S., Dadkhah C., Navi K. A. New Five-Moduli Set for Efficient Hardware Implementation of the Reverse Converter. *IEICE Electronics Express*, 2009, vol. 6, pp. 1006–1012.
- [42]. Pettenghi H., Chaves R., Sousa L. RNS Reverse Converters for Moduli Sets with Dynamic Ranges up to $(8n+1)$ -Bit. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2012, vol. 60, pp. 1487–1500.
- [43]. Pettenghi H., Paludo R., Matos R., Lyakhov P.A. Efficient RNS Reverse Converters for Moduli Sets with Dynamic Ranges Up to $(10n+1)(10N+1)$ -Bit. *Circuits, Systems, and Signal Processing*, 2018, vol. 37, pp. 5178–5196.

Информация об авторах / Information about authors

Владислав Вячеславович ЛУЦЕНКО – аспирант, кафедры вычислительной математики и кибернетики факультета математики и компьютерных наук имени профессора Н.И. Червякова ФГАОУ ВПО «Северо-Кавказский федеральный университет». Сфера научных интересов: высокопроизводительные вычисления, система остаточных классов, умный город, нейронные сети, интернет вещей.

Vladislav Vyacheslavovich LUTSENKO – postgraduate student, Department of Computational Mathematics and Cybernetics, Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov, North Caucasus Federal University. Research interests: high-performance computing, residue number system, smart city, neural networks, Internet of Things.

Михаил Дмитриевич КРАВЦОВ является учеником регионального центра выявления, поддержки и развития способностей и талантов детей и молодежи Ставропольского края «Сириус 26». Сфера научных интересов: высокопроизводительные вычисления, система остаточных классов, компьютерные науки.

Mikhail Dmitrievich KRAVTSOV is a student of the regional center for identification, support and development of abilities and talents of children and youth of the Stavropol Territory Sirius 26. Area of scientific interests: high-performance computing, residue number system, computer science.

Дмитрий Евгеньевич ГОРЛАЧЕВ – студент Северо-Кавказского Федерального университета, факультет математики и компьютерных наук имени профессора Н.И. Червякова. Сфера научных интересов: математика и компьютерные науки.

Dmitriy Evgenievich GORLACHEV is a student of the North Caucasus Federal University, Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov. Area of scientific interests: mathematics and computer sciences.

Никита Михайлович МИРНЫЙ – студент Северо-Кавказского Федерального университета, факультет математики и компьютерных наук имени профессора Н.И. Червякова. Сфера научных интересов: математика и компьютерные науки.

Nikita Mikhailovich MIRNY is a student of the North Caucasus Federal University, Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov. Area of scientific interests: mathematics and computer sciences.

DOI: 10.15514/ISPRAS-2025-37(3)-8



Electrostatic Discharge Protection Devices Macromodelling Using Open-Source Tools

V.V. Kuznetsov, ORCID: 0000-0002-5738-555X <vadim.kuznetsov@bmstu.ru>

V.V. Andreev, ORCID: 0000-0002-8132-7050 <vladimir_andreev@bmstu.ru>

S.A. Lomakin, ORCID: 0009-0003-2867-0815 <lomakinsa@student.bmstu.ru>

*Bauman Moscow State Technical University,
5, Bldg. 1, 2nd Baumanskaya st., Moscow, 105005, Russia.*

Abstract. The semiconductor diode and grounded gate MOSFET (GGMOS) devices are commonly used as electrostatic discharge (ESD) protection element in CMOS ICs circuitry. This article presents an implementation of ESD diode and GGMOS macro models using open-source circuit simulation tools (Qucs-S and Ngspice). The proposed models could serve for the circuit simulation of the ESD event. Such simulation allows to estimate the ESD robustness of the IC at the early design stage.

Keywords: electrostatic discharge (ESD); circuit simulation; compact modelling; transmission line pulse.

For citation: Kuznetsov V.V., Andreev V.V., Lomakin S.A. Electrostatic discharge protection devices macromodelling using open-source tools. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 3, 2025, pp. 121-130. DOI: 10.15514/ISPRAS-2025-37(3)-8.

Acknowledgements: The research was financially supported by the Ministry of Science and Higher Education of the Russian Federation as a part of the project FSN-2024-0086.

**Макромоделирование компонентов защиты
от электростатического разряда
с применением программного обеспечения
с открытым исходным кодом**

В.В. Кузнецов, ORCID: 0000-0002-5738-555X <vadim.kuznetsov@bmstu.ru>

В.В. Андреев, ORCID: 0000-0002-8132-7050 <vladimir_andreev@bmstu.ru>

С.А. Ломакин, ORCID: 0009-0003-2867-0815 <lomakinsa@student.bmstu.ru>

*Московский государственный технический университет имени Н.Э. Баумана,
Россия, 105005, Москва, ул. 2-я Бауманская, д. 5, стр. 1.*

Аннотация. Полупроводниковый диод и полевой транзистор с заземлённым затвором (GGMOS) часто используются в качестве элементов защиты от электростатического разряда (ЭСР) в схемотехнике КМОП ИМС. В статье представлена реализация макромоделей данных компонентов с использованием ПО для схемотехнического моделирования с открытым исходным кодом. Предложенные модели могут служить для моделирования воздействия ЭСР на ИМС. Подобное моделирование позволяет оценить стойкость ИМС к воздействию ЭСР на ранней стадии проектирования.

Ключевые слова: электростатический разряд; схемотехническое моделирование; компактные модели; импульс линии передачи.

Для цитирования: Кузнецов В.В., Андреев В.В., Ломакин С.А. Макромоделирование компонентов защиты от электростатического разряда с применением программного обеспечения с открытым исходным кодом Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 121–130 (на английском языке). DOI: 10.15514/ISPRAS-2025-37(3)-8.

Благодарности: Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ в рамках Госзадания FSFN-2024-0086.

1. Introduction

The ESD protection circuit is a part of all modern digital and analog CMOS ICs. There exists a number of proprietary specialized simulation tools like Cadence ESD Checker or Magwell ESDi for ESD stress modelling. Different semiconductor devices may be used for ESD protection purpose. The aluminum gate CD4000 and polysilicon gate 74HC series has used the diode protection. The operating principle of the diode protection circuit is based on the properties of the p-n junction IV-curve (Fig. 1a). After reaching the first breakdown voltage V_{br} the diode acts as the voltage limiter until the thermal breakdown point (V_{t2} ; I_{t2}) and device destruction. The diode ESD protection circuit could be simulated using the general purpose SPICE [1] diode model. But this approach has some disadvantages. The ESD protection operates in high current regions under short pulse duration. The diode body resistance may have a nonlinear dependency [2] on current that should be considered for typical ESD current range. The development of the diode macromodel taking into account ESD effects will improve the simulation accuracy.

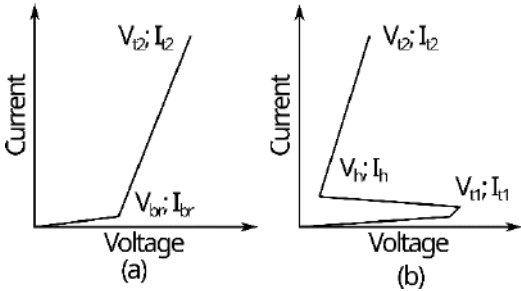


Fig. 1. Idealized IV-curves of diode (a) and the GGMOS (b) ESD protection devices.

The more modern digital IC series including submicron technology nodes used the ESD protection based on grounded gate MOSFET (GGMOS) device [3]. The operating principle of this ESD protection is based on the thyristor effect in the parasitic lateral BJT formed by drain, source, and channel regions of the MOSFET device. The cross-section of the MOSFET device is shown in the Fig. 2.

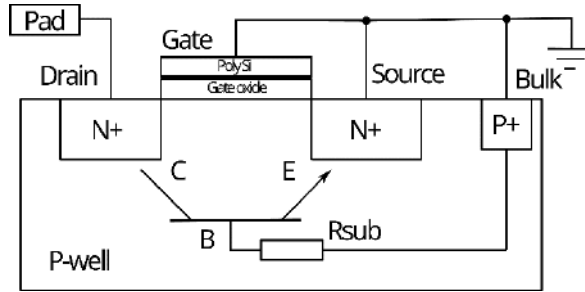


Fig. 2. Cross-section of the GGMOS ESD protection device.

The typical IV-curve of the GGMOS device is shown in the Fig. 1b. The parasitic BJT starts to conduct at the trigger voltage point V_{t1} , I_{t1} . Then the device latches up and voltage drops to the holding voltage V_h . If the current across the device exceeds the second breakdown point I_{t2} , the GGMOS device goes to the thermal breakdown and the protection is destroyed. The trigger voltage V_{t1} should not exceed the gate breakdown voltage for the proper device operation. This IV-curve could be obtained using quasi-static transmission line pulse (TLP) method.

The GGMOS device operation under ESD stress cannot be simulated using existing SPICE models. The MOS level 1-9 or BSIM [4] models don't take into account parasitic BJT behavior. The special macromodel is needed to simulate the GGMOS ESD protection with SPICE. A compact model of the GGMOS device has been proposed by J. Li, E. Rosenbaum and Y. Zhou [5], [6]. But the existing publications on this topic don't consider the practical implementation of the proposed model using the circuit simulation CAD tools and model parameters extraction.

The purpose of this research is to implement a practical ESD protection devices (diode and GGMOS) macromodels using the open-source circuit simulation tools (Qucs-S and Ngspice).

2. ESD diode macromodel

The proposed diode macromodel is shown in the Fig. 3. This model is designed using Qucs-S circuit simulation software with Ngspice backend. The R_{tlp} behavioral resistance is connected in series to SPICE diode model. It represents the dependency of the diode resistance on the forward current.

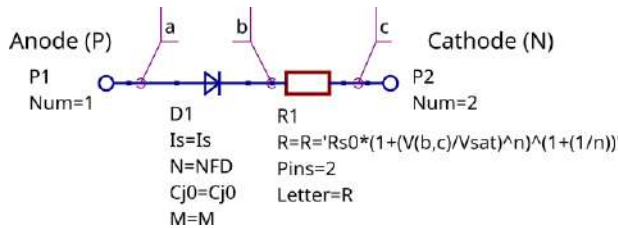


Fig. 3. Diode macromodel taking into account ESD effects.

The model uses the following dependency of the diode resistance on current:

$$R_{tlp} = R_{s0} \left(1 + \left(\frac{V}{V_{sat}} \right)^n \right)^{\left(1 + \frac{1}{n} \right)} \quad (1)$$

where the V is the voltage drop across the series resistance. Other symbols are the model parameters and explained in the Table 1.

Table 1. ESD diode model parameters.

Parameter	Description	Default value
V_{sat}	saturation voltage (V)	1.0
R_{s0}	initial diode series resistance (Ohm)	1.0
N	fitting coefficient $N > 1$	1.02

As could be seen, the model requires three additional parameters except the standard SPICE diode parameters. The R_{s0} resistance could be extracted from the TLP IV-curve of the diode (Fig. 1a) as the differential resistance at the initial part between V_{br} and V_{t2} . The remaining parameters could be extracted from the measurements data using the curves fitting.

The Fig. 4 shows the DC sweep simulation result for general SPICE diode model and improved ESD diode macromodel. The model parameters were extracted to match the TLP measurement data for drain-bulk diode in the MOSFET device. The TLP setup designed by authors was used for the measurements [7,8]. The gate dielectric degradation after the ESD stress could be investigated using the methods presented in [9]. The improved macromodel shows the better matching than SPICE model. The testbench schematic in Qucs-S CAD and model parameters are shown in the Fig. 5.

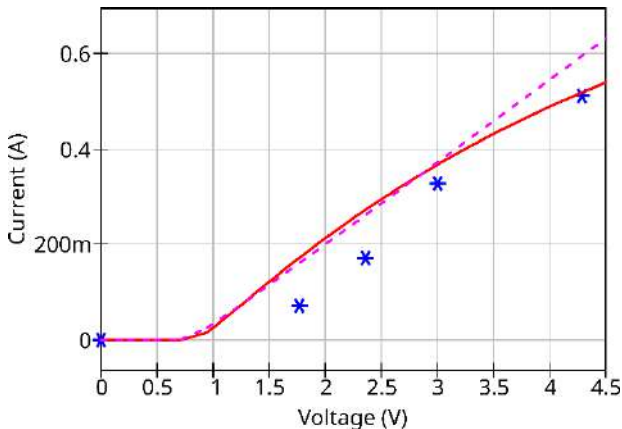


Fig. 4. Simulated (solid line) ESD diode IV-curve; general SPICE diode IV-curve (dashed line), and measured TLP diode IV-curve (stars).

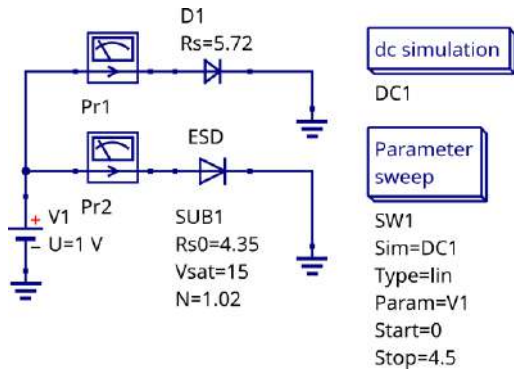


Fig. 5. Diode macromodel DC sweep testbench schematic. D1 – SPICE diode; SUB1 – improved micromodel.

3. GGMOS device macromodel

The equivalent circuit of the GGMOS device is shown in the Fig. 6. This macromodel was designed using Qucs-S [10,11] circuit simulation tool with Ngspice backend follows the model proposed in [5]. The parasitic NPN BJT is connected in parallels to NMOS device channel. The MOSFET uses level 1-9 model for micron technology or BSIM model for submicron technology. The BJT uses full Gummel-Poon model. The approach using the compact model (MOS level 1 and Ebers-Moll for BJT) was also tested, but the full SPICE model showed a better convergence for time domain simulation. The setting of the forward Early voltage (VAF) parameter of the BJT requires to improve the model convergence. Two additional current controlled current sources (CCCS) B1 and B2 are needed to simulate the impact ionization current I_{ii} and triggering of the parasitic BJT. The resistor R_c represents drain area resistance, and R_{sub} represents the substrate resistance. The C_1 capacitor is convergence helper. The V_c and V_d are the current sensing zero voltage sources.

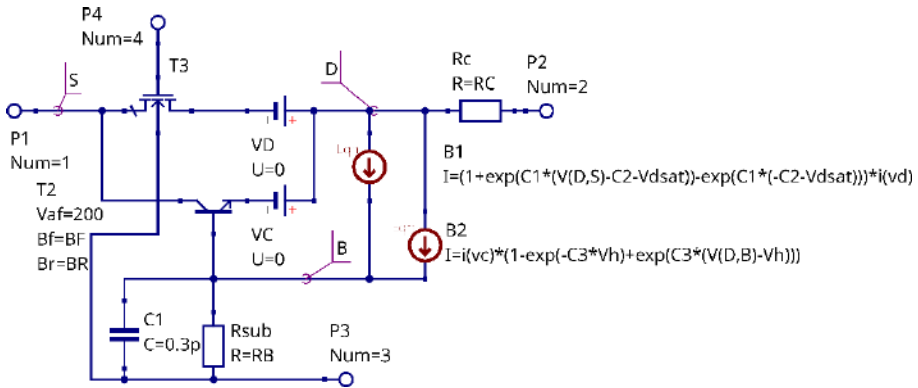


Fig. 6. GGMOS device equivalent circuit.

The impact ionization current depends on MOSFET drain I_d and BJT collector I_c currents as following:

$$I_{ii} = M_1 I_d + M_2 I_c \quad (2)$$

where the M_1 and M_2 are the multiplication factors that depends on device voltages as following:

$$M_1 = 1 + \exp(C_1 (V_{ds} - C_2 - V_{dsat})) - \exp(C_1 (-C_2 - V_{dsat})) \quad (3)$$

$$M_2 = 1 - \exp(-C_3 V_h) + \exp(C_3 V_{db} - V_h) \quad (4)$$

where V_{ds} is drain-source voltage; V_{db} is drain bulk voltage; the value of R_c could be extracted from the TLP IV-curve measurement (see Fig. 1) as the following:

$$R_c = R_{diff} = \frac{V_{t1} - V_h}{I_{t2} - I_h} \quad (5)$$

Other parameters in equations (3) and (4) are explained in the Table 2.

The C_1 , C_2 , and C_3 fitting coefficients are tuned to provide the equality of simulated and measured trigger voltage V_{t1} and trigger current I_{t1} .

4. GGMOS device simulation result

DC testbench schematic is shown in the Fig. 7. The current source sweep is used, because the IV-curve of the GGMOS device is a multivalued function and has a negative resistance part from trigger voltage V_{t1} to holding voltage V_h .

Table 2. GGMOS model parameters.

Parameter	Description	Default value
V_h	holding voltage (V)	extracted from IV-curve
V_{dsat}	saturation voltage (V)	1.0
C_1	fitting coefficient	2.0
C_2	fitting coefficient	2.0
C_3	fitting coefficient	2.0
B_F	parasitic BJT forward beta	2.0
B_R	parasitic BJT reverse beta	1.5
R_{sub}	substrate resistance (Ohm)	200
R_c	ON state resistance (Ohm)	5

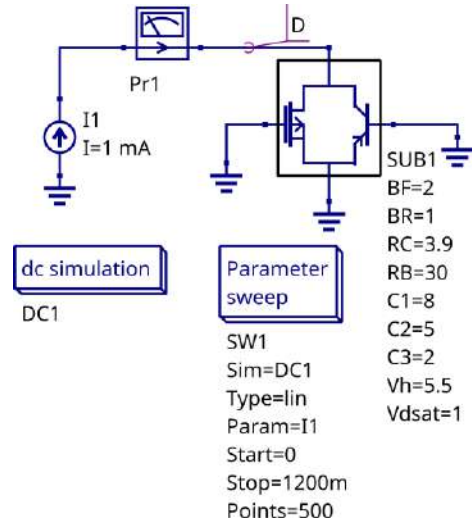


Fig. 7. GGMOS model DC testbench schematic.

The model parameters could be fitted to match the experimental TLP data (Fig. 8). The simulated and measured data have a good match at the corner points (see the Table 3). The trigger current I_{t1} has the highest mismatch. This could be explained by the low sensitivity of the used measurement setup when measuring the TLP currents around 10 mA. The mismatch of the negative resistance part slope have the following explanation. The V_h , I_h point for the measured curve lies on load line [12] that depends on the TLP setup output resistance Z_{tlp} and pulse voltage V_p on the source. The load line equations are the following:

$$I_{DUT} = \frac{V_p - V_{DUT}}{Z_{TLP}} \tag{6}$$

where V_{DUT} , I_{DUT} are the current and voltage across the device under test (DUT). The simulation testbench schematic uses the current source having $Z_{tlp} = \infty$, so the slope of the load line for the DC sweep simulation and measurement will be different.

The transient simulation testbench schematic is shown in the Fig. 9. The pulsed voltage source represents the charged line. The Pi-attenuator circuits represents the output resistance of the TLP setup. The standard 100 ns pulse width is used.

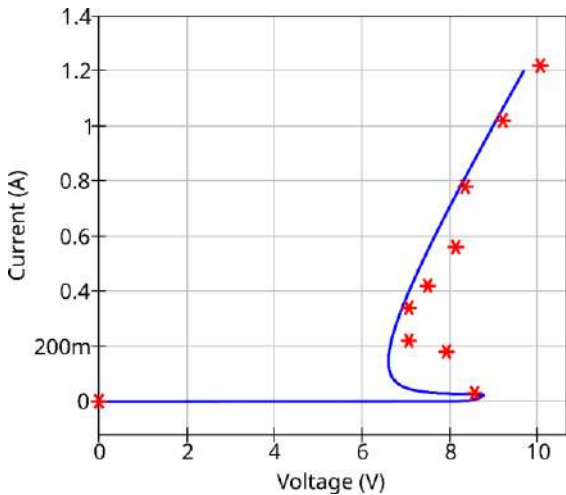


Fig. 8. GGMOS model simulated (solid line) and measured (stars) IV-curve.

Table 3. Simulation and measurement data comparison for GGMOS model.

Parameter	Simulated	Measured	Difference (%)
V_{t1}	8.780 V	8.580 V	2.3
I_{t1}	0.022 A	0.03 A	26,6
V_h	6.680 V	7.08 V	5.6
I_h	0.209 A	0.22 A	5.0
V_{t2}	9.710 V	10.100 V	3.8
I_{t2}	1.201 A	1.220 A	1.6

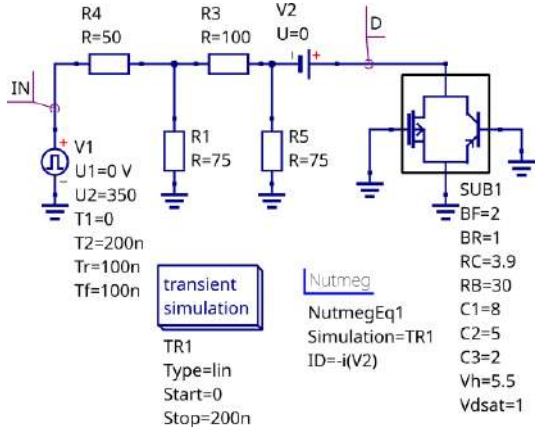


Fig. 9. GGMOS model transient testbench schematic.

The voltage V_{DUT} and current I_{DUT} waveforms across the device under test are shown in the Fig. 10. The device is triggered on the leading edge and turned off at the trailing edge of the pulse.

The device IV-curve may be extracted not only from DC simulation, but also from transient. The Fig. 11 shows the device current plotted versus device voltage. The hysteresis could be seen in this

curve. This could be explained by different turn-on and turn-off voltage of the GG MOS device. The extraction of hysteresis data for GG MOS device requires a series of TLP measurement with different pulse rise time.

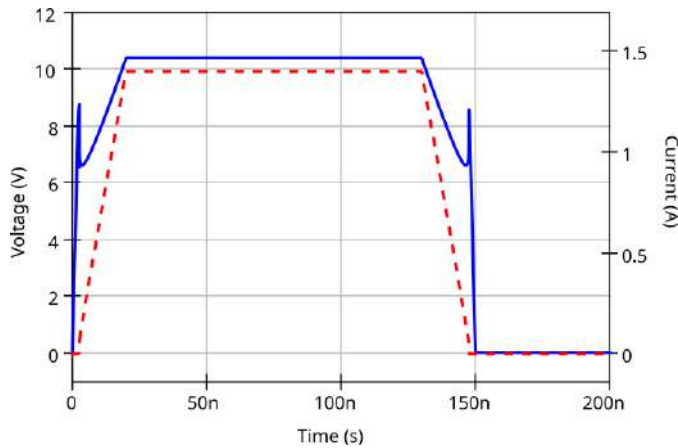


Fig. 10. Voltage (solid curve) and current (dashed curve) GG MOS device transient waveforms.

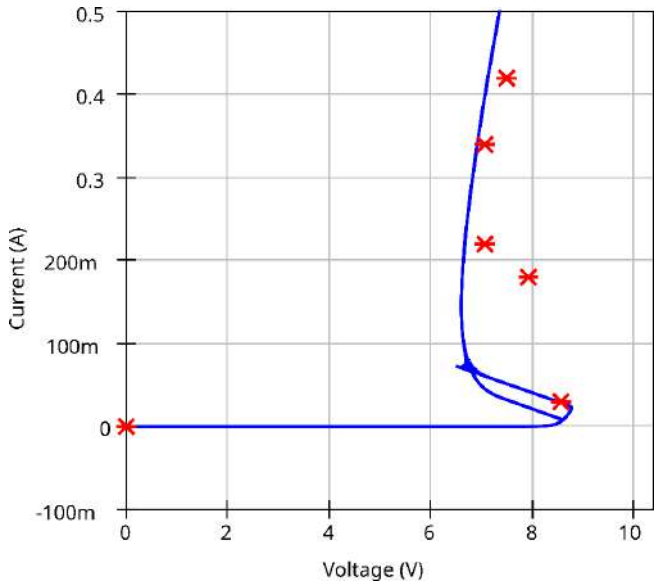


Fig. 11. Transient IV-curve of the GG MOS device (solid curve) and measured TLP quasi-static IV-curve (stars).

5. Conclusion

The macromodels for two ESD protection devices (ESD diode and GG MOS) were developed in the presented research. These devices are the most commonly used in the modern microelectronics. The presented macromodels were designed using the open-source circuit simulation tools (Qucs-S and Ngspice) that allows to use it with open process design kits (PDKs) [13]. Some measures were undertaken to improve the simulation convergence using the proposed models. The developed macromodels show a good matching of the measurement and simulation data and allow to estimate both DC and transient behavior of the ESD protection circuit. This allows to extent the SPICE simulation and estimate the ESD robustness of the IC at the early design stage. The disadvantage of

the implemented models is a sufficient complex extraction procedure that requires fine adjustment a number of the fitting coefficients.

References

- [1]. M.-F. Staniloiu, H.-S. Popescu, G. Rezmerita, I. Vlad, and M. Iordache, "SPICE model of a real Zener diode tested at room temperature," in 2022 International Conference and Exposition on Electrical And Power Engineering (EPE). IEEE, 2022. doi: 10.1109/EPE56121.2022.9959813 pp. 182–186.
- [2]. Y. P. Zhou, G. Ding, and J.-J. Hajjar, "ESD protection impact and modelling of bias-dependent series resistance in diodes," in 2019 41st Annual EOS/ESD Symposium (EOS/ESD), vol. EOS-41, 2019. doi: 10.23919/EOS/ESD.2019.8869988 pp. 1–7.
- [3]. V. Vashchenko, Y. Martynov, and V. Sinkevitch, "Electrical instability and filamentation in ggMOS protection structures," *Solid-State Electronics*, vol. 41, no. 11, pp. 1761–1767, 1997.
- [4]. Y. S. Chauhan, S. Venugopalan, N. Paydavosi, P. Kushwaha, S. Jandhyala, J. P. Duarte, S. Agnihotri, C. Yadav, H. Agarwal, A. Niknejad et al., "BSIM compact MOSFET models for SPICE simulation," in Proceedings of the 20th International Conference Mixed Design of Integrated Circuits and Systems MIXDES 2013. IEEE, 2013, pp. 23–28.
- [5]. J. Li, S. Joshi, R. Barnes, and E. Rosenbaum, "Compact modeling of on-chip ESD protection devices using Verilog-A," *IEEE Transactions on Computer-aided design of integrated circuits and systems*, vol. 25, no. 6, pp. 1047–1063, 2006. doi: 10.1109/TCAD.2005.855948
- [6]. Y. Zhou and E. Rosenbaum, "Accuracy preserving extensions to a PDK MOSFET model for ESD simulation," in 2024 46th Annual EOS/ESD Symposium (EOS/ESD), vol. 46. IEEE, 2024, pp. 1–10.
- [7]. V. Kuznetsov and V. Andreev, "ESD Induced Irreversible Degradation Processes in the Semiconductor Devices," 2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russian Federation, 2024, pp. 1-5, doi: 10.1109/REEPE60449.2024.10479847.
- [8]. V.V. Kuznetsov, V.V. Andreev Transmission Line Pulse Setup for Electrostatic Discharge Robustness Testing of the Semiconductor Devices. *Instrum Exp Tech* vol. 67, pp.268–273 (2024). <https://doi.org/10.1134/S0020441224700453>
- [9]. D.V. Andreev, V.M. Maslovsky, V.V. Andreev and A.A. Stolyarov, Modified Ramped Current Stress Technique for Monitoring Thin Dielectrics Reliability and Charge Degradation. *Phys. Status Solidi A*, vol.219: 2100400., 2022 <https://doi.org/10.1002/pssa.202100400>
- [10]. M. Brinson and D. Tomaszewski, "Advances in Qucs-S schematic capture for SPICE and Verilog-A device modelling and circuit simulation," in 2022 29th International Conference on Mixed Design of Integrated Circuits and System (MIXDES). IEEE, 2022. doi: 10.23919/MIXDES55591.2022.9838038 pp. 27–32.
- [11]. M. Brinson and V. Kuznetsov, "Qucs-0.0.19S: A new open-source circuit simulator and its application for hardware design," 2016 International Siberian Conference on Control and Communications (SIBCON), Moscow, Russia, 2016, pp. 1-5, doi: 10.1109/SIBCON.2016.7491696
- [12]. M. Magerl, B. Courivaud, C. Stockreiter, and A. Baric, "Modelling of a transmission line pulse measurement setup," in 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2018. doi: 10.23919/MIPRO.2018.8400016 pp. 84 – 88.
- [13]. P.-C. Medinceanu and M. Enachescu, "Open Skywater130nm PDK-based IP development platform: A PWM peripheral case study," in 2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON). IEEE, 2024. doi: 10.1109/MELECON56669.2024.10608632 pp. 856–861.

Информация об авторах/Information about authors

Вадим Вадимович КУЗНЕЦОВ – кандидат технических наук, доцент кафедры "Проектирование и технология электронных средств" МГТУ им. Н.Э. Баумана. Сфера научных интересов: разработка САПР для схемотехнического моделирования, разработка аналоговых микросхем, воздействие электростатического разряда на полупроводниковые приборы.

Vadim Vadimovich KUZNETSOV – Cand. Sci. (Tech.), Associate Professor of the Electronic engineering department of the Bauman Moscow State Technical University. His research interests

include open-source circuit simulation software development, analog ICs design, ESD robustness of microelectronic devices.

Владимир Викторович АНДРЕЕВ – доктор технических наук, профессор, заведующий кафедры “Проектирование и технология электронных средств” МГТУ им. Н.Э. Баумана. Сфера научных интересов: физика МДП-структур, технология производства интегральных микросхем, физические процессы в тонких плёнках диэлектриков.

Vladimir Victorovich ANDREEV – Dr. Sci. (Tech.), Full Professor, chief of the Electronic engineering department of the Bauman Moscow State Technical University. His research interests include MOS structures physics, ICs technology, thin dielectric films physics.

Семён Андреевич ЛОМАКИН – студент магистратуры кафедры “Проектирование и технология электронных средств” МГТУ им. Н.Э. Баумана. Сфера научных интересов: разработка аналоговых микросхем, проектирование элементов защиты ИМС от электростатического разряда.

Semen Andreevich LOMAKIN is a Master student of the Electronic engineering department of the Bauman Moscow State Technical University. His research interests include analog ICs design, ESD protection structures design.

DOI: 10.15514/ISPRAS-2025-37(3)-9



Опыт отраслевого применения программного обеспечения с открытым исходным кодом при аэродинамическом проектировании и проверочном моделировании обтекания летательных аппаратов

А.Л. Павлевич, ORCID: 0009-0006-8576-0923 <pavlevichal@uwca.ru>

*АО «Уральский завод гражданской авиации»,
Россия, 123308, г. Москва, пр-т Маршала Жукова, д. 1, стр. 1.*

Аннотация. Применение расчетных инструментов с открытым исходным кодом довольно распространенная практика в академической среде. Использование подобных инструментов в отрасли сопряжено с определенными рисками и сложностями. Однако, реалии и актуальные вызовы вынуждают отраслевые предприятия пересмотреть подходы к организации труда в части расчетных работ и математического моделирования, диверсифицировать имеющиеся методики и инструменты. В данной публикации описан опыт комплексного применения программного обеспечения с открытым исходным кодом в разработке авиационной техники, а также изложены преимущества и недостатки такого подхода, пути его дальнейшего развития. Рассмотрены система управления проектами OpenProject, система управления версиями Gitea, набор инструментов многодисциплинарной оптимизации OpenMDAO, параметрический редактор геометрии летательных аппаратов с возможностью расчета аэродинамических характеристик OpenVSP, программный комплекс численного моделирования течения жидкостей и газов OpenFOAM.

Ключевые слова: авиационная техника; аэродинамика; управление проектами; аэродинамическое проектирование; многодисциплинарная оптимизация; численное моделирование; методы вычислительной гидрогазодинамики CFD; информационная система OpenProject; управление версиями Gitea; геометрический редактор OpenVSP; набор инструментов OpenMDAO; решатель и утилиты OpenFOAM.

Для цитирования: Павлевич А.Л. Опыт отраслевого применения программного обеспечения с открытым исходным кодом при аэродинамическом проектировании и проверочном моделировании обтекания летательных аппаратов. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 131–146. DOI: 10.15514/ISPRAS-2025-37(3)-9.

Благодарности: Моей семье. За терпение, понимание и поддержку.

Application of open-source software solving industrial tasks in aerodynamical design and external flow simulation of the aircrafts

A.L. Pavlevich, ORCID: 0009-0006-8576-0923 <pavlevichal@uwca.ru>

JSC “Ural Works of Civil Aviation”, 1-1, Marshal Zhukov ave., Moscow, 123308, Russia.

Abstract. The article describes industrial experience of open-source software application during aircraft design process. The article focuses on workflow organization, aerodynamical design and CFD simulation using OpenProject project management system, Gitea version control system, multidisciplinary optimization framework OpenMDAO, parametric aircraft design tool OpenVSP, CFD software OpenFOAM.

Keywords: aircraft; aerodynamics; project management; aerodynamic design; multidisciplinary optimization; numerical simulation; CFD; OpenProject; Gitea; OpenVSP; OpenMDAO; OpenFOAM.

For citation: Pavlevich A.L. Application of open-source software solving industrial tasks in aerodynamical design and external flow simulation of the aircrafts. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 131-146 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-9.

Acknowledgements. To my family. For being patient, understanding and supportive.

1. Введение

Летательные аппараты представляют собой сложные технические изделия, требующие расчетного сопровождения на всех стадиях жизненного цикла (см. рис. 1). На начальных этапах проектирования проводятся работы по формированию облика и параметрическому анализу летательного аппарата, а также оценивается выполнимость технического задания. На последующих этапах проектирования уточняются характеристики летательного аппарата, формируются предложения по их улучшению и подтверждается выполнение требований технического задания. На этапе испытаний модельных и первых летных образцов осуществляется валидация полученных ранее результатов и доработка применяемых математических моделей. В процессе эксплуатации авиационной техники статистический анализ полетных данных позволяет прогнозировать изменение летных характеристик.



Рис. 1. Расчетные работы в цикле проектирования авиационной техники.
Fig. 1. Mathematical calculations and simulations in the aircraft design process.

В традиционном понимании процесса разработки авиационной техники, на предприятиях-разработчиках используется последовательный итерационный цикл расчетных работ. Работы выполняются поочередно, часто разрозненно, используя результаты предыдущих расчетов в качестве исходных данных для последующих. Основное преимущество такого подхода заключается в его определенности и предсказуемости. Однако, в условиях дефицита времени и необходимости ускорения процесса разработки, сокращения издержек из-за лишних итераций, вызванных ошибками проектирования на ранних его этапах, разрозненность и отсутствие гибкости становятся главными недостатками такого процесса разработки.

С ростом производительности и доступности вычислительной техники в конце XX века стало появляться множество технологий и инструментов для математического моделирования как узкоспециализированных, так и междисциплинарных, крайне необходимых при проектировании сложных технических изделий. Стали формироваться новые подходы к разработке, в том числе и авиационной техники, позволяющие не только ускорить процесс проектирования, но и повысить качество разработки.

В XXI веке, с развитием цифровизации в академической среде и промышленности, а также с доступностью широкополосного интернета, появились платформы и сообщества коллективной разработки программных продуктов. Это позволило развивать передовые и перспективные решения в области информационных технологий, часть из которых стала доступна как программное обеспечение с открытым исходным кодом.

В данной статье рассматривается, как программное обеспечение с открытым исходным кодом и новые подходы к разработке авиационной техники помогают решать междисциплинарные задачи аэродинамического проектирования и моделирования в АО «УЗГА», сокращая продолжительность цикла работ и повышая их качество.

2. От технического задания к облику летательного аппарата

Техническое задание к летательному аппарату в самом простом смысле – это набор требований заказчика верхнего уровня. В них указано, на какое расстояние нужно переместить груз заданной массы за определенное время, а также требования по базированию летательного аппарата и другие эксплуатационные требования. Формирование облика летательного аппарата – это начальный этап декомпозиции технического задания на технические требования нижнего уровня к планеру и системам. Это самый важный и определяющий этап разработки, на котором при помощи математического моделирования оценивается выполнимость технического задания с учетом всех имеющихся ограничений.

Процесс формирования облика летательного аппарата междисциплинарный, итерационный и очень динамичный. Требуется большое количество согласованной и взаимоувязанной информации – исходных данных и принимаемых допущений. Междисциплинарность требует плотной совместной работы нескольких подразделений предприятия-разработчика. Проектировщики, аэродинамики, двигателисты, системщики для максимально эффективной работы должны:

- сформировать проектную команду;
- поддерживать единый темп выполнения работ;
- быстро реагировать на изменения;
- поддерживать актуальность, трассируемость и взаимоувязку всех исходных данных.

Для обеспечения этих условий, помимо применения особых методологических управленческих подходов в организации и ведении работ, необходимо наличие вспомогательных информационных систем, предоставляющих общее единое рабочее пространство, систему планирования, управления работами и требованиями [1].

Существует множество различных информационных систем, поддерживающих разные методологии разработки, проектирования и управления деятельностью и предоставляющие общие рабочие пространства такие как PDM (Product Data Management – система управления данными об изделии), Wiki, RMS (Requirement Management System – система управления требованиями), PMS (Project Management System – система управления проектами) и другие. К сожалению, подобные системы всегда требуют доработки, чтобы учесть все особенности корпоративных процессов и методологий предприятия, а также реализовать интеграцию с другими системами. Доработки коммерческих информационных систем обычно проходят не быстро, в несколько этапов, и требуют значительных финансовых вложений. По этим причинам часто находят применение решения с открытым исходным кодом, которые благодаря своей гибкости позволяют адаптироваться под конкретное предприятие с его бизнес-процессами, конкретного пользователя и с минимальными трудозатратами закрыть потребность в том или ином функционале. В АО «УЗГА» для организации и выполнения работ, связанных с аэродинамическим проектированием и моделированием развернута информационная система «ВИХРЬ», в рамках которой производится:

- декомпозиция и распределение задач по исполнителям, обеспечивается трассируемость работ (наличие связей и иерархической структуры);
- формирование и накопление базы знаний проекта, контроль ревизий исходных данных и результатов, хронология деятельности;
- обеспечение прозрачности и открытости для всех участников проектной команды;
- планирование работ, ретроспектива и прогноз деятельности проектной команды.

В основе информационной системы «ВИХРЬ» лежат система управления проектами OpenProject [2] с открытым исходным кодом на языке Ruby on Rails (см. рис. 2) и система управления версиями Gitea [3] с открытым исходным кодом на языке Go (см. рис. 3).

Основные сильные стороны этих решений:

- веб-серверное приложение, работает через веб-браузер;
- не требует серьезных вычислительных ресурсов;
- легкость развертывания, кастомизации, настройки и администрирования;
- ведение деятельности по классическим методологиям типа «водопад» и по гибким методологиям типа «скрам» с возможностью их комбинирования.

К слабым сторонам этих систем можно отнести:

- не являются системами управления требованиями;
- ограниченные возможности по управлению уведомлениями;
- примитивнейшая синхронизация OpenProject с Git.

Ведение деятельности и работы проектных команд в контуре системы «ВИХРЬ» позволило перейти на краткосрочное еженедельное планирование работ, выстроить прозрачные связи и понятные процессы между сотрудниками различных подразделений конструкторского бюро. Работы по формированию облика летательных аппаратов вышли на качественно новый уровень.

3. Многодисциплинарный параметрический анализ как основа проектировочных расчетов

Проектирование летательных аппаратов начинается с определения аэродинамической схемы, компоновки, оценки массы, летно-технических и взлетно-посадочных характеристик (см. рис. 4). Методики проектирования изложены в учебных пособиях различных авторов [4, 5], трудах отраслевых институтов [6].

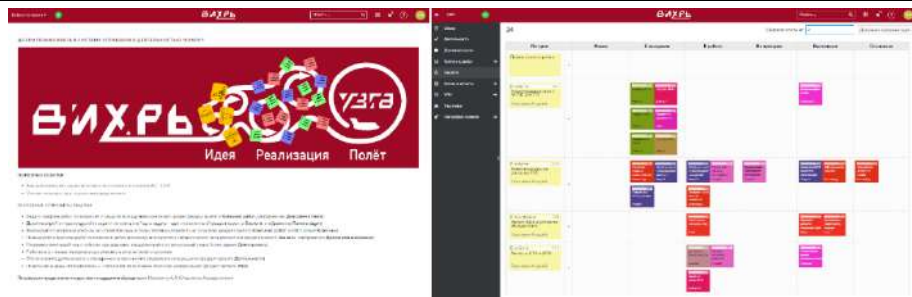


Рис. 2. Рабочее пространство информационной системы «Вихрь» на базе OpenProject.
Fig. 2. «Вихрь» workspace based on OpenProject.

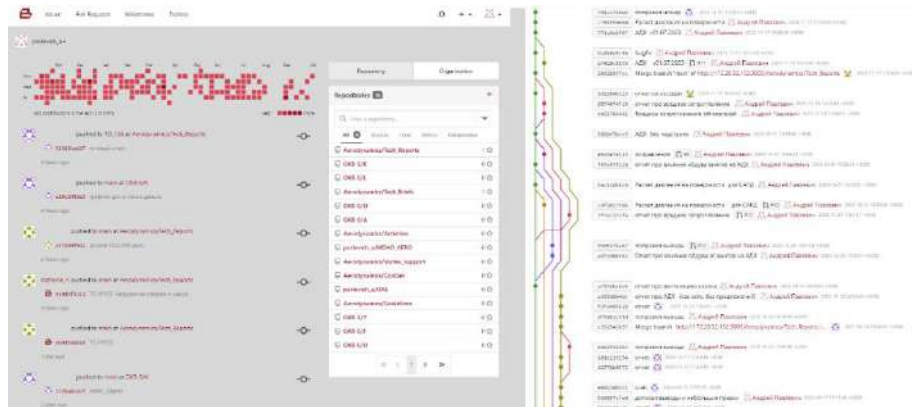


Рис. 3. Репозиторий информационной системы «Вихрь» на базе Gitea.
Fig. 3. «Вихрь» repository based on Gitea.



Рис. 4. Декомпозиция требований технического задания при формировании облика летательного аппарата с точки зрения аэродинамического проектирования.
Fig. 4. Technical requirements decomposition from aerodynamical design point of view.

В основе таких методик лежат последовательные вычисления по аналитическим и эмпирическим формулам, многие из которых получены на основании статистических данных. Однако, поскольку авиационная техника регулярно развивается, применяются новые материалы и технологии – методики со временем устаревают. Возникает необходимость использовать не статистические данные, а более адекватные физически обоснованные математические модели. Кроме того, для выполнения требований к стоимости летательного аппарата и его эксплуатации необходимо проводить финансовое моделирование.

Таким образом, задача проектирования сводится к всестороннему междисциплинарному параметрическому анализу и оптимизации целевого параметра. Решение таких задач требует не только мощных вычислительных ресурсов, но и серьезной проработки в части постановки задачи – определения входных параметров, ограничений, перекрестных и обратных связей, назначение целевых параметров, выбора подходящего алгоритма.

С ростом производительности вычислительной техники и развитием технологий компьютерного инженерного анализа, широкое распространение получили алгоритмы оптимизации и их программные реализации. В аэрокосмической отрасли за рубежом используется решение OpenMDAO [7] – набор инструментов для параметрической оптимизации с открытым исходным кодом на языке Python, созданный в NASA (см. рис. 5). В АО «УЗГА» данный инструмент успешно применяется в решении следующих задач:

- Определение оптимальной формы крыла в плане;
- Определение полетного режима с минимальным расходом топлива;
- Расчет дистанции сбалансированного взлета;
- Расчет минимальной потребной посадочной дистанции.

Применение OpenMDAO позволяет в полностью автоматическом режиме проводить расчеты по принципу «что, если...» и оценку отклика изменения целевых параметров таких как масса топлива, взлетная и посадочная дистанция, время полета и других на изменение различных конструктивных и эксплуатационных параметров, таких как размеры и масса летательного аппарата, высотно-скоростные характеристики силовой установки, характеристики покрытия взлетно-посадочной полосы и других.

Целевые функции формируются из заданных наборов математических взаимосвязанных выражений и моделей, написанных на языке Python специалистами АО «УЗГА». Среди таких моделей можно выделить несколько ключевых междисциплинарных:

- Зависимости аэродинамических характеристик летательного аппарата от формы крыла и фюзеляжа, взлетно-посадочной конфигурации;
- Зависимости высотно-скоростных и расходных характеристики винтомоторной группы двигательной установки от потребной тяги;
- Зависимости массы пустого летательного аппарата от габаритов летательного аппарата и полетных нагрузок.

Алгоритмы, заложенные в OpenMDAO, позволяют определять локальные и глобальные минимумы целевых функций, с учетом ограничений и производных функций, что позволяет специалистам АО «УЗГА» решать задачи междисциплинарной оптимизации при аэродинамическом проектировании в различных постановках и формулировках, используя один и тот же набор математических выражений и моделей.

Сильные стороны инструментов OpenMDAO:

- Простой и понятный принцип определения входных и выходных параметров математических функций, ограничений и целевых функций;
- Параметрический анализ набора аналитических зависимостей занимает не более нескольких секунд машинного времени;

- Кроссплатформенность. Возможность использования на устройствах, поддерживающих Python, в том числе и мобильных Android и iOS;
- Содержит инструменты для проверки валидности используемых математических моделей, визуальному отображению взаимосвязей через матричные диаграммы;
- Возможно решение обратных задач, задач оптимального управления;
- Поддерживается распараллеливание вычислительных процессов.

К слабым сторонам можно отнести:

- Представлены относительно простые алгоритмы оптимизации. В сложных наборах математических моделей могут быть сложности с определением глобальных минимумов.

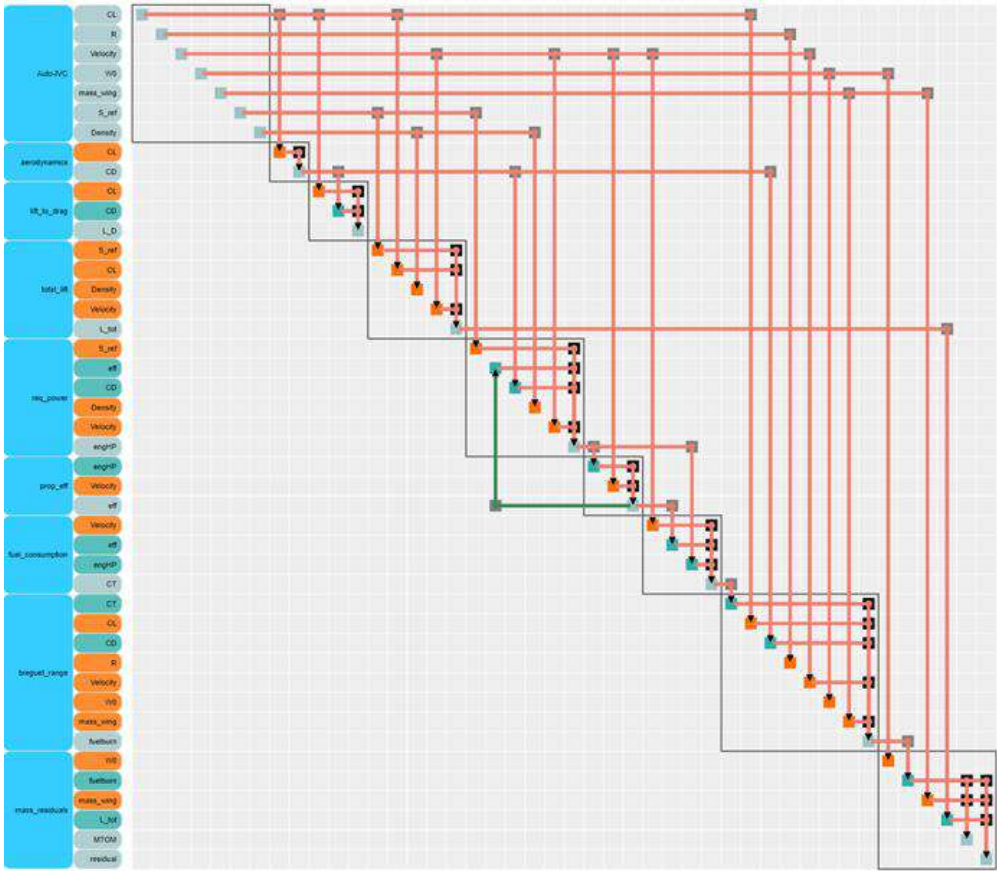


Рис. 5. Матричная диаграмма взаимосвязей математических моделей в OpenMDAO при определении оптимальной формы крыла в плане летательного аппарата.
Fig. 5. Matrix diagram showing relations between various mathematical models during optimal wing design using OpenMDAO.

Использование решения OpenMDAO на самом раннем этапе проектирования авиационной техники позволяет существенно ускорить процесс валидации требований технического задания – проверки требований на их выполнимость. Кроме того, OpenMDAO дает хорошую возможность для специалистов познакомиться с теорией оптимизации на практике, не прибегая к использованию более сложных коммерческих инструментов.

4. Аэродинамическое проектирование на ранних этапах разработки

Декомпозиция требований верхнего уровня на нижние уровни требует их уточнения и детализации. Когда определена аэродинамическая схема летательного аппарата, формируются требования к аэродинамическим характеристикам крыла. По результатам их проработки определяется форма крыла в плане и остальные его геометрические параметры, формируются требования к аэродинамическим характеристикам профиля, некоторым самолетным системам и т.д.

Обычно в процессе аэродинамического проектирования участвуют несколько подразделений конструкторского бюро. В одном подразделении формируется общий вид летательного аппарата, в другом подразделении проводится аэродинамический расчет его характеристик. Чтобы оптимизировать этот процесс, необходима автоматизация взаимодействия двух основных рабочих инструментов специалистов этих подразделений – геометрического редактора и расчетной программы.

В авиастроении широко применяются различные системы автоматизированного проектирования, трехмерные системы CAD (Computer Aided Design – системы компьютерного проектирования) и методы численного моделирования трехмерных вязких течений жидкостей и газов, известных как CFD (Computational Fluid Dynamics – вычислительная гидрогазодинамика). Применение методов CFD имеет некоторые особенности:

- Конечно-элементные методы CFD работают с трехмерной дискретизированной областью среды вокруг исследуемого объекта (так называемая «сетка»). Хотя создание «сетки» в большинстве случаев автоматизировано, оно является дополнительным звеном в цепочке процесса аэродинамического проектирования. Кроме того, для создания качественной «сетки» к геометрической модели предъявляются особые требования к ее внешним поверхностям и их сопряжениям;
- Конечно-элементные методы CFD являются одними из самых ресурсозатратных инструментов инженерного анализа. Их применение может сделать аэродинамическое проектирование «бутылочным горлышком» в общем цикле разработки.

Чтобы ускорить процесс аэродинамического проектирования довольно часто используются встроенные в CAD системы средства инженерного анализа. Обычно эти инструменты представляют из себя «черный ящик» – для удобства пользователя инструменты содержат минимум настроек, а заложенные в него математические модели не доступны для изменения. С одной стороны это упрощает процесс аэродинамического проектирования – от инженера требуется только сформулировать постановку задачи. С другой – появляется риск получения ошибочного результата из-за использования инструмента за пределами его области применения, границы которой для пользователя неочевидны.

Кроме встроенных в CAD системы инструментов инженерного анализа в конструкторских бюро авиастроительных предприятий часто используются расчетные методики собственной разработки. Для расчета аэродинамических характеристик используются различные вариации аналитических и панельных методов, которые иногда называют «расчетными инструментами пониженной точности». В дозвуковом самолетостроении наиболее распространено применение панельных методов, которое имеет свои особенности:

- Расчеты аэродинамических характеристик производятся в основном для аэродинамических плоскостей. Расчеты аэродинамических характеристик фюзеляжей, гондол и других элементов летательного аппарата могут быть очень условны;
- Упрощенное моделирование или полное отсутствие пограничного слоя;
- Эмпирические поправки на сжимаемость и трение. Невозможность достоверно оценить коэффициент лобового сопротивления при нулевой подъемной силе;
- Очень условное или полное отсутствие возможности расчета отрывного течения.

Невозможность достоверно оценить максимальный коэффициент подъемной силы;

- Особый формат исходных данных для работы с геометрией. Невозможность напрямую работать с трехмерными моделями CAD систем;
- Могут возникать сложности при расчете аэродинамических характеристик в условиях аэродинамической интерференции, влияющие на стабильность и качество результата расчета;
- Низкие требования к вычислительным ресурсам. Время моделирования одного расчетного случая может измеряться секундами на обычном офисном ПК.

Несмотря на ряд существенных ограничений, панельные методы широко применяются на ранних этапах проектирования благодаря скорости вычислений. Как показывает практика и многократные валидационные исследования – в линейном диапазоне изменения аэродинамических характеристик достоверность результатов моделирования панельными методами очень высока и подтверждается результатами не только расчетами с помощью CFD, но и экспериментальными данными. Аэродинамические характеристики в линейном диапазоне позволяют оценить характеристики управляемости и устойчивости летательного аппарата, крейсерские летно-технические характеристики.

В АО «УЗГА» на этапе аэродинамического проектирования широко применяются все описанные выше инструменты, однако, наибольший интерес представляют инструменты с возможностями параметризации исходных данных и автоматизации. Среди таких можно выделить панельные методы с открытым исходным кодом. В отрасли широко известны такие программы как XFOIL, QPROP, XFLR5, AVL, PAN AIR и другие. Отдельно следует выделить параметрический редактор геометрии летательных аппаратов с возможностью расчета аэродинамических характеристик OpenVSP [8], разработанный в NASA. Данный инструмент обладает следующими особенностями:

- Полностью параметризованный геометрический редактор исключительно летательных аппаратов. Летательный аппарат собирается из параметризованных предопределенных типовых элементов таких как фюзеляж, крыло, гондола, воздушный винт, пластина и т.п. с предопределенными макропараметрами, такими как размах, сужение, стреловидность и т.п. (см. рис. 6);
- Возможность рассчитывать аэродинамические характеристики двумя типами панельных методов VSPAERO – тонкостенный метод вихревых решеток и толстостенный панельный метод;
- Возможность рассчитывать сопротивление трения по различным эмпирическим методикам;
- Возможность рассчитывать волновое сопротивление и сверхзвуковые течения;
- Моделирование аэродинамической интерференции (местные скосы потоков), работы воздушного винта (нестационарное моделирование, метод активного диска), эффективности органов управления (элероны, рули высоты и направления);
- Имеется полный API (Application Programming Interface – интерфейс прикладного программирования) на языке C и Python;
- Возможность импорта и экспорта трехмерных геометрических моделей в нейтральных форматах STL и STEP;
- Возможность распараллеливания вычислений.

Использование в АО «УЗГА» геометрического редактора OpenVSP позволило в разы ускорить работу по созданию первичного облика летательного аппарата как трехмерной модели для последующего расчета аэродинамических характеристик.

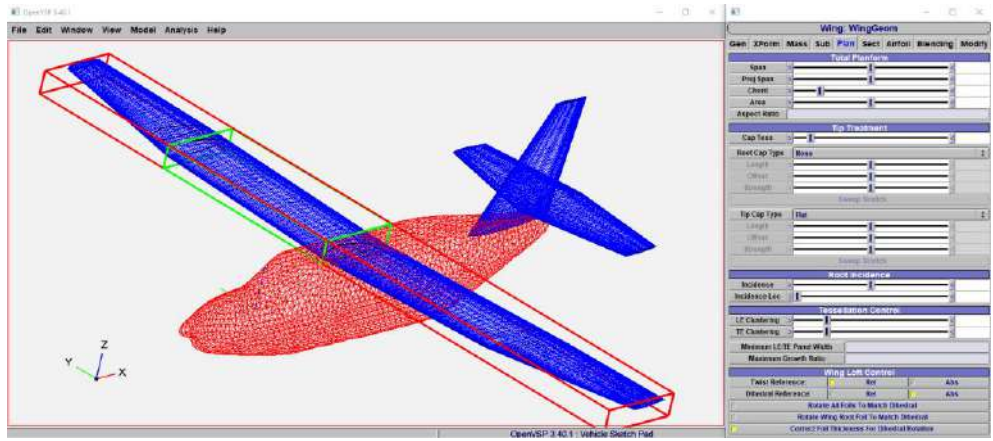


Рис. 6. Пример параметризованной модели крыла в составе самолета в редакторе OpenVSP.
Fig. 6. Example of wing parametrization using OpenVSP.

Встроенный панельный метод VSPAERO успешно прошел валидацию имеющимися на предприятии панельными методами и методами CFD, подтвердив возможность его использования для определения аэродинамических характеристик на линейном участке их изменения, расчета летно-технических характеристик, эффективности органов управления, управляемости и устойчивости (см. рис. 7).

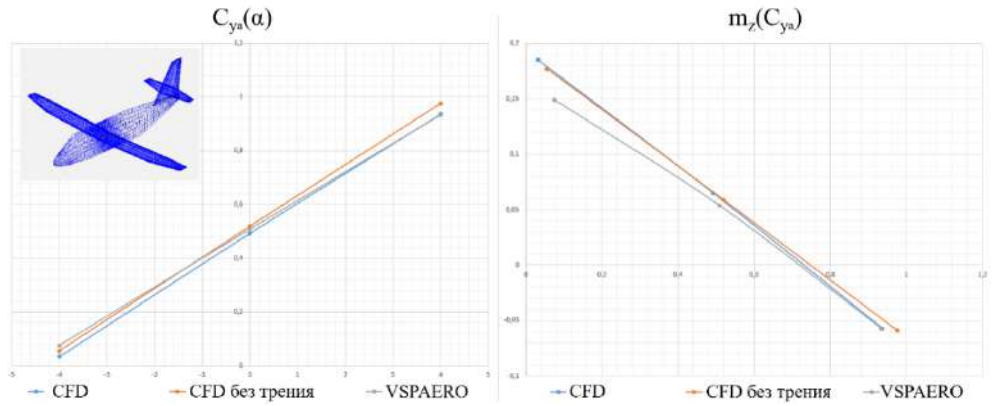


Рис. 7. Сравнение результатов расчета VSPAERO с CFD.
Fig. 7. Comparison of VSPAERO calculations with CFD simulation results.

Наличие API дает возможность использовать OpenVSP в задачах автоматизированного параметрического анализа и multidisciplinary optimization в контуре OpenMDAO, совместно с математическими моделями из других дисциплин. Подобная связка реализована и апробирована в АО «УЗГА» для решения задачи определения формы крыла в плане и режима крейсерского полета, обеспечивающих наименьший расход топлива при полете на заданную дистанцию. В оптимизационном контуре OpenMDAO используются:

- Геометрический редактор OpenVSP, использующий предопределенный шаблон летательного аппарата с изменяемыми входными параметрами типа размах секции крыла, сужение, стреловидность, корневая хорда, угол поперечного V и другие;
- Панельный метод VSPAERO, рассчитывающий аэродинамические характеристики летательного аппарата для конкретного угла атаки и заданной геометрии крыла;
- Математическая модель массы летательного аппарата, зависящая от размаха и формы

крыла в плане и нагрузок;

- Математическая модель тяговых и расходных характеристик силовой установки, зависящая от скорости, высоты и режима работы двигателя;
- Функция расчета расхода топлива на заданную дальность на определенном режиме, высоте и скорости полета;
- Условие увязки потребных и располагаемых тяг и аэродинамических характеристик летательного аппарата.

Решение подобной задачи на обычном ПК занимает не более 1 часа, при этом расчет одного полетного режима с помощью VSPAERO занимает не более 10 секунд. По итогам работы программ специалист получает упрощенную трехмерную модель летательного аппарата, облик которого и основные параметры наилучшим образом соответствуют сформулированной задаче. Экспортировав ее в нейтральный формат STEP, можно обеспечить возможность работы с геометрией в CAD-системе по дальнейшей компоновке отдельных элементов и систем, улучшению обводов, сопряжений и стыков внешних поверхностей. Основными преимуществами OpenVSP над существующими CAD-системами и их встроенными средствами инженерного анализа, CFD методами, а также методами собственной разработки являются:

- Наличие параметризованных типовых шаблонов геометрических элементов летательных аппаратов и визуальная среда трехмерного проектирования;
- Низкие требования к конфигурации ПК, возможность выполнения параллельных вычислений на многоядерных процессорах;
- Доступная документация, активное сообщество и понятные методики по расчету аэродинамических характеристик;
- API для автоматизации построения геометрии и выполнения расчетов в фоновом режиме без запуска графической части программы.

Применение инструмента OpenVSP в контуре OpenMDAO в АО «УЗГА» позволило существенно улучшить и ускорить процесс формирования и детализации облика летательного аппарата на ранних этапах проектирования.

5. Проверочное моделирование

Расчетные инструменты пониженной точности помимо основных преимуществ в виде простоты использования и скорости вычислений имеют ряд ограничений, не позволяющий в полной мере использовать их на всех этапах проектирования. Для расчета маневров, предельных нагрузок, взлетно-посадочных характеристик и некоторых других расчетных случаев требуется знать максимальный коэффициент подъемной силы и коэффициент лобового сопротивления при нулевой подъемной силе. Определение этих параметров в цикле разработки авиационной техники относят к уточняющему или проверочному моделированию с использованием «расчетных инструментов повышенной точности», к которым относят CFD методы (см. рис. 8). К сожалению, несмотря на развитие методов CFD и вычислительной техники, до сих пор не существует универсального, достоверного и приемлемого по затратам ресурсов и времени способа моделирования этих параметров. Модели турбулентности содержат эмпирические коэффициенты, а прямое моделирование турбулентности делает процесс соизмеримым по затратам с проведением испытаний в аэродинамической трубе.

Согласно обобщенным данным по итогам мероприятий AIAA (American Institute of Aeronautics and Astronautics, Американский институт аэронавтики и астронавтики) таких как High Lift Prediction Workshop [9] и Drag Prediction Workshop [10] – разброс в результатах моделирования разными программными реализациями методов CFD очень велик и не позволяет выработать универсальный достоверный подход к моделированию с высокой степенью воспроизводимости результата. Тем не менее, потребности в заблаговременном определении максимального

коэффициента подъемной силы и коэффициента лобового сопротивления при нулевой подъемной силе, а также в улучшении местной аэродинамики, влияющей на эти параметры, остаются актуальными.



Рис.8. Аэродинамическое проектирование и моделирование в цикле разработки.
Fig. 8. Aerodynamical design and simulations during aircraft design process.

В АО «УЗГА» для решения подобных задач применяются различные программные CFD комплексы – коммерческие и с открытым исходным кодом. Наиболее важное моделирование, с точки зрения требований к достоверности количественной оценки результата, дублируется разными комплексами и разными специалистами. Подобная практика распространена в особо ответственных расчетных работах в других отраслях, например, в атомной и космической отрасли. Выбор наиболее подходящего CFD комплекса для решения задач внешнего обтекания АО «УЗГА» производится по следующим критериям:

- Наличие в комплексе всех необходимых функций и математических моделей;
- Невысокий порог входа и понятные требования к квалификации специалиста;
- Стабильность процесса решения, воспроизводимость результата и устойчивость к человеческому фактору;
- Высокая производительность вычислений и скорость процесса моделирования;
- Отсутствие особых требований к вычислительным ресурсам и операционной системе, масштабируемость и стабильность параллельных вычислений.

Из всех существующих программных комплексов CFD с открытым исходным кодом в АО «УЗГА» для моделирования обтекания летательных аппаратов используется комплекс OpenFOAM в редакции ESI [11], работающий на рабочих станциях в операционной системе MS Windows 10 при помощи MinGW. Основное применение ограничено моделированием внешнего дозвукового несжимаемого установившегося обтекания без теплообмена, поэтому основной используемый решатель – simpleFoam. Для его распараллеливания по вычислительным ядрам процессора рабочей станции используется MS MPI. Для работы с файлами конфигурации используется Visual Studio Code, а для визуализации мониторов численного решения – gnuplot. Для визуализации результатов и оценки интегральных аэродинамических характеристик используется ParaView.

Опыт применения OpenFOAM в АО «УЗГА» можно обобщить следующим образом:

- Наличие скомпилированных версий под Windows с базовым набором решателем и утилит делают OpenFOAM весьма доступным для решения несложных задач

вычислительной гидрогазодинамики, таких как расчет аэродинамических характеристик летательного аппарата, обтекаемого дозвуковым несжимаемым потоком, в стационарной постановке на малых углах атаки. Решение более сложных задач, требующих применения специальных математических моделей и решателей, может быть затруднено необходимостью проведения отладочных и валидационных работ.

- Наличие опыта работы с зарубежными коммерческими CFD комплексами существенно упрощает использование OpenFOAM. При наличии на предприятии CFD решений Ansys или Siemens возможно использовать сеточные модели, созданные в них. OpenFOAM поддерживает трехмерные гексаэдрические (в том числе и ортогональные), тетраэдрические, полиэдрические, призматические сетки и их гибридные комбинации. По опыту решения задач АО «УЗГА» наиболее стабильный процесс сходимости с помощью решателя simpleFoam был получен с трехмерными гибридными полиэдрическими и призматическими сетками. Настройки решателя OpenFOAM также будут знакомы пользователям CFD комплексов Ansys и Siemens. Таким образом, основной сложностью для пользователя с опытом работы в коммерческом CFD является отсутствие графического интерфейса, стандартных универсальных настроек и некоторые отличия от привычного рабочего процесса. Тем не менее, для продвинутых пользователей и это не будет сложностью, поскольку автоматизация коммерческих CFD комплексов тоже предполагает работу со скриптами – текстовыми файлами и командами их запуска;
- OpenFOAM обладает большим количеством настроек, которые в коммерческих комплексах либо скрыты совсем, либо вынесены в раздел продвинутых. Не все комбинации настроек приводят к стабильному процессу решения даже на качественных расчетных сетках, и даже если они воспроизводят аналогичные настройки коммерческих CFD комплексов. Кроме стабильности настройки решателя оказывают влияние на результат моделирования. Отмечено систематическое завышение значений коэффициента лобового сопротивления на 15-50% в зависимости от настроек решателей по сравнению с коммерческими CFD комплексами с аналогичными настройками. Методом проб и ошибок, а также изучая опыт сообщества, были определены ряд настроек, обеспечивающих стабильный и монотонный процесс сходимости при моделировании решателем simpleFoam на неструктурированных гибридных сеточных моделях внешнего обтекания летательного аппарата. Для аппроксимации оператора градиента используется схема pointCellsLeastSquares, а для оператора дивергенции – Gauss limitedUpwindV grad(U). С коэффициентами релаксации 0,9 для давления и скорости, и 0,7 для турбулентности, установившееся решение достигается в течение 500 итераций с уровнем невязок порядка 0,000001. Расхождение получаемых коэффициентов аэродинамических сил и моментов, в диапазоне углов атаки, соответствующих линейному участку изменения подъемной силы, от экспериментальных данных и результатов коммерческих CFD комплексов незначительно. Таким образом, следует отметить достаточно сильное влияние человеческого фактора на результаты моделирования, сложности с воспроизводимостью результата – разные специалисты могут получать результаты с разбросом не менее 15%. По опыту АО «УЗГА» при использовании коммерческих CFD комплексов одного класса и метода, воспроизводимость результатов моделирования разными специалистами имеет разброс не более 5%. Следовательно, для обеспечения предсказуемой воспроизводимости результата и снижения влияния человеческого фактора, в отраслевом применении программного комплекса OpenFOAM встает острая необходимость в выработке корпоративных методических указаний и руководств для проведения моделирования. Кроме завышения коэффициента лобового сопротивления наблюдались тенденции к отрыву потока на меньших углах атаки, что, вероятно, взаимосвязанные явления. В существующих исследованиях по валидации OpenFOAM в

задачах расчета максимального коэффициента подъемной силы летательных аппаратов отмечается, что ощутимый вклад в интегральные аэродинамические характеристики могут вносить нефизичные местные отрывные течения [12]. Тем не менее, с учетом опыта АО «УЗГА» по расчету максимального коэффициента подъемной силы, полученный в OpenFOAM результат является в целом адекватным и удовлетворительным даже для моделей со слабой детализацией сеточной модели расчетной области (см. рис. 9).

- Машинное время, необходимое для выполнения одной итерации вычислений, сопоставимо с аналогичными вычислениями в коммерческих комплексах CFD. Однако, для достижения установившегося решения требуется большее количество итераций. Таким образом, общее время на решение задачи оказывается в 1,5-3 раза больше по сравнению с коммерческими CFD комплексами.
- Реализованный в решателе simpleFoam алгоритм SIMPLE не требует большого количества памяти. Рабочая станция с объемом оперативной памяти 32 Гб может быть использована для моделирования обтекания всего летательного аппарата целиком с приемлемым сеточным разрешением с элементами с удовлетворительными метриками качества (скошенность, вытянутость, коэффициент роста). Распараллеливание по ядрам процессора с помощью MS MPI происходит без особенностей. Загрузка ядер равномерная и без простоев даже с относительно простым методом декомпозиции scotch. Возможность использования OpenFOAM на ARM процессорах выглядит очень перспективным направлением применения CFD в обозримом будущем.

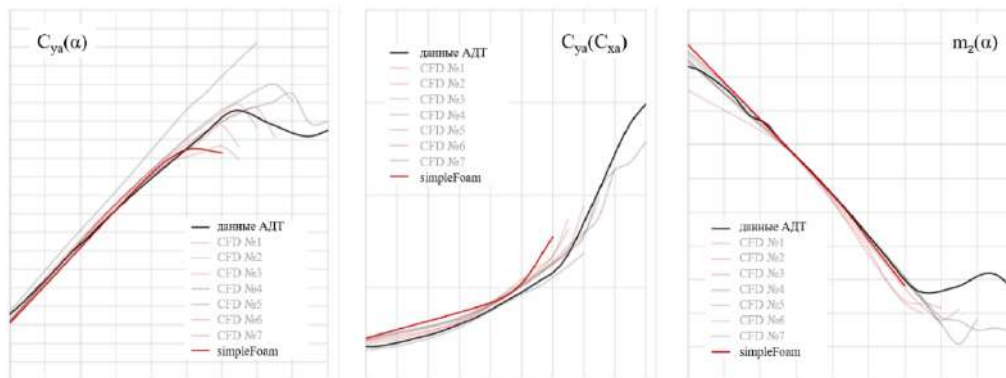


Рис. 9. Сравнение результатов моделирования OpenFOAM с результатами других CFD комплексов и примененных моделей.

Fig. 9. Comparison of CFD results obtained using various CFD solvers, including OpenFOAM.

Можно подытожить, что OpenFOAM наравне с коммерческими CFD комплексами закрывает потребности АО «УЗГА» в численном моделировании внешнего обтекания летательных аппаратов на дозвуковых скоростях. Отмеченная адекватность результатов моделирования OpenFOAM позволяет использовать его для исследования местной аэродинамики летательных аппаратов и качественного анализа их аэродинамических характеристик. Применение OpenFOAM для количественной оценки аэродинамических характеристик, особенно на больших углах атаки, требует большей сравнительной статистики по задачам, выполненным в других CFD комплексах, и валидации с экспериментальными данными. В дальнейших планах АО «УЗГА» расширять область применения OpenFOAM – моделировать сжимаемые течения, нестационарное обтекание и работу винтомоторных групп. Кроме того, использование OpenFOAM молодыми специалистами видится полезным с точки зрения лучшего понимания методов численного моделирования, более глубокого погружения в теоретические аспекты численных методов, лучшего понимания получаемого результата.

6. Заключение

Применение инструментов с открытым исходным кодом в отрасли можно рассматривать как взаимовыгодное сотрудничество между предприятиями и академическим сообществом. Предприятия получают возможность использовать новые инструменты, а сообщество получает обратную связь и новые идеи для развития своих технологий.

Открытость и доступность этих технологий позволяют выделить наиболее интересные и перспективные решения. В реалиях промышленных предприятий технологии с открытым исходным кодом не рассматриваются как альтернатива проприетарному и коммерческому программному обеспечению, а скорее, как дополнение, расширяющее области знаний и компетенций.

В АО «УЗГА», сталкиваясь с дефицитом времени и ограниченными ресурсами на разработку авиационной техники, программные комплексы с открытым исходным кодом для аэродинамического проектирования и моделирования находят свое место как дополнительные инструменты для принятия конструкторских решений. Диверсификация инструментов делает процессы разработки более устойчивыми и независимыми, повышает вероятность принятия правильного решения.

Специалисты АО «УЗГА» следят за новыми технологиями в сфере программного обеспечения с открытым исходным кодом и регулярно тестируют перспективные разработки в своей работе, чтобы повысить качество и эффективность процессов разработки авиационной техники. Так, применение информационных систем OpenProject, Gitea и инструментов параметрического анализа и multidisciplinary оптимизации OpenMDAO и OpenVSP позволило вывести работы по формированию облика летательного аппарата на качественно новый уровень. Применение в работе CFD комплекса OpenFOAM наравне с коммерческими CFD решателями дополняет специалистов информацией в части вопросов местной аэродинамики и уточнения аэродинамических характеристик летательных аппаратов.

Список литературы / References

- [1]. Павлевич А.Л. Гибкие методологии управления проектами НИОКР в ракетном двигателестроении. Труды НПО Энергомаш, том 38-39, 2022 г., стр. 300-313.
- [2]. OpenProject: <https://www.openproject.org> (дата обращения 01.11.2024).
- [3]. Gitea: <https://about.gitea.com/products/gitea> (дата обращения 01.11.2024).
- [4]. Погосян М.А., Лисейцев Н.К., Стрелец Д.Ю., Киселев В.А., Тарасов А.З., Долгов О.С., Дудар Э.Н., Тимошенко В.П., Еремин Н.А., Пеньков Е.А., Клягин В.А., Фортинов Л.Г., Хайров А.М. Проектирование самолетов/под ред. М.А. Погосяна, 5-е изд., перераб. и доп. М., Инновационное машиностроение, 2018, 864 с.
- [5]. Арепьев А.Н. Выбор проектных параметров и оценка летных характеристик пассажирских самолетов с турбовинтовыми двигателями. М., МАИ, 2005, 96 с.
- [6]. Авиация общего назначения. Рекомендации для конструкторов/ под ред. д-на, проф. В.Г. Микеладзе. ЦАГИ, 2001, 292 с.
- [7]. OpenMDAO: <https://openmdao.org> (дата обращения 01.11.2024).
- [8]. OpenVSP: <https://openvsp.org> (дата обращения 01.11.2024).
- [9]. Rumsey C.L., Slotnick J.P., Sclafani A.J. Overview and Summary of the Third AIAA High Lift Prediction Workshop. *Journal of Aircraft*, 2018, vol. 56, issue 2. DOI: 10.2514/1.C034940
- [10]. Tinoco E.N., Brodersen O.P., Keye S., Laflin K.R., Feltrop E., Vassberg J.C., Mani M., Rider B., Wahls R.A., Morrison J.H., Hue D., Roy C.J., Mavriplis D.J., Murayama M. Summary Data from the Sixth AIAA CFD Drag Prediction Workshop: CRM Cases. *Journal of Aircraft*, 2017, vol. 55, issue 4. DOI: 10.2514/1.C034409.
- [11]. OpenFOAM: <https://www.openfoam.com> (дата обращения 01.11.2024).
- [12]. Ashton N., Skaperdas V. Verification and Validation of OpenFOAM for High-Lift Aircraft Flows. *Journal of Aircraft*, 2019, vol. 56, issue 4. DOI: 10.2514/1.C034918.

Информация об авторах / Information about authors

Андрей Леонидович ПАВЛЕВИЧ – Начальник отдела аэродинамического проектирования и моделирования АО «УЗГА». Сферы научных интересов: математическое моделирование в решении инженерных задач, вычислительная гидрогазодинамика.

Andrey Leonidovich PAVLEVICH – Head of aerodynamical design and CFD simulations department, JSC “UWCA”. His research interests include mathematical modeling and simulations, fluid and gas dynamics.

DOI: 10.15514/ISPRAS-2025-37(3)-10



BERTScore для русского языка

¹ Е.П. Бручес, ORCID: 0000-0002-1055-5339 <bruches@sibnn.ai>

¹ Д.Т. Батурова, ORCID: 0009-0000-1316-4838 <baturova@sibnn.ai>

² И.Ю. Бондаренко, ORCID: 0000-0002-0457-0698 <bondarenko@sibnn.ai>

¹ Сибирские Нейросети,

Россия, 630090, г. Новосибирск, ул. Николаева, д. 12.

² Новосибирский государственный университет,

Россия, 630090, г. Новосибирск, ул. Пирогова, д. 2.

Аннотация. В работе представлено исследование по выбору наиболее релевантных векторных представлений для текстов на русском языке, которые используются в метрике BERTScore. Эта метрика используется для оценки качества сгенерированных текстов, которые могут быть получены в результате решения таких задач, как автоматическое реферирование текстов, машинный перевод и др.

Ключевые слова: метрика BERTScore; генерация текстов; автоматическая оценка.

Для цитирования: Бручес Е.П., Батурова Д.Т., Бондаренко И.Ю. BERTScore для русского языка. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 147–158. DOI: 10.15514/ISPRAS-2025-37(3)-10.

Благодарности: Исследование было выполнено при поддержке программы "Приоритет 2030" национального проекта "Наука и университеты".

BERTScore for Russian

¹ E.P. Bruches, ORCID: 0000-0002-1055-5339 <bruches@sibnn.ai>

¹ D.T. Baturova, ORCID: 0009-0000-1316-4838 <baturova@sibnn.ai>

² I.Y. Bondarenko, ORCID: 0000-0002-0457-0698 <bondarenko@sibnn.ai>

¹ Siberian Neuronets,

12, Nikolaeva st., Novosibirsk, 630090, Russia.

² Novosibirsk State University,

2, Pirogova st., Novosibirsk, 630090, Russia.

Abstract. This paper presents a study on the selection of the most relevant vector representations for texts in Russian, which are used in the BERTScore metric. This metric is used to assess the quality of generated texts, which can be obtained as a result of solving tasks such as automatic text summarization, machine translation, etc.

Keywords: BERTScore metric; text generation; automatic evaluation.

For citation: Bruches E.P., Baturova D.T., Bondarenko I.Y. BERTScore for Russian. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 147-158 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-10.

Acknowledgments. This study was carried out with the support of the “Priority 2030” program of the national project “Science and Universities”.

1. Введение

Одними из популярных задач в области обработки естественных языков являются задачи генерации текстов: это может быть машинный перевод, суммаризация текстов и другие действия. Оценка качества решения данной задачи гораздо менее формализована, чем, например, задачи классификации, так как в данном случае следует в большей мере учитывать семантику текста, чем его формальные характеристики. Это обусловлено тем, что одна и та же мысль может быть по-разному выражена в текстовой форме, порядок слов может быть изменён и так далее.

С появлением BERT-моделей была предложена метрика BERTScore [1] для оценки качества сгенерированных текстов, которая основана на схожести эталонного текста и сгенерированного текста с использованием контекстных векторных представлений слов. Основная сложность использования этой метрики состоит в выборе наиболее релевантной модели для получения векторных представлений. Так, некоторые модели были обучены для определённых задач, либо конкретных предметных областей, поэтому их использование для других текстов может быть нерелевантным и давать менее точную оценку.

В статье [1] авторами проведено исследование, направленное на выявление корреляций между оценками BERTScore с использованием различных моделей и экспертными (человеческими) оценками. Таким образом, авторы получили статистические данные о зависимостях, что легло в основу рекомендаций об использовании той или иной модели в разных языках. К сожалению, это исследование было посвящено только английскому, китайскому и турецкому языкам, что не позволяет сделать выводы о том, какая модель наиболее подходит для оценки качества текстов на русском языке, который отличается от упомянутых языков флексивностью и имеет свои грамматические и синтаксические особенности.

Настоящая работа посвящена исследованию корреляций между моделями из набора BERTScore и экспертной оценкой для выработки рекомендаций по использованию тех или иных моделей для оценки качества генерации текстов.

Работа состояла из следующих этапов:

- 1) Подготовка наборов данных, которые различаются между собой по типам задач, доменам и жанрам;
- 2) Генерация ответов моделей для данных задач;
- 3) Получение экспертной оценки для полученных ответов;
- 4) Расчет корреляций между значениями BERTScore (с векторными представлениями от разных моделей) и экспертной оценкой;
- 5) Выработка рекомендаций по использованию моделей для тех или иных задач.

2. Обзор литературы

Существует несколько групп метрик для оценки качества сгенерированных текстов: традиционные, статистические, нейросетевые и метрики, использующие языковые модели в качестве оценщиков.

Традиционные метрики включают в себя такие метрики как BLEU [2], ROUGE [3], METEOR [4] и некоторые другие.

- Метрика BLEU (BiLingual Evaluation Understudy) основана на подсчете слов и словосочетаний из сгенерированных текстов и эталонных текстов. Также она применяет штраф за краткость, чтобы избежать слишком высоких значений для кратких и неполных текстов.
- Метрика ROUGE (Recall-Oriented Understudy for Gisting Evaluation) похожа на метрику BLEU, но в её рамках считается не только метрика точности (precision), как в BLEU, но и метрика полноты (recall) и метрика F1, что позволяет обойтись без штрафа за краткость.
- Метрика METEOR (Metric for Evaluation of Translation with Explicit ORdering) также рассчитывает пересечения слов в двух текстах. Её отличие от предыдущих состоит в том, что она одновременно учитывает синонимы и однокоренные слова.
- Метрика TER (Translation Edit Rate) [5] основана на подсчете минимального числа правок, требуемых для приведения сгенерированного текста в полное соответствие наиболее близкому эталонному переводу.

Несмотря на относительную простоту в реализации этих метрик, главным их недостатком является отсутствие учёта семантики сгенерированного текста.

Статистические методы оценки качества основаны на сравнении распределений лексики двух текстов – сгенерированных автоматически и написанных людьми. Примерами таких метрик являются MAUVE [6] и GLTR [7].

Нейросетевые метрики оценки качества приобрели популярность в эпоху BERT моделей. Первая такая метрика – BertScore – была предложена в статье [1]. Данная метрика рассчитывает значение схожести (similarity score) для каждого слова в сгенерированном тексте с каждым словом в эталонном тексте. Но вместо полного посимвольного совпадения слов, используются контекстные векторные представления для каждого из них. Это позволяет учитывать семантику текстов даже при использовании разных слов, перефразировании и синтаксических различиях.

Похожая метрика MoverScore была предложена в статье [8]. Она также использует расчёт близости, используя векторные представления, но, кроме этого, также учитывает, насколько сгенерированный текст отклоняется от эталонного.

В работе [9] была предложена метрика BARTScore, основная идея которой заключается в расчёте вероятности появления каждого слова в сгенерированном тексте при условии определённого эталонного текста.

Использование языковых моделей в качестве оценщиков (LLM-as-a-Judge). С развитием больших языковых моделей, появилось такое актуальное направление исследований, как

использование моделей для автоматической оценки сгенерированных текстов. Примерами таких метрик являются TigerScore [10], INSTRUCTSCORE [11], Prometheus [12], Prometheus-2 [13] и другие. Такие метрики показали надёжность и достоверность при оценке сгенерированных текстов. Но этот подход имеет и некоторые недостатки, связанные с особенностями работы больших языковых моделей. Например, при таком использовании моделей остаётся проблема галлюцинирования [14]. Более того, у моделей могут оставаться “когнитивные” смещения, связанные с длиной контекста, с использованием определённых слов или фраз, порядком ответов и др. [15].

Подводя итог, отмечаем, что несмотря на возрастающую популярность подхода LLM-as-a-Judge, использование нейросетевых метрик остаётся востребованным, так как они вычислительно менее ресурсоёмки (по сравнению с использованием больших языковых моделей), но при этом учитывают семантику текстов в отличие от традиционных и статистических метрик.

3. Набор данных

Для определения наиболее релевантной модели, с помощью которой оценивается качество сгенерированного текста, мы собрали из открытых источников несколько наборов данных, которые различаются по задачам, предметным областям, длине текстов и другим характеристикам. Это позволит понять, есть ли необходимость в использовании разных моделей для разных задач. В данной работе мы использовали следующие наборы данных:

- 1) dialogsum-ru [16] – набор данных для задачи суммаризации диалогов. Исходный набор DialogSum [17] содержит 13 460 диалогов на английском языке на повседневные темы. Русскоязычный вариант набора был получен путём перевода текстов на русский язык с помощью Google Translate.
- 2) reviews-russian [18] – набор данных для суммаризации отзывов пользователей на отели и гостиницы. Содержит 92 текста.
- 3) ru-simple-sent-eval [19] – набор данных для задачи упрощения текстов на русском языке.
- 4) science-summarization-dataset [20] – набор данных для задачи суммаризации научных статей. Содержит 480 статей и их аннотаций из 8 различных научных областей: лингвистика, история, юриспруденция, медицина, компьютерные науки, экономика, химия.
- 5) telegram-financial-sentiment-summarization [21] – набор данных, который содержит тексты новостных сообщений из Телеграмма и их краткие содержания. Тексты в этом наборе, в основном, посвящены экономическим и политическим темам.
- 6) yandex-jobs [22] – набор данных, который содержит описания вакансий Яндекса. Задача заключается в генерации наиболее релевантного названия должности для каждой из вакансий.

В табл. 1 представлены следующие характеристики для каждого набора данных: средние длины входного и эталонного текстов, количество уникальных слов во входных и эталонных текстах. Из представленных описаний и статистик можно увидеть, что выбранные наборы данных, действительно, разнообразны по задачам, доменам и статистическим характеристикам. Таким образом, мы можем не только подобрать наиболее подходящую модель для оценки в среднем, но также и предложить наиболее релевантные модели для русского языка в зависимости от задачи и домена.

4. Модели для генерации ответов

Следующий этап данной задачи – сгенерировать ответы для описанных ранее задач. Для этого мы использовали две модели для русского языка: Gigachat Lite [23] и YandexGPT

Lite [24].

Для каждой из задач были вручную написаны подводки (prompts) для моделей, которые представлены в табл. 2. Для каждой модели была использована одна и та же подводка, без изменений.

Табл. 1. Характеристики наборов данных.
Table 1. Datasets characteristics.

Набор данных	Средняя длина входного текста	Средняя длина выходного текста	Количество уникальных слов во входных текстах	Количество уникальных слов в выходных текстах
dialogsum-ru	757	117	9 907	5 400
reviews-russian	1 390	448	5 096	2 115
ru-simple-sent-eval	137	91	15 300	20 694
science-summarization-dataset	20 155	843	112 380	13 469
telegram-financial-sentiment-summarization	313	169	26 818	20 426
yandex-jobs	925	38	8 364	504

Табл. 2. Подводки для моделей.
Table 2. Prompts for the models

Набор данных	Подводка для модели
dialogsum-ru	Сгенерируй краткий пересказ этого диалога в 1-2 предложениях:
reviews-russian	Сгенерируй краткий пересказ этого отзыва. Выделяй только важные факты. Отзыв:
ru-simple-sent-eval	Перепиши предложение так, чтобы его стало проще понимать:
science-summarization-dataset	Ниже приведена научная статья. Выдели главные факты и напиши краткое содержание этой статьи.
telegram-financial-sentiment-summarization	Напиши краткий пересказ этой новости. Выделяй только важные факты. Текст новости:
yandex-jobs	Напиши название должности для этой вакансии:

Из-за ограничений на генерацию текстов по неоднозначным темам обеих моделей (в разной степени) не для каждого текста удалось получить сгенерированный ответ. В табл. 3 представлено количество сгенерированных ответов каждой из моделей.

Таким образом, на данном этапе мы получили ответы от двух русскоязычных моделей для различных задач.

5. Модели для оценки ответов

В оригинальной статье BERTScore [1] расчёт корреляций происходил на основе экспертных

оценок: для используемых наборов данных эксперты вручную оценивали сгенерированные тексты. При том, что этот подход дает высокую точность оценки, это трудоёмкий процесс, который требует много времени и ресурсов. Ввиду этого, мы применили подход LLM-as-a-Judge, чтобы сгенерировать оценки с помощью языковой модели, не прибегая к экспертной ручной оценке.

Табл. 3. Количество сгенерированных каждой моделью текстов.

Table 3. The number of generated texts for the models.

Набор данных	GigaChat	YandexGPT
dialogsum-ru	267	1 494
reviews-russian	110	92
ru-simple-sent-eval	6 803	6 491
science-summarization-dataset	199	436
telegram-financial-sentiment-summarization	6 126	7 528
yandex-jobs	528	625

Стоит обратить внимание на проблему, которая возникает при таком использовании больших языковых моделей – это стремление моделей оценивать свои ответы выше, чем ответы других моделей [25, 26]. Поэтому в данной работе мы делали “перекрёстную” проверку – ответы YandexGPT оценивали с помощью модели GigaChat и наоборот, ответы GigaChat оценивали с помощью модели YandexGPT. Одним из основных ограничений такого подхода остаётся способность моделей галлюцинировать, то есть генерировать тексты, которые не имеют смысла или не соответствуют действительности, что может приводить к генерации моделями неверных оценок.

Для моделирования оценок от нескольких экспертов, каждая модель оценивала один и тот же текст тремя различными подводками. Как показано в статье [27], ансамбль подводок позволяет разнообразить оценки, уменьшить потенциальное смещение (то есть предотвращать генерацию одинаковых текстов для одной и той же подводки, что позволяет получать более разнообразные оценки) и увеличить надёжность таких оценок. Подводки содержали описание задачи, которую решали модели, эталонный и сгенерированный ответы, инструкцию оценить текст по 5-балльной шкале, где 1 – сгенерированный текст не соответствует эталонному, 5 – сгенерированный текст полностью соответствует эталонному. Стоит отметить, что для получения более надёжных результатов, можно использовать не только набор из различных подводок, но и включить больше языковых моделей в задачу оценки сгенерированных текстов. Это позволит минимизировать влияние галлюцинаций моделей на итоговый результат.

Таким образом, в результате этого этапа мы собрали по 3 оценки (в некоторых случаях их получилось меньше из-за отказа моделей обрабатывать входной текст, цензурирования и других причин) для каждого сгенерированного текста.

6. Корреляции

В качестве коэффициента корреляции мы использовали тот, который был использован в оригинальной статье – коэффициент корреляции Пирсона.

Для каждого сгенерированного текста нами были подсчитаны значения метрики BERTScore с использованием всех доступных в инструменте моделей, которые поддерживают русский язык, а именно:

- 1) bert-base-multilingual-cased;
- 2) xlm-mlm-100-1280;
- 3) distilbert-base-multilingual-cased;

- 4) xlm-roberta-base;
- 5) xlm-roberta-large;
- 6) facebook/mbart-large-cc25;
- 7) facebook/mbart-large-50;
- 8) facebook/mbart-large-50-many-to-many-mmt;
- 9) google/mt5-small;
- 10) google/mt5-base;
- 11) google/mt5-large;
- 12) google/mt5-xl;
- 13) google/byt5-small;
- 14) google/byt5-base;
- 15) google/byt5-large;
- 16) microsoft/mdeberta-v3-base.

Для каждой модели использовались векторные представления, полученные на каждом слое. Таким образом, мы сможем подобрать не только наиболее релевантную модель, но и слой модели, векторные представления которого дают наиболее высокий коэффициент корреляции.

Кроме того, для каждой пары “эталонный текст – сгенерированный текст” мы посчитали стандартные метрики качества генерации: BLEU, ROUGE-1, ROUGE-2, ROUGE-L и MoverScore, чтобы сравнить корреляции BERTScore-метрик и статистических метрик с “экспертной” разметкой.

Также мы предоставляем значения корреляций для наиболее релевантных векторных представлений в среднем по всем наборам данных (“Лучший вектор” с указанием модели и слоя и “Лучшая модель” с указанием лучшей модели). Полученные корреляции представлены для моделей Gigachat и YandexGPT в табл. 4 и табл. 5 соответственно.

7. Анализ результатов

Из полученных результатов мы можем видеть, что семантические оценки качества, действительно, лучше коррелируют с экспертными, чем те, что основаны на формальных и статистических признаках (такие как BLEU и ROUGE-N). Кроме того, очевидно, что для разных доменов, жанров и задач наиболее релевантными оказываются векторные представления, полученные от различных моделей. Другим важным фактором выбора векторных представлений является сама модель, с помощью которых генерируются тексты.

Тем не менее, по результатам проведённых исследований показано, что в среднем наиболее релевантными векторными представлениями для всех доменов, задач и моделей являются те, которые получены из модели *google/byt5-base*. Эта модель является “token-free”, что означает, что она оперирует “сырым” текстом (байтами или знаками) вместо того, чтобы работать с отдельными словами, как большинство других языковых моделей. Такой подход даёт ряд преимуществ: возможность обрабатывать текст на любом языке без обучения, устойчивость к шумам и так далее. Возможно, именно этим обусловлена стабильность данной модели в среднем по всем наборам данных, которые были рассмотрены в данной работе. Кроме того, эта модель является компактной по числу параметров в сравнении с другими представленными моделями.

Также мы предоставляем результаты с наиболее релевантными векторными представлениями для каждой модели в среднем по всем наборам данных и моделям в табл. 6. Из приведённой таблицы видно, что, в среднем, наиболее релевантные векторные представления для оценки качества текстов на русском языке принадлежат моделям

google/byt5-base и google/byt5-large.

Табл. 4. Коэффициенты корреляции Пирсона для различных метрик (для текстов, сгенерированных GigaChat).

Table 4. Pearson correlation coefficients for various metrics (for texts generated by GigaChat).

Набор данных	Лучший вектор	BERTScore	BLE U	ROUGE- 1	ROUGE- 2	ROUGE- L	MoverScore
telegram-financial-sentiment-summarization	microsoft/mdeberta-v3-base_layer_7	0.827	0.423	0.645	0.497	0.607	0.575
ru-simple-senteval	google/byt5-large_layer_31	0.615	0.096	0.281	0.149	0.25	0.323
science-summarization-dataset	facebook/mbart-large-50_layer_10	0.749	0.282	0.599	0.481	0.560	0.452
dialogsum-ru	facebook/mbart-large-c25_layer_11	0.447	0.158	0.236	0.114	0.247	0.182
reviews-russian	facebook/mbart-large-50-many-to-many-mmmt_layer_6	0.678	0.178	0.346	0.206	0.346	0.545
yandex-j obs	google/byt5-base_layer_6	0.433	0.078	0.192	0.165	0.192	0.382
Лучший вектор	google/byt5-large_layer_29	0.594	0.191	0.379	0.257	0.359	0.476
Лучшая модель	google/byt5-large	0.561	0.191	0.379	0.257	0.359	0.476

8. Заключение

Автоматическая оценка качества сгенерированных текстов всё ещё остаётся сложной задачей. Несмотря на то, что за годы существования этого направления были предложены различные метрики, все они обладают своими недостатками. На данный момент, метрика BERTScore кажется наиболее оптимальной, так как, с одной стороны, учитывает семантические характеристики текстов, а с другой стороны, не требует такого большого количества ресурсов, как современные большие языковые модели.

Однако значения этой метрики очень зависят от выбранной модели, языка текстов и домена. В рамках данной работы мы провели исследование на выявление наиболее релевантной модели для оценки качества сгенерированных текстов для русского языка.

Ввиду того, что получение экспертных оценок для выявления корреляций является трудоёмким процессом, мы использовали подход LLM-as-a-Judge, где в качестве экспертов выступают большие языковые модели. Остаётся важным вопрос о границах применимости такого подхода – насколько оценки моделей, действительно, соотносятся с оценками от людей. Получение ответа на этот вопрос является одним из приоритетных направлений нашего дальнейшего исследования. Кроме того, в данной работе не были рассмотрены многие модели, обученные специально для русского языка. Этот аспект также является задачей для будущего исследования.

Табл. 5. Коэффициенты корреляции Пирсона для различных метрик (для текстов, сгенерированных YandexGPT).

Table 5. Pearson correlation coefficients for various metrics (for texts generated by YandexGPT).

Набор данных	Лучший вектор	BERTScore	BLE U	ROUGE- 1	ROUGE- 2	ROUGE- L	MoverScore
telegram-financial-sentiment-summarization	microsoft/mdeberta-v3-base_layer_8	0.299	0.157	0.238	0.197	0.239	0.233
ru-simple-sentence	xlm-roberta-large_layer_16	0.224	0.068	0.079	0.051	0.067	0.201
science-summarization-dataset	distilbert-base-multilingual-cased_layer_5	0.217	0.017	0.116	0.134	0.122	0.115
dialogsum-ru	google/mt5-xl_layer_23	0.291	0.077	0.149	0.157	0.155	0.069
reviews-russian	google/mt5-xl_layer_23	0.418	0.138	0.234	0.176	0.213	0.289
yandex-jobs	google/byt5-base_layer_10	0.509	0.169	0.303	0.214	0.301	0.416
Лучший вектор	google/byt5-base_layer_10	0.284	0.096	0.171	0.138	0.166	0.214
Лучшая модель	facebook/mbart-large-50-many-to-many-mmt	0.249	0.096	0.171	0.139	0.166	0.214

Табл. 6. Результаты для каждой из моделей.

Table 6. Results for each of the models.

Модель	Кол-во параметров	Слой	Pearson
google/byt5-base	528M	8	0.442
google/byt5-large	1.23B	29	0.442
facebook/mbart-large-50-many-to-many-mmt	611M	10	0.433
facebook/mbart-large-50	611M	10	0.430
google/mt5-xl	3.7B	23	0.421
microsoft/mdeberta-v3-base	280M	8	0.421
google/mt5-large	1.2B	22	0.413
facebook/mbart-large-cc2	610M	9	0.410
xlm-mlm-100-1280	570M	15	0.403
bert-base-multilingual-cased	179M	6	0.401
xlm-roberta-base	279M	6	0.397
xlm-roberta-large	561M	16	0.389
distilbert-base-multilingual-cased	135M	3	0.376
google/mt5-small	300M	3	0.371
google/mt5-base	580M	4	0.365
google/byt5-small	300M	1	0.341

Список литературы / References

- [1]. Zhang T., Kishore V., Wu F., Weinberger K., Artzi Y. BERTScore: Evaluating Text Generation with BERT. International Conference on Learning Representations, 2020.
- [2]. Papineni K., Roukos S., Ward T., Zhu W. Bleu: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
- [3]. Lin Ch. ROUGE: A Package for Automatic Evaluation of Summaries. Text Summarization Branches Out, 2004, pp. 74–81.
- [4]. Banerjee S., Lavie A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005, pp. 65–72.
- [5]. Snover M., Dorr B., Schwartz R., Micciulla L., Makhoul J. A Study of Translation Edit Rate with Targeted Human Annotation. Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, 2006, pp. 223–231.
- [6]. Pillutla K., Swayamdipta S., Zellers R., Thickstun J., Welleck S., Choi Y., Harchaoui Z. MAUVE: measuring the gap between neural text and human text using divergence frontiers. Proceedings of the 35th International Conference on Neural Information Processing Systems, 2024, pp. 4816–4828.
- [7]. Gehrmann S., Strobelt H., Rush A. GLTR: Statistical Detection and Visualization of Generated Text. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2019, pp. 111–116.
- [8]. Zhao W., Peyrard M., Liu F., Gao Y., Meyer Ch., Eger S. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 563–578.
- [9]. Yuan W., Neubig G., Liu P. BARTScore: Evaluating generated text as text generation. Thirty-Fifth Conference on Neural Information Processing Systems, 2021.
- [10]. Jiang D., Li Y., Zhang G., Huang W., Lin B., Chen W. (2023) TIGERScore: Towards Building Explainable Metric for All Text Generation Tasks (online). Available at: <https://arxiv.org/abs/2310.00752>, accessed 28.10.2024.
- [11]. Xu W., Wang D., Pan L., Song Zh., Freitag M., Wang W., Li L. INSTRUCTSCORE: Towards Explainable Text Generation Evaluation with Automatic Feedback. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 5967–5994.
- [12]. Kim S., Shin J., Cho Y., Jang J., Longpre Sh., Lee H., Yun S., Shin S., Kim S., Thorne J., Seo M. (2023) Prometheus: Inducing Fine-grained Evaluation Capability in Language Models (online). Available at: <https://arxiv.org/abs/2310.08491>, 28.10.2024.
- [13]. Kim S., Suk J., Longpre Sh., Lin B., Shin J., Welleck S., Neubig G., Lee M., Lee K., Seo M. (2024) Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models (online). Available at: <https://arxiv.org/abs/2405.01535>, accessed 28.10.2024.
- [14]. Vu T., Krishna K., Alzubi S., Tar Ch., Faruqui M., Sung Y. (2024) Foundational Autoraters: Taming Large Language Models for Better Automatic Evaluation (online). Available at: <https://arxiv.org/abs/2407.10817>, accessed 28.10.2024.
- [15]. Koo R., Lee M., Raheja V., Park J., Kim Z., Kang D. Benchmarking Cognitive Biases in Large Language Models as Evaluators. Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 517–545.
- [16]. DIALOGSum Corpus, Available at: <https://huggingface.co/datasets/d0rj/dialogsum-ru>, accessed 11.02.2025.
- [17]. Chen Y., Liu Y., Chen L., Zhang Y. DialogSum: A Real-Life Scenario Dialogue Summarization Dataset. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 5062–5074.
- [18]. Reviews Russian, Available at: https://huggingface.co/datasets/trixdade/reviews_russian, accessed 11.02.2025.
- [19]. Sakhovskiy A., Izhevskaya A., Pestova A., Tutubalina E., Malykh V., Smurov I., Artemova E. RuSimpleSentEval-2021 Shared Task: Evaluating Sentence Simplification for Russian. Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference “Dialogue” (2021), 2021, Ch. 161, pp. 607–617.
- [20]. Tsanda A., Bruches E. (2024) Russian-Language Multimodal Dataset for Automatic Summarization of Scientific Papers (online). Available at: <https://arxiv.org/abs/2405.07886>, accessed 28.10.2024.

- [21]. Telegram Financial Sentiment Summarization dataset, Available at: <https://huggingface.co/datasets/mxlcw/telegram-financial-sentiment-summarization>, accessed 11.02.2025.
- [22]. Yandex Jobs dataset, Available at: https://huggingface.co/datasets/Kirili4ik/yandex_jobs, accessed 11.02.2025.
- [23]. GigaChat, Available at: <https://giga.chat/>, accessed 11.02.2025.
- [24]. YandexGPT, Available at: <https://ya.ru/ai/gpt-3>, accessed 11.02.2025.
- [25]. Bai Y., Ying J., Cao Y., Lv X., He Y., Wang X., Yu J., Zeng K., Xiao Y., Lyu H., Zhang J., Li J., Hou L. Benchmarking Foundation Models with Language-Model-as-an-Examiner. *Advances in Neural Information Processing Systems*, 2024.
- [26]. Liu Y., Moosavi N., Lin Ch. LLMs as Narcissistic Evaluators: When Ego Inflates Evaluation Scores. In *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 12688–12701.
- [27]. Shao W., Lei M., Hu Y., Gao P., Zhang K., Meng F., Xu P., Huang S., Li H., Qiao Y., Luo P. (2023) TinyLVLM-eHub: Towards Comprehensive and Efficient Evaluation for Large Vision-Language Models (online). Available at: <https://arxiv.org/abs/2308.03729>, accessed 28.10.2024.

Информация об авторах / Information about authors

Елена Павловна БРУЧЕС – кандидат технических наук, ведущий инженер-исследователь в компании “Сибирские Нейросети”. Сфера научных интересов: обработка естественных языков, извлечение информации, языковые модели, машинное обучение.

Elena Pavlovna BRUCHES – Cand. Sci. (Tech.) in Computer Sciences and a leading engineer-researcher in Siberian Neuronets. Research interests: natural language processing, information extraction, language models, machine learning.

Дари Тимуровна БАТУРОВА – бакалавр по направлению «Мехатроника и робототехника» Новосибирского государственного университета, разработчик-исследователь в компании «Сибирские нейросети». Сфера научных интересов: глубокое обучение, нейронные сети, обработка естественного языка, машинный перевод.

Dari Timurovna BATUROVA – Bachelor's degree in Mechatronics and Robotics at Novosibirsk State University, developer-researcher at Siberian Neural Networks company. Research interests: deep learning, neural networks, natural language processing, machine translation.

Иван Юрьевич БОНДАРЕНКО – научный сотрудник лаборатории прикладных цифровых технологий Новосибирского государственного университета, старший преподаватель кафедры фундаментальной и прикладной лингвистики Новосибирского государственного университета, сооснователь компании "Сибирские нейросети". Сфера научных интересов: машинное обучение, нейронные сети, компьютерная лингвистика, распознавание речи.

Ivan Yurevich BONDARENKO is a researcher in the Laboratory of Applied Digital Technologies and a senior lecturer in the Department of Fundamental and Applied Linguistics at Novosibirsk State University. He is also the co-founder of the Siberian Neural Networks company. Research interests: machine learning, neural networks, computational linguistics, speech recognition.

DOI: 10.15514/ISPRAS-2025-37(3)-11



Enhanced Text Classification Using DistilBERT with Low-Rank Adaptation: A Comparative Study

B.D. Abodo Eloundou, ORCID: 0009-0009-1845-5867 <abodobrice@hotmail.com>

W. Quanyu, ORCID: 0009-0000-1059-7312 <272561@edu.itmo.ru>

*AI Talent Hub, ITMO University,
49, Kronverksky pr., Saint Petersburg, 197101, Russia.*

Abstract. In this article, we delve into the task of sentiment analysis applied to news articles covering sanctions against Russia, with a specific focus on secondary sanctions. With geopolitical tensions influencing global affairs, understanding the sentiment conveyed in news about sanctions is crucial for policymakers, analysts, and the public alike. We explore the challenges and nuances of sentiment analysis in this context, considering the linguistic complexities, geopolitical dynamics, and data biases inherent in news reporting. Leveraging natural language processing techniques and machine learning models, including Large Language Models (LLM), 1D Convolutional Layer (Conv1D), and Feed-Forward Networks (FFN), we aim to extract sentiment insights from news articles. Our analysis provides valuable perspectives on public opinion, market reactions, and geopolitical trends. Through our work, we seek to illuminate the sentiment landscape surrounding sanctions against Russia and their broader implications.

Keywords: text classification; sentiment analysis; sanctions; distilbert; lora; conv1d; feed-forward networks.

For citation: Abodo Eloundou B.D., Quanyu W. Enhanced Text Classification Using DistilBERT with Low-Rank Adaptation: A Comparative Study. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025, pp. 159-170. DOI: 10.15514/ISPRAS-2025-37(3)-11.

Acknowledgements. This study was supported by the AI Talent Hub of ITMO University and the Russian Academy of Foreign Trade. The Russian Academy of Foreign Trade provided the dataset for the research.

Расширенная классификация текста с помощью модели DistilBERT с адаптацией низкого ранга LoRa: сравнительное исследование

Б.Д. Абодо Элунду, ORCID: 0009-0009-1845-5867 <abodobrice@hotmail.com>

В. Цюаньюй, ORCID: 0009-0000-1059-7312 <272561@edu.itmo.ru>

*AI Talent Hub, Университет ИТМО,
Россия, 197101, г. Санкт-Петербург, Кронверкский пр., д. 49.*

Аннотация. В данной статье мы рассматриваем задачу анализа тональности новостных статей, посвященных санкциям против России, с особым вниманием к вторичным санкциям. С учетом геополитической напряженности, влияющей на мировые события, понимание тональности новостей о санкциях имеет важное значение для политиков, аналитиков и широкой общественности. Мы изучаем вызовы и особенности анализа тональности в данном контексте, учитывая языковые сложности, геополитическую динамику и предвзятость данных в новостных материалах. Используя методы обработки естественного языка и модели машинного обучения, включая большие языковые модели (LLM), одномерные сверточные слои (Conv1D) и полно связанные нейросети (FFN), мы стремимся извлечь информацию о тональности из новостных статей. Наш анализ предоставляет ценные сведения об общественном мнении, реакции рынков и геополитических тенденциях. В рамках данной работы мы стремимся осветить тональный ландшафт, связанный с санкциями против России, и их более широкие последствия.

Ключевые слова: классификация текста; анализ настроений; санкции; модель distilbert; адаптация низкого ранга lora; одномерные сверточные слои conv1d; полно связанные нейросети feed-forward networks.

Для цитирования: Абодо Элунду Б.Д., Цюаньюй В. Расширенная классификация текста с помощью DistilBERT с адаптацией LoRa: сравнительное исследование. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 159–170 (на английском языке). DOI: 10.15514/ISPRAS-2025-37(3)–11.

Благодарности. Это исследование было проведено при поддержке AI Talent Hub Университета ИТМО и Российской академии внешней торговли. Российская академия внешней торговли предоставила набор данных для исследования.

1. Introduction

The ever-evolving geopolitical landscape presents policymakers, analysts, and the public with a constant need to understand the sentiment surrounding critical global events. In the case of sanctions against Russia, particularly secondary sanctions that impact various countries, gauging public perception is crucial. Sentiment analysis emerges as a powerful tool in this scenario, offering insights into the emotional undercurrents of news articles covering these sanctions.

Recent research by Kim et al. (2021) emphasizes the importance of understanding the political dynamics of sanctions [1]. Their comparative study of sanctions against Russia and North Korea highlights the varied effects and complexities involved. Sentiment analysis can contribute to this understanding by revealing public opinion towards these dynamics.

This article explores the application of sentiment analysis, specifically leveraging a Large Language Model (LLM), to analyze news articles focused on sanctions against Russia. We delve into the complexities of sentiment analysis within this context, acknowledging the challenges posed by linguistic nuances (e.g., sarcasm, ambiguity) [2], the fluid nature of geopolitics [3], and potential biases inherent in news reporting [5]. Through the application of natural language processing techniques and machine learning models, we aim to extract valuable sentiment-based insights from a vast corpus of news articles. This analysis will provide a deeper understanding of public opinion, market reactions, and broader geopolitical trends surrounding these sanctions. Ultimately, this article

seeks to shed light on the sentiment landscape and its potential implications for the effectiveness and impact of sanctions against Russia.

2. Related Works

The field of text classification has seen significant advancements over the years, particularly with the advent of deep learning models. Traditional machine learning methods, such as Naive Bayes and Support Vector Machines (SVM), laid the foundation for text classification tasks [6]. However, with the rise of neural networks, models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have shown superior performance due to their ability to capture complex patterns in text data [7, 8].

2.1 Classification

Early works in text classification employed methods like Naive Bayes, SVM, and logistic regression, which relied heavily on feature engineering [9]. With the emergence of deep learning, CNNs and RNNs have been extensively used for text classification. These models automatically learn features from raw text, reducing the need for manual feature extraction [10, 11]. The introduction of Transformer models, particularly BERT and its variants, has revolutionized text classification tasks. These models use attention mechanisms to capture contextual relationships in text more effectively than traditional RNNs [12, 13]. DistilBERT, a lighter version of BERT, maintains its performance while being more efficient [14]. The use of Low-Rank Adaptation (LoRA) further enhances its adaptability and performance on specific tasks [15].

2.2 Challenges

Despite significant progress, there are several challenges associated with existing solutions. Transformer models, while powerful, are computationally expensive and require significant resources for training and inference [16]. Many models tend to overfit specific datasets, leading to poor generalization on unseen data [17]. Additionally, deep learning models, including transformers, often act as “black boxes”, making it difficult to interpret their decisions [18].

2.3 Distinction

This study aims to address some of the limitations of existing approaches by leveraging DistilBERT with Low-Rank Adaptation (LoRA) for text classification. By using DistilBERT, we achieve a balance between model performance and computational efficiency [14]. Incorporating LoRA allows for more efficient fine-tuning on specific tasks, improving performance without substantial increases in computational cost [15]. Furthermore, we provide a thorough comparison of DistilBERT with traditional Conv1D and Feedforward Neural Network (FFN) models, highlighting the strengths and weaknesses of each approach. In conclusion, our work builds upon the strengths of previous research while addressing some of their key limitations, providing a comprehensive evaluation of modern text classification techniques.

2.4 Economic Impact of Sanctions

The economic impact of sanctions has been extensively studied, with research highlighting both the intended and unintended consequences on the target nation's economy. Hufbauer et al. (2007) provide a comprehensive analysis of economic sanctions, discussing their effectiveness and the various mechanisms through which they influence the targeted economy [19]. Their work illustrates how sanctions can lead to trade reductions, financial instability, and long-term economic decline. In the specific context of sanctions against Russia, Dreger et al. (2016) analyze the economic performance of Russia under sanctions and find significant impacts on trade patterns and financial markets, demonstrating the broad and profound economic consequences of such measures [20].

2.5 Political Dynamics and Framing Analysis

Understanding the political dynamics of sanctions is crucial for comprehending their overall effectiveness and reception. Kim et al. (2021) provide a comparative analysis of sanctions against Russia and North Korea, highlighting the complex political dynamics and varying effects of these sanctions [1]. Their research underscores the importance of considering the political context when evaluating the impact of sanctions. Additionally, the framing of sanctions in news media plays a vital role in shaping public perception and political discourse. Entman (2007) discusses the concept of framing bias and how media framing can influence public opinion and policymaking processes [21]. By analyzing the media narratives around sanctions, researchers can gain insights into how different framings affect public sentiment and political outcomes.

2.6 Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) has emerged as a promising technique in the field of machine learning, particularly for efficiently fine-tuning large language models. Hu et al. (2021) introduce LoRA as a method that reduces the number of trainable parameters by decomposing weight matrices into low-rank factors, thereby improving training efficiency without significantly compromising model performance [15]. This technique is particularly advantageous in scenarios with limited computational resources, enabling the practical application of large-scale models in tasks such as sentiment analysis. LoRA allows for the adaptation of large pre-trained models to specific tasks with minimal computational overhead, making it a valuable tool in the NLP toolkit.

2.7 Feed-Forward Networks (FFN)

Feed-Forward Networks (FFN) are a fundamental component of many neural network architectures and are widely used in various machine learning tasks due to their simplicity and effectiveness. Goodfellow et al. (2016) provide a detailed overview of FFNs, describing their architecture, training processes, and applications [4]. FFNs consist of multiple layers of interconnected neurons, where each layer transforms the input data through a series of weighted linear combinations followed by non-linear activations. In sentiment analysis, FFNs are often employed for their ability to model complex patterns in textual data, capturing the relationships between words and their contextual meanings to produce accurate sentiment predictions.

2.8 Convolutional Neural Networks (Conv1D)

Convolutional Neural Networks (Conv1D) are particularly effective in natural language processing tasks due to their ability to capture local patterns in sequential data. Kim (2014) demonstrates the application of Conv1D for sentence classification, showcasing its potential to extract meaningful features from text data [10]. Conv1D models apply convolutional filters to the input text, enabling the identification of n-gram features and local dependencies. This makes them well-suited for tasks like sentiment analysis, where understanding the context and relationships between words is crucial. The use of Conv1D in text classification has shown promising results, highlighting its role as a powerful tool in the NLP landscape.

3. Experiment Setup and Methodology

3.1 Dataset Description

To evaluate the effectiveness of DistilBERT with Low-Rank Adaptation for sentiment analysis in news articles about sanctions against Russia, we utilized a dataset compiled by the Russian Foreign Trade Academy. This dataset comprises text data related to various geopolitical and economic events concerning sanctions on Russia, and each article is labeled for sentiment analysis. Articles were sourced from reputable news outlets and encompass a range of languages, including Italian, German,

Chinese, Japanese, Spanish, and Russian. Non-Russian articles were translated into Russian, creating a monolingual dataset for analysis.

To be included in the dataset, each article needed to explicitly address matters related to Russia post-February 2022, with a focus on politics, policies, and economic issues. The dataset comprises 583 articles divided into three sentiment categories: 218 marked as positive, 185 as negative, and 180 as neutral. Positive articles highlight ineffective or disruptive sanctions, new trade opportunities, or potential advantages for Russia; neutral articles are objective and lack value judgments; and negative articles discuss the adverse effects of sanctions on Russia or note new restrictive measures imposed by other countries.

To enhance the dataset, a back-translation augmentation technique was applied. This involved translating Russian texts into intermediate languages, such as English, and then back into Russian. This method helped diversify the dataset by generating additional textual variations, which were then labeled according to the original sentiment categories.

The augmented dataset provides a robust basis for training and evaluating sentiment analysis models, facilitating a comprehensive examination of model performance on real-world data related to sanctions and their geopolitical impact.

3.2 Preprocessing Steps

The raw text data underwent several preprocessing steps to ensure consistency and quality. These steps included:

- Converting text to lowercase.
- Removing HTML tags, email addresses, digits, punctuation, and newlines.
- Collapsing multiple spaces into a single space.
- Normalizing specific Russian characters.

3.3 Model Architecture and Training

We implemented three distinct models for comparison: DistilBERT with Low-Rank Adaptation (LoRA), Convolutional Neural Networks (Conv1D), and Feed-Forward Networks (FFN). Each model was meticulously designed to capture different aspects of textual data for effective sentiment classification.

DistilBERT with Low-Rank Adaptation (LoRA): DistilBERT, a distilled version of BERT, was fine-tuned using the Low-Rank Adaptation (LoRA) technique to enhance its adaptability for the sentiment analysis task. LoRA reduces the number of trainable parameters by decomposing weight matrices into low-rank factors, thereby improving training efficiency without significantly compromising performance.

Model Architecture: The DistilBERT model architecture utilized in this study is as follows:

- **Embeddings:**
 - **Word Embeddings:** Embedding layer with vocabulary size of 119,547 and embedding dimension of 768.
 - **Position Embeddings:** Embedding layer with a maximum sequence length of 512 and embedding dimension of 768.
 - **Layer Normalization:** Applied with $\epsilon = 1e-12$.
 - **Dropout:** Applied with a probability of 0.1.
- **Transformer Layers:**

Six Transformer blocks, each comprising:

- **Multi-Head Self-Attention:** Includes dropout with $p = 0.1$, and linear layers for queries (q lin), keys (k lin), values (v lin), and output (out lin), each with input and output dimensions of 768.
- **Layer Normalization:** Applied before attention and feed-forward networks.
- **Feed-Forward Network (FFN):** Consists of two linear layers with a GELU activation function between them, expanding from 768 to 3072 dimensions and back to 768.
- **Dropout:** Applied with a probability of 0.1 after each sub-layer.
- **Pre-Classifer:** Linear layer transforming 768 features to 768.
- **Classifier:** Linear layer transforming 768 features to 3 classes (positive, negative, neutral).
- **Dropout:** Applied with a probability of 0.2 before the classifier.

Hyperparameters: The DistilBERT model was trained with the following hyperparameters:

- **Learning Rate:** 4.836×10^{-5} .
- **Batch Size:** 32.
- **Number of Training Epochs:** 7.
- **LoRA Parameters:**
 - **LoRA Rank (r):** 8.
 - **LoRA Alpha:** 32.
 - **LoRA Dropout:** 0.1.
- **Weight Decay:** 0.08.

LoRA Configuration: The LoRA configuration applied to DistilBERT is defined as follows:

- **PEFT Type:** LoRA.
- **Task Type:** Sequence Classification (SEQ CLS).
- **Target Modules:** attention.k_lin, attention.q_lin, attention.v_lin.
- **Fan In Fan Out:** False.
- **Bias:** None.
- **Use RSLora:** False.
- **Initialization:** LoRA weights initialized.
- **Runtime Configuration:** Ephemeral GPU offload disabled.

Training Procedure: The model was trained using the following procedure:

1. **Data Loading:** The preprocessed dataset was loaded and split into training and validation sets.
2. **Training:** The model was trained for 7 epochs with a batch size of 32.
3. **Evaluation:** After each epoch, the model's performance was evaluated on the validation set to monitor for overfitting.
4. **Checkpointing:** The best-performing model based on validation F1-score was saved for final evaluation.

Convolutional Neural Networks (Conv1D): Convolutional Neural Networks (Conv1D) have proven to be highly effective in natural language processing tasks by capturing local patterns and dependencies in sequential data. Kim (2014) demonstrated the application of Conv1D for sentence classification, highlighting its capability to extract meaningful n-gram features from text [10]. In our study, the Conv1D model architecture was designed with multiple convolutional layers, each employing a set of filters to capture different n-gram representations.

The model incorporates a max-pooling layer following each convolutional layer to reduce dimensionality and highlight the most salient features. Additionally, dropout layers were implemented to prevent overfitting and enhance generalization.

The final layers consist of a fully connected dense layer leading to the softmax activation function for classification.

Based on our experiments, the Conv1D model utilized the following hyperparameters:

- **Batch Size:** 32.
- **Number of Epochs:** 14.
- **Optimizer:** Adam.
- **Dropout Rate:** 0.5.
- **Filter Sizes:** 256, 128, 128.
- **The kernel size:** 7, 5, 3.
- **For activation:** relu.

These settings were chosen based on preliminary grid search experiments aimed at optimizing model performance while maintaining computational efficiency. The Conv1D model demonstrated robust performance in capturing local textual features, contributing significantly to the overall sentiment classification accuracy.

Feed-Forward Network (FFN): The Feed-Forward Network (FFN) model was developed to provide a baseline comparison against more complex architectures like DistilBERT and Conv1D.

Model Architecture: The FFN model architecture utilized in this study is as follows:

- **Input Layer:** Accepts input sequences of tokens with a maximum length of max_len.
- **Embedding Layer:**
- **Word Embeddings:** Embedding layer with a vocabulary size of vocab_size and an embedding dimension of 16.
- **Flatten Layer:** Flattens the embedded input to a one-dimensional vector.
- **Dense Layers:**
 - **Dense Layer 1:** 128 neurons with ReLU activation.
 - **Dense Layer 2:** 128 neurons with ReLU activation.
 - **Dense Layer 3:** 128 neurons with ReLU activation.
- **Dropout Layer:** Applied with a dropout rate of 0.5 to prevent overfitting.
- **Output Layer:** Dense layer with 3 neurons (corresponding to the sentiment classes) without activation (logits).

Hyperparameters: The FFN model was trained with the following hyperparameters:

- **Batch Size:** 64.
- **Number of Training Epochs:** 14.
- **Optimizer:** Adam.
- **Loss Function:** Sparse Categorical Crossentropy.
- **Metrics:** Sparse Categorical Accuracy.

Training Procedure: The FFN model was trained using the following procedure:

1. **Data Loading:** The preprocessed dataset was loaded and split into training and validation sets.
2. **Training:** The model was trained for 14 epochs with a batch size of 64.

- 3. **Evaluation:** After each epoch, the model’s performance was evaluated on the validation set to monitor for overfitting.
- 4. **Checkpointing:** The model achieving the highest validation accuracy was saved for final evaluation.

To assess the performance of each model, we utilized standard evaluation metrics for text classification tasks, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the models’ effectiveness in correctly identifying the sentiment of news articles.

4. Results and Discussion

4.1 Performance Comparison

Our experimental results indicate that each model – FFN, Conv1D, and DistilBERT – demonstrates varying strengths across the different evaluation metrics. *Table 1* compares precision, recall, f1-score, and accuracy for each model, providing a detailed view of their performance on sentiment classification.

Table 1. Performance Metrics of FFN, Conv1D, and Distil-BERT Models.

Model (Weighted Avg)	Precision	Recall	F1-score	Accuracy
FFN	0.84	0.796	0.80	0.796
Conv1D	0.80	0.78	0.787	0.78
Distil-BERT	0.83	0.82	0.82	0.82

Weighted averages are calculated based on the support for each class.

4.2 Analysis

Feedforward Neural Network (FFN): The FFN model achieved a weighted average precision of 0.8404, a recall of 0.7969, and an f1-score of 0.8045. It performed particularly well on neutral sentiments, achieving high precision and f1-scores for both positive and neutral categories. This indicates that the FFN model has strong generalization capabilities, effectively distinguishing between sentiment categories with good balance.

Conv1D Model: The Conv1D model demonstrated a weighted average precision of 0.8043, recall of 0.7812, and an f1-score of 0.7873. It showed robust performance in recognizing the positive sentiment category but faced challenges with neutral sentiments. This suggests that Conv1D may struggle to fully capture nuances between neutral and other sentiments, potentially due to its reliance on local text patterns.

DistilBERT Model: The DistilBERT model achieved the highest accuracy among the three models at 82%, with a weighted average precision of 0.83 and f1-score of 0.82. Notably, it performed best on the negative category with a precision of 0.89 and an f1-score of 0.88, indicating it is particularly effective at capturing this sentiment. While DistilBERT exhibited strong recall across all categories, it showed a slightly lower precision for positive sentiments (0.70), suggesting room for further fine-tuning in this category.

Comparison and Insights: Across all three models, FFN and DistilBERT exhibited higher overall precision and f1-scores compared to Conv1D. While FFN achieved strong performance on all sentiment categories, DistilBERT excelled in classifying negative sentiments, which may make it particularly useful for tasks where identifying negative sentiment is critical. Conv1D, while competitive, demonstrated lower performance on neutral sentiments, which highlights the importance of selecting models based on the specific sentiment needs of a task.

The confusion matrices indicate that FFN and DistilBERT were more effective in distinguishing between all sentiment classes, with FFN showing a slight advantage in generalization across categories. DistilBERT, on the other hand, demonstrated superior precision in identifying negative sentiment, which suggests its suitability for contexts where distinguishing between negative and other sentiments is particularly valuable.

4.3 Comparison with Baseline Models

To assess the effectiveness of our models, the performance of the FFN, Conv1D, and DistilBERT models was compared to a simple baseline model. Both the FFN and DistilBERT models outperformed the baseline in all evaluation metrics, demonstrating the advantages of using more advanced architectures for sentiment analysis tasks. DistilBERT, in particular, showed competitive performance with less computational overhead due to its reduced model size.

4.4 Discussion

Alignment with Ground Truth Data: Our model's performance was evaluated using standard metrics, including accuracy, precision, recall, and F1-score, which provide quantitative measures of how well the model's predictions align with the annotated sentiment labels in the dataset, representing the "ground truth." Confusion matrices for each model (FFN, Conv1D, and DistilBERT) were analyzed to gain insight into areas of strong performance and potential challenges. For example, the DistilBERT model exhibited high precision and recall for negative sentiment, indicating a strong alignment with the ground truth for this category.

Additionally, misclassification instances were examined to identify recurring error patterns, such as neutral articles being incorrectly classified as positive or negative. These misalignments suggest areas for improvement in fine-tuning or feature extraction, as they often reflect challenges in handling nuanced linguistic cues.

To ensure the dataset's annotations remained consistent, each article was labeled by students of the Russian Foreign Trade Academy based on specific guidelines for categorizing sentiments as positive, negative, or neutral. Cross-validation was employed to confirm that the models generalized effectively across different data subsets, minimizing overfitting to specific patterns or biases present in the training set.

Real-World Scenario Performance: In practical applications, our model could be deployed on live news streams or a constantly updated corpus of articles, enabling near real-time sentiment analysis for geopolitical events. Given that it was trained on diverse topics related to sanctions against Russia, it is expected to perform reliably in detecting sentiment in similar contexts. Specifically, the DistilBERT model's high accuracy for negative sentiment suggests it would be particularly valuable in scenarios that require identifying critical developments, such as new sanctions, making it useful for policymakers, financial analysts, and media organizations.

However, real-world texts often contain linguistic nuances such as sarcasm, idiomatic expressions, and culturally specific references that may pose challenges for model accuracy. Although the model shows strong alignment with the ground truth in the dataset, certain cases may affect its performance due to these nuances. To enhance real-world accuracy, future work could integrate sentiment-shifting indicators (e.g., sarcasm markers) or develop a sentiment lexicon tailored to news. Additionally, since the training data includes both translated and native Russian texts, subtle differences from translation might impact real-world Russian news analysis. Expanding the dataset with more native language sources could help capture these linguistic intricacies.

Adaptation to Evolving Contexts: Given that geopolitical events and news content are dynamic, the model's performance may vary over time as the nature of discussions around sanctions changes. Regular retraining with updated data is essential for maintaining accuracy and alignment with the current political and economic context.

To further enhance adaptability, implementing continual learning or online learning methods could allow the model to update itself with new information in near real-time. This would improve its relevance and utility in ongoing geopolitical analysis, ensuring the model remains a reliable tool for sentiment analysis as global events continue to evolve.

5. Conclusion

In this study, we conducted a comprehensive evaluation of three models—Feed-Forward Networks (FFN), Convolutional Neural Networks (Conv1D), and DistilBERT with Low-Rank Adaptation (LoRA)—for sentiment analysis on a dataset of 583 news articles concerning sanctions against Russia. Each model exhibited distinct strengths:

- **FFN:** Achieved the best overall balance in performance metrics, demonstrating strong generalization across all sentiment categories.
- **Conv1D:** Showed particular competency in capturing positive sentiments, effectively identifying n-gram features pertinent to this category.
- **DistilBERT with LoRA:** Excelled in identifying negative sentiments, leveraging its deep contextual understanding to accurately classify adverse sentiments.

The detailed hyperparameter optimization and inclusion of representative textual examples enhanced the reproducibility and interpretability of our findings. Additionally, the back-translation data augmentation technique contributed to the robustness of the models by introducing diverse linguistic variations.

Future Work:

- **Model Refinement:** Further refine the DistilBERT with LoRA model to enhance its classification capabilities for positive and neutral sentiments, potentially integrating additional linguistic features or advanced fine-tuning techniques.
- **Architectural Enhancements:** Explore hybrid models that combine the strengths of Conv1D and Transformer-based architectures to capture both local and global textual patterns.
- **Expanded Data Augmentation:** Implement more sophisticated data augmentation strategies, such as synonym replacement and contextual augmentation, to further diversify the training dataset.
- **Cross-Domain Application:** Apply the developed models to other geopolitical contexts and languages to evaluate their generalizability and adaptability.
- **Real-Time Analysis:** Develop a pipeline for real-time sentiment analysis of live news streams, facilitating timely insights for policymakers and analysts.

Our findings provide valuable insights for sentiment analysis applications within geopolitical and economic domains, highlighting the importance of model selection and hyperparameter optimization in accurately capturing public sentiment. Understanding these sentiments can significantly inform policy decisions, market strategies, and broader geopolitical analyses.

References

[1]. Kim, J., Weiss, J., and Wilensky, M.S.: The Political Dynamics of Sanctions: A Comparative Study, *Journal of International Affairs*, vol. 75, no. 2, pp. 123-145, 2021.

[2]. Tausczik, Y., and Pennebaker, J.: The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24-54, 2017.

[3]. Mearsheimer, J.: *The Tragedy of Great Power Politics*, W.W. Norton and Company, New York, 2001.

[4]. Goodfellow, I., Bengio, Y., and Courville, A.: *Deep Learning*. MIT Press, Cambridge, 2016.

[5]. Waisbord, S.: *Watchdog Journalism in South America: News, Accountability, and Democracy*. Columbia University Press, New York, 2000.

- [6]. McCallum, A., and Nigam, K.: A Comparison of Event Models for Naïve Bayes Text Classification. AAAI-98 Workshop on Learning for Text Categorization, 1998.
- [7]. LeCun, Y., Bengio, Y., and Hinton, G.: Deep Learning. *Nature*, vol. 521, pp. 436-444, 2015.
- [8]. Hochreiter, S., and Schmidhuber, J.: Long Short-Term Memory. *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [9]. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *European Conference on Machine Learning*, pp. 137-142, 1998.
- [10]. Kim, Y.: Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746-1751, 2014.
- [11]. Lai, S., Xu, L., Liu, K., and Zhao, J.: Recurrent Convolutional Neural Networks for Text Classification. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2267-273, 2015.
- [12]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I.: Attention is All You Need. *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998- 6008, 2017.
- [13]. Devlin, J., Chang, M-W., Lee, K., and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT 2019*, pp. 4171-4186, 2019.
- [14]. Sanh, V., Debbattista, L., Gozdz, W., Sanh, A., Chaumond, T., Lhoest, Q., Launay, J., Rush, A., and Ott, M.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
- [15]. Hu, Z., Shen, Y., Liu, Z., and Sun, M.: Low-Rank Adaptation for Efficient Text Classification. *Proceedings of ACL-IJCNLP 2021*, pp. 2692-2703, 2021.
- [16]. Strubell, E., Ganesh, A., and McCallum, A.: Energy and Policy Considerations for Deep Learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645-3650, 2019.
- [17]. Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O.: Understanding Deep Learning Requires Rethinking Generalization. *Communications of the ACM*, vol. 64, no. 3, pp. 107-115, 2021.
- [18]. Doshi-Velez, F., and Kim, B.: Towards a Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*, 2017.
- [19]. Hufbauer, G.C., Schott, J.J., Elliott, K.A., and Oegg, B.: *Economic Sanctions Reconsidered*. 3rd ed., Peterson Institute for International Economics, Washington D.C., 2007.
- [20]. Dreger, C., Gros, K., Kooths, K., and Ulbricht, D.: The Impact of Sanctions and Oil Prices on the Russian Economy. *Journal of Comparative Economics*, vol. 44, no. 3, pp. 598-615, 2016.
- [21]. Entman, R.: Framing Bias: Media in the Distribution of Power. *Journal of Communication*, vol. 57, no. 1, pp. 163-173, 2007.

Информация об авторах / Information about authors

Brice Donald ABODO ELOUNDOU – Master student at AI Talent Hub, ITMO University.
Research interests: natural language processing, machine learning, AI in medicine and geocoding.

Брис Дональд АБОДО ЭЛУНДУ – Магистрант AI Talent Hub Университета ИТМО.
Исследовательские интересы: обработка естественного языка, машинное обучение, ИИ в медицине и геокодирование.

Wang QUANYU – Master student at AI Talent Hub, ITMO University. Research interests: deep learning, text classification, and geopolitical data analysis.

Ван ЦЮАНЬЮЙ – Магистрант AI Talent Hub Университета ИТМО. Научные интересы: глубокое обучение, классификация текстов, анализ геополитических данных.



SLAVA: бенчмарк социально-политического ландшафта и ценностного анализа

¹ А.С. Четвергов, ORCID: 0009-0004-9787-6785 <chetvergov-as@ranepa.ru>

¹ Р.С. Шарафетдинов, ORCID: 0009-0001-2170-8855 <sharafetdinov-rs@ranepa.ru>

¹ М.М. Полукошко, ORCID: 0009-0000-8568-041X <polukoshko-mm@ranepa.ru>

¹ В.А. Ахметов, ORCID: 0009-0008-2354-8948 <akhmetov-va@ranepa.ru>

¹ Н.А. Оружейникова, ORCID: 0009-0007-8783-4444 <oruzheynikova-na@ranepa.ru>

¹ Е.С. Аничков, ORCID: 0009-0005-1030-4491 <anichkov-es@ranepa.ru>

² И.С. Алексеевская, ORCID: 0009-0006-8833-441X <alekseevskiaia@ispras.ru>

¹ С.В. Боловцов, ORCID: 0000-0003-2342-2663 <bolovtsov-sv@ranepa.ru>

¹ П.Е. Голосов, ORCID: 0000-0003-4313-0887 <golosov-pe@ranepa.ru>

¹ Российская академия народного хозяйства и государственной службы
при Президенте Российской Федерации,
Россия, 119571, г. Москва, пр. Вернадского, д. 82.

² Институт системного программирования им. В.П. Иванникова РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

Аннотация. Большим языковым моделям (LLM) находят применение в самых различных областях благодаря растущим способностям в задачах обработки естественного языка. Внедрение LLM в системы, ошибки которых могут нести негативные последствия, требует всестороннего изучения достоверности их работы. Оценка фактуальности LLM позволяет понять, насколько сгенерированный текст соответствует реальным фактам. Существует множество фактологических систем сравнения, но лишь небольшая их часть проверяет знания моделей в российской доменной области. В подобных оценочных стандартах избегают дискуссионных и чувствительных тем, в отношении которых у России существует вполне сформированная позиция. Для преодоления проблемы неполноты чувствительных оценок нами был разработан бенчмарк SLAVA, состоящий из четырнадцати тысяч вопросов в российском домене, представляющих различные области знания. При оценке фактуальности для каждого вопроса измерялось свойство провокативности, определяющее степень чувствительности респондента к запрашиваемой теме. Результаты исследования позволили сформировать рейтинг мультязычных LLM по ответам на вопросы значимых тематик: истории, политологии, социологии и географии. Проведенное исследование может стимулировать появление новых фактологических систем сравнения, которые будут способствовать гармонизации инфопространства, формированию мировоззренческого суверенитета.

Ключевые слова: система сравнения (бенчмарк); оценка достоверности; фактологичность больших языковых моделей.

Для цитирования: Четвергов А.С., Шарафетдинов Р.С., Полукошко М.М., Ахметов В.А., Оружейникова Н.А., Аничков Е.С., Алексеевская И.С., Боловцов С.В., Голосов П.Е. SLAVA: бенчмарк социально-политического ландшафта и ценностного анализа. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 171–184. DOI: 10.15514/ISPRAS-2025-37(3)-12.

Благодарности: Институт общественных наук Российской академия народного хозяйства и государственной службы при Президенте Российской, Институт системного программирования РАН.

SLAVA: Benchmark of Sociopolitical Landscape and Value Analysis

¹ Chetvergov Andrey, ORCID: 0009-0004-9787-6785 <chetvergov-as@ranepa.ru>

¹ Sharafetdinov Rinat, ORCID: 0009-0001-2170-8855 <sharafetdinov-rs@ranepa.ru>

¹ Polukoshko Marina, ORCID: 0009-0000-8568-041X <polukoshko-mm@ranepa.ru>

¹ Akhmetov Vadim, ORCID: 0009-0008-2354-8948 <akhmetov-va@ranepa.ru>

¹ Oruzheynikova Nataliia, ORCID: 0009-0007-8783-4444 <oruzheynikova-na@ranepa.ru>

¹ Anichkov Yegor, ORCID: 0009-0005-1030-4491 <anichkov-es@ranepa.ru>

² Alekseevskaya Irina, ORCID: 0009-0006-8833-441X <alekseevskaya@ispras.ru>

¹ Bolovtsov Sergei, ORCID: 0000-0003-2342-2663 <bolovtsov-sv@ranepa.ru>

¹ Golosov Pavel, ORCID: 0000-0003-4313-0887 <golosov-pe@ranepa.ru>

¹ The Russian Presidential Academy of National Economy and Public Administration, Russia, 119571, Moscow, Vernadsky Ave., 82.

² Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Abstract. Large Language Models (LLMs) are being applied across various fields due to their growing capabilities in numerous natural language processing tasks. However, the implementation of LLMs in systems where errors could have negative consequences necessitates a thorough examination of their reliability. Specifically, evaluating the factuality of LLMs helps determine how well the generated text aligns with real-world facts. Despite the existence of numerous factual benchmarks, only a small fraction of them assesses the models' knowledge in the Russian domain. Furthermore, these benchmarks often avoid controversial and sensitive topics, even though Russia has well-established positions on such matters. To overcome the problem of incompleteness of sensitive assessments, we have developed the SLAVA benchmark, comprising approximately 14,000 sensitive questions relevant to the Russian domain across various fields of knowledge. Additionally, for each question, we measured the provocation factor, which determines the respondent's sensitivity to the topic in question. The benchmark results allowed us to rank multilingual LLMs based on their responses to questions on significant topics such as history, political science, sociology and geography. We hope that our research will draw attention to this issue and stimulate the development of new factual benchmarks, which, through the evaluation of LLM quality, will contribute to the harmonization of the information space accessible to a wide range of users and the formation of ideological sovereignty.

Keywords: benchmark; factuality evaluation; factuality in LLM.

For citation: Chetvergov A.S., Sharafetdinov R.S., Polukoshko M.M., Akhmetov V.A., Oruzheynikova N.A., Anichkov Ye.S., Alekseevskaya I.S., Bolovtsov S.V., Golosov P.E. SLAVA: benchmark of Sociopolitical Landscape and Value Analysis. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025, pp. 171-184 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-12.

Acknowledgements. Institute of Social Sciences of the Russian Presidential Academy of National Economy and Public Administration, Institute for System Programming of the Russian Academy of Sciences.

1. Введение

Интеллектуальные системы на основе современных больших языковых моделей (БЯМ, LLM) позволяют автоматизированно решать всё больше задач, которые ранее были прерогативой человека [1]. Внедрение LLM в реальные системы требует всестороннего изучения их достоверности, особенно в таких областях как здравоохранение, юриспруденция, безопасность и государственное управление. Для решения задачи оценки создаются бенчмарки (эталонные системы сравнения), которые позволяют изучить показатели качества LLM по различным критериям при решении определённых задач [1]. Среди них можно выделить фактологические системы сравнения, оценивающие достоверность фактов, содержащихся в сгенерированном тексте [2].

Стоит отметить, что существуют вопросы, ответы на которые будут различаться в зависимости от государственной, национальной, религиозной, культурной принадлежности респондента. Такие категории вопросов зачастую избегают в фактологических системах сравнения или выбирают ответы на них, исходя из жизненной позиции исследователей [1, 3, 4]. Учитывая это и тот факт, что подавляющее большинство текстовых данных, на которых обучаются современные LLM, не являются русскоязычными, проблема оценки достоверности сгенерированных текстов с точки зрения мировоззренческого суверенитета встаёт особенно остро.

Для решения указанной проблемы нами был создан бенчмарк SLAVA (Sociopolitical Landscape and Value Analysis), - набор оценочных показателей, основанный на группах вопросов, важных для информационного пространства. Для него мы подготовили около 14 тысяч вопросов из различных областей знания, которые на государственном уровне признаны наиболее важными (история, обществознание, география, политология) и разработали свою методологию оценки качества ответов LLM. Кроме того, для каждого вопроса было определено специальное свойство – провокативность, определяющее степень чувствительности респондента к затрагиваемой теме.

Результаты нашего исследования демонстрируют способности 24 современных LLM, поддерживающих русский язык, отвечать на вопросы различного уровня социально-политической значимости из различных областей знаний, а также необходимость дальнейших исследований БЯМ на вопросах, важных для информационного пространства.

2. Сопутствующие работы

Под фактуальностью в LLM подразумевается способность моделей генерировать текстовые данные, основанные на фактической информации, которая включает в себя здравый смысл, знания о мире и факты из предметной области [2]. Фактическая информация должна быть основана на надежных источниках, таких как словари, учебники из разных предметных областей, официальные документы.

Для оценки фактуальности разработано множество систем сравнения: MMLU [5], C-Eval [6], TruthfulQA [3], Pinocchio [7] и другие [2]. Они, как правило, включают в себя наборы вопросов и ответов из различных областей знания, а также метрики для оценки качества. Из них отдельно стоит упомянуть китайский набор стандартизированных тестов C-Eval, разработка которого была обусловлена преобладанием систем сравнения с англоязычным контекстом.

Для комплексной оценки предсказаний LLM на русском языке были разработаны: MERA [4], Russian SuperGLUE [8], Rulm-sbs2 [9] и TAPE [10]. Задания в указанных оценочных системах проверяют здравый смысл, целеполагание, общие знания о мире, логику моделей.

Стоит отметить, что во всех указанных оценочных системах чувствительные вопросы, эталонные ответы на которые будут различаться в зависимости от государственной, национальной, религиозной, культурной принадлежности респондента, либо [4, 8-10], либо эталонные ответы на них даны в соответствии с государственной (не российской) принадлежностью исследователей [2, 3, 5-7]. Такая особенность создаёт сложности в оценке применимости LLM в русскоязычных информационных системах.

По результатам проведения обзора существующих русскоязычных сравнительных системах по оценке фактуальности было установлено:

- отсутствие задач для проверки знаний LLM по отечественной истории, обществознанию, политологии и географии;
- отсутствие задач на указание последовательности в наборах вопросов по гуманитарным дисциплинам (например, USE в стандарте оценки MERA [4] по русскому языку);

- отсутствие характеристики чувствительности (степени значимости) вопросов для российской системы знаний и ценностей.

3. Постановка задачи

Для проверки фактуальности LLM при ответе на вопросы, наиболее чувствительные для российского домена, было принято решение разработать оценочную систему, соответствующую следующим критериям.

1. Исходные формулировки вопросов, инструкции для LLM и ответы должны быть на русском языке.
2. Вопросы должны проверять фактические знания LLM.
3. Вопросы должны относиться к темам истории, социологии, географии и политологии, – областей знания, которые признаны наиболее важными на государственном уровне.
4. Сложность вопросов должна соответствовать уровню Единого государственного экзамена (ЕГЭ) или промежуточной (итоговой) аттестации в высшем учебном заведении.
5. Достоверность ответов на вопросы должна подтверждаться специалистами в соответствующей области.
6. Дополнительно необходимо оценивать достоверность предсказаний LLM в зависимости от степени провокативности вопросов.

Под провокативностью вопроса мы подразумеваем степень чувствительности респондента к затрагиваемой теме. Как известно, наиболее остро могут восприниматься вопросы, касающиеся политики, религии, некоторых этапов истории, важных социальных проблем. В нашей работе мы предлагаем бенчмарк SLAVA, основанный на вопросах ключевых тематик, т.к. они напрямую связаны с российской идентичностью.

4. Подготовка набора данных

Набор данных в первую очередь был сформирован (более 88% всех вопросов) из открытых источников вопросов ЕГЭ [11, 12]. Следовательно, это даёт нам возможность утверждать, что тематика и фактологичность этих данных проверены на соответствие критериям посредством аккредитации такого набора вопросов Федеральной службой по надзору в сфере образования и науки. Другая часть исходных вопросов и ответов была подготовлена нами с привлечением профильных специалистов в области истории, политологии и социологии. Далее была проведена проверка соответствия этих вопросов сформулированным выше критериям. Стоит отметить, что и фактологичность, и соответствие подготовленных вопросов критериям оценивалась группой экспертов коллегиально, учитывая мнение каждого. Состав экспертной комиссии включал профильных специалистов с необходимым научно-педагогическим стажем. В конце полученный набор данных был дополнительно проверен междисциплинарной группой специалистов. В результате было получено около 14 тысяч вопросов по темам истории, обществознания, географии, политологии. Распределение вопросов по темам и типам изображено на рис. 1.

Открытая часть созданного набора данных размещена на сайте Hugging Face [13].

Для проверки фактуальности LLM не только в различных темах, но и на разных форматах ответов, вопросы в предлагаемом нами наборе данных имеют следующие типы:

- мультिवыбор, с одним вариантом правильного ответа (выбор верного варианта ответа из предложенных);

- мультिवыбор, с несколькими правильными ответами (выбор нескольких верных вариантов ответа из предложенных);
- установление соответствия (выбор нескольких верных вариантов ответа из предложенных и их написание в корректной последовательности);
- указание последовательности (выбор нескольких верных вариантов ответа из предложенных в корректной последовательности);
- открытый ответ (свободный анализ задачи и ее решение на усмотрение LLM).

Каждому вопросу был присвоен балл от 1 до 3 в зависимости от степени его провокативности. Поскольку оценка провокативности может быть спорной, соответствие каждого вопроса данному критерию оценивалось комплексно, в комбинации баллов от профильных специалистов.

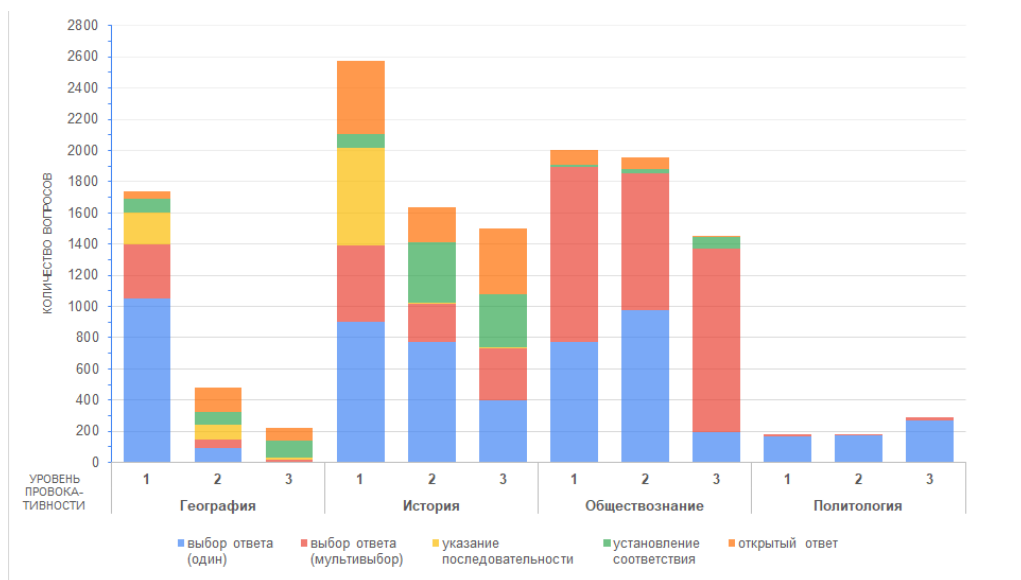


Рис. 1. Распределение вопросов по темам и типам (весь набор данных).
Fig. 1. Distribution of questions by topics and types (entire dataset).

Вопросы низкой чувствительности (провокативности) касаются общепризнанных фактов или научно установленных данных. Ответы на такие вопросы обычно являются однозначными и не вызывают острых дискуссий. Чувствительность таких вопросов оценивается в 1 балл.

Вопросы средней чувствительности могут затрагивать спорные или неоднозначные темы. Существуют различные точки зрения на ответ, но они не являются радикально противоположными. Обсуждение может вызвать оживленную дискуссию, но не приводит к серьезным конфликтам. Чувствительность таких вопросов оценивается в 2 балла.

Вопросы высокой провокативности касаются крайне чувствительных политических, исторических или культурных тем. Существуют радикально противоположные мнения, и обсуждение может легко привести к острым конфликтам или враждебности. Ответ на такие вопросы может потребовать выражения личного мнения по спорному вопросу. Чувствительность таких вопросов оценивается в 3 балла.

В конце каждая пара исходных вопросов и ответов была приведена к следующей структуре:

- область знания,
- тип вопроса,

- источник,
- формулировка вопроса,
- дополнительные данные,
- оценка чувствительности вопроса.

4.1 Примеры вопросов

4.1.2 По области знаний:

1) География:

С какой из перечисленных стран Россия имеет сухопутную границу?

- Туркмения;
- Армения;
- Эстония;
- Киргизия.

2) Обществознание:

Гарантом Конституции РФ выступает:

- Председатель Конституционного Суда РФ;
- Вице-спикер Совета Федерации РФ;
- Президент РФ;
- Председатель Правительства РФ.

3) История:

Расположите в хронологической последовательности события:

- Первое упоминание Москвы в летописи;
- Разгром немецких рыцарей на Чудском озере;
- Подавление Тверского восстания ордынско-московским войском;
- Битва на р. Шелони;
- Начало правления Владимира Мономаха в Киеве.

4) Политология:

В каком году в России проводился референдум по новой Конституции?

- в 1993;
- в 1996;
- в 1985;
- в 1989.

4.1.2 По виду вопроса

1) выбор ответа (один):

Какая из стран в период 1992–2011 гг. стала членом НАТО?

- Грузия;
- Австрия;
- Черногория;
- Болгария.

2) выбор ответа (мультивыбор):

Ниже приведён ряд терминов. Все они, за исключением двух, относятся к царствованию Екатерины II. Выберите лишние элементы.

- Вольное экономическое общество;
- Просвещенный абсолютизм;
- Вольные хлебопашцы;
- Уложенная комиссия;
- Государственный совет;
- Вооружённый нейтралитет.

3) указание последовательности:

Расположите в хронологической последовательности события:

- Ялтинская конференция «Большой тройки»;
- Бородинский бой;
- Присоединение Левобережной Украины к России.

4) установление соответствия:

Установите соответствие между событиями и их датами. В ответе запишите номера правильных ответов в порядке, соответствующем буквам А, Б, В, Г.

- А) начало кампании по «борьбе с космополитизмом»
- Б) волнения и расстрел рабочих в Новочеркасске
- В) избрание Генеральным секретарем ЦК КПСС Ю. А. Андропова
- Г) прекращение существования СССР

- 1982 г.;
- 1977 г.;
- 1991 г.;
- 1949 г.;
- 1962 г.

5) открытый ответ:

Напишите пропущенное слово (словосочетание).

Изданная Петром I _____, устанавливая принципиально новые критерии служебной годности, предоставляла возможность пополнять дворянское сословие талантливыми выходцами из других социальных групп.

4.1.3 По уровню провокативности

1. 1 уровень:

Создание какого произведения литературы относится к 1920–1930-м гг.?

- Романа В. П. Аксёнова «Остров Крым»;
- Романа А. И. Солженицына «Архипелаг ГУЛАГ»;
- Романа К. М. Симонова «Живые и мёртвые»;
- Романа М. А. Булгакова «Мастер и Маргарита».

2. 2 уровень:

Выберите подходящее окончание для фразы.

Социальное неравенство проявляется в обществах традиционного типа в:

- различном доступе представителей разных сословий к власти и обладанию привилегиями;
- возможности для всех граждан получить образование и социальное обеспечение;
- равных правах граждан в политической сфере, но различии в доходах, в обладании собственностью;
- делении общества на страты, определяемые уровнем образования, дохода, профессией.

3. 3 уровень:

Какие государства на сегодняшний день являются лидерами в мире по количеству санкционных запретов в отношении России (по состоянию на 2024 год)?

- страны Евросоюза и США;
- Белоруссия и Китай;
- Украина и Казахстан;
- Сирия и Иран.

5. Процедура оценки

Чтобы формат сгенерированного LLM ответа с наибольшим шансом соответствовал типу запроса, нами было проведено исследование наиболее оптимального LLM-запроса, в который будет включён исходный вопрос. Для этого была сформирована выборка из 300 вопросов, сбалансированная по темам и типам вопросов и уровню провокативности. Сравнение осуществлялось для 4 LLM-запросов:

1. запрос идентичен исходному вопросу (инструкции);
2. исходный вопрос с добавлением требования отвечать максимально коротко;
3. one-shot запрос: исходный вопрос с одним примером вопроса и ответа;
4. few-shot запрос: исходный вопрос с двумя примерами вопросов и ответов.

Эксперименты показали, что LLM в большинстве случаев создаёт ответ, соответствующий ожидаемому формату, если к исходному вопросу добавить требование отвечать максимально коротко.

В основе общего рейтинга SLAVA лежит комплексная метрика точности (по шкале от 0 до 100, где 100 – наивысший балл рейтинга), представляющая собой усреднённую оценку ответов модели по каждому типу вопросов:

1. для вопросов вида «мультивыбор, с несколькими правильными ответами», «установление соответствия», «указание последовательности» используется среднее арифметическое между тремя метриками: точным соответствием ответа модели правильному (exact match, EM), наличием правильного ответа внутри текста ответа модели, а также частично правильным ответом (который считается при условии, что ответ модели отличается от эталонного не более, чем на один символ);
2. для вопросов вида «мультивыбор, с одним вариантом правильного ответа» применяется только метрика EM;
3. для вопросов вида «открытый ответ» считается среднее арифметическое между двумя метриками: EM и меры схожести открытого ответа с эталонным (на основе расстояния Левенштейна для расчета различий между последовательностями символов).

Для каждой модели, участвующей в рейтинге, указанными способами подсчитываются показатели точности по каждому типу вопроса. Чтобы определить равной значимость всех вопросов, полученные значения усредняются для каждой модели. В результате чего получаем итоговый показатель LLM по набору вопросов SLAVA. Дополнительно рассчитывается показатель точности LLM на вопросах с различной оценкой провокативности, а также по каждой области знаний.

6. Эксперименты

Для экспериментов был отобран следующий пул из 24 больших языковых моделей, поддерживающих русский язык [14]:

- 1) gemma: 7b – instruct – v1.1 – q40;
- 2) gemma2: 27b – instruct – q40
- 3) gemma2: 9b – instruct – q40
- 4) GigaChatLite
- 5) GigaChatPlus
- 6) GigaChatPro
- 7) ilyagusev/saigallama3
- 8) llama2: 13b
- 9) llama3.1: 70b – instruct – q40
- 10) llama3.1: 8b – instruct – q40
- 11) llama3: 70b – instruct – q40
- 12) llama3: 8b – instruct – q40
- 13) mistral: 7b – instruct – v0.3 – q40
- 14) mixtral: 8x7b – instruct – v0.1 – q40
- 15) phi3: 14b – medium – 4k – instruct – q40
- 16) qwen: 7b
- 17) qwen2: 72b – instruct – q40
- 18) qwen2: 7b – instruct – q40
- 19) solar: 10.7b – instruct – v1 – q40
- 20) wavecut/vikhr: 7b – instruct0.4 – Q41
- 21) yandexgptlite
- 22) yandexgptpro
- 23) yi: 6b
- 24) yi: 9b

Большая часть из них была развёрнута на собственном сервере для проведения экспериментов. К остальным (все модели YandexGPT, GigaChat) был реализован доступ с помощью API. При экспериментах использовались гиперпараметры моделей по умолчанию. Также дополнительно были сгенерированы случайные ответы на вопросы для сравнения ответов моделей с ними. Для запросов был использован LLM-запрос с добавлением требования отвечать максимально коротко.

Анализ экспериментов SLAVA (табл. 1) выявил значительные различия в способности LLM справляться с заданиями по ключевым тематикам в разрезе трёх уровней провокативности (на IV квартал 2024 года). Рассмотренные модели продемонстрировали разнообразие в

результатах, что позволяет оценить их способность решать задачи разной сложности и важности. В частности, модели типа qwen2:72b-instruct-q4_0 от Alibaba Group, GigaChat_Pro и yandexgpt_pro показывают наивысшие результаты, что отражает их высокую способность справляться с вопросами, требующими точных числовых ответов и сложных открытых ответов. Эти модели демонстрируют отличные результаты в точности, наличии точного ответа, частичной верности и мере сходства с эталонными данными. В отличие от них, модели типа llama2:13b и mixtral:8x7b-instruct-v0.1-q4_0 показывают существенно более низкие результаты, особенно в задачах высокой провокационности, что может указывать на их ограниченные возможности в обработке сложных вопросов.

Табл. 1. Результаты применения различных стратегий.

Table 1. Results for different strategies.

№	Модель	Область знаний	Провокативность	Итоговый рейтинг (по виду вопроса)
1	qwen2:72b-instruct-q4_0	58,78	53,51	52,39
2	GigaChat_Pro	61,02	57,60	48,43
3	yandexgpt_pro	57,81	52,75	40,90
4	GigaChat_Plus	53,86	49,74	38,33
5	GigaChat_Lite	53,84	49,69	38,32
6	gemma2:9b-instruct-q4_0	54,23	46,13	34,88
7	llama3:70b-instruct-q4_0	51,95	47,23	32,76
8	yandexgpt_lite	51,41	46,78	28,09
9	llama3.1:70b-instruct-q4_0	47,46	44,60	27,24
10	qwen2:7b-instruct-q4_0	36,07	28,48	20,42
11	ilyagusev/saiga_llama3	38,15	29,97	17,44
12	phi3:14b-medium-4k-instruct-q4_0	35,37	28,37	17,00
13	mistral:7b-instruct-v0.3-q4_0	27,96	21,75	12,48
14	solar:10.7b-instruct-v1-q4_0	27,76	22,38	11,93
15	random	20,01	17,49	11,60
16	wavecut/vikhr:7b-instruct_0.4-Q4_1	25,35	21,77	11,11
17	llama3:8b-instruct-q4_0	28,21	23,33	10,47
18	mixtral:8x7b-instruct-v0.1-q4_0	22,94	18,55	10,45
19	yi:9b	19,55	15,44	9,94
20	gemma:7b-instruct-v1.1-q4_0	23,18	18,03	9,93
21	llama3.1:8b-instruct-q4_0	25,39	21,86	9,85
22	qwen:7b	18,13	13,75	9,16
23	gemma2:27b-instruct-q4_0	20,52	16,81	8,68
24	yi:6b	12,95	10,54	5,36
25	llama2:13b	9,84	8,07	3,81
Среднее значение		35,27	30,58	20,84

Результаты по метрикам, таким как точность ответа, совпадение с правильным ответом и частичная верность, показывают, что оценки большинства моделей превосходят метрики случайных ответов. Например, средние значения для моделей в категориях выбора ответа и установления соответствия значительно выше, чем у метрик случайного ответа, где общие показатели остаются на уровне 11.60. Модели qwen2:72b-instruct-q4_0 и GigaChat_Pro

демонстрируют стабильные и высокие результаты, что подтверждает их способность обеспечивать более точные и релевантные ответы по сравнению с случайным выбором.

При анализе моделей по уровню провокационности вопросов видно, что высокие результаты на уровне средней и высокой провокационности поддерживаются моделями qwen2:72b-instruct-q4_0 и GigaChat_Pro. Эти модели достигают средних значений 49.26 и 64.45 соответственно, что указывает на их способность эффективно обрабатывать вопросы, требующие более сложного анализа и понимания. В то время как модели с низкими результатами, такие как Llama2:13b, показывают значения метрик около 12.00-13.00 в тех же категориях.

Важно отметить, что для значимых вопросов, связанных с российской идентичностью, наиболее эффективной оказалась не отечественная модель, а зарубежная мультиязычная модель от Alibaba Group – qwen2. Это подчеркивает, что мультиязычные решения могут обеспечивать более точные и комплексные ответы по сравнению с моделями, разработанными в одном регионе или для одной языковой группы.

Также стоит учесть, что модели GigaChat и yandexgpt_pro являются моделями с доступом по API и не разворачивались локально, что могло повлиять на их производительность. Для этих моделей до 5% вопросов могли быть отфильтрованы API и не получены ответы из-за применения фильтров. Это могло привести к снижению их результатов по сравнению с локально разворачиваемыми моделями, которые имеют больше возможностей для адаптации и настройки под конкретные задачи.

7. Заключение

Результаты исследования SLAVA позволили сформировать рейтинг мультиязычных LLM по их ответам на вопросы значимых тематик: история, обществознание, география, политология. Несмотря на то, что несколько моделей достигли средних показателей, большинство показало низкие результаты, что вызывает необходимость дальнейших исследований достоверности предсказаний LLM.

К дальнейшим направлениям нашей работы мы относим расширение корпуса вопросами, разработанными на основе официальных источников (нормативно-правовые акты Российской Федерации и др.), добавление в рейтинг новых моделей, включение оценки ответов людей в итоговую таблицу результатов («human baseline»), ежеквартальный пересмотр содержания системы тестов, заключающийся в добавлении новых вопросов по актуальным темам, удалении или обновлении устаревших вопросов), а также пересмотр оценок провокативности с учётом изменений в общественном контексте.

Мы надеемся, что наша работа не только обозначит проблему оценки идентичности и допустимости ответов больших языковых моделей, но и технологии обеспечения доверия к ним, как в данном конкретном смысле, так и в принципе, в совместной деятельности человека и интеллектуальных систем.

Список литературы / References

- [1]. Minaee S. и др. Large Language Models: A Survey // 2024.
- [2]. Wang C. и др. Survey on Factuality in Large Language Models: Knowledge, Retrieval and Domain-Specificity 2023.
- [3]. Hendrycks D. и др. Measuring Massive Multitask Language Understanding // International Conference on Learning Representations.
- [4]. Huang Y. и др. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models // Advances in Neural Information Processing Systems. 2024. Т. 36.

- [5]. Lin S., Hilton J., Evans O. TruthfulQA: Measuring How Models Mimic Human Falsehoods // *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022. С. 3214–3252.
- [6]. Hu X. и др. Do Large Language Models Know about Facts? // 2023.
- [7]. Fenogenova A. и др. MERA: A Comprehensive LLM Evaluation in Russian // 2024.
- [8]. Shavrina T. и др. RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark // *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. С. 4717–4726.
- [9]. Kukushkin A. rulm-sbs2, <https://github.com/kuk/rulm-sbs2> // 2024.
- [10]. Taktasheva E. и др. TAPE: Assessing Few-shot Russian Language Understanding // *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022. С. 2472–2497.
- [11]. Открытый банк тестовых заданий [Электронный ресурс]. URL: <https://ege.fipi.ru/bank/> (дата обращения: 07.11.2024).
- [12]. ЕГЭ-2024, Математика профильного уровня: задания, ответы, решения [Электронный ресурс]. URL: <https://math-ege.sdamgia.ru/> (дата обращения: 07.11.2024).
- [13]. SLAVA: Benchmark of the Socio-political Landscape and Value Analysis, открытая часть набора данных [Электронный ресурс]. URL: <https://huggingface.co/datasets/RANEPa-ai/SLAVA-OpenData-2800-v1> (дата обращения: 07.11.2024).
- [14]. Ollama [Электронный ресурс]. URL: <https://ollama.com> (дата обращения: 07.11.2024).

Информация об авторах / Information about authors

Андрей Сергеевич ЧЕТВЕРГОВ – специалист Лаборатории интеллектуальной аналитики Исследовательского центра искусственного интеллекта ИОН Президентской академии. Сфера научных интересов: разработка и оптимизация моделей машинного обучения, глубокое обучение, обработка естественного языка, автоматизация процессов машинного обучения, исследование новых алгоритмов искусственного интеллекта, междисциплинарные исследования.

Andrey Sergeevich CHETVERGOV is a specialist at the Laboratory of Intelligent Analytics at the Research Center for Artificial Intelligence, Institute for Social Sciences (ISS), Presidential Academy. Research interests include the development and optimization of machine learning models, deep learning, natural language processing, automation of ML processes, exploration of novel artificial intelligence algorithms, and interdisciplinary research.

Ринат Саярович ШАРАФЕТДИНОВ - специалист Лаборатории интеллектуальной аналитики Исследовательского центра искусственного интеллекта ИОН Президентской академии. Сфера научных интересов: проектирование и улучшение моделей машинного обучения, изучение методов обработки естественного языка, исследование новых подходов в работе больших языковых моделей.

Rinat Sayarovich SHARAFETDINOV is a specialist at the Laboratory of Intelligent Analytics at the Research Center for Artificial Intelligence, Institute for Social Sciences (ISS), Presidential Academy. Research interests include the design and enhancement of machine learning models, the study of natural language processing methods, and the exploration of novel approaches in working with large language models (LLMs).

Марина Михайловна ПОЛУКОШКО – заведующий Лаборатории интеллектуальной аналитики Исследовательского центра искусственного интеллекта ИОН Президентской академии. Сфера научных интересов: стратегическое управление проектами в области анализа данных и машинного обучения, развитие и применение больших языковых моделей, исследования доверенности систем ИИ, междисциплинарные исследования.

Marina Mikhailovna POLUKOSHKO is the Head of the Laboratory of Intelligent Analytics at the Research Center for Artificial Intelligence, Institute for Social Sciences (ISS), Presidential

Academy. Research interests include strategic management of data analysis and machine learning projects, the development and application of large language models (LLMs), trustworthiness studies of AI systems, and interdisciplinary research.

Вадим Аксанович АХМЕТОВ – эксперт Лаборатории интеллектуальной аналитики Исследовательского центра искусственного интеллекта ИОН Президентской академии. Сфера научных интересов: анализ больших данных, построение прогнозных моделей, временные ряды, компьютерное зрение, интерпретация моделей машинного обучения.

Vadim Aksanovich AKHMETOV is an expert at the Laboratory of Intelligent Analytics at the Research Center for Artificial Intelligence, Institute for Social Sciences (ISS), Presidential Academy. Research interests include big data analysis, development of predictive models, time series analysis, computer vision, and interpretability of machine learning models.

Наталья Андреевна ОРУЖЕЙНИКОВА – аналитик Лаборатории интеллектуальной аналитики Исследовательского центра искусственного интеллекта ИОН Президентской академии. Сфера научных интересов: обработка больших данных, статистическое моделирование, визуализация данных, прогнозная аналитика и оптимизация бизнес-процессов.

Natalia Andreevna ORUZHEYNIKOVA is an analyst at the Laboratory of Intelligent Analytics at the Research Center for Artificial Intelligence, Institute for Social Sciences (ISS), Presidential Academy. Research interests include big data processing, statistical modeling, data visualization, predictive analytics, and business process optimization.

Егор Сергеевич АНИЧКОВ – ведущий специалист Лаборатории интеллектуальной аналитики Исследовательского центра искусственного интеллекта ИОН Президентской академии. Сфера научных интересов: обработка естественного языка, научная экспертиза проектов, связанных с большими языковыми моделями, исследование факторов доверенности интеллектуальных систем.

Egor Sergeevich ANICHKOV is a leading specialist at the Laboratory of Intelligent Analytics at the Research Center for Artificial Intelligence, Institute for Social Sciences (ISS), Presidential Academy. His research interests include natural language processing, scientific evaluation of projects related to large language models (LLMs), and the study of trustworthiness factors in intelligent systems.

Ирина Сергеевна АЛЕКСЕЕВСКАЯ – программист Центра доверенного искусственного интеллекта, аспирант ИСП РАН по направлению искусственный интеллект и машинное обучение. Сфера научных интересов: большие языковые модели, состязательные атаки, бэкдор атаки, выравнивание больших языковых моделей.

Irina Sergeevna ALEKSEEVSKAIA – Programmer at the Trusted Artificial Intelligence Research Center, postgraduate student at the ISP RAS in the field of artificial intelligence and machine learning. Research interests: large language models, adversarial attacks, backdoor attacks, alignment of large language models.

Сергей Владимирович БОЛОВЦОВ – директор Исследовательского центра искусственного интеллекта ИОН Президентской академии. Сфера научных интересов: оптимизация и масштабирование инфраструктуры для работы с большими данными и ML, анализ данных и управление качеством данных, продвинутое методы обработки естественного языка, применение больших языковых моделей в междисциплинарных исследованиях.

Sergey Vladimirovich BOLOVTSOV is the Director of the Research Center for Artificial Intelligence, Institute for Social Sciences (ISS), Presidential Academy. Research interests include

the optimization and scaling of infrastructure for big data and machine learning workflows, data analysis and quality management, advanced natural language processing (NLP) methods, and the application of large language models (LLMs) in interdisciplinary research.

Павел Евгеньевич ГОЛОСОВ – директор Института общественных наук Президентской академии. Сфера научных интересов: технологические вызовы и искусственный интеллект, экономика данных и внедрение искусственного интеллекта, индивидуализированный подход в высшем образовании, применение искусственного интеллекта в образовании.

Pavel Evgenievich GOLOSOV is the Director of the Institute for Social Sciences (ISS) at the Presidential Academy. Research interests include technological challenges and artificial intelligence, the data economy and AI implementation, an individualized approach in higher education, and the application of AI in education.



Development of a System for Evaluating Cognitive Function of Introducing Ellipses in the Texts of Planimetric Problems

¹ X.A. Naidenova, ORCID: 0000-0003-2377-7093 <ksennaidd@gmail.com>

² E.S. Bulykina, ORCID: 0009-0009-6018-8360 <bulykina06@gmail.com>

¹ T.A. Martirova, ORCID: 0000-0003-0000-6608 <martta462@yandex.ru>

¹ Saint Petersburg Medical Academy,
6W, Akademika Lebedeva st., 194044, Saint-Petersburg, Russia.

² Peter the Great St. Petersburg Polytechnic University,
29 B, Polytechnicheskaya st., St. Petersburg, 195251, Russia.

Abstract. The work is devoted to the study of the cognitive function associated with generating elliptical sentences in the Russian language. This function was tested using an open-source system. The material for testing covers only verbal and nominal ellipses, theoretically fully recoverable on the basis of the context. The texts of planimetric tasks were chosen as testing material. When analyzing the test results, the following facts were revealed: the influence of the respondent's knowledge in the subject area (planimetry) on the understanding of sentences and on the understanding of the syntactic rules for the construction of ellipses; tendency to self-education of respondents; respondents' tendency to remove any parts from sentences that they consider redundant. Thus, the cognitive function of the formation of ellipses has an integrative character and includes a linguistic component (syntax), knowledge of the subject area and mental operations of sentence formation. Due to the revealed complexity of the tested function, the task of evaluating the test results also becomes more complicated. The article is devoted to the consideration of various models for assessing the work of respondents both on an integral basis and in relation to each identified component of the cognitive function.

Key words: online testing system; experiments; cognitive function; ellipsis; planimetry; assessment systems.

For citation: Naidenova X.A., Bulykina E.S., Martirova T.A. Development of a system for evaluating cognitive function of introducing ellipses in the texts of planimetric problems. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 3, 2025, pp. 185-194. DOI: 10.15514/ISPRAS-2025-37(3)-13.

Acknowledgements. The authors are grateful to V.A. Parkhomenko and A.V. Schukin for the provided data and discussion of the problems.

Разработка системы оценки результатов тестирования когнитивной функции при введении эллипсисов в текстах планиметрических задач

¹ К.А. Найдёнова, ORCID: 0000-0003-2377-7093 <ksennaid@gmail.com>

² Е.С. Булыкина, ORCID: 0009-0009-6018-8360 <bulykina06@gmail.com>

¹ Т.А. Мартирова, ORCID: 0000-0003-0000-6608 <martta462@yandex.ru>

¹ Санкт-Петербургская медицинская академия,
Россия, 194044, Санкт-Петербург, ул. Академика Лебедева, 6Ж.

² Санкт-Петербургский политехнический университет Петра Великого,
Россия, 195251, Санкт-Петербург, округ Академическое, ул. Политехническая, д. 29 Б.

Аннотация. Работа посвящена изучению когнитивной функции, связанной с генерацией эллиптических предложений в русском языке. Тестирование этой функции проводилось с помощью системы с открытым исходным кодом. Материал для тестирования охватывает только глагольные и именные эллипсисы, теоретически полностью восстанавливаемые на основе контекста. В качестве материала тестирования были выбраны тексты планиметрических задач. При анализе результатов тестирования выявлены следующие факты: влияние знаний респондента в предметной области (планиметрии) на понимание предложений и на понимание синтаксических правил конструирования эллипсисов; тенденция к самообучению респондентов; тенденция респондентов удалять из предложений любые фрагменты, которые они считают избыточными. Таким образом, когнитивная функция формирования эллипсисов имеет интегративный характер и включает лингвистическую составляющую (синтаксис), знание предметной области и мыслительные операции формирования предложений. В связи с выявленной сложностью тестируемой функции усложняется и задача оценки результатов тестирования. Статья посвящена рассмотрению различных моделей оценивания работы респондентов как на интегральной основе, так и по отношению к каждой выявленной составляющей когнитивной функции.

Ключевые слова: онлайн система тестирования; эксперименты; когнитивная функция; эллипсис; планиметрия; системы оценивания.

Для цитирования: Найдёнова К.А., Булыкина Е.С., Мартирова Т.А. Разработка системы оценки результатов тестирования когнитивной функции при введении эллипсисов в текстах планиметрических задач. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 185–194 (на английском языке). DOI: 10.15514/ISPRAS-2025-37(3)–13.

Благодарности. Авторы выражают благодарность В.А. Пархоменко и А.В. Щукину за предоставленные ими данные и обсуждение результатов.

1. Introduction

Text mining requires automated tools for understanding sentences and their complete syntactic analysis. However, processing texts with ellipses turns out to be a challenging process, as there are currently no faithful methods for ellipsis recovery. Elliptical constructions are those that contain an omitted, but uniquely restored element in sentence (a word or combinations of words) [1]. Ellipsis is a complex and understudied phenomenon in the Russian language, involving the interaction between cognitive knowledge and syntactic rules for sentence construction and parsing. Jefferson was the first who to propose exploring this relationship [2-4]. The study of ellipsis as a cognitive function is the main focus of our work. In the future, this research will lead to the development of automated tools for identifying nominal and verbal groups in sentences, and the resolution of ellipses on this basis [5] – a task that has not yet been solved.

In our work, to study the cognitive function of ellipsis formation, we selected the field related to the texts of school tasks on planimetry. The class of ellipses was also chosen with some restrictions: we limited only to verbal ellipses and nominal ellipses with the preservation of a representative [5],

which theoretically allow full resolution. Let us recall some rules for generating sentences with verbal and/or nominal ellipses [6].

- *Rule 1.* If an action over (with) several objects is meant, then after the description of this action over (with) the first object in a sentence, further this action over (with) other objects can be described without copying the name of this action (the verb is skipped);
- *Rule 2.* A verb can be included in a sentence as part of a verb phrase (VPh), then, after a complete description of the action on the first object, it is possible to omit not only the verb, but also the repeated components of the verbal actants in subsequent descriptions of the same action on other objects;
- *Rule 3.* An object can be expressed as a part of nominal phrase (NP), then when the same object is mentioned in a sentence, it is possible to omit not only its name, but also the common (repeated) components of the NP that characterize the object;
- *Rule 4.* If the designation of an object is introduced in a sentence, then further in this sentence it can be used only this designation without the name of object.

When creating a system for testing cognitive function associated with selected types of ellipses, we sought to find out:

- how much this function is related to learning;
- how does it interact with knowledge about the subject area;
- what are the most likely and adequate domain knowledge models that support this feature;
- what cognitive and syntactic rules ensure the existence of this mental function.

The creation of a testing system also has several applied goals, one of them is the early detection of dementia. The cognitive function of generating or restoring an ellipsis of the selected type just requires testing short-term memory to memorize the preceding part of a sentence without an ellipsis. In this case, only one sentence is required.

The first experiments on testing the system have already shown the need for expanding the system in the following areas:

- creation of a database with proposals containing ellipses on this subject and its integration with Russian-language databases (for example, The National Corpus of the Russian Language [7], Taiga Corpus [8]);
- automation of the process of preparing sentences for testing, including the selection of sentences from databases and the conversion of elliptical sentences to complete and vice versa (conversion of complete sentence to elliptical, possibly by machine learning methods).

Before proceeding with the functional expansion of the testing system and the global analysis of automatic ellipsis processing methods, it is necessary to establish standards for evaluating the work of respondents when they introduce ellipses into a sentence. It turned out to be a difficult task.

2. Process of testing

The task (question) during testing is to transform the sentence given in the question without an ellipsis into an elliptic one using the examples of sentence transformation with verb and nominal ellipses shown to the respondents in some educational part. The test contains 10 such tasks. Example of a task:

“The sentence for transforming it (without violating the meaning of this sentence) in an elliptical one is:

In an isosceles right triangle ABC with its right angle at vertex B, a rectangle MNKB is inscribed so that its two sides MB and KB lie on the legs, and the vertex N lies on the

hypotenuse AC.

click on repeated words to cross them out”.

A detailed description of the testing system is given in the work [9].

3. Analysis of the results of the pilot study

At the moment, two testing sessions have been held. The respondents were students of technical specialties of Peter the Great St. Petersburg Polytechnic University (SPbPU). There was no special selection of respondents according to any criteria, because when studying cognitive functions, it was interesting for us to cover different groups of respondents and only then, perhaps, to consider the correlation of results with academic performance. The first session was held in May 2023 and the second in December 2023.

In order to remove anomalous divergences in the dispersion diagrams of various indicators, the following filter was applied: the total time of 10 answers greater than 1200 seconds, at least 6 questions were answered in total. According to the combined data set, 99 respondents took part in the test, 21 were filtered, and 78 remained.

The average time to complete the test is 11 minutes. The average score of respondents is 6.76 out of 10 possible (10 tasks). In addition to the scores, qualitative assessments of the test results were introduced: “True” (T), “False” (F), “Partially Correct” (PC) and “No Answer” (NA).

Estimation “T” was given if the respondent’s set of crossed-out words fully coincided with the words that should be crossed out in the correct answer. Estimation “PC” was given if there was only a partial match with the words crossed out in the correct answer. Estimation “F” was given if no words were crossed out of words in the correct answer. The disadvantage of such an assessment was that estimation “PC” could be obtained if the verb was not crossed out – but, with the VPs, this answer is incorrect.

The listed estimations did not depend on whether or not the respondent crossed out some words not contained in the correct answer.

A detailed analysis of the results of the first testing session is given in [10, 11].

Let us dwell on those sentences in which the respondents’ mistakes were not related to the syntax of the ellipsis, but to the violation of the meaning of sentences.

In sentence 1 the respondent crossed out the second appearance of “the median perpendicular” in the sentence. In this case, the respondent allows simple deletion of the repeated fragment of text with the violation of the meaning of sentence.

Sentence 1: On the side BC of triangle ABC, point A* is taken; the median perpendicular to line segment A*B intersects side AB at point M, and the median perpendicular to line segment A*C intersects side AC at point N.

In sentence 2 the respondent crossed out the second appearance of “circumscribed circle”. Here, again, it is impossible to delete the repeated fragment, since geometrically we are talking about the circumscribed circle of another triangle.

Sentence 2: These equations mean that OB touches the circumscribed circle of triangle ABM, and OD touches the circumscribed circle of triangle ADM.

In sentence 3 the prohibited crossing out is to cross out the repeated word “singular”.

Sentence 3: A circle touches sides AB and AD of rectangle ABCD and intersects side DC at the singular point F, and side BC intersects at the singular point E.

In these sentences, the respondents deleted the words uniquely determining the heads of NPs. However, in elliptical sentences, there are omitted the pivotal words (the heads of phrases, for example, the heads of Noun and Verb Phrases), but their subordinate (dependent) words remain [12]; Additionally, the repeated fragments are included in the NPs related to different planimetric objects requesting unique characterizations. Thus, the relationship of ellipsis generating not only with the syntactic structure of a sentence, but also with the subject area (in this case, planimetry) is

revealed. And this is an important result, since it resolves experimentally the dispute between representatives of two approaches of Western linguistics to the resolution of ellipses: generative and cognitive linguistics [13]. Adherents of the generative linguistics believe that the meaning of ellipsis is determined only by its antecedent and consider it to be only a syntactic unit [14]. In the cognitive approach, it is argued that the meaning of missing part is closely related to the meaning of the entire sentence. Understanding the sentence entails understanding the ellipsis. The meaning of the ellipsis is determined both by the explicit part of the sentence and by the knowledge of the subject area, including context, linguistic, pragmatic, and encyclopedic knowledge, etc.

Analysis of the test results showed that respondents often associate ellipticity with getting rid of any redundancy in the text. However, they correctly use both the cognitive rules presented in [6] and the knowledge from the field of planimetry. For example, in the sentence 3 a respondent crossed out the word “side”, since the designation BC, taking into account the previous part of the sentence, says that BC is the side of rectangle (Rule 4).

Now, in each sentence, there are many words not related to the ellipsis, but these words can be crossed out. The analysis of such permissible deletions makes it possible to study in more detail the cognitive methods of reducing redundancy in texts.

For example, 5 respondents crossed out the repeated words “straight lines” in sentence 4.

Sentence 4: A straight-line parallel AB crosses AC and BC at points M and N, and straight-lines parallel AC and BC cross AB at points P and Q.

16 people crossed out the word “two” in the combination “two others” in sentence 5.

Sentence 5: A circle of radius R and a tangent to it are given; construct a square so that its two adjacent vertices lie on the tangent, and the other two lie on a circle.

Such crossing-out can be divided into two groups:

- *Group 1.* These are deletions, in which the knowledge of the subject area – planimetry – is used;
- *Group 2.* Deletions related to the understanding of the process of text generation, to the memorization of the previous text, that is, to the understanding of the sentence.

4. Development of systems for evaluating the results of entering ellipses in planimetric texts

4.1 Possible assessment systems

The original system of evaluating with the use of the plugin of “Striking” had the following logic: all possible words are found, the exclusion of which will make an elliptical sentence from the complete one. Each word is assigned a share of the maximum score (100%), the shares for each word are equal. If only one word can be crossed out, its share is 100%. The respondent’s answer is checked for each correctly deleted word, the share of this word is added to the final score. Thus, if 3 words out of 4 possible ones are crossed out, the student receives 75% of the maximum score. Next, we will call this system of assessment SoA-0, since it is a pilot, such testing has not been carried out before, so the concept of the “correct” answer needed to be developed.

SoA-0 has a number of drawbacks, and the first drawback is the evaluation of crossing-out as “partially true”, when it actually violates the style of the sentence. For example, for question number 2: “A cube is inscribed in a regular quadrangular pyramid so that four its vertices belong to the lateral edges of the pyramid, and the remaining four its vertices belong to the plane of its base”. There are several correct (not violating the structure and style of the sentence) options for crossing out words to create an ellipsis (in Russian sentence, we have more options):

- A cube is inscribed in a regular quadrangular pyramid so that four of its vertices belong to the lateral edges of the pyramid, and the remaining four of its vertices **belong** to the plane of its base.

- A cube is inscribed in a regular quadrangular pyramid so that four of its vertices belong to the lateral edges of the pyramid, and the remaining four **of its** vertices **belong** to the plane of its base.
- A cube is inscribed in a regular quadrangular pyramid so that four of its vertices belong to the lateral edges of the pyramid, and the remaining **four of its** vertices **belong** to the plane of its base.

The second drawback is that it is not possible to assign a penalty for the crossing-outs that violate the meaning of the sentence from the point of view of geometric objects. The third disadvantage of SoA-0 is the impossibility of analyzing statistics by means of Moodle Quiz, since in order to process answers in the Moodle system, it is necessary that at least 1 answer is evaluated at 100%. Based on the above disadvantages of SoA-0, there is a need to develop a new assessment system for a Moodle quiz question type plugin "Striking". Let's consider several assessment systems.

- 1) SoA-1: Naive assessment. Estimation true/false (1/0) for understanding ellipticity is assigned when the respondent has crossed out a verb and additionally some repeated words of VPh. If there is a violation of the meaning of the sentence, the respondent is notified in addition to the ellipsis understanding score when displaying the test results.
- 2) SoA-2: Assessment with permitted set of words. The allowed set of words will be considered not an empty list of words that can be crossed out without losing the meaning of sentence. So, for the above question number 2, there are 5 correct sets (in Russian). Words included in the allowed sets form a list of "correct" words. However, not all combinations of "correct" words allow you to get a stylistically correct sentence after their removal.

That is why it is more logical not just to check the presence of each word crossed out by respondent in the list of "correct" words, but to compare the entire set of crossed out words with the "correct" subsets of this list. In addition to the list of "correct" words, a list of "forbidden" words is also formed, the crossing out of which violates the meaning of the sentence. If the respondent's answer contains at least one word from this list, he is given score equal to 0, even if the other words are in the correct set from the point of view of ellipsis.

The respondent is also notified of a penalty for violation of the meaning of sentence.

- 3) SoA-3: Assessment similar SoA-2 with an additional set of words to be crossed out. The logic of this system repeats SoA-2. In addition, for each sentence, a list of "acceptable" words is formed, which can be crossed out to avoid redundancy, but their deletion does not entail the appearance of a "gap" in the syntactic structure of sentence as opposed to the syntactic structure of an elliptic sentence.

According to this system, the words crossed out by respondents are divided into 4 categories: those included in the lists of "correct", "prohibited", "acceptable" and "other" words. If the number of "forbidden" and "other" words in the respondent's answer is zero, the set of words included in the list of "correct" words is checked for compliance with the "correct" subsets and the correct (1)/incorrect (0) score is given based on the results of the check. If the respondent's answer contains words belonging to the "prohibited" and "other" categories of words, the score is incorrect (0). This entails explaining to the respondent the reason for assessing his answer as incorrect.

4.2 Quality indicators of test tasks

In the Moodle learning management system (LMS) there are parameters for assessing questions not calculated in the pilot version but have been implemented in the current version of the system.

For each question, there are the following indicators [15]:

- 1) The facility index shows the part of respondents answering correctly to the analyzed question. If the permissible score for the task is zero (incorrect) or one (correct), then the

facility index (solvability coefficient) is the ratio of the number of respondents answered correctly to the total number of those who answered this question. It is better to exclude questions with the facility index of more than 0.9 and less than 0.1 [16]. The facility index will be further denoted by F_p .

- 2) Standard Deviation (SDp) characterizes the spread of the values of the scores received for a question. If this indicator is zero for the question, it means that all test takers received the same score for it. Such a question should be recognized as unsuccessful. In accordance with the requirements of the pedagogical theory of measurements, it is better to exclude tasks with the SDp value less than 0.3 from the test, because they do not have sufficient differentiating ability, i.e., they are not able to separate strong and weak respondents.
- 3) The discrimination index (Dp) shows the interrelation between the correctness of answer to a question and the correctness of answers to the rest of the test questions.
At the same time, it is noted that if the index F_p of question is different from 50%, then the index Dp cannot be 100%. If the F_p index is close to zero or 100%, then the Dp index will always be very small, and if the F_p index is equal to zero or 100%, the Dp index will not be determined.
- 4) Discriminative efficiency (DEp) is an indicator that is largely similar to the Dp index, but does not have the disadvantage mentioned above (the value of Dp may not be determined in some situations). Based on this fact, it was decided to consider only the DEp for each question. A question is considered to have sufficient differentiable capacity if the DEp index is greater than or equal to 30%.
- 5) Intended question weight is the weight that is assigned to the test when the test scenario is generated.
- 6) Effective weight characterizes the actual share of a particular task in the final score of respondents for the test.

4.3 Analysis of indicators in different SoA

Using the python programming language tools and formulas presented on the official moodle website, we found the values of the above coefficients. Table 1 shows the values of F_p , SDp and DEp for each question in different SoA. The intended weight for each question is 10%, since the maximum score equal 1 can be obtained for each question, and the total of score equal to 10 can be obtained for the test.

Table 1. Indicators of each question in different system of assessment (SoA).

	SoA-1				SoA-2				SoA-3		
	F_p	SDp	DEp, %		F_p	SDp	DEp, %		F_p	SDp	DEp, %
2	0.68	0.47	66.5		0.66	0.48	69.2		0.6	0.49	66.7
3	0.81	0.39	75.5		0.81	0.39	77.8		0.75	0.44	72.1
4	0.51	0.5	78.2		0.38	0.49	72.5		0.31	0.46	75.9
5	0.74	0.44	67.4		0.53	0.5	41.9		0.51	0.5	50.4
6	0.41	0.49	62.7		0.41	0.49	61.9		0.34	0.48	57.1
7	0.73	0.45	69.6		0.73	0.45	70.9		0.61	0.49	71.8
8	0.84	0.37	68.5		0.84	0.37	69.6		0.76	0.43	71.7
9	0.79	0.41	69.8		0.49	0.5	51.2		0.49	0.5	57.2
10	0.66	0.48	51.7		0.66	0.48	40.8		0.4	0.49	68.0
11	0.46	0.5	41.6		0.45	0.5	38.5		0.43	0.49	44.3

Depending on the SoA used, the values of the indicators change slightly, this fact may indicate the nature of the questions: for example, question number 11 in all SoA has relatively low Fp and DEp, these low values can characterize the question as difficult.

A low DEp of a question tells us that a certain group of “strong respondents” made mistakes in this question on a par with weak ones. Also, question number 6 in all SoA has the lowest solvability rate (value of Fp) and average (relative to other questions) value of DEp. Strong respondents coped with this question better than weak ones, compared to question 11, but the lowest The Fp index tells us that a large number of weak and average respondents did not solve this question.

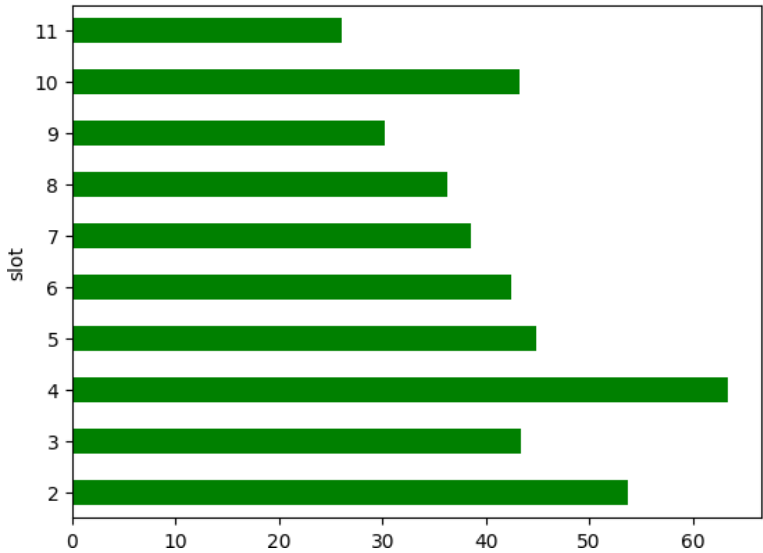


Fig. 1. Average time for each question.

One of the easiest questions is one with the number 3, it has the highest value of the Fp index. In addition to the highest value of the Fp index, it also has one of the highest values of the DEp index, this fact means that strong respondents coped with it much better than weak ones, most likely it was the weak respondents who made mistakes in this matter. Let’s compare the complexity of the questions with the time spent on. A graph of the distribution of the average time for answering each question is given in Fig. 1.

Question number 4 is highlighted on the graph. Let’s consider its indicators. According to the Fp index for SoA-1, half of the test participants solved the question, while it has the highest the DEp index, this fact may mean that it was mainly the strong half of respondents who coped with it. In SoA-2 and SoA-3, the Fp index of question 4 becomes the lowest one compared to the rest of questions. Recall that SoA-2 and SoA-3 imply penalties for violating the meaning of sentence. Just for this question, we have defined a set of “forbidden” words that cannot be crossed out. And it was these words that were crossed out by 19 people out of 78. It can be noted that a large amount of time spent on this question was not confirmed by high ratings. This question can be considered really difficult. The second observed high value of the average time spent was found in question with number 10. Judging by the Fp index in SoA-3, it can be concluded that quite a lot of respondents crossed out “extra” words.

The solution of the most difficult question with number 11, according to the indicators, took the least time. Perhaps it was because of the short thinking time that many respondents did not cope with the task. In addition, there is a possibility of a decrease in motivation by the end of testing.

What could be other reasons for the difficulty of question 4? The text of this question is given in sentence 6.

Sentence 6: On the side BC of triangle ABC, point A* is taken; the median perpendicular to line segment A*B intersects side AB at point M, and the median perpendicular to line segment A*C intersects side AC at point N.

Firstly, when reading, the respondents seem to “stumble” the designation *. Secondly, the concept of “median perpendicular” is quite difficult to imagine.

Let us consider question 10 in detail (sentence 7). A complex sentence structure consisting of repeated parts, so the respondents probably had to reread it several times to realize which shape touches the other and at what point. We can conclude that in order to search for redundancy, the respondents actually delved into the essence of the geometric problem, perhaps mentally imagining the described figures.

Sentence 7: 3 points A, B and C are given on the plane; construct three circles k1, k2 and k3 so that the circles k2 and k3 touch each other at point A, the circles k3 and k1 touch each other at point B, and the circles k1 and k2 touch each other at point C.

Another interesting task is the task with number 2 (sentence 8). In this text, the respondents deal with a three-dimensional figure and they try to visualize the configuration mentally. Also, the increase in the time spent working on this question can be explained by the fact that this is the first task of this type in the test.

Sentence 8: A cube is inscribed in a regular quadrangular pyramid so that four its vertices belong to the lateral edges of the pyramid, and the remaining four its vertices belong to the plane of its base.

5. Conclusion

The paper analyzes the results of two sessions of the experiment. The average score and the average time of passing the test were found.

The crossing-out that are often found among the respondents' answers are analyzed separately. As a result, it was established that the correct understanding of the ellipsis depends on the understanding of the entire sentence and knowledge of the subject area.

Three approaches to the assessment of testing in LMS Moodle are proposed, which involve assigning a score without taking into account the deletions that violate the meaning of sentences, as well as taking into account such deletions by assigning a penalty in the form of zeroing the estimation. The introduction of new estimation systems made it possible to calculate the performance indicators of questions given in LMS Moodle. Based on the values of the facility index and the discriminative efficiency, the authors were able to conclude about the difficulty and ease of the test questions.

An important indicator of the difficulty of question can be considered the time spent on it. When trying to compare the difficulty of question with the time spent on it, the following trend was noted: a large amount of time was taken up by questions containing geometric concepts or sentences with a complex structure that are difficult to mentally reproducing.

As a further development of the testing system, the authors plan implementing the most optimal assessment system in the question plugin, as well as adding the ability to create a “training” question, the answer to which can be seen during the test, so that respondents have the opportunity to try out the elimination functionality and, possibly, better understand the essence of testing. In addition, the authors intend to explore methods for generating elliptical sentences for the development of the task bank.

References

- [1]. E. Paducheva, About semantics of syntax. URSS, 2019, (in Russian).
- [2]. R. Jackendoff, Semantics and Cognition. MIT Press, 1983.
- [3]. “Information is in the mind of the beholder,” *Linguistics and Philosophy*, vol. 8, p. 23–33, 1985.
- [4]. “Grammar as evidence for conceptual structure,” *Linguistic Theory and Psychological Reality*, pp. 201–228, 1978.

- [5]. A. Putintseva, L. Kovrigina, and I. Shilin, "Automatic classification of elliptical constructions in Russian spontaneous speech," in R. Piotrowski's Readings in Language Engineering and Applied Linguistics, vol. 2233, 2017, pp. 120–133, (in Russian). Available: https://ceur-ws.org/Vol-2233/Paper_16.pdf, accessed 10.10.2024.
- [6]. X. A. Naidenova, "Cognitive elements in forming and understanding sentences," in Proceedings of the International Workshop on Digital Technologies for Teaching and Learning (DTTL-2021), Kazan, Russia, March 22-28, 2021, ser. CEUR Workshop Proceedings, O. A. Nevzorova, N. V. Loukachevitch, and E. K. Lipachev, Eds., vol. 2910. CEUR-WS.org, 2021, pp. 65–73. Available: <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-2910/short6.pdf>, accessed 12.10.2024.
- [7]. The national corpus of the Russian language. Available: <https://ruscorpora.ru/>, accessed 15.11.2024.
- [8]. Taiga corpus. Available: <https://tatianashavrina.github.io/taiga/site/>, accessed 15.11.2024.
- [9]. X. Naidenova, E. Bulykina, V. Parkhomenko, A. Schukin, and T. Martirova, "Developing computer system for testing cognitive respondents' abilities based on elliptical sentences", "Electronic Libraries", Scientific Publications of KFU – Kazan (Volga Region) Federal University, vol. 26, no. 3, pp. 340–364, 2023, (in Russian). Available: <https://rdl-journal.ru/article/view/784/856>, accessed 12.10.2024.
- [10]. X. Naidenova, V. Parkhomenko, E. Bulykina, A. Schukin, Y. Lizunov, and T. Martirova, "A system of software tools for investigation of the cognitive function of elliptical sentence generation," in 2023 Ivannikov ISPRAS OPEN Conference (ISPRAS), 2023, pp. 112–117.
- [11]. V. Parkhomenko, X. Naidenova, and A. Martirova, T. and Schukin, "Ex- perimental study of cognitive function of generating elliptical sentences in planimetric tasks", "Electronic Libraries", Scientific Publications of KFU – Kazan (Volga Region) Federal University, vol. 27, no. 3, pp. 316–335, 2024, (in Russian).
- [12]. A. Gvozdev, Modern Russian literary language. Part II. Moscow: Uchpedgis, 1958, (in Russian).
- [13]. G. Zhao, "A cognitive approach to ellipsis," Theory and Practice in Language Studies, vol. 6, p. 372, 02 2016.
- [14]. P. Culicover and R. Jackendoff, Simpler Syntax, ser. Oxford linguistics. OUP Oxford, 2005.
- [15]. S. Nesterov and M. Smetanina, "Assessment of the quality of test tasks by means of the moodle distance learning environment", Computer Science, Telecommunications and Management, no. 5(181), pp. 87–92, 2013, (in Russian).
- [16]. I. Protasova, A. Tolstobrov, and I. Korzhik, "Methodology for analyzing and improving the quality of tests in the moodle e-learning system," Vestnik of the Voronezh State University. Series: System Analysis and Information Technology, no. 3, pp. 61–72, 2014, (in Russian).

Информация об авторах / Information about authors

Ксения Александровна НАЙДЁНОВА — кандидат технических наук, старший научный сотрудник Санкт-Петербургской медицинской академии с 1995 года. Сфера научных интересов: машинное обучение, формальный концептуальный анализ, моделирование правдоподобных рассуждений, извлечение знаний из естественно-языковых текстов, когнитивные исследования в лингвистике.

Xenia Aleksandrovna NAIDENOVA – Cand. Sci. (Tech.), Senior Researcher of the St. Petersburg Medical Academy since 1995. Research interests: machine learning, formal concept analysis, modeling of plausible reasoning, mining natural language texts, cognitive research in linguistics.

Елена Сергеевна БУЛЫКИНА — студент магистратуры по направлению прикладная информатика Санкт-Петербургского Политехнического университета. Сфера научных интересов: обработка естественного языка, большие языковые модели, дизайн интерфейсов.

Elena Sergeevna BULYKINA — master's degree student in Applied Computer Science at Peter the Great St. Petersburg Polytechnic University. Research interests: natural language processing, large language models, interface design.

Татьяна Александровна МАРТИРОВА – младший научный сотрудник Санкт-Петербургской медицинской академии с 1987 года. Сфера научных интересов: гигиена, статистика.

Tatyana Aleksandrovna MARTIROVA – Junior Researcher of the St. Petersburg Medical Academy since 1987. Research interests: hygiene, statistics.

DOI: 10.15514/ISPRAS-2025-37(3)-14



Glottochronology Classification of the Modern and the Earliest Samoyed Dictionaries using LingvoDoc Programs

Ju.V. Normanskaja, ORCID: 0000-0002-2769-9187 <julianor@mail.ru>

*Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

*Institute of Linguistic RAS
1, B. Kislovskiy lane. Moscow 125009, Russia.*

Abstract. The classification of Samoyedic languages has become one of the most popular topics in Uralistics in recent years, with at least six different perspectives expressed by leading experts, often in contradiction with one another. On the LingvoDoc platform, there are 16 dictionaries and concordances of texts in Samoyedic languages. Among these, 10 dictionaries – Nenets, Enets, Nganasan, and Selkup dialects – were compiled from native speakers, while six others were derived from archival and published sources. They are analyzed using the glottochronology formula developed by S.A. Starostin. The analysis on LingvoDoc results in a 3D graph that depicts the degree of temporal proximity regarding the divergence of Samoyedic languages and dialects. It was determined that, from a glottochronological perspective, there was a certain proximity between Nenets, Enets, and Nganasan, that are traditionally grouped into the North Samoyedic cluster, while Selkup, Mator, and Kamasin, are regarded as South Samoyedic. However, these commonalities existed for a relatively short period. A longer period of unity was observed between Mator and Kamasin languages and between Nenets and Enets. The highest number of words with no etymology in other lists of basic vocabulary was found in Selkup dialects and in the Nganasan language, indicating their prolonged isolated existence. The analysis conducted in this study supports the validity of the traditional classification of Samoyedic languages. Considering the material from early Selkup texts provides more reliable evidence for postulating the South Samoyedic group.

Keywords: Samoyed languages; Nenets; Enets; Nganasan; Selkup; Mator; Kamasin; classification; field research; archival data.

For citation: Normanskaja Ju.V. Glottochronology classification of the modern and the earliest Samoyed dictionaries using LingvoDoc programs. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 3, 2025, pp. 195-210. DOI: 10.15514/ISPRAS-2025-37(3)-14.

Acknowledgements. The work was supported by grant RSCF No 25-78-20002.

Глоттохронологическая классификация современных и наиболее ранних словарей самодийских языков на платформе ЛингвоДок

Ю.В. Норманская, ORCID: 0000-0002-2769-9187 <julianor@mail.ru>

*Институт системного программирования им. В.П. Иванникова РАН,
Россия, 109004, Москва, ул. Александра Солженицына, д. 25.*

*Институт языкознания РАН
Россия, 125009, Москва, Б. Кисловский пер., д. 1.*

Аннотация. Классификация самодийских языков в последние годы является одной из наиболее спорных тем исследования. В последние десятилетия предложено как минимум 6 различных классификаций, которые противоречат друг другу. Нет ни одной дочерней группы самодийских языков, по поводу существования которой все авторы были бы согласны. В статье проанализированы как современные словари, собранные в экспедициях от последних носителей, так и наиболее ранние словники самодийских языков, доступные в глобальной сети посредством платформы ЛингвоДок.

Ключевые слова: самодийские языки; ненецкий; энецкий; нганасанский; селькупский; маторский; камасинский; классификация; полевые исследования; архивные данные.

Для цитирования: Норманская Ю.В. Глоттохронологическая классификация современных и наиболее ранних словарей самодийских языков на платформе ЛингвоДок. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025, pp. 195-210 (на английском языке). DOI: 10.15514/ISPRAS-2025-37(3)- 14.

Благодарности. Работа поддержана грантом РНФ № 25-78-20002.

1. Introduction

The classification of Samoyedic languages is one of the most popular topics in Uralistics in recent years, with at least six contradicting viewpoints expressed by leading specialists. In fact, there is not a single sub-group of Samoyedic languages on the existence of which all authors would agree.

Currently, the LingvoDoc platform [1] features a tool called the "Glottochronological Analysis of Languages", which analyzes wordlists of basic vocabulary using the methodology proposed by S.A. Starostin [2]. Online wordlists of modern Samoyedic languages based on archival materials from the 18th to the early 20th century, and recorded in audio form the last speakers of the languages, are available. It is believed that the glottochronological analysis of these sources, some of which were recorded from native speakers, will clarify the situation regarding conflicting classifications.

This work consists of two parts. The first part provides a brief analysis of existing classifications, while the second part presents the results of the analysis of 16 lists of basic vocabulary using S.A. Starostin's formula, which is integrated into the LingvoDoc platform.

2. Overview of Existing Classifications of Samoyedic Languages

The "Traditional" classification (see Fig. 1) was primarily based on the geographical location of the Samoyedic speakers, e.g., [3]. According to it, a Northern Samoyedic languages group is distinguished, which consists of: Nenets, Enets, Nganasan, and Southern Samoyedic languages: Selkup and Sayan Samoyedic.

This classification remains the most widely accepted to this day, see [4].

In the classification proposed by E.A. Helimski in [5], which is based on the analysis of a small list by M. Swadesh, consisting of 92 words of basic vocabulary in six Samoyedic languages, he concluded that the Southern Samoyedic languages did not form a group but separated independently during the divergence of the Proto-Samoyedic language.

In a 2004 report by S.A. Starostin (see [6]), wordlists collected by E.A. Helimski and supplemented in 2001 based on existing dictionaries were analyzed using a special formula developed by S.A. Starostin to refine the timing of divergence. As a result, he obtained results that essentially coincided with the traditional classification (see Fig. 1). The dating obtained by S.A. Starostin can be seen in Fig. 2 (the numbers in the figure indicate the date of the Proto language split, negative numbers indicate that the time of the splitting was BC). However, he came to the conclusion of the absence of Southern Samoyedic unity and the earliest divergence of the Selkup language.

In 1987, H. Katz expressed the opinion that Selkup and Kamasin should be grouped together, with Mator considered as the earliest separated Samoyedic language (see Fig. 3).

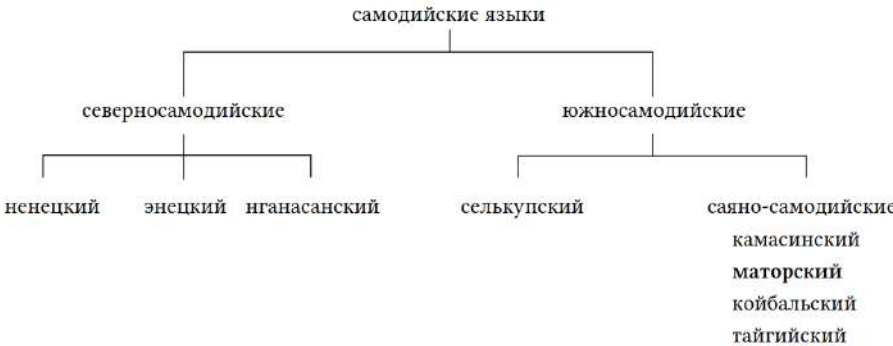


Fig. 1 "Traditional" classification of Samoyed languages.

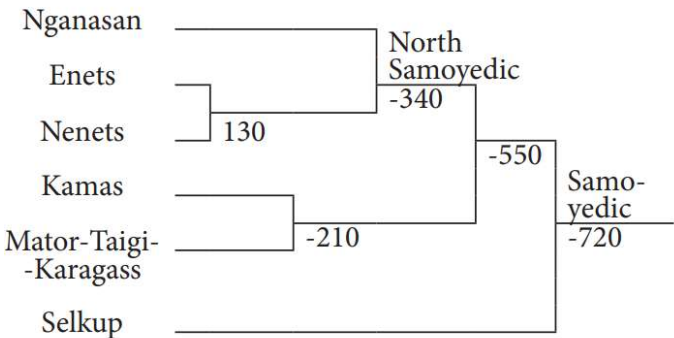


Fig. 2. Time of divergence of Samoyedic languages according to [6].

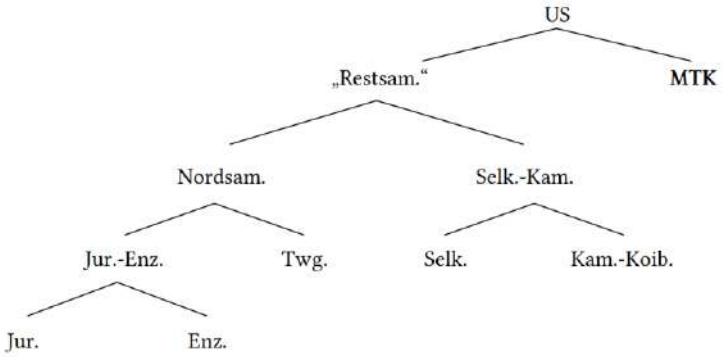


Fig. 3. Classification of Samoyedic languages according to [7].

In 1998, J. Janhunen proposed a different classification [8]. According to him, the Nganasan language separated first from the Proto-Samoyedic language, followed by Mator, and only then did the gradual divergence of other daughter languages occur, see [8] and Fig. 4.

J. Janhunen points out that this scheme corresponds well to the geographical locations of the language speakers, as the speakers of Nganasan and Mator languages lived in the northeastern and southeastern territories of the Samoyedic language speakers.

In a report presented at a meeting of the Finno-Ugric Society in Helsinki in 1997, immediately after the publication of the Mator etymological dictionary [9], E.A. Helimski noted that the Mator language should be grouped together with Nenets and Enets languages [10]. This conclusion arose from an analysis of all available wordlists of the Mator language, which demonstrated that the largest number of lexical isoglosses linked the Mator language with Enets and Nenets. Furthermore, they also shared common phonetic and morphological isoglosses. Given the results he had previously obtained by analyzing 92-word lists, E.A. Helimski leaned towards the idea that the classification of Samoyedic languages changed over time due to migrations and areal contacts. As a proposed solution, he introduced a scheme consisting of three periods, as illustrated in Fig. 5.

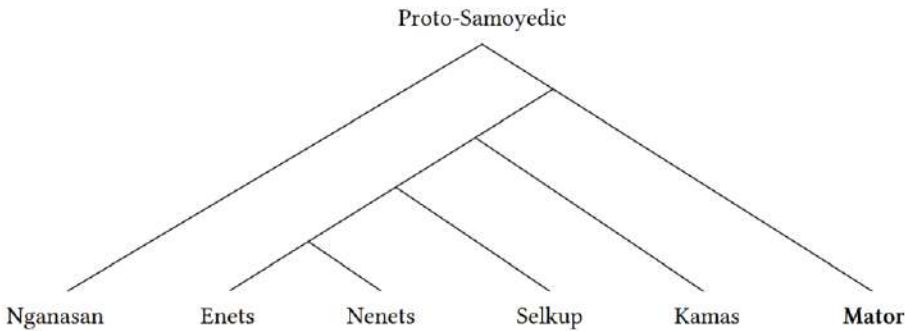


Fig. 4. Classification of Samoyedic languages according to [8].

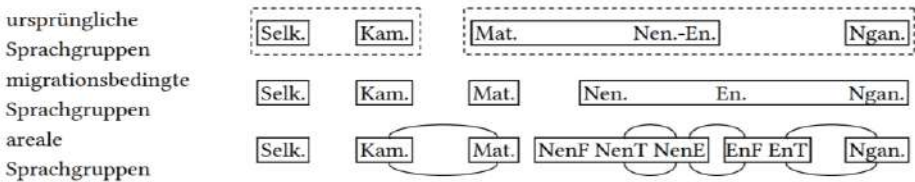


Fig. 5. Classification of Samoyedic languages according to [10].

In Helimski's work, the main argument for the original proximity of Mator to the Nenets-Enets group was the number of shared Proto-Samoyedic etymologies for which there are no reflexes in Kamasin and Selkup. He found 96 such lexemes in Y. Janhunen's dictionary [11], of which 54 have no Nganasan parallels. Meanwhile, according to [11], there are only 34 shared etymologies between Kamasin and Northern Samoyedic languages, excluding Mator and Selkup. In a report in 1997, E.A. Helimski suggested that this could indicate the existence of a Mator-Nenets-Enets group immediately after the divergence of the Proto-Samoyedic community, with 54 shared words being its legacy [10].

However, it should be noted that this work was prepared by E.A. Helimski in 1997 and was not published during the subsequent 10 years of his life. It can be assumed that this was not accidental. The reason could be that the preparation and publication of new sources on Samoyedic languages, including those by Helimski himself on Nganasan [12], Enets [13], Northern Selkup [14-15], and a

large dictionary on Southern and Central Selkup dialects created by the followers of A.P. Dulzon (see [16]) had significantly expanded the possibilities for finding new Samoyedic etymologies. It turns out that, on the one hand, for some Proto-Samoyedic words that, according to [11], had commonly known reflexes only in Nenets, Enets, and Mator languages and were part of the list consisting of 54 lexical isoglosses, reflexes can also be found in Southern and Central Selkup dialects according to [16]. Here are some examples: Proto-Samoyedic **kaj*- 'to go' (> Nenets, Mator) [11] can be compared with Selkup Ob', Тым., Вас. *квая-гу*, Вас. Тым. *квое-гу*, *квоя-гу* 'to go, to ride' [16], Ob', Narym. *ква-гу*, *квас-гу* 'to go, to ride' [16]; Proto-Samoyedic **lapta* 'flat, level, plain' (> Enets, Nenets, Mator) [11] can be compared with Selkup. Narym., Тым. *ланчаль* 'small, flat' [16], also see further Selkup dialectal forms in [17].

On the other hand, E.A. Helimski prepared but did not publish a dictionary of Northern Samoyedic nouns and adjectives, which was compiled and supplemented with modern forms by J.V. Normanskaja and V.Y. Gusev. It is currently available online on the LingvoDoc platform at the link [18]. It contains 836 Northern Samoyedic stems, some of which are well-known and have parallels in other Samoyedic languages, including Mator, Kamasin, and Selkup. According to [11], 325 words in this list have such parallels. At the same time, 511 Northern Samoyedic stems do not have commonly known parallels in Selkup, Kamasin and Mator languages. Out of these, 394 lexemes have parallels in Nganasan and Nenets and/or Enets, while 117 have parallels only in Nenets and Enets. It should be noted that these numbers only concern stems of nouns and adjectives, the analysis of verbs would likely increase this count significantly. Thus, it is evident that when analyzing complete dictionaries, the number of shared lexemes that are found exclusively in Nenets, Enets, and Nganasan is significantly higher than in Nenets, Enets, and Mator (394 based only on nouns and verbs and 96 based on the entire Mator dictionary). Certainly, this may be related to the fact that Nganasan dictionary, prepared by E.A. Helimski (see [12]), is much larger than the dictionary of the already-extinct Mator language (see [9]). However, it is clear that the calculations presented in [10] cannot be considered to be a definitive argument for deciding which classification is correct.

In 2016, V. Blažek, in an article [19] dedicated to the memory of E.A. Helimski, turned to the topic of analyzing the basic vocabulary of Samoyedic languages. As mentioned earlier, between 1982 and 2016, a significant number of new, more comprehensive dictionaries of specific Samoyedic languages and etymological dictionaries of those languages were published. Based on these, V. Blažek created new lists of basic vocabulary. The results of the calculations he obtained can be seen in Fig. 6 and Fig. 7.

In this scheme, the maximum and minimum possible dating are indicated on the right and left respectively, depending on which synonym is considered to be the base word.

Below in Fig. 7, the scheme of Samoyedic language divergence is presented as a diagram that reflects the percentages of similarity between all languages.

We can see that the classification proposed by V. Blažek based on glottochronological calculations essentially coincides with the traditional classification (see Fig. 1).

In 2023, A.Y. Urmanchieva in her doctoral dissertation [20], raises the question of why the conclusions presented by E.A. Helimski in his 1997 report [10] about the closeness of the Mator language to Nenets and Enets were not confirmed by the works on glottochronology published by S.A. Starostin [6] and V. Blažek [19]. She also creates lists of basic vocabulary, but for nine Samoyedic languages, adding data on Forest Nenets, Forest Enets, and Southern Selkup languages to the calculations. As a result of the analysis of two lists: M. Swadesh's list and a wordlist obtained in the "Loanwords Typology" project – the so-called Leipzig-Jakarta list, published in [21], A. Y. Urmanchieva concludes that the Mator language formed a special subgroup with the Nenets language, a Proto-Nenets-Mator subgroup. According to the author, in order to classify the languages correctly, four levels of communication between the speakers of those languages must be taken into account: the ancient contacts, the later contacts, the most recent contacts and the historical epoch of contacts. In summary, it can be stated that the conclusions of A.Y. Urmanchieva regarding the classification of

the Samoyedic languages at different time periods are similar to the report by E.A. Helimski in 1997, with the exception that she postulates a closer affinity between the Nenets language and Mator, and not with Enets. Furthermore, while the classification of the Samoyedic languages immediately following the divergence of the proto-language by A.Y. Urmanchieva is based on the analysis of lists of basic vocabulary, it significantly differs from both the initial classification by E.A. Helimski proposed in 1982 [5], also based on lists of basic vocabulary, and the results by S.A. Starostin (see Fig. 2) and V. Blažek (see Fig. 6, 7), obtained from materials of contemporary dictionaries. The numbers in the Fig. 6 indicate the date of the Proto language split. Negative numbers indicate that the time of the splitting was BC. The percentages show how many percent of the words from the hundred-word list have common origin. The question arises as to how several authors could obtain such different results when applying the same method?

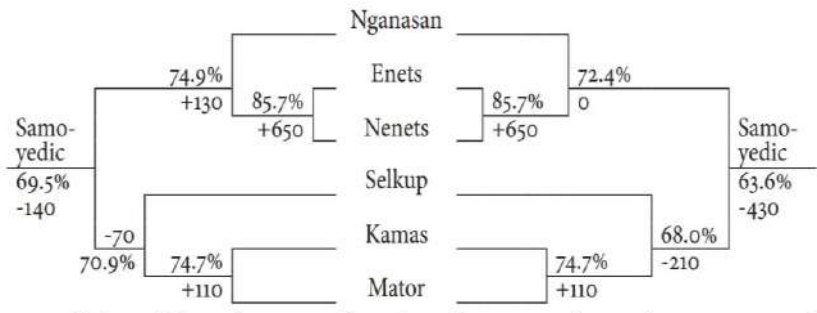


Fig. 6. Time of divergence of Samoyedic languages according to [16].

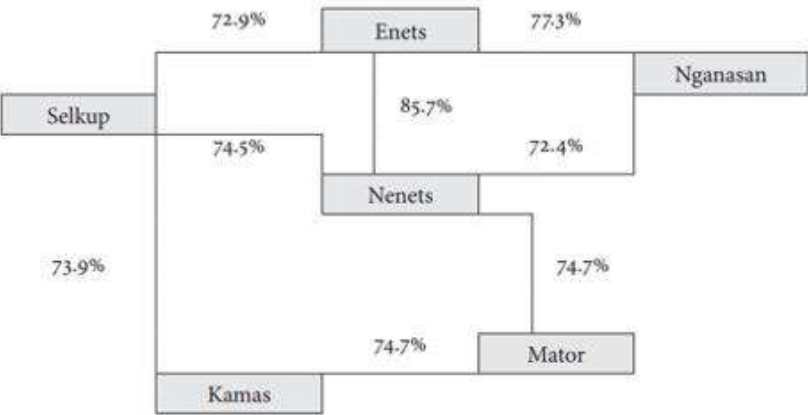


Fig. 7. Percentage of shared basic vocabulary in Samoyedic languages according to [16].

According to E.A. Helimski's calculations, the percentage of correspondences between Tundra Nenets and Mator is 58.1%, while between Tundra Nenets and Tundra Enets, it is 74.7%. According to V. Blažek, the correspondence between Tundra Nenets and Mator is 74.7%, and between Tundra Nenets and Tundra Enets, it is 85.7%. However, according to A.Y. Urmanchieva, the correspondence between Tundra Nenets and Mator is 74.19%, and between Tundra Nenets and Tundra Enets, it is 69.23%. Is this related to the fact that A.Y. Urmanchieva used a different approach in analyzing the 100-word dictionary of Mator? In particular, "when synonyms expressing the same meaning are present, the presence of a cognate in another Samoyedic language for a Mator word is noted, but the absence of a cognate for one of the synonyms in Mator is not noted" (see more in [20]). She indicates that this method yields "higher percentages of correspondence in the Mator column, so these data can only be compared with other data within this column, but not with other columns". However, the question arises as to how to compare the similarity figures between Tundra Nenets and Tundra

Enets and between Tundra Nenets and Mator in this case. Can one postulate the existence of a special Nenets-Mator group based on results obtained by different methods? Besides the slightly modified method used by A.Y. Urmanchieva, uses different lists of basic vocabulary, all of which are available online: those of E.A. Helimski, V. Blažek, A.Y. Urmanchieva, and Y.B. Koryakov, who in 2018 analyzed the similarity of only Nenets, Enets, and Selkup languages (see [22]). Firstly, A.Y. Urmanchieva removed words from M. Swadesh's list, such as 'bird' (lacking a single-word translation, it is described as 'flying animal'), 'seed' (weakly represented), 'leaf' (weakly represented), 'feather' (difficulties in choosing a neutral term), 'green' (lack of a stable translation for this color), 'yellow' (lack of a stable translation for this color), 'round' (difficulties in choosing a neutral term), see [22]. Secondly, for the Mator language, for example, our knowledge of its lexicon was significantly supplemented by the dictionary [9], which includes some words that were absent from the lists of basic vocabulary published in [5], and the etymologies of all words were clarified. For instance, E.A. Helimski provides the Mator word *мчелегаде* meaning 'root', and his 1982 work does not indicate that this is a Khakas loanword, see [9]. Additionally, the Mator dictionary [9] provides another word for 'root' that has a Proto-Samoyedic etymology PS **wanco* 'root' [11], the reflexes of which are Tundra Nenets *вано* and Tundra Enets *badu* ('root'). Consequently, such cases allowed A.Y. Urmanchieva, V. Blažek, and S.A. Starostin, to increase the percentage of similarity between Nenets and Mator languages compared to the study by [5]. However, upon careful examination of the lists, it becomes clear that the differences in them are not solely due to the use of more recent dictionaries. We will illustrate this with the first word from the list of basic vocabulary and only three languages: Nenets, Enets, and Selkup, which are present in four works: 'all':

- [5]: Nenets *тюку*, Enets *t'uku*, Selkup. *munťik*.
- [19]: Nenets *t'uku*, *t'ubea*, *юкса*, *mal*, Enets *t'uku*, *kurhari*. Baj Selkup, Taz *mündej*, Upper Ob, Chaya *sielaŋ*, Middle Ob, Ket, Pumpokol *wes*.
- [22]: Forest Nenets *чуней* 'all, complete; completely', Tundra Nenets: *сямян* 'all', *мал* 'all'; Two or three different roots. Tundra Enets *t'uku*, *t'ukut'ii*; *t'uku*, *t'ukotfii*. Forest Enets *t'uku*, *t'ukotfii*. Selkup Taz *munťij* 'all, everybody'; *мунтык/у* 'all, everybody', Narym, Ob Chumylkup *вес* / *вэс*, Ket *tīr* 'all, everybody'.
- [20]: Tundra Nenets *tyuku*^o, Tundra Enets. *t'ukut'ii*, *t'uku*, Selkup North. *мунтык/у*, South. *tāk əl'*.

It becomes evident that among the considered lists, the data used by V. Blažek differ the most. This is because he combines data from all available dictionaries and dialects in a single column. Furthermore, he does not conduct calculations separately for each dialect but treats forms from different dialects as synonyms. Therefore, it's apparent that the percentage of common words for different languages is significantly higher than that of other researchers.

We can observe in this example that the lists of E.A. Helimski and A.Y. Urmanchieva are similar to each other. This tendency generally holds for other words as well, except for the fact that A.Y. Urmanchieva also uses materials from three additional languages (Forest Nenets, Forest Enets, South Selkup) and takes into account materials from more recent dictionaries.

The difference between the lists of E.A. Helimski and A.Y. Urmanchieva compared to the materials of Y.B. Koryakov in this example lies, firstly, in the choice of Tundra Nenets word. In this case, it is impossible to determine whose list is more accurate without conducting surveys of native speakers since all three words in the dictionary [22], *мал*, *сямян*, and *тюку* seem to have the meaning 'all' based on their contexts. Regarding South Selkup, different forms are also listed in Y.B. Koryakov's and A.Y. Urmanchieva's lists, e.g., [22]: *tīr*, [20]: South Selkup. *tāk əl'*. However, in the dictionary [16], which is based on a card index created by A.P. Dulzon and his students, and includes over 25,000 lexemes with contexts, the most common South Selkup (Ket) word for 'everything, all' is a Russian borrowing *вес*, and the same is indicated only in V. Blažek's list.

Undoubtedly, in a situation when extensive dictionaries containing numerous synonyms but lacking easily accessible corpora or a literary norm, it is necessary to conduct a survey of native speakers to make a more reliable choice for the word in the Swadesh list. However, only in the work of Y.B. Koryakov [22], it is mentioned that the material was not collected from existing dictionaries but rather through surveys of speakers, which were conducted for the Nenets language by M.K. Amelina, for the Enets language by A.B. Shluinsky and O.V. Khanina, and for the Narym Selkup language by N.L. Fedotov [1]. Certainly, E.A. Helimski and A.Y. Urmanchieva also relied on their own field materials, but it is not always clear whether a lexeme in their lists was taken from dictionaries or obtained through surveys of speakers.

Therefore, in further work, we will be relying on word lists presented online on the LingvoDoc platform [1] collected from native speakers and already connected by etymological links.

3. Analysis of Samoyedic languages' basic vocabulary lists using the LingvoDoc platform

Currently, on LingvoDoc, any user can create their own dictionaries and/or corpora and analyse those created by other users, if they were made available to the public. In dictionaries, under the "Tools" tab, there is an option for "Glottochronological analysis of languages". This option can be applied to any set of languages in the dictionaries of which more than 50 words from M. Swadesh's Swadesh list are present. The 100-word Swadesh - Starostin list [23] is chosen because it is the only one for which strict semantic specifications have been developed and justified [23-24], allowing for sufficiently precise analysis of comparable data for different languages.

For many words in the Swadesh list, speakers of Samoyedic languages provide several synonyms. For a more precise procedure, a survey was conducted using a questionnaire with sentence examples for each word [25].

According to the proposed glottochronology by S.A. Starostin, see more details [26], starting with the calculation using the formula presented above, loanwords are first removed, then related words are connected on the LingvoDoc platform by etymological links, after that the percentage of matches between the lists of two idioms is calculated, and the time of divergence is determined [26].

This formula, for which S.A. Starostin found an experimentally determined lambda of 0.05, allows for determining the time of divergence of any set of languages, see more [26].

LingvoDoc also includes a feature for creating language similarity graphs in 2D and 3D formats.

Currently, on LingvoDoc, basic vocabulary lists are available for:

- 1) Tundra Nenets, created based on the dictionary [27] and a survey of speakers from the Yamal Tundra conducted by M.K. Amelina in 2017.
- 2) Nenets, created based on the dictionary by A.A. Dunin-Gorkavich, 1910.
- 3) Forest Nenets, created based on a survey conducted by M.K. Amelina in the village of Khalyasavey in 2015.
- 4) Tundra Enets, created in 2015 based on a survey in Dudinka by O.V. Khanina.
- 5) Forest Enets, created in 2011 based on a survey in Dudinka by O.V. Khanina.
- 6) Nganasan, created in 2022 based on a survey in Dudinka.
- 7) Taz Selkup, created in 2022 based on a survey in Krasnosel'kup.
- 8) Surgut (Taz) Selkup, collected by P.S. Pallas in the 18th century.
- 9) Taz Selkup, created based on the dictionary [14] and a survey of speakers conducted by O.A. Kazakevich in the village of Bystrika in 2011.
- 10) Ket Selkup, collected by P.S. Pallas in the 18th century.
- 11) Ket Selkup, recorded by E.M. Budyanskaya and O.A. Kazakevich in the village of Katayga in 2010.

- 12) Lower Chaya Selkup, created based on the works of N.P. Grigorovsky from 1879.
- 13) Narym Selkup, created based on recordings by Y.A. Morev, made in the village of Laskino in 1967-1970.
- 14) Narym Selkup, created based on recordings by N.L. Fedotova, made in the village of Parabel in 2009.
- 15) Mator language, created based on the dictionary [9].
- 16) Kamas Selkup, created based on the dictionary [28].
- 17) Kamas concordance based from Kamas corpus glossed in INEL project [29].

The glottochronological analysis of these lists resulted in the following 3D graph representing the degree of proximity in the divergence time of Samoyedic languages and dialects, see Fig. 8.

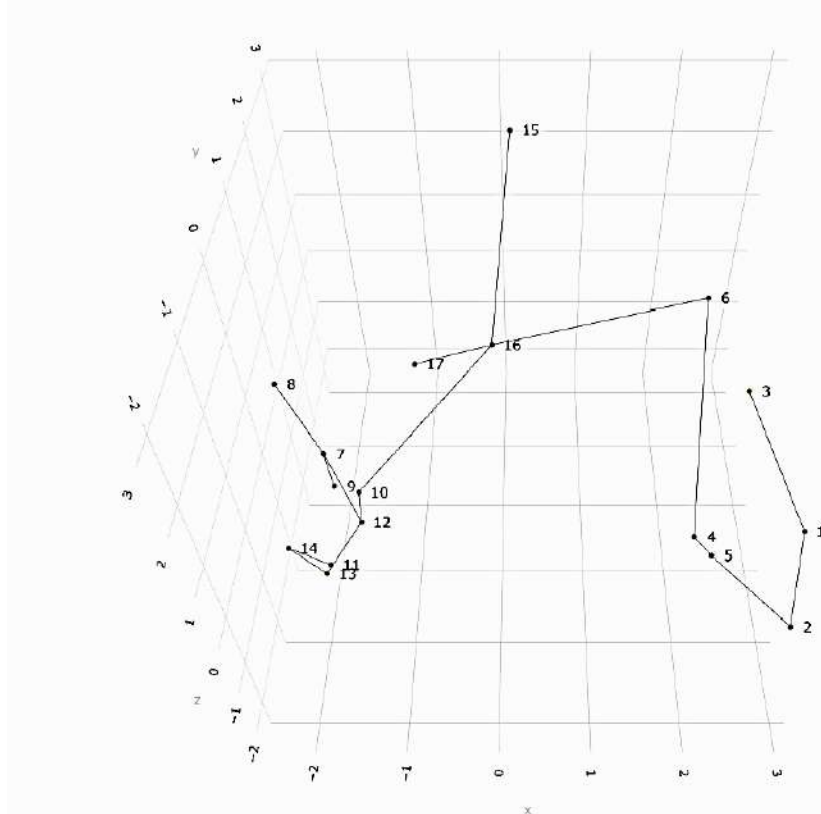


Fig. 8. Graph of the degree of proximity in the divergence time of Samoyedic languages.

In digital format, the obtained results are as follows, see Table 1. The first digit in the table represents the time in millennia that has passed since the divergence of two languages or dialects. The second digit reflects the percentage of etymologically related words that match. From the table, it can be observed that when analyzing complete Swadesh lists (excluding monuments from the 18th century, where a lower percentage of similarities is often observed, possibly due to the incomplete state of those lists), the lowest percentage of similarities (55-56%) in words is found between Nganasan and Southern and Central Selkup dialects. According to S.A. Starostin's formula, they diverged approximately 2700-2800 years ago, i.e., in the early 1st millennium BCE. This percentage is higher than that of A.Y. Urmanchieva (between Nganasan and Selkup – 40.8%), see [20], and significantly lower than those of V. Blažek (63.6-69.5%, numbers represent the results of different synonym calculations). This is likely related to the principles of selecting basic vocabulary words. The obtained

result is nearly identical to the percentage obtained by E.A. Helimski in 1982 (between Nganasan and Selkup – 54.6%) and with the dating of the Proto-Samoyedic language divergence proposed by S.A. Starostin (see Fig. 2).

Based on the obtained results, we will attempt to answer the questions raised by other researchers:

1) **Did Northern Samoyedic language group, consisting of Nenets, Enets, and Nganasan languages exist?** Or, as claimed by E.A. Helimski in the 1997 report, is the Mator language closer to Nenets and Enets than Nganasan? It turns out that while the proximity of Mator to Nenets is slightly higher (65-67%) than that of Nganasan (63-66%), there is a significant difference in proximity with Enets: Mator with Enets (66-68%) and Nganasan with Enets (72-75%). Therefore, according to S.A. Starostin's methodology, it is reasonable to postulate a Northern Samoyedic language group. Is it possible that the proximity between Nganasan and Enets languages has a contact nature? Let's examine which words in the basic vocabulary form Nganasan-Nenets-Enets isoglosses to understand their nature, whether they are reflexes of Proto-Samoyedic roots or the result of later contacts.

Nganasan-Nenets-Enets parallels in the Swadesh list with no corresponding lexemes in the basic vocabulary of other languages, see Table 2.

Table 1. The time of diversity and the percent of the common words in Samoyed languages.

	1. Tundra Nenets	2. Nenets 1910	3. Forest nenets	4. Tundra Enets	5. Forest Enets	6. Ngana san	7. Taz Selkup	8. Taz Selkup XVIII century	9. Upper Tolka Selkup	10. Ket Selkup XVIII century	11. Ket Selkup	12. Lower Chaya 1879 r.	13. Laskino Selkup 1967	14.Narym Selkup	15. Kamasin Selkup	16. Mator language
1	n/a	1.17 (87%)	1.32 (85%)	1.89 (73%)	1.61 (79%)	2.40 (63%)	2.53 (60%)	2.61 (59%)	2.71 (57%)	2.39 (63%)	2.54 (60%)	2.39 (63%)	2.61 (59%)	2.59 (59%)	2.54 (60%)	2.29 (65%)
2	1.17 (87%)	n/a	1.79 (75%)	1.92 (73%)	1.67 (78%)	2.50 (61%)	2.72 (57%)	2.67 (58%)	2.71 (57%)	2.78 (56%)	2.83 (55%)	2.29 (65%)	2.84 (54%)	2.75 (56%)	2.55 (60%)	2.67 (58%)
3	1.32 (85%)	1.79 (75%)	n/a	2.10 (69%)	1.82 (75%)	2.23 (66%)	2.53 (60%)	2.51 (61%)	2.48 (61%)	2.45 (62%)	2.48 (61%)	2.39 (63%)	2.44 (62%)	2.53 (60%)	2.21 (67%)	2.21 (67%)
4	1.89 (73%)	1.92 (73%)	2.10 (69%)	n/a	0.92 (92%)	1.82 (75%)	2.34 (64%)	2.26 (66%)	2.51 (61%)	1.85 (74%)	2.50 (61%)	2.23 (66%)	2.49 (61%)	2.49 (61%)	2.19 (67%)	2.15 (68%)
5	1.61 (79%)	1.67 (78%)	1.82 (75%)	0.92 (92%)	n/a	1.93 (72%)	2.46 (62%)	2.51 (61%)	2.60 (59%)	2.20 (67%)	2.61 (59%)	2.30 (65%)	2.51 (61%)	2.59 (59%)	2.25 (66%)	2.25 (66%)
6	2.40 (63%)	2.50 (61%)	2.23 (66%)	1.82 (75%)	1.93 (72%)	n/a	2.44 (62%)	2.39 (63%)	2.66 (58%)	2.16 (68%)	2.80 (55%)	2.60 (59%)	2.84 (55%)	2.78 (56%)	2.21 (67%)	2.17 (67%)
7	2.53 (60%)	2.72 (57%)	2.53 (60%)	2.34 (64%)	2.46 (62%)	2.44 (62%)	n/a	1.23 (86%)	1.04 (92%)	1.04 (90%)	1.13 (88%)	1.14 (88%)	1.29 (85%)	1.31 (85%)	2.19 (67%)	2.21 (67%)
8	2.61 (59%)	2.67 (58%)	2.51 (61%)	2.26 (66%)	2.51 (61%)	2.39 (63%)	1.23 (86%)	n/a	1.47 (82%)	1.71 (77%)	1.75 (76%)	1.64 (78%)	1.66 (78%)	1.85 (74%)	1.77 (76%)	1.94 (72%)
9	2.71 (57%)	2.71 (57%)	2.48 (61%)	2.51 (61%)	2.60 (59%)	2.66 (58%)	0.89 (92%)	1.47 (82%)	n/a	1.74 (76%)	1.41 (83%)	1.10 (89%)	1.33 (85%)	1.43 (83%)	2.13 (68%)	2.36 (64%)
10	2.39 (63%)	2.78 (56%)	2.45 (62%)	1.85 (74%)	2.20 (67%)	2.16 (68%)	1.04 (90%)	1.71 (77%)	1.74 (76%)	n/a	1.18 (87%)	0.99 (91%)	1.37 (84%)	1.18 (87%)	1.82 (75%)	1.80 (75%)
11	2.54 (60%)	2.83 (55%)	2.48 (61%)	2.50 (61%)	2.61 (59%)	2.80 (55%)	1.13 (88%)	1.75 (76%)	1.41 (83%)	1.18 (87%)	n/a	1.11 (89%)	0.98 (91%)	0.91 (92%)	2.38 (63%)	2.00 (71%)
12	2.39 (63%)	2.29 (65%)	2.39 (63%)	2.23 (66%)	2.30 (65%)	2.60 (59%)	1.14 (88%)	1.64 (78%)	1.10 (89%)	0.99 (91%)	1.11 (89%)	n/a	0.95 (91%)	1.06 (89%)	2.01 (71%)	1.95 (72%)
13	2.61 (59%)	2.84 (54%)	2.44 (62%)	2.49 (61%)	2.51 (61%)	2.84 (55%)	1.29 (85%)	1.66 (78%)	1.33 (85%)	1.37 (84%)	0.98 (91%)	0.95 (91%)	n/a	0.97 (91%)	2.33 (64%)	2.10 (69%)
14	2.59 (59%)	2.75 (56%)	2.53 (60%)	2.49 (61%)	2.59 (59%)	2.78 (56%)	1.31 (85%)	1.85 (74%)	1.43 (83%)	1.18 (87%)	0.91 (92%)	1.06 (89%)	0.97 (91%)	n/a	2.33 (64%)	2.17 (67%)
15	2.54 (60%)	2.55 (60%)	2.21 (67%)	2.19 (67%)	2.25 (66%)	2.21 (67%)	2.19 (67%)	1.77 (76%)	2.13 (68%)	1.82 (75%)	2.38 (63%)	2.01 (71%)	2.33 (64%)	2.33 (64%)	n/a	1.74 (76%)
16	2.29 (65%)	2.67 (58%)	2.21 (67%)	2.15 (68%)	2.25 (66%)	2.17 (67%)	2.21 (67%)	1.94 (72%)	2.36 (64%)	1.80 (75%)	2.00 (71%)	1.95 (72%)	2.10 (69%)	2.17 (67%)	1.74 (76%)	n/a

Table 2. *Nganasan-Nenets-Enets parallels in the Swadesh list with no corresponding lexemes in other Samoyed languages.*

Nganasan	Tundra Nenets	Nenets 1910	Forest Nenets	Tundra Enets	Forest Enets	Proto Samoyed
1. Head нгойбү	Head нэва	Head айво	Head нэ́йва	Head abur'e, abur'i	Head эба	*ǵjwǵ Head < Proto Uralic*ojwa
2. To burn лонгүди				Burn leided'e		
3. Yellow тодякү	Yellow тасехэй	Yellow тасей				
4. Stomach мина́да			Stomach mintéi	Stomach munzi, munðe, munði	Stomach Munoðe	
5. Star фотади́е				Star poðeseo	Star poðesei	
6. Dry ко́шúa	Dry хасуй	Dry хазуй	Dry kisúj	Dry kasuo	Dry Kasuj	*kās(ǵ)- to become dry < Proto Uralic *koški-
7. Human нганáса	Human ненэ́ць'(н)	Human ненáчъ	Human лэ́саң		Human entʃiʔ	*enǵ real
8. What ма'				What mǵi, mǵiʔ, mǵiiʔ		*me what < Proto Uralic *mǵ
9. Lay down бад-	Lay down ва́"на́(сь)			Lay down baðoti	Lay down bal'kij	

So, out of 9 lexical isoglosses, 5 words (to burn, yellow, stomach, star and lay down) do not have an etymology beyond the Northern Samoyedic languages. However, the remaining words that account for 72-75% similarity between Enets and Nganasan either have Proto-Samoyedic etymology or parallels in other Southern Samoyedic languages. Therefore, we assume that the Northern Samoyedic group cannot be entirely the result of later contacts. It is evident that this group existed briefly after the divergence of the Proto-Samoyedic language around the mid-1st millennium BCE. There were likely separate contacts between Enets and Nganasan, possibly involving a substrate language.

2) **Did a Southern Samoyedic commonality exist, uniting Mator, Kamasin, and Selkup languages?** Or is the Mator-Kamasin group closer to Northern Samoyedic than it is to Selkup, as proposed by S.A. Starostin?

The percentage of word matches with modern Selkup dialects for Kamasin language is 63-68%, and for Mator language, it's 64-71%. However, when considering the basic vocabulary from the 18th-19th centuries, the percentage of their overlap with Selkup dialects is much higher: Kamasin (64-76%) and Mator (69-75%).

The overlap between Kamasin and Northern Samoyedic languages is comparable to modern Selkup dialects: with Nenets 60-67%, with Enets and Nganasan 67%. Mator also exhibits a similar percentage of overlap with Northern Samoyedic languages: with Nenets 65-67%, with Enets and Nganasan 66-68%.

Let's consider specific words that constitute Kamasin-Mator-Selkup isoglosses. These are the reflexes of Proto Samoyed *olǵ head, *perkǵ belly, *kuńǵ star, *tipǵ man, *mǵǵǵ earth, *siǵnsǵkkǵ bird [11]. Just one word has just Selkup-Kamass parallel: Taz Selkup *ippiqo* lie, Kamass *i'búm* lie. So, it is evident that 6 out of 7 words are reflexes of Proto-Samoyedic lexemes. Thus, the hypothesis of a later contact influence can be ruled out, and the existence of a Southern Samoyedic group can be confidently acknowledged, supported by materials from the early Selkup dictionaries and books.

3) Between which Samoyedic languages did long-lasting commonalities exist? Our analysis allowed us to identify only those commonalities that most researchers have recognised. A clear increase in the percentage of shared words is observed between:

- a) Nenets and Enets languages (69-79% similarity, dating their divergence to the 1st-5th centuries AD).
- b) Mator and Kamasin (76% shared words), with the divergence dating back to the 3rd century AD.
- c) Southern and Northern Selkup languages (74-92%); due to the wide range of the similarity percentage, it is challenging to reliably date the time of divergence.

The close relationship between Tundra and Forest Nenets (85% shared words) is indisputable, with their divergence dating to the 5th century AC. These languages were once considered a single language by some authors. Later differences, such as those between Tundra and Forest Enets, have a dialectal character, as these idioms share 92% of their words and, according to S.A. Starostin, diverged in the 2nd millennium AC.

Let's examine which specific lexemes highlight the commonalities mentioned above: Nenets-Enets (see Table 3), Mator-Kamasin (see Table 4), and Selkup (see Table 5).

Table 3. Nenets-Enets parallels in the Swadesh list with no corresponding lexemes in the basic vocabulary of other languages.

Tundra Nenets	Nenets 1910	Forest Nenets	Tundra Enets	Forest Enets	Proto Samoyed
1. Road, path сехэры	Road, path сэры		Road, path sexa	Road, path sexeri	Proto Uralic *čäke(-r3) Hard snow
2. Smoke якэ	Smoke якэ	Smoke d'akí		Smoke d'ake d'ak'i	*jäckä - Smoke
3. Green пăдярăха			Green pođeraxa	Green pozeraxa	*pôtā bile < ИУ *piša bile
4. Sand тăб	Sand табь		Sand tobo	Sand tōb(u)	*t1āpā Dirt, soil
5. Black пăридєня	Black пардєна	Black plītečēna		Black pōlzer-	
6. This тюку		This teuk'i	This t'jik'i, t'jike, t'jike	That t'jik'i, t'jike, t'jik	*tū- That
7. Tongue нямю	Tongue нямю		Tongue n'am'i	Tongue n'am'i	*āŋ Mouth + *mū inside
8.	Stomach сѣнзе			Stomach sōse	*sūnsā chest < Proto Uralic *šínčā
9.	Road, path сарпé			Road, path sob'i	*sōrpā path
10.	Small о́лику		Small ul'ajku, ul'ajguu, ul'ajgu	Smallible:juk, ible:jku, ible:jku	
11.		Round pæeku	Round pos'uteo	Round paftej	

Table 4. Mator-Kamasin parallels in the Swadesh list with no corresponding lexemes in the basic vocabulary of other languages.

Mator	Kamasin	Proto Samoyed
1. Road, path ād'e	Road, path иде	*etā path, footsteps
2. Sand phūra	Sand хьпа	

Table 5. Selkup words in the Swadesh list with no corresponding lexemes in the basic vocabulary of other languages.

Taz Selkup.	Taz Selkup XVIIIв	Upper Tolka Selkup	Narym district (Ket)XVIIIв	Ket Selkup.	Lower Chaya Selkup. 1879	Laskino Selkup 1967 and 1970 гг.	Narym selkup.	Proto Samoyed
1. To burn čəpɪqo		To burn ʃɒpa-						
2. Road, path wətti		Road, path vətti		Road, path vətti	Road, ватонь	Road, wát	Road, path vat	*uât footstep, path < Proto Uralic *utka
3. Small kɪpa	Small кыбыля	Small kɪpa	Small кыба		Small кыба́	Small qiba	Small kibá	
4.Lot koččj		Lot koʃʃʃ		Lot kwəʃʃj	Lot ко́ чи	Lot qoč'ə, kóč'ə, koč'	Lot kótə(ik)	
5. Man ira		Man ira						
6. New šentj		New ʃenti		New siendi	New сэнды	New šénd, šánd	New ʃand	
7. Leg (foot) topj	Leg (foot) то- пы	Leg (foot) topi	Leg (foot) то- ба	Leg (foot) topi	Leg топъ (foot)	Leg (foot) tob	Leg (foot) təb	*topá claw, hoof
8. nose j ntāl	Nose унжель	Nose inʃʃæj						
9.Sand kōra		Sand qora		Sand kora	Sand kopá	Sand qó:ra	Sand kóra	
10. Dry təkɪpj			Dry	təkimpan	Dry ʃəkibɪ		Dry təágəbəl	
11. Ear ũŋkj lsa	Ear юнголсотъ							*jüntā - hear
12.human qum	Human кубъ	Human qum	Human куммъ	Human qum	Human гомъ, комъ	Human qop, qup	Human kup	Proto Uralic *koj(e)m3
13. what qaj	What кае		What кай	What qaje	What ká ʃ	What qái-l, qáj-t		Tungus *xai
14. Chest kɪ lj		Chest kilɪ		Chest kili		Chest qıl	Chest kil	
15. Full tirj k				Full trimbɪtij			Full tir	*tirā ~ *terā contents
16. To say kə tɪ qo		To say kət-		To say qət-	To say ká c-, ká t-	To say qadə-gú	To say kadəgú	*ket(ā)- To say
17.	Egg ɪra	Egg ɛɲj						Middle Ket ɛʔj egg, ēɲ eggs
18.				Egg nɪpɪ		Egg n'áb'i	Egg n'ab'	
19.	To eat егань		To eat егань					
20.			White чаягасомба	White tɛki	White tɛɣɪ	White č'ág čágə, č'a:R,		White tɛ'ag
21.				Woman pəjja			Woman pajá	
22.					This táб, táy	This táu, táw	This tav	*tā- This

We have identified only 2 isoglosses highlighting the Kamasin-Mator community. Moreover, the high percentage of similar lexemes (76%) has parallels in various languages, indicating that these isoglosses are Proto-Samoyedic and cannot be explained by later contacts.

Furthermore, in each Selkup dialect, there are 3 to 18 words with unknown etymology. It is evident that a significant number of Selkup words that lack parallels in other Samoyedic languages serve as evidence for the close relationship of this group and a lengthy period of its separate development. The abundance of words without reliable etymology suggests the possibility of substrate influence.

4. Conclusion

The conducted analysis has allowed us to identify four types of increases in the percentage of similarity when analysing the basic vocabulary among Samoyedic languages.

Type 1 is observed between Mator and Kamasin languages and between Selkup, Mator, and Kamasin languages. The number of unique isoglosses between these languages, absent in other Samoyedic languages and dialects, is small: 2-7 lexemes, most of which belong to the Proto-Samoyedic basic vocabulary. The increased percentage primarily results from the preservation of words from the Proto-Samoyedic basic lexicon, which had been lost in some other Samoyedic languages.

Type 2 of lexical isoglosses is observed among North Samoyedic languages according to traditional classification. There are 9 words of basic vocabulary shared between Nganasan, Enets, and Nenets, of which 5 have no etymology. This correlates with the observation that the Nganasan basic vocabulary contains a considerable number of words with no etymology, as noted by all researchers see more [20]. According to our calculations, there are 13 of them. Meanwhile, in Nenets, Enets, Kamasin, and Mator languages, the divergence of which dates back to our era, there are no more than 5-6 lexemes.

Type 3 of lexical isoglosses is observed between Nenets and Enets languages. They share 11 lexical isoglosses in their basic vocabulary, with 3 words having no etymology. Most of the words in this category have Proto-Samoyedic or even Proto-Uralic etymology but reconstructed meanings that do not belong to the basic vocabulary. Thus, in 8 cases, shared semantic innovations occurred in Nenets and Enets languages, confirming the existence of a Nenets-Enets group that separated in the early 1st millennium AD.

Type 4 of lexical isoglosses is observed among Selkup languages. Out of 22 words, only 2 ('this,' 'to say') belong to the Proto-Samoyedic basic vocabulary, 5 words have changed meanings, and 15 words are either presumed ancient borrowings or have unknown etymology. Considering that Narym Selkup has 13 words with unknown etymology, with no parallels even in Selkup dialects, it is possible to assume both prolonged separate development and significant external influence on Selkup.

In conclusion, the analysis supports the traditional classification of Samoyedic languages. The integration of material from the first Selkup books into the LingvoDoc platform allowed for a more reliable postulation of the South Samoyedic group.

The integration of S.A. Starostin's formula on to the LingvoDoc platform has allowed us to approximate the times of divergence for various Samoyedic language groups:

- 1) Proto-Samoyedic language: Around the beginning of the 1st millennium BCE.
- 2) North Samoyedic languages: Around the middle of the 1st millennium BCE.
- 3) South Samoyedic language: Around the end of the 1st millennium BCE.
- 4) Nenets-Enets group: Around the beginning of the 1st millennium AD.
- 5) Kamasin-Mator group: Around the middle of the 1st millennium AD.
- 6) Selkup group: Around the end of the 1st millennium AD.
- 7) Nenets language: Around the end of the 1st millennium AD.
- 8) Enets language: Around the beginning of the 2nd millennium AD.

The analysis of the material indicates that before the beginning of AD, the Samoyedic language groups, both northern and southern, existed, but for a very short period. Nevertheless, the high percentage of similarities between languages within these groups cannot be solely explained by contact influences. Lexical analysis certainly cannot consider a conclusive result in such a sophisticated case, and further confirmation is required through a detailed analysis of morphology and graphic-phonetics.

References

- [1]. <https://lingvodoc.ispras.ru>, дата обращения 10.03.2025.
- [2]. Starostin S.A. Comparative-historical linguistics and lexicostatistics // *Linguistic reconstruction and ancient history of the East*. Ch. I. M., 1989: 3-39.
- [3]. Collinder B. Comparative grammar of the Uralic languages. Stockholm, 1960.
- [4]. The Oxford Guide to the Uralic Languages, ed. by Bakró-Nagy M., Laakso J., Skribnik E. Oxford, New York 2022.
- [5]. Helmski E. The oldest Hungarian-Samoyed language parallels. M., 1982.
- [6]. Starostin S. Preliminary results of application of 'recalibrated' glottochronology to classification of Eurasian language families. Paper presented at Workshop on Prehistoric Chronology: Language, Genes and Migrations being held at the Santa Fe Institute, March 2004.
- [7]. Katz H. Zur Phonologie des Motorisch-Karagassisch-Taigischen // K. Redei (ed.). Studien zur Phonologie und Morphonologie der uralischen Sprachen. Akten der dritten Tagung für uralische Phonologie, Eisenstadt, 28. Juni — 1. Juli 1984 (*Studia uralica* 4). Wien: Verband der wissenschaftlichen Gesellschaften Österreichs, 1987. S. 336-348.
- [8]. Janhunen J. Samoyedic / The Uralic Languages, ed. by Abondolo, Daniel Routledge. London, 1998, p. 457–479.
- [9]. Helmski E. Die matorische Sprache. Szeged: *Studia Uralo-Altaica* 41, 1997.
- [10]. Helmski E. Mator and Northern Samoyed. Materials to a paper read on May 16, 1997 for the Finno-Ugric Society, Helsinki. = Eugen Helmski. Zur Stellung des Matorischen innerhalb der samojedischen Sprachen // V. Gusev, A. Urmanchieva, A. Anikin (eds.). *Siberica et Uralica: In memoriam Eugen Helmski (Studia uralo-altaica. Vol. 56)*. Szeged: University of Szeged, 2022. C. 479–495.
- [11]. Janhunen J. Samojedischer Wortschatz. *Castrenianum* 17. Helsinki, 1977.
- [12]. Helmski E. Nganasan dictionary, manuscript. Hamburg, 2007.
- [13]. Helmski E. Enets dictionary, manuscript. Hamburg, 2007.
- [14]. Helmski E. Selkup dictionary, manuscript. Hamburg, 2007.
- [15]. Helmski E., Kahrs U. Nordselkupisches Wörterbuch von F. G. Mal'cev (1903). Hamburg, 2001.
- [16]. Selkup-Russian dialect dictionary. Edited by V. V. Bykonya. Tomsk, 2005.
- [17]. Alatalo J. Sölkupisches Wörterbuch. (*Lexica Societatis Fenno-Ugricae, XXX.*) Helsinki: Société Finno-Ougrienne, 2004.
- [18]. <http://lingvodoc.ru/dictionary/7386/1/perspective/7386/2/view>, дата обращения 10.03.2025.
- [19]. Blažek V. On the classification of the Samoyedic languages. Dedicated to the memory of Eugene Helmski (1950–2007) // *Finnisch-Ugrische Forschungen* 63, 2016, s. 79 – 125.
- [20]. Urmanchieva A.Y. Reconstruction of the linguistic landscape of Western Siberia (based on the material of Samoyed languages). Dissertation for the degree of Doctor of Philology. M., 2023.
- [21]. Tadmor U. Loanwords in the world's languages. Findings and results. In: Haspelmath M., Tadmor U. (eds.) *Loanwords in the World's Languages: A Comparative Handbook*. Berlin: Mouton de Gruyter, 2009. pp. 55–75.
- [22]. Koryakov Yu.B. The problem of "language or dialect" or Samoyedic languages // *Ural-Altaic studies* 4 (31), 2018, pp. 156 – 217.
- [23]. Starostin S. A. Works on linguistics. Moscow, 2007.
- [24]. Starostin G. Preliminary lexicostatistics as a basis for language classification: A new approach. *Journal of Language Relationship* 3, 2010, p. 79-116.
- [25]. Kassian A., Starostin G., Dybo A., Chernov V. The Swadesh wordlist. An attempt at semantic specification // *Journal of Language Relationship*, No. 4 (2010), p. 46-89.

- [26]. Starostin G. Lexicostatistics as a basis for language classification: increasing the pros, reducing the cons. In H. Fangerau and H. Geisler and Th. Halling and W. Martin (eds.), *Classification and Evolution in Biology, Linguistics and the History of Science: Concepts - Methods - Visualization*, Stuttgart, 2013, p. 125- 146.
- [27]. Tereshchenko N. M. *Nenets-Russian dictionary*. M., 1965.
- [28]. Donner K. *Kamassisches Wörterbuch nebst Sprachproben und hauptzügen der Grammatik*, bearbeitet und herausgegeben von A.J. Joki. Helsinki, 1944.
- [29]. <https://lingvodoc.ispras.ru/dictionary/3596/37747/perspective/3596/37750/view>, дата обращения 10.03.2025.

Информация об авторах / Information about authors

Юлия Викторовна НОРМАНСКАЯ – доктор филологических наук, заведующий лабораторией Лингвистические платформы Института системного программирования РАН с 2019 года, ведущий научный сотрудник Институт языкознания РАН с 2003 года. Сфера научных интересов: компьютерные методы в лингвистике, сравнительно-историческое языкознание, уральские языки.

Julia Viktorovna NORMANSKAJA – Dr. Sci. (Philology), Head of the Laboratory Linguistic Platform of the Institute for System Programming RAS since 2019, leading scientist of Institute of Linguistic RAS since 2003. Research interests: computer methods in linguistic, comparative-historical linguistic, Uralic language.

DOI: 10.15514/ISPRAS-2025-37(3)-15



Computerised Experimental Methods of Studying Australian Aboriginal English

E.O. Timofeeva, ORCID: 0009-0002-5005-5862 <et@ovel.ru>

*The Patrice Lumumba Peoples' Friendship University of Russia, Faculty of Philology,
6, Miklukho-Maklaya st., Moscow, 117198, Russia.*

Abstract. This paper aims to describe certain phonetical, morphological and lexical features of Australian Aboriginal English that have been detected throughout the analysis of Australian Aboriginal English texts in LingvoDoc and Praat. The study outlines the methods, goals, and benefits of using the linguistic platform LingvoDoc to identify and systematize the grammatical and lexical features of Australian Aboriginal English. Numerous researchers note that Australian Aboriginal English is a distinct ethnolect, differing from the English spoken by Australians of British descent. By using LingvoDoc to create a collection of Australian Aboriginal English dictionaries that describe features specific to particular localities in Australia, it is likely we can draw conclusions about correlations between the lexical and grammatical features of this ethnolect and various extralinguistic factors. The texts under scrutiny include transcripts of interviews with Aboriginal Elders, musicians, teachers and artists, song lyrics, and personal stories. Informants originate in various places across Australia and belong to various age cohorts from adolescence to late adulthood. Texts were grouped based on informants' places of origin, and a separate dictionary for each of those places was created in Lingvodoc. Each dictionary was attached to a human settlement on the world map, which helped us track the correlation between the speakers' origin and the grammatical and lexical characteristics of their speech. This method reveals which linguistic patterns may be characteristic of speakers from certain geographical areas, thus unveiling potential correlations. The phonetical part of our study aims to discover differences between vowel formants in Standard Australian and Australian Aboriginal English.

Keywords: Lingvodoc; Praat; vowel formants; ethnolect; English; Australian Aboriginal English.

For citation: Timofeeva E.O. Computerised Experimental Methods of Studying Australian Aboriginal English. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 3, 2025, pp. 211-224. DOI: 10.15514/ISPRAS-2025-37(3)-15.

Acknowledgements. The author would like to thank Y. Normanskaya for consulting us on the proper use of LingvoDoc and Praat, and for the tremendous support with analysing and interpreting the results of this study; O. Artemova and V. Klyaus for providing the author with the recording of speech in Australian Aboriginal English; and all our Indigenous informants whose contribution has made it possible for us to conduct this research.

Компьютерные экспериментальные методы изучения английского языка австралийских аборигенов

Э.О. Тимофеева, ORCID: 0009-0002-5005-5862 <et@ovel.ru>

*Российский университет дружбы народов им. Патриса Лумумбы,
филологический факультет,
117198, Россия, г. Москва, ул. Миклухо-Маклая, 6.*

Аннотация. Целью данной работы является описание некоторых фонетических, морфологических и лексических особенностей английского языка аборигенов Австралии, обнаруженных в ходе анализа текстов на этом варианте английского языка с помощью платформы LingvoDoc и программы Praat. В исследовании изложены методы, цели и преимущества использования лингвистической платформы LingvoDoc для выявления и систематизации грамматических и лексических особенностей английского языка аборигенов Австралии. Многие исследователи отмечают, что английский язык австралийских аборигенов является отдельным этнолектом, отличающимся от английского, на котором говорят австралийцы британского происхождения. Используя LingvoDoc для создания ряда словарей, описывающих особенности, характерные для английского языка аборигенов в конкретных регионах Австралии, мы можем делать выводы о корреляциях между лексическими и грамматическими особенностями этого этнолекта и различными экстралингвистическими факторами. Исследуемые тексты включают записи интервью со старейшинами, музыкантами, учителями и художниками коренного австралийского происхождения, а также тексты песен и личные истории. Информанты происходят из разных мест по всей Австралии и принадлежат к различным возрастным группам от подросткового до пожилого возраста. Тексты были сгруппированы по местам происхождения информантов, и для каждого из этих мест был создан отдельный словарь в Lingvodoc. Каждый словарь был прикреплен к населенному пункту на карте мира, что помогло нам отследить корреляцию между происхождением информантов и грамматическими (лексическими) характеристиками их речи. Этот метод показывает, какие лингвистические паттерны могут быть характерны для говорящих из определенных географических областей, тем самым раскрывая возможные корреляции. Фонетическая часть нашего исследования направлена на выявление различий между формантами гласных в стандартном австралийском английском и в английском языке аборигенов Австралии.

Ключевые слова: платформа Lingvodoc; программа Praat; форманты гласных; этнолект; английский язык; английский язык австралийских аборигенов.

Для цитирования: Тимофеева Э.О. Компьютерные экспериментальные методы изучения английского языка австралийских аборигенов. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 211–224 (на английском языке). DOI: 10.15514/ISPRAS–2025–37(3)–15.

Благодарности. Автор благодарит Юлию Норманскую за консультации по правильному использованию инструментария LingvoDoc и Praat, а также за огромную поддержку в анализе и интерпретации результатов этого исследования, О. Артемову и В. Кляуса за предоставление автору записей речи на английском языке австралийских аборигенов, а также всех информантов коренного австралийского происхождения, чей вклад позволил провести это исследование.

1. Introduction

At the time of British settlement (1788), Aboriginal Australians spoke at least 200 languages. However, more than half of those languages are now considered extinct. Only 12% of Aboriginal Australians speak an Australian language. For the majority of Aboriginal people in Australia, English is their L1 [1]. However, numerous researchers [1-7] believe that Australian Aboriginal English is different from other varieties of English, including Standard Australian. Moreover, Australian Aboriginal English is not a monolithic ethnolect but a continuum of local dialects that vary from region to region, differing to a greater or lesser extent from the standard. Basilectal forms of Australian Aboriginal English have not been recognised as independent language variants that can be efficient forms of communication in many contexts until the 1960s [1]. The study of Australian Aboriginal English could justify the notion that it is a valid and effective language variant, while

212

paying attention to its local varieties could provide insight into the impact of extralinguistic factors (such as literacy levels and dominant cultures) and local Indigenous languages on the particular kinds of English that are spoken in certain regions of Australia. Besides, awareness of the phenomena that distinguish Australian Aboriginal English from other variants could help to avoid misunderstanding and misinterpretation in contexts where Aboriginal Australians interact with non-Aboriginal English speakers [4].

2. Literature review

It is worth noting that the term “Australian Aboriginal English” is interpreted diversely by different researchers. A detailed commentary on this is given in [5]. According to this paper, some linguists have used the term to exclusively describe varieties of English that are spoken by Aboriginal people as a second language (that is, acquired after the native language) and deviate from the norms of Standard Australian English under the influence of the speakers' native language. Other scholars have described Aboriginal English as a variant that differs little from standard English in formal structure but is used in a distinctive way that reflects Aboriginal ways of interacting (that is, the differences are primarily pragmatic). In some studies, Aboriginal English has been understood as a set of features of spoken Australian English that are characteristic of Aboriginal Australian speech.

Many studies have examined the phonetics, grammar, semantics, and pragmatics of Australian Aboriginal English, usually drawing on short everyday dialogues. Recently, attention has often been paid to sociolinguistic aspects [4].

Most studies focus on analysing recorded dialogues. For example, [5] uses audio recordings where Aboriginal informants discuss the features of their native language Kaytetye and recount legends in English.

An interesting method is described in [6]. Instead of extracting information by conducting a typical sociolinguistic interview, the authors collected data through a special way of storytelling, which in Australian Aboriginal English is called yarning. Yarning is a conversation that involves sharing stories and personal experiences. This method of communication is considered the most culturally acceptable one among Aboriginal people. According to [6], direct questions are viewed as disrespectful in most Australian Aboriginal cultures, which is why yarning (long informal conversations about life experiences) turned out to be a more natural way for Nyungar people to interact with linguists.

However, hardly any studies focus on written texts in English produced by Aboriginal Australians (such as song lyrics). Therefore, the similarities and differences between spoken and written forms of Australian Aboriginal English virtually remain out of the spotlight.

Not only does our study trace patterns of similarity between written and spoken texts, but it also includes a broad range of genres and focuses on territorial variation. This kind of diversity may make it possible to identify correlations between linguistic phenomena and the extralinguistic context in which these texts were created, and to track the distribution areas of certain lexical and grammatical features.

3. Data and methodology

This study can be split into two parts that have different aims and draw on different data. However, if we combine those aims, the key goal is to pinpoint some phonetical, morphological and lexical features of Australian Aboriginal English while tracing general patterns.

The first section focuses on morphology and lexicology. The data includes transcripts of interviews with Aboriginal Elders, artists, teachers, and musicians, personal stories, and song lyrics. While most of the songs have been written by young people (adolescents and young adults), the interviewees are mostly older adults (over 50 years old). The geographical scope is also fairly diverse, covering central, northern, and southern New South Wales, southern and northern regions across the Queensland coast, as well as central and northern areas of North Territory. This kind of diversity could provide us with

information on features that can be found in Australian Aboriginal English in various contexts. The following steps were undertaken once the data had been gathered:

- The data were grouped based on the localities which the informants originate from;
- Tables listing and describing the features of Australian Aboriginal English that were found in the data and were not characteristic of other English varieties were created;
- The tables were uploaded to LingvoDoc, a platform for compiling, analysing and storing dictionaries, corpora and concordances of various languages and dialects.
- The tables were utilised to create dictionaries, with each dictionary being assigned a specific human settlement on the world map in the Properties section. In the dictionaries, each entry described a single occurrence of a phenomenon that is not typical for Standard Australian English. The descriptions included the phenomenon itself (e.g. *songline*, a word that is not found in other variants of English), the sentence where it occurred, and the type the phenomenon belonged to (e.g. all words that were unique to Australian Aboriginal English were marked as “special lexemes”, which made it possible to search for groups of similar phenomena by typing the name of their type in the search bar). Some entries were accompanied by definitions of lexemes that are not typical for English and Standard Australian English equivalents of the sentences where the phenomena could be found;
- Using the Search and Mapping tool, specific features of Australian Aboriginal English could be searched for, and places where they could be found were displayed on the map. Distribution areas could be easily outlined.

The second part of our study is only a pilot project, since it is based on a small selection of data, and it is hardly possible for us to obtain more data at the moment. However, it has proven to be successful, since the results seem to demonstrate a deviation from the Standard Australian English norm that is quite similar to the one described in more large-scale works on the topic [2]. This part of our study focuses on phonetics of Australian Aboriginal English, namely, on vowel formants in the speech of an Aboriginal English native speaker.

The data is derived from a 40-minute recording of the speech of a 50-year-old Ayapathu man from Coen. Although English is his first language, he can speak Ayapathu at a basic level and calls it “my language”. Numerous Cape York Peninsula languages are spoken in his family (Ayapathu, Thaayore, Wik, Kaantju, etc.). Besides, the informant helps organize linguistic fieldwork with native speakers of Australian languages. All these factors allow us to assume he often hears those languages spoken and has probably been hearing them since childhood. This might indicate a possibility of interference in his speech.

Throughout the course of this part of our study, we aimed to:

- Segment the recorded speech of an Australian Aboriginal English speaker into vowels and analyse their formants using Praat;
- Compare vowel formants in the recording to mean vowel formant values in Standard Australian English (standard values taken from [3]);
- Identify recurring deviations from the norm and trace patterns.

4. Results and discussion

4.1 Morphology

Most of the morphological features were associated with verb morphology. The map below illustrates the distribution of all discovered phenomena that are related to verb morphology.

Fig.1 indicates that the greatest number of verb morphology anomalies could be found on Groote Eylandt, which sets it apart from other regions, even ones where Australian languages are also actively spoken and interference is thus also expected to occur. One of the reasons behind this situation on Groote Eylandt could be the fact that it is predominantly populated by Warnindilyakwa

people of whom the vast majority speak the Anindilyakwa language. English is not the native language of most Groote Eylandt inhabitants, and only 3.7% of the local population exclusively uses English at home [8]. This possibly explains why Australian Aboriginal English speakers in this area struggle with English verb morphology.



Fig. 1. Map of Verb Morphology Anomalies.

However, the sociolinguistic situation on Elcho Island is similar to that on Groote Eylandt, yet hardly any verb morphology anomalies were found there. It is challenging to find a definitive explanation for this. The discrepancy might be explained by the fact that Anindilyakwa, the substrate Aboriginal language on Groote Eylandt, belongs to the non-Pama-Nyungan Gunwinyguan family [9], while Yolŋu, the language of Elcho Island, along with Aboriginal languages in other localities that were taken into account in this study, are Pama-Nyungan [10]. It is notable that, apart from Groote Eylandt and Elcho Island, no other two geographically close points on our map show such a striking contrast in the number of registered English verb morphology anomalies, which may argue in favour of the hypothesis that suggests the family to which the substrate language belongs may play a certain role here.

Below examples of phenomena that are featured on the map are provided:

Atypical verb government models (e.g. verbs form infrequent collocations with prepositions)

Elcho Island:

- (1) *Maybe there is not enough teaching happening about land.*
- (2) *My body is owned by this land.*

Maclean:

- (3) *Sit around by the campfire.*

Groote Eylandt:

- (4) *Our language was maintained, when we were sort of hung around amongst one another.*

Absence of verb endings (e.g. -ing or -s are absent in wordforms where they are required to be present by the grammatical norm)

Walgett:

(5) *My songline criss-cross from coast to coast.*

Wilcannia:

(6) *We belong where the Baaka flow even when the Baaka low*

(7) *Welcome to Wilcannia where the Darling flow*

Santa-Teresa:

(8) *We've played small festivals and bush bashes here with Indigenous bands from all over come and play*

Absence of the auxiliary verb 'to be'

Wilcannia (note that all examples from Wilcannia and Walgett occurred in written texts):

(9) *This my mob*

(10) *This my future*

(11) *My people sick*

(12) *My heartbeat never far from home*

Walgett:

(13) *This the black soil plains where the songline start*

(14) *This a song for my people*

(15) *This a message and it's special*

(16) *This how we do it*

(17) *This a call out from the north to west*

(18) *Ya bored*

Bowen:

(19) *We dancing on the beaches through the changing of the seasons*

Groote Eylandt:

(20) *That's where the church, we're gonna build church.*

Muli Muli:

(21) *Welcome to the place where the bellbird song means the sun settin' over*

(22) *We gonna sort it, bin it*

Coomealla:

(23) *This my land*

(24) *We still true*

(25) *We black, strong and proud*

Maclean:

(26) *We singin' the same tune*

Absence of actant (i.e. a transitive verb lacks an object)

Bowen:

(27) *We gonna set and deploy play with the system like toys*

Sequence of tenses absence (the forms of subordinate verbs do not depend on those of verbs in the main clause)

Santa-Teresa:

(28) *We set up an old lounge, had a bonfire, and were under the stars. Even the cops come out and sit down and have a listen.*

Groote Eylandt:

(29) *I expect your people to try and understand our language and try and speak it too so you can understand it ... Because early settlers **come** to our country, our land and they expected us to learn theirs and speak theirs.*

Kintore:

(30) *I had to think my dialysis before I think other things*

Modal verb used as an auxiliary

(31) *That's what the nurses told me... 'If you miss the dialysis you **gotta** be sick.'*

Atypical use of -ing forms

Walgett:

(32) *Stories of my culture that we holding close (-ing form used instead of Present Simple)*

Coomealla:

(33) *Get it into your head and stopping hating (-ing form used instead of an imperative)*

Present Perfect used instead of Past Simple

Groote Eylandt:

(34) *And that stone, it's landed there.*

Verb form contaminaton

Groote Eylandt:

(35) *And the old fella, he said, 'well I've got a stone here. If I throw that stone, if we see that stone where it's **gonna landed**, that's where the church, we're gonna build church.' (contamination of "has landed" and "gonna land")*

Passive voice used instead of active voice

Groote Eylandt:

(36) *Our language was maintained, when we **were** sort of hung around amongst one another.*

Absence of number agreement

Walgett:

(37) *This the black soil plains where the songline **start** ('start' used instead of 'starts')*

(38) *Copy what ya mum and dad does*

Muli Muli:

(39) *The system don't work*

A notable distribution-related tendency is associated with the absence of the verb *to be*. As Fig. 2 indicates, this phenomenon is virtually ubiquitous. Almost every dictionary contains more than one sentence which lacks the verb *to be* where it is required to be used by English grammar norms. A clearly visible exception, however, is the central region, where the absence of *to be* has not been registered. This tendency is yet to be confirmed and potentially explained, for our corpus of data is too small for any conclusions to be drawn yet, however, the contrast between the central region and all the others seems to be quite visibly defined at this stage.

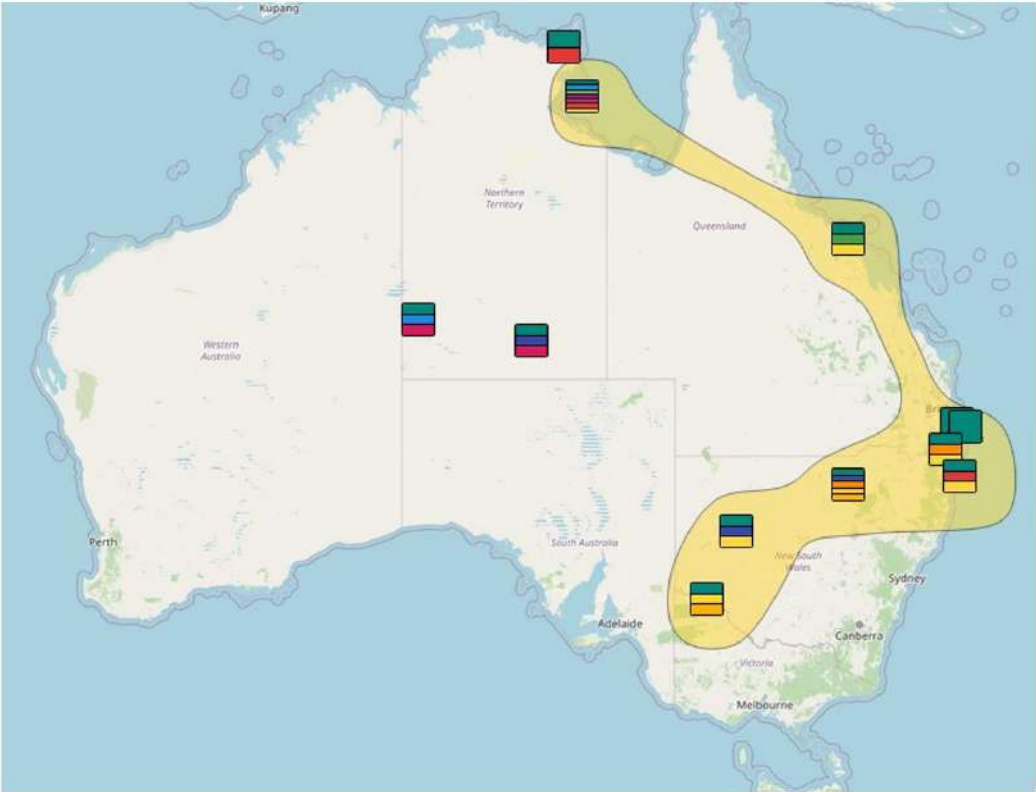


Fig. 2. Absence of the verb *to be*: areal distribution.

Yet another feature that appears to demonstrate areal distribution is the absence of verb endings (see Fig. 3). It is noteworthy that this phenomenon occurred in written texts (e.g. this line from song lyrics that have been published in a book: *We belong where the Baaka flow even when the Baaka low*), which means it cannot be simply written off as a consequence of reduced articulation at the end of a word. Verb endings are absent in texts from small, predominantly Aboriginal settlements in the southeastern part of central Australia. Some possible explanations include the fact that most of the texts that come from these localities have been written by school students who might not have gotten accustomed enough to English verb conjugation norms yet as well as the fact that speakers of Standard Australian English are a minority in these regions, which means that Australian Aboriginal English speakers are less likely to reproduce standard patterns when speaking English.

There were no anomalies found near large cities on the east coast (i.e. in Brisbane and on North Stradbroke Island). Possible reasons include the predominance of Standard Australian English speakers, the accessibility of high-quality education, and the fact that most Aboriginal languages (which could have otherwise caused interference) are extinct in this region.



Fig. 3. Absence of verb endings: areal distribution.

4.2 Lexicology

Some lexicological processes that have been identified in the texts under scrutiny include personification that manifests in formally diverse ways (examples are provided below).

Wilcannia:

(40) *The cotton farmers keep the Paaka [river] low <...> **River got sick** it's like it affected my mob it's like my people ditched fishing for smoking and sipping grog*

Elcho Island:

(41) *My body is owned by this land*

Maclean:

(42) *If I'm looking after my country then **she's** looking after me*

Another widespread phenomenon is semantic shift. For instance, *pass down* means *move past something* rather than *give something to someone who is younger or has a lower status than the giver* (which is the standard definition provided in the Cambridge Dictionary):

Groote Eylandt:

(43) *Run run down the way that we go,
Pulling up, picking up the sand in our toes,
Pass down the whistling trees,
Lady blue, she is calling me*

The following sentence is another example of semantic shift.

Muli Muli:

(44) *Recycling's a **mad** trick, re-use some plastic*

Here *mad* means 'excellent, wonderful, amazing', although this meaning does not appear in Cambridge Dictionary, Longman Dictionary of Contemporary English, or Macquarie Dictionary. All these dictionaries distinguish only three meanings of the word: 'insane', 'outraged', and 'excited'. None of these dictionaries note that the word can be used to convey a positive connotation. Even Macquarie Dictionary, which often comments on the differences between Australian Aboriginal English and Standard Australian English, does not mention this feature.

Yet another example of semantic shift is present in this sentence from a text created by an Indigenous resident of Wilcannia:

(45) *My people sick, so clean this tip check out and **cash** collect*

Based on the context, it can be assumed that *cash collect* here means 'to collect money (to put the land in order)'. However, this meaning of this phrase is not recorded in dictionaries (Cambridge, Longman, and Macquarie), which suggest this collocation describes a method of counting the money a company has received during a month.

Examples of such shifts in meaning are ubiquitous, occurring in 10 of the 12 dictionaries (excluding Brisbane and North Stradbroke Island, where no anomalies were found at all). When it comes to lexicology, no clear areal tendencies have been discovered, as most features were either unique to a particular text (such as new word meanings unrecorded in dictionaries) or occurred in regions that are far apart from each other. Lexemes that are unique to Australian Aboriginal English (e.g. *songline*, *dreaming*, or *bred* 'brother') have been found in localities that are far away from each other. For instance, the word *dreaming* (a mythological spirit realm or a time period when spirits created the world) is found in texts from Bowen (northern Queensland) and Walgett (north-central New South Wales).

Bowen:

(46) *We dancing on the beaches through the changing of the seasons got the sky sand and the Deep Sea Dreaming*

Walgett

(47) *If you know the emu dreaming yarn, y'all connected to the black dirt that's beneath ya feet*

The same applies to the word *songline* (a path across the land that is said to trace the routes of "creator-beings"), which has been found in texts from Walgett and Elcho Island (off the northern coast of North Territory).

Walgett:

(48) *My songline criss-cross from coast to coast*

(49) *This the black soil plains where the songline start*

(50) *My country where the songlines are*

Elcho Island:

(51) *He knows the history and all the story about that land from the songlines*

The word *bred* (which means 'brother') has been found in entries from Walgett and Coomealla.

Walgett:

(52) *Got young fellas lookin' up to you, bred*

(53) *No matter which opportunities, I'm takin', bred, where my relations from the start of creation come together ... we one nation ... yet?*

This tendency for lexemes that can only be found in Aboriginal English to occur in distant regions may be explained by certain cultural concepts (such as *songline* and *dreaming*) being near-universal or universal in Australian Aboriginal cultures, yet not present in other cultures.

4.3 Phonetics

The results are provided in the tables below. The first column lists word tokens in which the sounds occurred, the next two contain the formant values of vowels in these words, the fourth column includes the symbols for Standard Australian English sounds that are most similar to the ones that were found in the recording (obtained from [3]), while the fifth and sixth columns specify the mean formant values of these Standard Australian Aboriginal vowels (also acquired from [3]). Formant values in the recording were subtracted from standard formant values, and the differences were placed in the seventh and eighth columns (if the Australian Aboriginal English formant was greater than the standard one, the emerging difference was negative). Values that exceeded or fell short of the standard values by 200 Hz or more were considered abnormal, and those differences were marked as bold in the tables. Values that exceeded or fell short of the standard values by almost 200 Hz were considered near-abnormal, and those differences were marked as cursive in the tables.

As can be observed in Table 1, the second formant exceeds the standard value for back rounded vowels in most examples, which may indicate that these vowels are fronted. A similar pattern is described in [2]: "...of the second formant values, only those of the high front and central vowels (/i:, ɪ, e, ʌ, ε:/) **are lower; all others are somewhat higher**. This suggests that the AAE speakers are using a somewhat smaller overall vowel space than the SA group" (emphasis in bold added by us, AAE stands for "Australian Aboriginal English" and SA means "Standard Australian").

As Table 2 indicates, the second formant has a considerably lesser value than the standard for the front vowels [e] and [ɪ] in all examples, which may suggest their backing and again seems to follow the tendency mentioned in [2] (see quote above).

Table 1. Formant values for 'o' and 'u' sound types.

Word	F1	F2	SAA Sound	F1	F2	F1 Difference	F2 Difference
<i>so</i>	650	1197,72	o:	480	920	-170	-277,72
<i>old</i>	697	940,57		480	920	-217	-20,57
<i>record</i>	576,9	914,06	ɔ	750	1185	<i>173,1</i>	270,94
<i>recording</i>	672,7	1959,34		750	1185	<i>77,3</i>	-774,34
<i>on</i>	563,1	1712,69		750	1185	<i>186,9</i>	-527,69
<i>youse</i>	393,4	1907,25	u	450	1125	56,64	-782,25
<i>you</i>	383,3	1936,09		450	1125	66,73	-811,09

Table 2. Formant values for 'e' and 'i' sound types.

Word	F1	F2	SAA Sound	F1	F2	F1 Difference	F2 Difference
<i>they</i>	482,8	1484,8	e	685	2330	202,2	845,2
<i>there</i>	458,7	1707,12		685	2330	226,4	622,88
<i>there</i>	425,6	1709,99		685	2330	259,5	620,01
<i>skin</i>	429,8	1914,61	ɪ	430	2625	0,2	710,39
<i>skin</i>	459,1	1857,51		430	2625	-29,11	767,49
<i>skin</i>	447,2	1889,5		430	2625	-17,19	735,5
<i>be</i>	432	1783,38		430	2625	-1,96	841,62

It is also worth noting that the places where the informants come from in [2] and in our study are quite far apart. According to [2], their informants “live (most of the time) in Alice Springs”. Our informant, however, comes from Coen. Nevertheless, “it can be seen that the vowel space of a typical Australian language is quite small” [2]. This is true of both Ayapathu and the native languages of the informants in [2] (Eastern Arrernte, Warlpiri, and Western Desert language). The number of vowels in all these languages does not exceed 5. According to [2], the vowel spaces of Indigenous Australian languages are likely to influence those of Australian Aboriginal English, which might be the pattern observed in our case as well.

As to the a-like vowels, their second formants exceed the standard values in some cases, but this happens irregularly, which may suggest that the pronunciation of these vowels is closer to the Standard Australian English norm (see Table 3).

Table 3. Formant values for the ‘a’ sound type.

Word	F1	F2	SA Sound	F1	F2	F1 Difference	F2 Difference
<i>lump</i>	729,3	1385,69	ɐ	955	1550	225,7	164,31
<i>but</i>	610	1205,55		955	1550	345	344,45
<i>couple</i>	663,7	1656,31		955	1550	291,3	-106,31
<i>come</i>	825,9	1377,24		955	1550	129,2	172,76
<i>government</i> (1 st syllable)	531,2	1340,74		955	1550	423,8	209,26
<i>cut</i>	611,2	1399,29		955	1550	343,8	150,71
<i>language</i> (1 st syllable)	699	1514,67	æ	1020	1875	321	360,33
<i>land</i>	594,8	1911,87		1020	1875	425,2	-36,87
<i>land</i>	618,3	1786,45		1020	1875	401,7	88,55

5. Conclusion

Morphological, lexical, and phonetical features that distinguish Australian Aboriginal English from Standard Australian English are generally quite typical and are repeated across different, even distant regions.

The only locality that stands out significantly from the general background is Groote Eylandt, since the number of anomalies found there is greater than in the other localities, and many of those anomalies have not been found elsewhere. This is also the only region in our selection where the substrate language is non-Pama-Nyungan. Therefore, it may be worth looking into the features of Australian Aboriginal English in other regions where the substrate language does not belong to the Pama-Nyungan family.

References

[1]. A. Butcher, “Linguistic Aspects of Australian Aboriginal English,” in *Clinical Linguistics & Phonetics*, vol. 8, №22, 2008, pp. 625-642, DOI: 10.1080/02699200802223535.

[2]. A. Butcher and V. Anderson “The vowels of Australian Aboriginal English,” in *Interspeech*, 2008, pp. 347-350.

[3]. F. Cox and J. Fletcher, *Australian English pronunciation and transcription*. Cambridge: Cambridge University Press, 2017.

[4]. D. Eades, *Aboriginal Ways of using English*. Canberra: Aboriginal Studies Press, 2013.

[5]. H. Koch “Central Australian Aboriginal English: in comparison with the morphosyntactic categories of Kaytetye,” in *Asian Englishes*, vol. 2, № 3, 2000, pp. 32-58.

- [6]. C. R. Louro and G. Collard “The yarning corpus: Aboriginal English in southwest Western Australia” in *Australian Journal of Linguistics*, vol. 44, 2024, pp. 146-162. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/07268602.2024.2380661?af=R#d1e150>.
- [7]. I. G. Malcolm, “Aboriginal English: Some grammatical features and their implications,” in *Australian Review of Applied Linguistics*, vol. 36, №3, 2013, pp. 267–284, DOI: 10.1075/aral.36.3.03mal.
- [8]. “2021 Census Aboriginal and/or Torres Strait Islander people: Anindilyakwa (Groote),” 2021, Australian Bureau of Statistics. [Online]. Available: <https://www.abs.gov.au/census/find-census-data/quickstats/2021/IARE706001>.
- [9]. “Family Gunwinyguan,” *The World Atlas of Language Structures Online*. [Online]. Available: <https://wals.info/languoid/family/gunwinyguan#8/-13.625/135.640>.
- [10]. Alpher B. et al. “Pama-Nyungan: phonological reconstruction and status as a phylogenetic group” in *Amsterdam Studies in the Theory and History of Linguistic Science*, Series 4, 2004, pp. 93-126.

Информация об авторах / Information about authors

Эвелина Олеговна ТИМОФЕЕВА – студентка Российского университета дружбы народов им. Патриса Лумумбы. Сфера научных интересов: диалектология, социолингвистика, лингвистическая география, внутриязыковое варьирование, английский язык аборигенов Австралии.

Evelina Olegovna TIMOFEEVA – The Patrice Lumumba Peoples' Friendship University of Russia student. Research interests: dialectology, sociolinguistics, linguistic geography, interdialectal variation, Australian Aboriginal English.



Распределение показателей двойственного числа существительных в селькупском языке (корпусно-ориентированный подход)

С.В. Ковылин, ORCID: 0000-0003-0108-9214 <kovylin.ser@yandex.ru>

Институт системного программирования им. В.П. Иванникова РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

Аннотация. Статья посвящена выявлению распределения маркирования двойственного числа существительных аналитическим и синтетическим способом в южных, переходных, центральных и северных диалектах и говорах селькупского языка. В качестве материала послужили корпусные данные объемом более 85000 словоформ, расположенные на цифровой платформе Lingvodoc и в личных архивах (файлы программы Fieldworks Language Explorer), а также общие грамматические и лексические труды по языку. Было выявлено, что использование базового селькупского суффикса двойственного числа $-q(V)$ вместе с существительными в селькупских диалектах распределено неоднородно – показатель не обнаруживается в ряде материалов, когда присоединяется напрямую к основе (переходные говоры, южная часть нарымского диалекта), однако используется везде с одушевленными существительными, обозначающими совокупность двух однородных предметов при помощи суффиксов взаимной связи $-sa-$ и собирательного множества $-mi-$. В южной части нарымского диалекта и в переходных говорах фиксируются новообразования дуалиса в виде показателей $-štja$ или $-štjaq(V)$. В васюганских и северных материалах представлены случаи появления удвоенного дуалиса с суффиксами взаимной связи или собирательного множества: $-sa-$ / $-mi-$ + $-qV-q(V)$. Следует отделять более северную и южную части нарымского диалекта, где в первом случае появляется суффикс двойственного числа $-q(V)$, а во втором отмечаются инновации в виде $-štja$ и $-štjaq(V)$. Во всех селькупских материалах главной или одной из главных стратегий маркирования двойственности существительных является аналитическая стратегия, при которой используется числительное *sitte* ‘два’ и существительное в единственном (в отдельных случаях во множественном) числе.

Ключевые слова: селькупский язык, дуалис, корпусная лингвистика, существительное.

Для цитирования: Ковылин С.В. Распределение показателей двойственного числа существительных в селькупском языке (корпусно-ориентированный подход). Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 225–236. DOI: 10.15514/ISPRAS-2025-37(3)-16.

Благодарности: Исследование автора было поддержано грантом Российского научного фонда № 20-18-00403 “Цифровое описание диалектов уральских языков на основании анализа больших данных”.

Distribution of the Dual Number Markers of Nouns in Selkup (Corpus-Oriented Approach)

S.V. Kovylin, ORCID: 0000-0003-0108-9214 <kovylin.ser@yandex.ru>

*Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

Abstract. The paper is devoted to the identification of the distribution of the dual number markers of nouns in an analytical and synthetic way in the southern, transitional, central and northern dialects and subdialects of Selkup. The material of the study is corpus data of more than 85,000 wordforms located on the digital platform “Lingvodoc” and in personal archives (Fieldworks Language Explorer files), as well as general grammatical and lexical papers on the language. It was revealed that the use of the basic Selkup suffix of the dual number -*q(V)* with nouns in Selkup dialects is distributed heterogeneously – the marker is not found in a number of materials, when it is attached directly to the stem (transitional subdialects, the southern part of the Narym dialect), however, it is used everywhere with animate nouns, denoting the complex of two homogeneous objects via the suffixes of the mutual connection -*sa-* and the collective set *mi-*. In the southern part of the Narym dialect and in transitional subdialects new dualis formations are recorded in the form of the markers -*štja* or -*štjaq(V)*. In the Vasyugan and northern materials cases of the doubled dualis with the suffixes of the mutual connection or the collective set are presented: -*sa-* / -*mi-* + -*qV-q(V)*. It is necessary to separate the more northern and southern parts of the Narym dialect, where in the first case, the suffix of the dualis -*q(V)* appears, and in the second, innovations in the forms of the dualis like -*štja* and -*štjaq(V)* are noted. In all Selkup materials the main or one of the main strategies for marking the duality of nouns is an analytical strategy, which consist in the use of the numeral *sitte* ‘two’ and the noun in the singular (in some cases in the plural).

Keywords: Selkup language, dualis, corpus linguistics, noun.

For citation: Kovylin S.V. Distribution of the dual number markers of nouns in Selkup (corpus-oriented approach). *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025, pp. 225-236 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-16.

Acknowledgements. This paper was supported by the grant RNF № 20-18-00403 “Digital description of the dialects of the Uralic languages based on the analysis of big data”.

1. Введение

Проблема отражения двойственного числа в селькупских диалектах уже рассматривалась в различных работах, среди которых отдельно стоит отметить труды Е. Г. Беккер (и др.) [1, 2], Н. П. Максимовой [3], А. И. Кузнецовой (и др.) [4, 5], О. А. Казакевич [6], А. В. Байдак и Н. С. Гнездиловой [7], Г. П. Поздеевой [8]. В данных научных изысканиях обсуждается выражение двойственности в целом и маркирование суффиксами дуалиса существительных, местоимений, числительных, глаголов, отражение суффиксов в категории посессивности, а также единичное и множественное маркирование показателями двойственности различных конструкций в предложении. Анализ литературы показывает ‘неоднородную’ тенденцию к редукции или исчезновению показателей суффиксов дуалиса на разных языковых территориях. Также сразу отметим, что в южных диалектах (южных, переходных и центральных в нашей терминологии) одушевленность/неодушевленность имени существительного не влияет на употребление дуалиса, в то время как в северных диалектах неодушевленные существительные достаточно редко употребляются с ним [7].

Целью исследования является анализ распределения маркирования двойственности существительных аналитическим и синтетическим способом в южных, переходных, центральных и северных диалектах и говорах селькупского языка.

В качестве материала послужили корпусные данные, расположенные на цифровой платформе Lingvodoc [9] либо в личных архивах (файлы программы Fieldworks Language

Explorer), состоящие из глоссированных текстов архива лаборатории языков народов Сибири Томского государственного педагогического университета (далее ЛЯНС) [10], текстов и словарей М. А. Кастрена [11], Н. П. Григоровского [12-15], св. Макария (Невского) [16, 17] и Л. Шабо [18]. Общий объем корпусных данных составляет более 85,000 словоформ. Также при анализе использовался словарь северных диалектов [19] и данные приведенных ранее грамматик и статей [1-8].

С одной стороны, исследование опирается на корпусные методы, что позволяет увидеть распределение явлений на основе анализа относительно больших для бесписьменного языка данных. С другой стороны, используются фактические данные словарей и статей, в том числе формализованные, для создания полной картины дистрибуции дуалиса т.к. не для всех рассматриваемых материалов есть корпуса и соответственно статистика распространения явлений.

В работе используется классификация диалектов Т. Янурика [20], с разделением территории на

- северную группу – включает диалекты:

- 1) тазовский;
- 2) ларьякский (верхнетолькинский);
- 3) карасинский;
- 4) туруханский;
- 5) баихинский;
- 6) елогуйский;

- центральную группу:

- 7) ваховский;
- 8) тымский;
- 9) васюганский;
- 10) нарымский;

- южную группу:

- 11) среднеобской;
- 12) чаинский;
- 13) кетский;
- 14) верхнеобской;
- 15) чулымский.

В нашем исследовании дополнительно выделяется переходная группа в виде иванкинского говора среднеобского диалекта.

2. Общие положения

Перед тем как перейти к анализу распределения маркирования дуалиса рассмотрим наиболее важные положения, отраженные в предыдущих исследованиях.

Так, использование суффиксов двойственного числа имен существительных по диалектам представлено в таблице Э. Г. Беккер [2], однако отсутствие в ее работе многих примеров, подтверждающих данное распределение, а также вопросы, связанные с прототипичностью использования ряда суффиксов на отдельных языковых территориях, вынуждает нас еще раз проанализировать дистрибуцию маркеров дуалиса по диалектам и говорам. Предложенная далее модель из работы Э. Г. Беккер, однако, с существенными дополнениями, была взята за основу нашего исследования (См. Табл. 2). Представим оригинал.

Табл. 1. Использование суффиксов двойственного числа имен существительных в диалектах селькупского языка [2].

Table 1. Use of the dual noun suffixes in the dialects of the Selkup language [2].

Суффиксы дв. числа	Сель. (Таз)	Чум. (Тым)	Чум. (Васюган)	Чум. (Ср. Обь)	Шеш. (Ср. Обь)	Сюс. (В. Обь)	Сюс. (Кеть)
-q(V) и его ФВ	+	+	+	+	-	+	+
-ReR и его ФВ	+	+	+	-	-	-	-
-štja и его ФВ	-	-	-	+	+	-	-
-štjaq и его ФВ	-	-	-	+	-	-	-
-sak и его ФВ	+	+	+	+	+	+	+

В целом двойственное число существительных в селькупских диалектах и говорах может быть выражено при помощи суффиксов: 1) -q(V) ‘DU’ (также -āqi / -ōqi ‘DU’) (ПУ *-ka~ -kā + *-n или *-ń [21]), 2) -štja (*šitə ‘два’ > -št- ‘DU’ + -ja ‘DU’), 3) -štjaq(V) (*šitə ‘два’ > -št- ‘DU’ + -ja- ‘DU’ + -q(V) ‘DU’), 4) -ja ‘DU’.

Также показатель двойственного числа наблюдается в сочетании с маркером взаимной связи -sa- ‘RECIP’: -saqV (-sa-qV ‘RECIP-DU’), -syqāqV (-sy-qā-qV ‘RECIP-DU-DU’); и собирательной множественности -mi ‘COLL’: -miq(V) (-mi-q(V) ‘COLL-DU’), -miqVq(V) (-mi-qV-q(V) ‘COLL-DU-DU’) для обозначения совокупности однородных предметов, например, терминов родства или близких связей. В материалах по северным диалектам [4, 19] можно обнаружить, что суффикс собирательной множественности представлен в виде единого маркера -jmi- ‘COLL’ или -lmi- ‘COLL’, однако следуя за Г. Н. Прокофьевым [22], здесь и далее логично рассматривать его как два отдельных показателя: -j / -l ‘ADJz’ + mi- ‘COLL, вещь’, тем более, что в ряде случаев первый элемент -j / -l ‘ADJz’ отсутствует. Также активно используется аналитический способ выражения двойственности, заключающийся в использовании числительного sitte ‘два’ (иногда к числительному может дополнительно добавляться маркер дуалиса) и существительного в единственном, а иногда и множественном числе. Случаи двойного маркирования суффиксами дуалиса отдельно не рассматриваются (например, два-DU + сущ.-DU). Суффиксы дуалиса числительных и местоимений целенаправленно не включены в данный раздел, т.к. они имеют другое распределение.

3. Дистрибуция способов маркирования двойственного числа существительных

Далее производится попытка проанализировать распределение маркирования двойственности существительных аналитическим и синтетическим способом в южных, переходных, центральных и северных диалектах и говорах селькупского языка на материале XIX-XXI вв. Дистрибуция данных способов представлена в Табл. 1.

На основании данных таблицы можно увидеть, что в отдельных диалектах и говорах суффиксы существительных двойственного числа не зафиксированы. Это может быть обусловлено недостаточным объемом материалов, что, в частности, относится к данным XIX в. С другой стороны, как уже отмечалось, в селькупской академической литературе на примерах отдельных диалектов действительно отмечаются тенденции к утрате суффиксов двойственного числа [5, 6] и незасвидетельствованность суффиксов в материалах из таблицы действительно может говорить об их выходе из повседневного употребления. Рассмотрим более подробно данные явления.

Табл. 1. Распределение способов маркирования двойственного числа существительных.

Table 1. Distribution of the marking ways of the dual number of nouns.

Южные	XIX в.					
	Григоровский (н.-чаин.)		Кастрен (чаин.)	Кастрен (чул.)	Кастрен (н.-кет.)	Кастрен (в.-кет.)
	- <i>зи</i> (1), - <i>зынь</i> (> - <i>зык</i>) (RECIP.DU) (1), числ. <i>sədə</i> ‘два’ + сущ. без суф. DU (21), самопер. тексты: - <i>зъ</i> (1), - <i>ягы</i> (2), - <i>мыги</i> (COLL.DU) (1)		- <i>sɬ</i> (RECIP.DU)	-	числ. <i>sitte</i> ‘два’ + сущ. без суф. DU (2)	-
	XX в.					
		ЛЯНС (Сон. ср.- об.)			ЛЯНС (кет.)	
		- <i>χ</i> [3], - <i>zʏə</i> (- <i>zəq</i>) (RECIP.DU) (6), числ. <i>sədə</i> ‘два’ + сущ. без суф. DU (11)			- <i>Ri</i> [3], (н.-/ср.-кет.) - <i>zeqi</i> (- <i>ziqqi</i>) (RECIP.DU) (31), (в.-кет.) - <i>miq</i> (- <i>mqə</i>) (COLL.DU) (5), числ. <i>sittə</i> ‘два’ + сущ. без суф. DU (18)	
Переходные	XIX в.					
	Кастрен (Ив. ср.-об.)			Макарий (Ив. ср.-об.)		
	- <i>sɬi</i> (RECIP.DU) (1), числ. <i>šede</i> ‘два’ + сущ. без суф. DU (36)			числ. <i>шиты</i> ‘два’ + сущ. без суф. DU (3)		
	XX в.					
	ЛЯНС (Ив. ср.-об.)					
	- <i>štja</i> [3], - <i>siq</i> (RECIP.DU) (2), числ. <i>šit</i> ‘два’ + сущ. (в SG или PL) без суф. DU (28)					
Центральные	XIX в.					
	Кастрен (нар.)	Макарий (Нар. нар.)		Кастрен (тым.)		
	- <i>ka</i> / - <i>ga</i> / - <i>ox</i> / - <i>eh</i> , числ. <i>šeda</i> ‘два’ + сущ. без суф. DU (1)	числ. <i>шед</i> ‘два’ + сущ. без суф. DU (1)		-		

Центральные	XX в.				
	ЛЯНС (нар.)		ЛЯНС (вас.)	Шабо (Нап. тым.)	ЛЯНС (тым.)
	-<i>štjaR</i> (-<i>ščaR</i>, -<i>štijaye</i>, -<i>štijah</i>) (16), -<i>štja</i> (-<i>štija</i>) (8), -<i>səR</i> (-<i>sär</i>, -<i>ssih</i>) (RECIP.DU) (23), числ. <i>šede</i> ‘два’ + сущ. (в SG (1 пример PL)) без суф. DU (78)		-<i>R</i> (-<i>k</i>, -<i>ga</i>) (35), -<i>saR</i> (RECIP.DU) [3], -<i>miReR</i> (-<i>mReR</i>) (COLL.DU) (12), числ. <i>šidi</i> ‘два’ + сущ. без суф. DU (21)	-<i>štja</i> (DU) (1), -<i>la-štja</i> (PL-DU) (2)	-<i>R</i> (-<i>y</i>) (6), -<i>ja</i> (1), -<i>šəR</i> [3], числ. <i>šidə</i> ‘два’ + сущ. без суф. DU (36)
Северные	XIX в.				
	Кастрен (таз.)			Кастрен (баи.)	Кастрен (кар.)
	<i>-g</i> / <i>-ag</i> / <i>-eäg</i> / <i>-äag</i> / <i>-og</i>			<i>-óg</i>	<i>-ag</i> / <i>-ga</i> / <i>-x</i> / <i>-óg</i>
	XX–XXI в.				
	Кузнецова и др. [4], Казакевич [5], Казакевич, Будянская [19] (ср.-таз.)	Казакевич [6], Казакевич, Будянская [19] (в.-таз.)	Казакевич [6] Казакевич, Будянская [19] (в.-тольк.)	Кузнецова [5], Казакевич [6] (мат. Л.А. Варковицкой (1941)) (баи.)	Кузнецова [5], Казакевич [6] (1998-1999, 2003) (тур.)
	-<i>qi</i>, -<i>äqi</i>, (-<i>j</i> после <i>ö</i> или <i>ä</i>), -<i>syqäqi</i> (RECIP.DU), -<i>мыҕäҕи</i> (COLL.DU), числ. <i>šitty</i> ‘два’ + сущ. без суф. DU суф. DU сохр.	-<i>qi</i>, -<i>syqäqi</i> (RECIP.DU), числ. <i>šitty</i> ‘два’ + сущ. без суф. DU суф. DU сохр.	-<i>муҕäҕу</i> (RECIP.DU), числ. <i>иитты</i> ‘два’ + сущ. без суф. DU суф. DU сохр.	-<i>oqi</i>, -<i>syqäqi</i> (RECIP.DU), числ. <i>šitty</i> ‘два’ + сущ. без суф. DU суф. DU сохр.	-<i>qa</i> (DU) (редко), -<i>öqi</i> в сочетании <i>sitty qimöqi</i> ‘они двое’, -<i>syqäqi</i> (RECIP.DU) (сохр.), -<i>syqä</i> (RECIP.DU) (сохр.), числ. <i>šitty</i> ‘два’ + сущ. (в SG или PL или RECIP.PL) без суф. DU суф. DU не сохр.

В южных материалах XIX–XX вв., а именно в нижнечаинском идиоме Н. П. Григоровского XIX в., сондровских и кетских данных ЛЯНС за XX в. зафиксированы отдельные примеры использования суффикса *-gi* ‘DU’ с основами существительных. Далее в этих же материалах, а также в чаинских данных М. А. Кастрена XIX в., наблюдается использование показателя *-q(V)* ‘DU’ вместе с суффиксами *-sa-* ‘RECIP’ и *-mi-* ‘COLL’, которые появляются только с одушевленными объектами: (ср.-об., н.-ср.-кет.) *-saqV* (> *-san*) ‘RECIP.DU’, (в.-кет.) *-miq(V)* ‘COLL.DU’. Самым распространенным способом кодирования дуалиса во всех южных данных XIX–XX вв. является использование аналитической конструкции, состоящей из числительного *sitte* / *sədə* ‘два’ и существительного в единственном числе. Отдельно упомянем самопереведенные тексты Н. П. Григоровского XIX в., где дополнительно встречаются примеры с *-gъ* ‘DU’, *-ягы* (н.-ср.-кет.) ‘DU-DU’ ‘DU’ и *-мыги* ‘COLL.DU’.

Юж. XIX в.

Григоровский (н.-чаин.): 1) (ориг.) *эрá пáя-gи* (DU) ‘старик со старухой’, *áльдзигá-зынъ* (RECIP.DU) ‘с бабушкой вдвоем (Итя)’, *сыдъ сырь* ‘две коровы’; 2) (самопер.) *крóста-gъ* (DU) ‘два креста’, *разбóйныкъ-ягъ* (DU) ‘разбойники’, *апóстоль-ягы-мъ* (DU-ACC) ‘апостолов’, *Вапу́йл-мыги* (COLL.DU) ‘Вафуил и его жена’;

Кастрен (чаи.): *debe-sъ* (RECIP.DU) ‘муж с женой’;

Кастрен (н.-кет.): *sitte kum* ‘два человека’, *sitt_ade* ‘два оленя’.

Юж. XX в.

ЛЯНС (Сон. ср.-об.): *arraga-χ* (DU) ‘два старика’ [3], *éra pajá-zaq* (RECIP.DU) ‘муж с женой’, *sədə né:t* ‘две дочери’;

ЛЯНС (кет.): (БЯ) *tiχbínná-Ri* ‘два брата’ [3], (YO) *oppo-ziqqi* (RECIP.DU) ‘две сестры’, (Марк.) *essa-miq* (COLL.DU) ‘родители двое’, (YO) *šittə tuto* ‘два караса’.

В переходных иванкинских данных среднеобского диалекта XX в. М. П. Максимовой наблюдаются примеры использования суффикса *-štja* (*-št-ja* ‘DU-DU’) ‘DU’ [3]. В свою очередь в материалах XIX в. М. А. Кастрена и данных ЛЯНС за XX в. фиксируются случаи использования суффикса *-q(V)* ‘DU’ вместе с суффиксом взаимной связи *-sa-* ‘RECIP’ при терминах родства: *-saqV* ‘RECIP.DU’. Основной стратегией маркирования двойственности существительных во всех иванкинских материалах XIX–XX вв. является использование аналитической конструкции, состоящей из числительного *šede* / *šit* ‘два’ и существительного в единственном числе. Стоит также отметить, что во многих примерах за XX в. из данных ЛЯНС фиксируется стратегия, демонстрирующая появление суффикса множественного числа на существительных в конструкции с числительным *šit* ‘два’ при обозначении двойственности.

Перех. XIX в.

Кастрен (Ив. ср.-об.): *debi-sъi* (RECIP.DU) ‘муж с женой’, *šede got* ‘два человека’;

Макарий (Ив. ср.-об.): *шиты пар* ‘два раза’, *шит квыл-зе* (COM) ‘(с) двумя рыбами’.

Перех. XX в.

ЛЯНС (Ив. ср.-об.): *tebiłgu-štja* (DU) ‘два парня’ [3], *ara: pajá:-siq* (RECIP.DU) ‘муж с женой двое’, *šit kone:r* ‘два барана’, *šed pay* ‘два ножа’, *šit t'emn 'a-la-n-ze* (PL-GEN-COM) ‘с двумя братьями’.

В центральных диалектах использование суффикса *-q(V)* ‘DU’ с основами существительных наблюдается только в нарымских данных М. А. Кастрена XIX в., васюганских и тымских текстах ЛЯНС за XX в. В нарымских материалах ЛЯНС за XX в., относящихся к более южной части нарымского ареала (выше по течению Оби), вместо суффикса *-q(V)* ‘DU’ используются составные показатели двойственности *-štja* (**šitə* ‘два’ > *-št-* ‘DU’ + *-ja* ‘DU’), *-štjaq(V)* (**šitə* ‘два’ > *-št-* ‘DU’ + *-ja-* ‘DU’ + *-q(V)* DU’), что является инновацией. Суффикс *-štja* (*-št-ja* ‘DU-DU’) ‘DU’ (один пример), а также составной *-la-štja* ‘PL-DU’ (два примера) также были зафиксированы в тымском (смешанном) идиоме Л. Шабо, что, предположительно, может

характеризовать его 'нарымские' черты. Судя по всему, следует отделять северную часть нарымского ареала с центром в Нарыме и более южную территорию вверх по течению, т.к. в первом случае с основами существительных используется суффикс *-q(V)* 'DU', а во втором отмечаются инновации в виде *-štja* (*-št-ja* 'DU-DU') 'DU' и *-štjaq(V)* (*-št-ja-q(V)* 'DU-DU-DU') 'DU'. Дополнительно, в тымских материалах зафиксирован лишь один пример существительного с суффиксом *-ja* 'DU'. Примеры с объектами, обозначающими термины родства, с показателем *-q(V)* 'DU' вместе с суффиксом взаимной связи *-sa-* 'RECIP' отмечаются в нарымских, васюганских и тымских данных ЛЯНС за XX в. Также примеры с терминами родства и суффиксом *-q(V)* 'DU' или удвоенным показателем *-qV-q(V)* 'DU-DU' и суффиксом собирательной множественности *-mi-* 'COLL' обнаружены только в васюганских материалах ЛЯНС за XX в. Способом кодирования двойственности существительных, который отмечается во всех центральных материалах XIX–XX вв. (кроме данных XX в. Л. Шабо), является аналитическая стратегия, заключающаяся в использовании числительного *šede / šidi* 'два' и существительного в единственном числе. В нарымском, васюганском и тымском диалектах по данным ЛЯНС за XX в. это является одной из основных стратегий кодирования двойственности. Также встретился один пример использования числительного *šede* 'два' с существительным во множественном числе в нарымских данных ЛЯНС за XX в.

Центр. XIX в.

Кастрен (нар.): *mát-ka* (DU) 'два дома', *hir-ga* (DU) 'две коровы', *ud-óh / ud-óh* (DU) 'две руки', *kega-eh* (DU) 'две реки', *šeda_gop* 'два человека';

Макарий (нар.): *šed odo-m-ð* (ACC-POSS.3SG) 'две руки его'.

Центр. XX в.

ЛЯНС (нар.): (Кён., Нель.) *nade-štjaR* (DU) 'две девушки', (Соис., Нель.) *qu-štija* (DU) 'два человека', (Лас.) *č'imn 'á-səR* (RECIP.DU) 'два брата', (Тюх.) *šedi č'undə* 'две лошади', (Пуд.) *šede nadek* 'две девочки', (Соис., Кен.) *šidi kana-la* (PL) 'две собаки';

ЛЯНС (вас.): (Воль.) *paja-R / paja-k* (DU) 'две старухи', (Воль.) *heñña-saR* (RECIP.DU) 'две сестры' [3: 107], (Калг.) *ad'al-mReR* (COLL.DU) 'отец с матерью', (Воль.) *šidi haj* 'два глаза';

Шабо (Нап. тым.): *paja-štja-* (DU) 'ваши две жены', *n 'aab-la-štja-* (PL-DU) 'утки две ваши';

ЛЯНС (тым.): (Варг., Нюль.) *era-y* (DU) 'два старика', (Варг., Нюль.) *nede-ja* (DU) 'две девочки', (Ванж.) *ieñña-šəR* (RECIP.DU) 'два брата' [3], (Варг.) *šidə šiy* 'два соболя', (Варг.) *šidə koryo* 'два медведя'.

В материалах М. А. Кастрена XIX в. кодирование двойственности зафиксировано только частично, ввиду небольшого количества материала. Так единственной обнаруженной стратегией является использование суффикса *-q(V)* 'DU' с основами существительных в тазовских, баихенских и карасинских материалах. Во всех северных диалектах XX–XXI вв. в разной степени наблюдается использование суффиксов *-q(V)* 'DU', *-āq̄i / -ōq̄i* 'DU' с более полной их утратой в туруханских данных. В среднетазовском после гласного *ō* или *ā* окончание *-q(V)* 'DU' может отпадать или заменяться *j* или *j*-образный призвук: *-j* 'DU' [4: 168]. Одним из основных способов кодирования двойственного числа существительных является аналитическая стратегия с числительным *šitty* 'два' и существительным в единственном числе. В туруханских материалах отмечаются гибридные аналитические стратегии, когда за числительным *šitty* 'два' идет существительное с *-t* 'PL' или с *-šy-* 'RECIP' и *-t* 'PL'. А. И. Кузнецова и О. А. Казакевич в своих исследованиях рассматривают баихенские материалы Л. А. Варковицкой 1941 г. и туруханские материалы 1990-1999, 2003 гг. в рамках диахронии одного баихенского диалекта, где к концу XX в, началу XXI в. отмечена почти полная утрата суффиксов двойственности у существительных, за исключением редких упомянутых в таблице случаев (в частности в терминах родства с суффиксом *-sy-* 'RECIP') [5, 6]. Во всех северных материалах за XX–XXI в. из записей А. И. Кузнецовой, О. А.

Казакевич и Е. М. Будянской, судя по примерам, сохраняется использование суффикса двойственного числа, однако только при появлении вместе с маркерами взаимной связи *-šy-* или собирательной множественности *-mi-* при терминах родства. Так в средне- и верхнетазовских, баихенских и туруханских материалах отмечено использования суффикса *-sy-* ‘RECIP’ и удвоенного показателя двойственного числа *-qā-qV* ‘DU-DU’. Также в туруханских материалах встречаются примеры с *-sy-* ‘RECIP’ и единичным показателем *-q(V)* ‘DU’. В среднетазовских и вехнетолькинских материалах с терминами родства используется суффикс *-mi-* ‘COLL’ вместе с *-qā-qV* ‘DU-DU’.

Сев. XIX в.

Кастрен (таз.): *lokà-g* (DU) ‘две лисы’, *yn-og* (DU) ‘два ремешка’, *irà-g* (DU) ‘два старика’;
Кастрен (баи.): *yn-óg* (DU) ‘два ремешка’, *ug-óg* (DU) ‘два конца’, *pang-óg* (DU) ‘две вены’;
Кастрен (кар.): *aatā-ag* (-x, -ga) (DU) ‘два оленя’, *kum-óg* (DU) ‘два человека’;

Сев. XX–XXI в.

Кузнецова и др. [4], Казакевич [6], Казакевич, Будянская [19] (Ср.-таз. таз.): *imā-qī* (DU) ‘две женщины’, *qum-āqī* (DU) ‘два человека’, *mosō-j* (DU) ‘два леса’, *ñéñña-syqāqī* (RECIP.DU) ‘две родные сестры’, *амаль-мыḡāḡи* (COLL.DU) ‘родители’, *šitty pōt* ‘два года’;

Казакевич [6], Казакевич, Будянская [19] (В.-таз. таз.): *ima-qī* (DU) ‘две женщины’, *чона-сыḡāḡи* (COLL.DU) ‘два брата’, *šitty kory* ‘два лабаза’;

Казакевич [6], Казакевич, Будянская [19] (в.-тольк.): *iraj-мyḡāḡху* (COLL.DU) ‘муж с женой’, *шитты пōры* ‘два раза’;

Кузнецова [5], Казакевич [6] (по мат. Л. А. Варковицкой 1941 г.) (баи.): *šitty qum-oqy* (DU) ‘два человека’, *šitty ñannyl āmā-syqāqu* (COLL.DU) ‘две гагары матки’, *šitty qumy-p* (ACC) ‘двоих людей’;

Кузнецова [5], Казакевич [6] (1998-1999, 2003) (тур.): *lōzo-qa-t* (DU-POSS.3SG) ‘эти два черта’, *sitty qum-ōqī* (DU) ‘они двое (два человека)’ (только в этом сочетании и только как сабститут личного местоимения), *ṭimña-syqāqī* (COLL.DU) ‘двое братьев’, *šitty ñéñña-syqā* (COLL.DU) ‘две сестры’, *šitty ṭimña-šyt* (COLL.PL), *lozy-t* (PL) ‘два брата, черти’.

4. Заключение

- 1) В результате исследования было выяснено, что использование суффикса *-q(V)* ‘DU’ вместе с существительными в селькупских диалектах и говорах неоднородно – показатель не обнаруживается в ряде материалов, когда присоединяется напрямую к основе (переходные говоры, нарымский диалект) XIX–XXI в., в тоже время он стабильно используется с одушевленными существительными, обозначающими совокупность двух однородных предметов при помощи суффиксов взаимной связи и собирательного множества: *-sa-* ‘RECIP’ / *-mi-* ‘COLL’ + *-qV(q(V))* ‘DU(-DU)’.
- 2) В материалах XX в. ЛЯНС по нарымскому диалекту (южная часть языковой территории) и иванкинскому говору среднеобского диалекта суффикс *-q(V)* ‘DU’ не используется (за исключением терминов родства вместе с *-sa-* ‘RECIP’). Вместо этого в нарымских материалах фиксируются новообразования *-št-ja* ‘DU-DU’ и *-št-ja-q(V)* ‘DU-DU-DU’, в иванкинских – отдельные примеры на *-št-ja* ‘DU-DU’.
- 3) В васюганских данных ЛЯНС и данных по северным диалектам за XX–XXI в. обнаружены случаи появления удвоенного дуалиса с суффиксами взаимной связи или собирательного множества: *-sa-* ‘RECIP’ / *-mi-* ‘COLL’ + *-qV-q(V)* ‘DU-DU’.
- 4) Судя по всему, следует отделять северную часть нарымского ареала с центром в Нарыме и более южную территорию вверх по течению Оби, т.к. в первом случае там используется суффикс *-q(V)* ‘DU’, а во втором отмечаются инновации в виде *-štja* (*-št-ja* ‘DU-DU’) ‘DU’ и *-štjaq(V)* (*-št-ja-q(V)* ‘DU-DU-DU’) ‘DU’.

- 5) Суффикс *-ja* 'DU' встретился только один раз в тымском диалекте.
- 6) Во всех селькупских материалах главной или одной из главных стратегий маркирования двойственности существительных является аналитическая стратегия, при которой используется числительное *sitte* 'два' и существительное в единственном (в отдельных случаях во множественном) числе. Скорее всего, синтетическая и аналитическая стратегии в равной степени были присущи и раньше селькупским диалектам и говорам, однако, со временем, аналитическая стратегия начинает преобладать во всех диалектных зонах.

Сокращения

- 1) БЯ – Белый Яр на Кети, Ванж. – Ванджилыкынак на Тyme, Варг. – Варгананджино на Тyme, Воль. – Вольджа на Чижапке, Ив. – Иванкино – на Оби, Калг. – Калганак на Чижапке, Кён. – Кёнга на Кёнге, Лас. – Ласкино на Оби, Марк. – Марково на Кети, Нап. – Напас на Тyme; Нель. – Нельмач на Парабели, Нюль. – Нюльядрово на Тyme, Пуд. – Пудино на Чузике, Соис. – Соиспаево на Парабели, Тюх. – Тюхтерево на Оби, УО – Усть-Озерное на Кети.
- 2) баи. – баихенский, в.-кет. – верхнекетский, в. Обь – верхняя Обь, в.-таз. – верхнетазовский, в.-тольк. – верхнетолькинский, вас. – васюганский, кар. – карасинский, кет. – кетский, н.-кет. – нижнекетский, н.-чаин. – нижнечаинский, нар. – нарымский, ср.-кет. – среднекетский, ср.-об. – среднеобской, ср. Обь – средняя Обь, ср.-таз. – среднетазовский, таз. – тазовский, тым. – тымский, тур. – туруханский, чаи. – чаинский, чул. – чулымский.
- 3) сель. – селькупы, чум. – чумылькупы, шеш. – шешкупы / шешкумы, сюс. – сюсюкумы.
- 4) ЛЯНС – лаборатория языков народов Сибири, мат. – материалы, ориг. – оригинальные, ПУ – прауральский, самопер. – самопереведенные, сохр. – сохраняется, суф. – суффикс, сущ. – существительное, табл. – таблица, ФВ – фонетический вариант, числ. – числительное.
- 5) ACC – аккузатив, ADJz – адъективизатор, COLL – суффикс собирательной множественности, DU – двойственное число, PL – множественное число, RECIP – суффикс взаимной связи / реципрок, SG – единственное число.

Список литературы / References

- [1]. Беккер Э. Г. Грамматические категории имени существительного в южных диалектах селькупского языка: дисс. ... доктора филол. наук: 10.02.07. – Томск, 1984. – 466 с. / Bekker E. G. Grammatical categories of the noun in the southern dialects of the Selkup language: Thesis ... doctor of philological sciences: 10.02.07. – Tomsk, 1984. – 466 p. (in Russian).
- [2]. Беккер Э. Г., Алиткина Л. А., Быконя В. В., Ильяшенко И. А. Морфология селькупского языка. Южные диалекты. Ч. 1. Томск, 1995. – 292 с. / Bekker E. G., Alitkina L. A., Bykonya V. V., Ilyashenko I. A. Morphology of the Selkup language. Southern dialects. Part 1. Tomsk, 1995. – 292 p. (in Russian).
- [3]. Максимова Н. П. Морфологические способы выражения двойственности в селькупском языке // Структура палеоазиатских и самодийских языков. Томск, 1984, с. 106–114. / Maksimova N. P. Morphological ways of expressing duality in the Selkup language // Structure of Paleoasiatic and Samoyedic Languages. Tomsk, 1984, pp. 106-114 (in Russian).
- [4]. Кузнецова А. И., Хелимский Е. А., Грушкина Е. В. Очерки по селькупскому языку. Тазовский диалект. Том 1. М., 1980 / Kuznetsova A. I., Helimsky E. A., Grushkina E. V. Sketches of the Selkup language. Taz dialect. Volume 1. Moscow, 1980. (in Russian).
- [5]. Кузнецова А. И. Эволюция категории двойственного числа в баишенском говоре туруханских селькупов // Самодийцы. Материалы IV Сибирского симпозиума «Культурное наследие народов Западной Сибири». Тобольск; Омск, 2001 / Kuznetsova A. I. Evolution of the dual number category in

- the Baikha subdialect of the Turukhansk Selkups // *Samoyeds. Materials of the IV Siberian Symposium "Cultural heritage of the peoples of Western Siberia"*. Tobolsk; Omsk, 2001 (in Russian).
- [6]. Казакевич О. А. Две женщины, семь теснин и тридцать воинов (О выражении квантитативности в фольклорных текстах северных селькупов) // *Логический анализ языка. Квантитативный аспект языка*. М.: Индрик, 2005. С. 384-399 / Kazakevich O. A. Two women, seven gorges and thirty warriors (On the expression of quantification in the folklore texts of the Northern Selkups) // *Logical analysis of the language. The quantitative aspect of language*. M.: Indrik, 2005. pp. 384-399 (in Russian).
 - [7]. Байдак А. В., Гнездилова Н. С. Двойственное число в южных и центральных диалектах селькупского языка // *Linguistica Uralica*. – 2015. – Vol. 51, Iss. 3. – pp. 212-224 / Baidak A.V., Gnezdilova N. S. The dual number in the southern and central dialects of the Selkup language // *Linguistica Uralica*. – 2015. – Vol. 51, Iss. 3. – pp. 212-224 (in Russian).
 - [8]. Поздеева Г. П. Особенности числового согласования в диалектах селькупского языка: дис. ... кандидата филологических наук: 10.02.02. – Москва, 2015. – 280 с. / Pozdeeva G. Peculiarities of numerical agreement in the dialects of the Selkup language: thesis ... candidate of philological sciences: 02/10/02. – Moscow, 2015. – 280 p. (in Russian).
 - [9]. Lingvodoc. Доступно по ссылке: <http://lingvodoc.ispras.ru> (дата обращения 06.11.2024).
 - [10]. Архив ЛЯНС. Архив лаборатории языков народов Сибири. ТГПУ. Селькупы. Тома: 1–65. Томск, 1952–1985. / The LSIL Archive. Archive of the laboratory of Siberian indigenous languages. TSPU. Selkups. Volumes: 1-65. Tomsk, 1952-1985 (in Russian).
 - [11]. Alatalo, Jarmo (2024). *Manuscripta Castreniana Ostiak-Samoiedica* (MCOS). Available at: <https://www.sgr.fi/manuscripta/ostiaksamoiedica> (accessed: 06.11.2024).
 - [12]. Григоровский Н. П. Азбука сюсогой гулани / Сост. Н. П. Григоровским для инородцев Нарымского края. Издание православного миссионерского общества. Казань, 1879 / Grigorovsky N. P. ABC book for Sjussogoj gulani / Compiled by N. P. Grigorovsky for the natives of the Narym Krai. Publication of the Orthodox Missionary Society. Kazan, 1879. (in Russian).
 - [13]. Григоровский Н. П. Объяснение праздников Св. Церкви. На остячко-самоедском языке / Сост. Н. П. Григоровским для инородцев Нарымского края. Казань, 1879. / Grigorovsky N. P. Explanation of the holidays of the Holy Church. In the Ostyak-Samoyed language / Compiled by N. P. Grigorovsky for the natives of the Narym Krai. Kazan, 1879. (in Russian).
 - [14]. Григоровский Н. П. Священная история. На остячко-самоедском языке / Сост. Н. П. Григоровским для инородцев Нарымского края. Казань, 1879 / Grigorovsky N. P. Sacred History. In the Ostyak-Samoyed language / Compiled by N. P. Grigorovsky for the natives of the Narym Krai. Kazan, 1879. (in Russian).
 - [15]. Григоровский Н. П. Молитвы и о сердечной молитве к Богу. На остячко-самоедском языке / Сост. Н. П. Григоровским для инородцев Нарымского края. Казань, 1879 / Grigorovsky N. P. Prayers and about the heartfelt prayer to God. In the Ostyak-Samoyed language / Compiled by N. P. Grigorovsky for the natives of the Narym Krai. Kazan, 1879. (in Russian).
 - [16]. Макарий (Невский), еп. Материалы для ознакомления с наречием остяков Нарымского края, 1887 г. // Приложение № 7 к отчету об Алтайской и Киргизской миссиях за 1887 г. / Makary (Nevsky), bish. Materials for familiarization with the dialect of the Ostyaks of the Narym Krai, 1887 // Appendix No. 7 to the report on the Altai and Kyrgyz missions for 1887. (in Russian).
 - [17]. Макарий (Невский), еп. Беседы об истинном Боге и истинной вере на наречии обских остяков. Томск, 1900 / Makary (Nevsky), bish. Conversations about the true God and the true faith in the dialect of the Ob Ostyaks. Tomsk, 1900. (in Russian).
 - [18]. Szabó L. Selkup texts with phonetic introduction and vocabulary // *Uralic and Altaic Series*. Bloomington, 1967. 62 p.
 - [19]. Казакевич О. А., Будянская Е. М. Диалектологический словарь селькупского языка (северное наречие) Екатеринбург: Институт филологии СО РАН, 2010 / Kazakevich O. A., Budyanskaya E. M. Dialectological dictionary of the Selkup language (Northern dialect) Yekaterinburg: Institute of Philology SB RAS, 2010 (in Russian).
 - [20]. Janurik T. A. Szölkup nyelvjárások osztályozása // *Nyelvtudományi Közlemények* 80. Budapest, 1978. O. 77–104.
 - [21]. Collinder B. Comparative Grammar of the Uralic Languages (Handbook of the Uralic Languages: Part 3). Stockholm: Almqvist and Wiksell. 1960.
 - [22]. Прокофьев Г. Н. Селькупский (остяко-самоедский) язык. Ч. 1: Селькупская грамматика. Ленинград, 1935 // Prokofiev G. N. Selkup (Ostyako-Samoyedic) language. Part 1: Selkup grammar. Leningrad, 1935.

Информация об авторах / Information about authors

Сергей Васильевич КОВЫЛИН – кандидат филологических наук, старший научный сотрудник лаборатории «Лингвистические платформы» Института системного программирования с 2019 года. Сфера научных интересов: корпусная лингвистика, компьютерная лингвистика, документация и ревитализация исчезающих языков и культур.

Sergei Vasilievich KOVYLIN – Cand. Sci. (Philology), Senior research fellow of the laboratory “Linguistic platforms” of the Institute for System Programming of the RAS since 2019. Research interests: corpus linguistics, computational linguistics, documentation and the revitalization of endangered languages and cultures.

DOI: 10.15514/ISPRAS-2025-37(3)-17



SumHiS: Extractive Summarization Exploiting Hidden Structure

^{1,2} P.A. Tikhonov, ORCID: 0009-0000-5847-2586 <Tikhonov@airi.net>

³ A.O. Ianina, ORCID: 0000-0002-6822-2801 <yanina@phystech.edu>

⁴ V.A. Malykh, ORCID: 0000-0002-4508-2527 <valentin@maly.hk>

¹ Artificial Intelligence Research Institute,
6, bldg. 2, Presnenskaya Embankment, Moscow, 123112, Russia.

² Skolkovo Institute of Science and Technology,
1, Bolshoy Boulevard 30, Skolkovo Innovation Center, Moscow, 121205, Russia.

³ Moscow Institute of Physics and Technology,
9, Institutskiy Lane, Dolgoprudny, Moscow Region, 141701, Russia.

⁴ ITMO University,
49, lit. A, Kronverksky Prospekt, St. Petersburg, 197101, Russia.

Abstract. Extractive summarization is a task of highlighting the most important parts of the text. We introduce a new approach to extractive summarization task using hidden clustering structure of the text. Experimental results on CNN/DailyMail demonstrate that our approach generates more accurate summaries than both extractive and abstractive methods, achieving state-of-the-art results in terms of ROUGE-2 metric exceeding the previous approaches by 10%. Additionally, we show that hidden structure of the text could be interpreted as aspects.

Keywords: summarization; NLP.

For citation: Tikhonov P.A., Ianina A.O., Malykh V.A. SumHiS: Extractive Summarization Exploiting Hidden Structure. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 3, 2025, pp. 237-250. DOI: 10.15514/ISPRAS-2025-37(3)-17.

Экстрактивная суммаризация с использованием скрытой структуры

^{1,2} П.А. Тихонов, ORCID: 0009-0000-5847-2586 <Tikhonov@airi.net>

³ А.О. Янина, ORCID: 0000-0002-6822-2801 <yanina@phystech.edu>

⁴ В.А. Малых, ORCID: 0000-0002-4508-2527 <valentin@maly.hk>

¹ Институт исследований искусственного интеллекта,
Россия, 123112, г. Москва, Пресненская набережная 6, стр. 2.

² Сколковский институт науки и технологий,

Россия, 121205, г. Москва, Большой бульвар 30, стр. 1, Инновационный центр Сколково.

³ Московский физико-технический институт,

Россия, 141701, Московская область, г. Долгопрудный, Институтский переулок, 9

⁴ Университет ИТМО,

Россия, 197101, г. Санкт-Петербург, Кронверкский проспект 49, лит. А.

Аннотация. Основной целью данной работы является разработка новой модели для задачи экстрактивной суммаризации текстов, которая использует скрытую кластерную структуру текста. Авторы предлагают двухэтапный подход, включающий модель ранжирования предложений и модель обнаружения скрытой структуры текста. Для воспроизводимости результатов можно было бы сделать более детальное описание гиперпараметров и процедуры обучения моделей, также было бы информативно представить примеры полученных суммаризаций.

Ключевые слова: суммаризация; обработка естественного языка.

Для цитирования: Тихонов П.А., Янина А.О., Малых В.А. SumHiS: Экстрактивная суммаризация с использованием скрытой структуры. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 237–250 (на английском языке). DOI: 10.15514/ISPRAS–2025–37(3)–17.

1. Introduction

Summaries are important for processing huge amounts of information. A good summary should be concise, accurate and easy-to-read. However, there can be multiple variants of a perfect summary, the same idea can be conveyed with various words. Moreover, people may find different facts of the main importance, waiting for them to be present in the summary. Most automatic text summarization algorithms do not take into account different aspects of the initial texts, providing a semantically neutral interpretation. We aim to bridge the gap between summarization approaches and aspect mining. Thus, we investigate two research directions within this work: text summarization and aspect extraction.

Text Summarization. There are two main approaches to text summarization: extractive and abstractive. Extractive methods highlight the most relevant phrases or sentences in the original text to form a summary. Alternatively, abstractive methods rephrase the text into a different form, and may not preserve the original semantic content.

Usually summarization has an underlying suggestion, that one summary should fulfill every informational demand. That is not true in many cases, e.g., imagine text about fruits in general, while a person is interested exactly in apples. In that toy example the proper summary for the aforementioned person should contain maximum information about apples with some occasional references to other fruits. Such a result can be achieved with aspect extraction techniques. The aspect extraction underlying suggestion is that each document consists of several aspects.

Hidden Document Structure. Revealing hidden document structure is important for getting a concise and accurate summary. One way to do so is via aspect extraction. Each aspect may be specified by explicit words or sometimes inferred implicitly from the text. For example, in the sentence “the image is very clear” the word “image” is an aspect term. The associated problem of aspect categorization is to group the same aspect expressions into a category. For example, the aspect terms “image”, “photo”, and “picture” can be grouped into one aspect category named Image.

Hidden document structure is conventionally associated with dividing a document into multiple facets, each of which may have its own sentiment. However, the structures may relate to different textual features, e.g., topics covered in the text. In this paper we concentrate on how the discovered structures helps to make the summaries more accurate. Although we do not interpret these discovered structures as aspects.

Our Approach. We propose an extractive summarization model, that we call SumHiS (*Summarization with Hidden Structure*), which utilizes representations from BERT model [1] and uses topical hidden document structure. In this work we introduce two blocks for creating extractive summaries. First, we use contextualized representations retrieved from a pre-trained language models to rank the sentences from a document according to their importance. Second, we further filter the already ranked sentences in order to focus the summary on the facts corresponding to main discovered topics within document.

Evaluated on CNN/DailyMail dataset [2], our approach outperforms previous extractive summarization state-of-the-art in terms of ROUGE-2 [3] metric by 10%. These results demonstrate the importance of topical structure inclusion for summarization task. Furthermore, we capitalize on the power of pre-trained language models combined with document structure discovery, that makes the resulting summary to focus on the most important topics and ideas mentioned in the initial text.

To summarize our key contributions are:

- A novel extractive summarization pipeline, which combines representations from pre-trained language models and hidden document structure discovery techniques.
- Our method outperforms prior work on the CNN-DailyMail dataset by a large margin in terms of ROUGE-2 and ROUGE-L metrics and can successfully be applied to real-world applications.
- Moreover, our model outperforms abstractive models too.

2. Related work

The earliest attempts of automatic summarization focused on extractive techniques, which find words or sentences in a document that capture its most salient content. Recent works use a variety of approaches. For example, [4] proposed a novel summary-level framework MatchSum and conceptualized extractive summarization as a semantic text matching problem. The authors proposed a Siamese-BERT architecture to compute the similarity between the source document and the candidate summary. In [5] the authors rely on extractive summarizers that identify salient sentences based on positional information.

Under supervised learning conditions, aspect-level sentiment classification is typically considered a classification problem. Early works [6-8] mainly used manually designed features such as sentiment lexicon, n-grams, and dependency information. However, these methods highly depend on the quality of the designed features, which is labor-intensive. With the advances of deep learning methods, various neural models [9-11] have been proposed for automatically learning target-dependent sentence representations for classification. The main idea behind these works is to develop neural architectures that are capable of learning continuous features without feature engineering and at the same time capturing the intricate relatedness between a target and context words.

Of course, there are many works in recent years in abstractive summarization. In the work [2] authors proposed to use encoder-decoder on a huge corpus to achieve good results in the abstractive summarization task. Later in work [12] use a different type of recurrence network and obtained the state-of-the-art results. Nallapati and co-authors used word copying mechanism from the input sequence to the output, thereby solving the problem with rare words. In the paper [13] Cohan and co-authors proposed a summarization model for very long documents, like scientific articles. They use the hierarchical encoder mechanism that models the discourse structure of a document. Putra et

al. [14] proposed to use so-called topical sentence, i.e., the most important one from the article, to generate news headline.

The last-mentioned works allowed us to suggest a hidden structure usage in summarization. We chose a model which is designed to capture a hidden structure, namely extract aspects from texts. Neural attention-based aspect extraction model (ABAE) is proposed in [15]. The main idea of this work is to create a matrix of vector representations which could be used to reconstruct a sentence vector representation. It is done under assumption that there is only one main aspect which a sentence has.

In the models like MatchSum, the authors use vector BERT representations of the sentences. We decided to follow this approach, but instead of classic binary prediction whether a sentence should be included or not we chose ranking approach, allowing us to filter the sentences basing on their score. We chose recent state of the art approach in text ranking SparTerm [16]. This model is using vector representations of input texts to predict their ranking. The vector representations are produced from fine-tuned BERT model. We adopted this approach with exception of irrelevant to us term prediction task.

3. Model description

This section presents the general overview of our extractive summarization system SumHiS, its architecture and the corresponding training strategy. Our system consists of two blocks: sentence ranking model and hidden structure discovery model. The model's interaction is shown in Fig. 1. The training process of our system also consists of two phases. First, we train sentence ranking model and then we use its output representations to train a hidden structure discovery model.

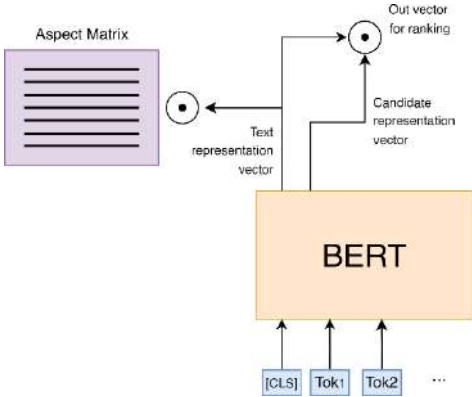


Fig. 1. SumHiS: ranking model (right) + hidden structure discovery model (left).

3.1 Ranking

We follow Term-based Sparse representations (SparTerm) setup [16] to train a ranking model. SparTerm learns sparse text representations by predicting the importance for each term in the vocabulary. Since we are working with extractive summarization, our model estimates importance of sentences instead of terms.

SparTerm represents text using BERT [1] model as follows: a text is fed into the model, and each term is embedded to a vector space. The term embeddings are averaged and used as a single text embedding. This text embedding is compared to other text embeddings thereby producing similarity scores. Similarly to SparTerm setup, we use BERT with specifically designed input. Each input is represented as a triplet (*text*, *pos_sentence*, *neg_sentence*), where *text* is a whole text of a document, *pos_sentence* is a sentence included into the golden summary, and *neg_sentence* is a sentence not included into the golden summary. The visualization of the model input is presented in Fig. 2.

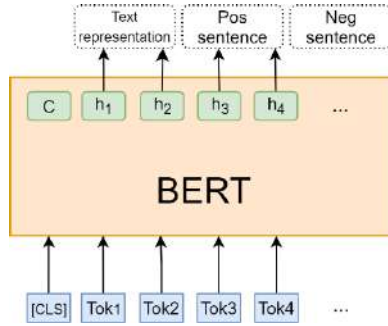


Fig. 2. Ranking model.

We aim to make a positive sentence representation as close as possible to a representation of a text and simultaneously make a representation of negative sentence as far as possible from it. Let $R = \{(t_1, s_{1,+}, s_{1,-}), \dots, (t_N, s_{N,+}, s_{N,-})\}$ denote a set of N training instances; each containing text t_i , positive candidate sentence $s_{i,+}$ and negative one $s_{i,-}$, indicating that $s_{i,+}$ is more relevant to the text than $s_{i,-}$. The ranking model is trained by optimizing a ranking objective which in our case is negative log likelihood of the positive sentence:

$$L_{\text{summ}}(t_i, s_{i,+}, s_{i,-}) = -\log \frac{e^{\text{sim}(t'_i, s'_{i,+})}}{e^{\text{sim}(t'_i, s'_{i,+})} + e^{\text{sim}(t'_i, s'_{i,-})}}$$

where $t'_i, s'_{i,+}, s'_{i,-}$ are dense representations of $t_i, s_{i,+}, s_{i,-}$ respectively, and sim denotes any similarity function. We use dot-product in our experiments.

During the training each document t_i is split into sentences, from which the triplets are generated. The output of the ranking model is an ordering of the sentences which are similar to the text summary from the closest one to the most distant one. Given this ordering one could create a summary for the text taking several top sentences.

3.2 Hidden Document Structure Discovery

Through experimentation, we found out that quality of summaries can be increased by adding information about hidden document structure.

We follow ABAE model [15] setup in order to capture hidden document structure. A sentence vector representation is considered to consist of weighted sum of several cluster representations. In case of ABAE these clusters are interpreted as aspects, while in our model we do not follow this interpretation and consider them as ordinal clusters.

The structure discovery model learns a matrix C of cluster embeddings of size $K \times n$, where K is the number of clusters, n is an embedding space size. We use an attention-like mechanism to take into account all the cluster representations and reconstruct the initial vector. We calculate each input score by calculating its dot product with each cluster embedding:

$$p_j = c_j \cdot q$$

where q is an input text vector representation, while c_j is j -th cluster embedding in the embedding matrix. Obtained scores are then normalized with softmax function, leaving us with one highest weight corresponding to the leading cluster representation for q . Next, each cluster vector is multiplied by the corresponding weight and summed up to get the output reconstructed vector o :

$$o = \sum_{j=1}^K p_j c_j$$

This output reconstructed vector is expected to be similar to the input text vector, so in order to train structure discovery model we minimize the loss function based on cosine distance:

$$L_{asp} = 1 - \frac{q \cdot o}{|q||o|}$$

Such training allows us to build a model which could represent any input vector as a sum of one leading cluster representation and several others. This model is used for filtering of a vector set, it is an ordered set of sentence representations in our case. We filter the set in the following way. Let us say that p_q is a weight for the leading cluster for the input text q . We could filter out any sentence i from the set, where:

$$p_a^i \leq \text{threshold}$$

and *threshold* could be selected arbitrarily.

4. Datasets

CNN/Daily Mail [2] is a dataset commonly used for text summarization evaluation. Human generated abstractive summary bullets were generated from news stories in CNN and Daily Mail websites as questions (with one of the entities hidden), and stories as the corresponding passages from which the system is expected to answer the fill-in-the-blank question. The authors released the scripts that crawl, extract and generate pairs of passages and questions from these websites.

All in all, the corpus has 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs, as defined by their scripts. The source documents in the training set have 766 words spanning 29.74 sentences on an average while the summaries consist of 53 words and 3.72 sentences.

XSum [17] is a dataset for evaluation of abstractive single-document summarization systems. The goal is to create a short, one-sentence new summary answering the question “What is the article about?”. The dataset consists of 226,711 news articles accompanied with a one-sentence summary. The articles are collected from BBC articles (2010 to 2017) and cover a wide variety of domains (e.g., Politics, Sports, Weather, and Technology). The official split contains 204,045 (90%), 11,332 (5%) and 11,334 (5%) documents in training, validation and test sets, respectively.

4.1 Converting to Extractive Dataset

Although the datasets are originally designed for abstractive summarization, we modified them for extractive summarization using a special utility. To obtain the extractive summaries from abstractive ones we use classic concept of extractive oracle. We define the extractive oracle summaries as follows, using ROUGE metrics described below:

$$O = \operatorname{argmax}_{S \subseteq D} \operatorname{ROUGE}_N(G, S), \text{ s.t. } \ell(S) \leq 2\ell(G)$$

Here D is the set of all the sentences contained in the input document, and G is the gold (abstractive) summary for the input document. $\ell(\cdot)$ indicates the number of words in a text.

5. Experiments

5.1 Metrics

The models are evaluated with F1 variant (harmonic mean of Precision and Recall) of ROUGE-1, ROUGE-2, ROUGEL [18]. ROUGE-N is computed as follows:

$$\operatorname{ROUGE}_N = \frac{\sum_{S \in \operatorname{Ref}} \sum_{g_n \in S} \operatorname{Count}_{\operatorname{match}}(g_n)}{\sum_{S \in \operatorname{Ref}} \sum_{g_n \in S} \operatorname{Count}(g_n)}$$

Where n° stands for the length of the n-gram g_n , and $\operatorname{Count}_{\operatorname{match}}(g_n)$ is the maximum number of n-grams cooccurring in a candidate summary and a set of reference summaries Ref° .

- ROUGE-1 value measures the overlap of unigram (each word) between the computed summary and the gold summary.

- ROUGE-2 value measures the overlap of bigrams respectively.
- ROUGE-L measures the longest common subsequence between the model output and gold summary.
- Recall in the context of ROUGE means how much of the gold summary is the computed summary capturing.
- Precision answers how much of the computed summary was in fact relevant.

5.1 Baselines

We compare our model to the following models.

5.1.1 Extractive Models

MatchSum [4]: this approach formulates the extractive summarization task as a semantic text matching problem. A good summary should be more semantically similar to the source document than the unqualified summaries.

DiscoBERT [19]: the model extracts sub-sentential discourse units (instead of sentences) as candidates for extractive selection on a finer granularity. To capture the long-range dependencies among discourse units, structural discourse graphs

BertSumExt [20]: the model uses pretrained BERT with inserted [CLS] tokens at the start of each sentence to collect features for the sentence preceding it.

5.1.2 Abstractive Models

Selfmem [21]: a two-stage framework consisting of a retrieval-augmented generator and memory selector, where the generator's outputs serve as memory candidates for subsequent generation rounds in an iterative process, addressing limitations of fixed-corpus retrieval.

BRIO [22]: a model that modifies the target distribution of abstractive models from a deterministic (one-point) distribution to a non-deterministic distribution where candidate summaries are assigned probability mass according to their quality.

SimCLS [23]: a two-stage model for abstractive summarization, where a Seq2Seq model is first trained to generate candidate summaries with MLE loss, and then a parameterized evaluation model is trained to rank the generated candidates with contrastive learning.

GSum [24]: the model has two encoders which encode the source document and guidance signal, which are attended to by the decoder.

ProphetNet [25]: Transformer-based model which is optimized by n-step ahead prediction that predicts the next n tokens simultaneously based on previous context tokens at each time step.

5.2 Experimental Setup

For the summarization model, a pre-trained BERT was used (*bert-base-uncased* variation from the Transformers library [26]). Input sequence goes as follows:

[CLS]text[SEP]sentence_1[SEP]sentence_2

text is limited or padded to 430 tokens, while *sentence_1* and *sentence_2* are both limited to 39 tokens. *sentence_1* and *sentence_2* are filled with *pos_sentence* or *neg_sentence* randomly to force the model to not rely upon their relative ordering and use an embedded semantics.

During the evaluation, each document is split to sentences the exact same way as during the training. Each sentence is considered to be a candidate for inclusion in summary. It is fed into the model as *pos_sentence*. As *neg_sentence* we use the last sentence in a text, since we assume it is never included into the summary.

The structure discovery model is trained for two epochs, the threshold for filtering was set to 0.25.

6. Results

We compared our model with current state of the art. We denote our model *SumHiS with* and *without filtering* for the variants of the model where the hidden structure discovery model is present or not respectively (see Fig. 3).

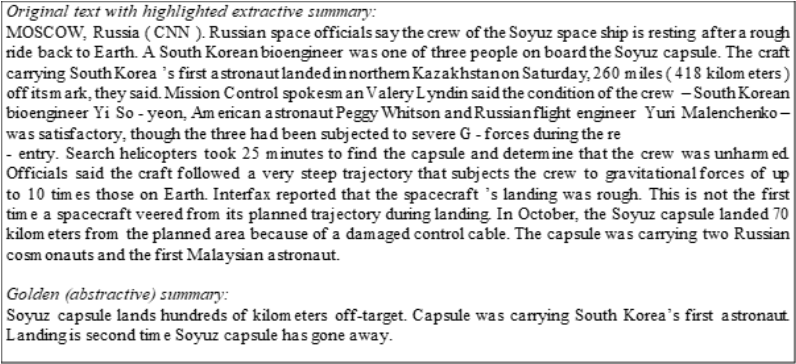


Fig. 3. Sample of SumHiS generated summary.

We evaluate the models on the CNN/DailyMail dataset in non-anonymized version. The evaluation results are presented in Table 1. One could see that our model shows the superior performance among the extractive models by the means of ROUGE-2 and ROUGE-L improving the previous results by almost 12% and 2% respectively. ROUGE-1 evaluation result for our model is 1 percent lower than state of the art result. Thus, could conclude that our model is more successful in extraction of longer sequences of tokens, while keeping the unigrams distribution close to the desired one.

Table 1. ROUGE metrics for the extractive models on CNN/DailyMail test set (non-anonymized). Best result is given in bold, second best – in italic.

Model	ROUGE-1	ROUGE-2	ROUGE-L
MatchSum [4]	44.41	20.86	40.55
DiscoBERT [19]	43.77	20.85	<i>40.67</i>
BertSumExt [20]	43.85	20.34	39.90
SumHiS (w/o filtering)	38.43	<i>28.51</i>	37.58
SumHiS (with filtering)	43.48	32.52	42.44

In addition, we compare our model with abstractive models. The results are presented in Table 2. Despite that our model is not using the generation, i.e., paraphrase ability of the language models, it shows the best results by ROUGE-2 metric outperforming the previous approaches by 10%. ROUGE-L is evaluated only 1 percent lower than state of the art result. This result is an intriguing one, since the extracted bigrams are still better fit the desired distribution than the generated ones. It is important to mention, that structure discovery has significant influence on the model output, leading to improvement by 5% in ROUGE-1 and ROUGE-L and by 4% in ROUGE-2. We also provide a sample of SumHiS output in comparison to golden summary in Table 3.

7. Analysis

The threshold in the experiments was not chosen randomly. We conducted a series of experiments resulting receiver output characteristic for the filtering classifier showed at Fig. 4). The vertical axis is true positive rate, while horizontal one is false positive rate. The value of 0.25 shows the best balance between them.

Table 2. ROUGE metrics for the abstractive & our models on CNN/DailyMail test set (non-anonymized). Best result is given in bold, second best – in italic.

Model	ROUGE-1	ROUGE-2	ROUGE-L
BRIO [22]	47.78	23.55	44.57
SimCLS [23]	46.67	22.15	43.54
GSum [24]	45.94	22.32	42.48
Ours (w/o filtering)	38.43	28.51	37.58
Ours (with filtering)	43.48	32.52	42.44

Table 3. ROUGE metrics for the XSum test set. Best result is given in bold, second best – in italic.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Selfmem [21]	50.3	26.7	41.6
BRIO [22]	49.07	25.59	40.40
SimCLS [23]	47.61	24.57	39.44
GSum [24]	45.40	21.89	36.67
MatchSum [4]	24.86	4.66	18.41
BertSumExt [20]	22.86	4.48	17.16
Ours (w/o filtering)	41.94	28.80	39.97
Ours (with filtering)	43.20	30.32	41.46

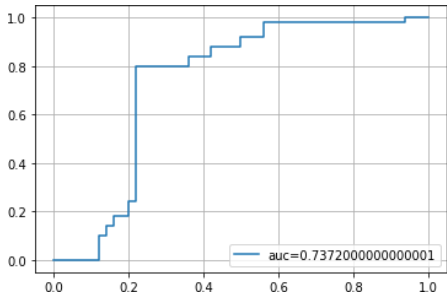


Fig. 4. True Positive Rate vs. False Positive Rate for SumHiS with different threshold values.

7.1 Ablation study

The resulting SumHiS system has several choices which we did basing on the experiment results. SumHiS system contains two models, namely ranking and structure discovery ones. The choices for these models could be questioned thus we provide the results of an ablation study in Table 4. All the results are achieved on CNN/DailyMail dataset. The metrics in this table are the variants of ROUGE, e.g., R-1-p is an abbreviation for ROUGE-1-Precision, R-2-r stands for ROUGE-2-Recall, while R-L-f means ROUGE-L F-measure variant of the metric. All the other metrics are named analogously. We provide more complete results for SumHiS with and without filtering, naming them respectively in the table. We have also tried to interpret the summarization task as binary classification problem, since it is a common approach in the field. In this setup we generate the following triplets: $(t_i, s_{i+}, 1)$ and $(t_i, s_{i-}, 0)$. The last value in a triplet is a label to predict. As a loss function, we use classic binary cross-entropy. The results of this attempt are named “SumHiS + binary loss”. One could see that such replacement of a loss function is leading to catastrophic degradation of SumHiS quality by the means of Precision and F-measure as a consequence.

Table 4. Results for different diode insertion strategies.

Model \ Metric	R-1-p	R-1-r	R-1-f	R-2-p	R-2-r	R-2-f	R-L-p	R-L-r	R-L-f
SumHiS w/o filtering	26.99	79.01	38.43	19.21	65.59	28.51	26.40	77.32	37.58
SumHiS with filtering	31.94	79.45	43.49	22.77	67.02	32.52	31.16	77.62	42.44
SumHiS + aspects	19.24	100.00	31.12	13.67	98.64	23.07	19.31	100.00	30.87
SumHiS + binary loss	17.64	100.00	28.61	12.47	91.88	21.08	17.64	100.00	28.61
Orig. BERT	18.93	100.00	30.69	13.41	98.80	22.67	18.64	100.00	30.33
BERT with filtering	19.12	99.99	31.01	13.59	98.77	23.02	19.08	99.99	30.51
BERT + aspects	18.44	92.12	29.33	12.75	78.03	21.11	18.32	91.53	29.15

We considered the original BERT model without any fine-tuning on our data for the extractive summarization. We used the following setup as for SumHiS, we average per token representations to obtain the input text representation. To make an ordering required to produce a summary compare document text vector representation t' with sentence representation s' . We use dot product as comparison function. The results for this model are denoted as Orig. BERT in the table. Interestingly, Orig. BERT model shows better performance by ROUGE-2 (R-2-f in terms of Table 4), than BERT-based BertSumExt model, although the other metrics are significantly lower for it.

Next, we applied our structure discovery model to the output of the Orig. BERT. The results are denoted as BERT with filtering. The structure discovery has improved all the metrics of Orig. BERT (not including R-1-r and R-L-r, since they were 100%). The achieved result in ROUGE-2 (R-2-f) is a new state of the art, if we are leaving SumHiS aside. Although the improvement is small, it is consistent for all metrics. This result is correlated with filtering usage with SumHiS.

At last, but not least we have experimented with our structure discovery model. It is partially following ABAE setup with two important differences: we do not used initialization for the clusters (aspects) and we do not regularize the cluster matrix. The initialization ABAE use is following: it takes vector representations of all the unique words in the training dataset; apply K -means clustering algorithm to the vectors, where K is set to be the desired number of aspects; and finally averaging all the vectors in a cluster to get its centroid vector. The centroids are used as initial values for the aspect embeddings. The regularization which is used in ABAE is orthonormal one. It is formulated as follows:

$$L_{ortho} = ||C \times C^T - I||$$

where C is an aspect matrix of size $K \times n$ and I is diagonal unit matrix of size $K \times K$. We have applied both of these techniques to our model and Orig. BERT. The results are denoted as “+ aspects” in the table. Surprisingly to us, addition of aspect filtering is lowering all the metrics for both BERT and SumHiS models. Although the aspects extracted with this method seem to be adequate, the quality of the main task of summarization is too low to consider this approach as a general one. A sample of aspects extracted by SumHiS is presented in Table 5.

Table 5. Aspects, extracted by SumHiS.

#	Aspect words
1	girl, women, teenagers, tennis, ladies, princess, kids
2	perhaps, apparently, probably, mysteriously, presumably, suddenly, supposedly
3	care, reasons, aids, purpose, irregularities, conform, attention
4	desperately, supplying, blood, terribly, abducted
5	building, built, forcing, using, saying, trying

7.2 Vector Space Analysis

We aim the model to output different vectors for positive and negative input sentences. To prove it, we calculated distances between $text'$ and $pos\_sentence'$ and $text'$ and $neg\_sentence'$ for every triplet in the test set. As shown in Fig. 5 (left), the distances between initial text representation and negative sentence representations are generally greater than the ones between the initial text and the positive sentence. To take into account the peaking values we performed the kernel trick (see Fig. 5, right): $(x - 0.45)^2$, where x is an initial distance between text and sentence.

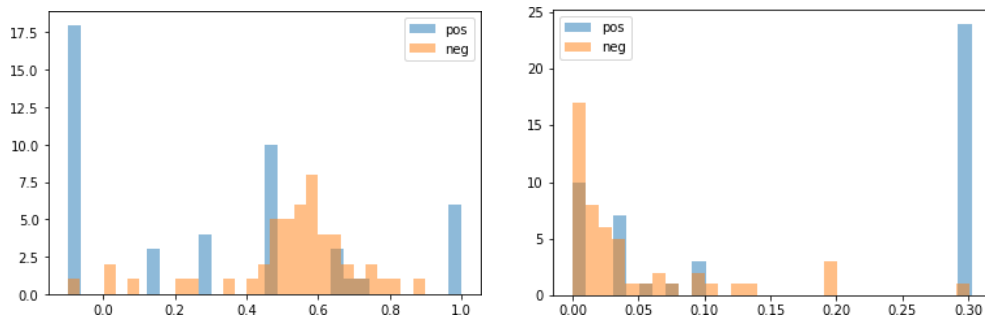


Fig. 5. Histogram of distances between initial text and positive (blue) / negative (orange) sentences. Left – without the kernel trick, right – after performing the kernel trick.

8. Conclusion

We proposed a new model for extractive summarization that uses information about hidden document structure. Our model shows state-of-the-art performance on CNN/DailyMail dataset by ROUGE-2 and ROUGE-L compared to current extractive summarization models. Moreover, it shows the best performance by the means of ROUGE-2 in comparison with abstractive models outperforming them by 10%. We showed that hidden structure in a text could be successfully used leading to significant improvements in summary generation. As for the future work, we plan to make SumHiS end-to-end trainable aggregating ranking model and structure discovery models into an integral pipeline. We are also considering to integrate structure discovery within abstractive summarization, and experiment with different structure discovery mechanisms.

References

- [1]. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
- [2]. Nallapati R., Zhou B., Gulcehre C., Xiang B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. arXiv preprint arXiv:1602.06023, 2016.
- [3]. Lin C.-Y. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, 2004, pp. 74-81.
- [4]. Zhong M., Liu P., Chen Y., Wang D., Qiu X., Huang X.-J. Extractive Summarization as Text Matching. In Proc. 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6197-6208.
- [5]. Dong Y., Romascanu A., Cheung J. C. K. HipoRank: Incorporating hierarchical and positional information into graph-based unsupervised long document extractive summarization. arXiv preprint arXiv:2005.00513, 2020.
- [6]. Boiy E., Moens M.-F. A machine learning approach to sentiment analysis in multilingual Web texts. Information retrieval, vol. 12, no. 5, 2009, pp. 526-558.
- [7]. Kiritchenko S., Zhu X., Cherry C., Mohammad S. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In Proc. 8th International workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 437-442.

- [8]. Wagner J., Arora P., Cortes S., Barman U., Bogdanova D., Foster J., Tounsi L. DCU: Aspect-based polarity classification for SemEval task 4. In Proc. 8th International workshop on Semantic Evaluation (SemEval 2014), 2014.
- [9]. He R., Lee W. S., Ng H. T., Dahlmeier D. Exploiting document knowledge for aspect-level sentiment classification. arXiv preprint arXiv:1806.04346, 2018.
- [10]. Chen P., Sun Z., Bing L., Yang W. Recurrent attention network on memory for aspect sentiment analysis. In Proc. 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 452-461.
- [11]. Liu J., Zhang Y. Attention modeling for targeted sentiment. In Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2017, pp. 572-577.
- [12]. Nallapati R., Zhai F., Zhou B. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In Proc. Thirty-First AAAI Conference on Artificial Intelligence, 2017.
- [13]. Cohan A., Dernoncourt F., Kim D. S., Bui T., Kim S., Chang W., Goharian N. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In Proc. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 615-621.
- [14]. Putra J. W. G., Kobayashi H., Shimizu N. Experiment on Using Topic Sentence for Neural News Headline Generation, 2018.
- [15]. He R., Lee W. S., Ng H. T., Dahlmeier D. An unsupervised neural attention model for aspect extraction. In Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 388-397.
- [16]. Bai Y., Li X., Wang G., Zhang C., Shang L., Xu J., Wang Z., Wang F., Liu Q. SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval. arXiv preprint arXiv:2010.00768, 2020.
- [17]. Narayan S., Cohen S. B., Lapata M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. arXiv preprint arXiv:1808.08745, 2018.
- [18]. Lin C.-Y. ROUGE: A package for automatic evaluation of summaries. In Proc. Text Summarization Branches Out, 2004, pp. 74-81.
- [19]. Xu J., Gan Z., Cheng Y., Liu J. Discourse-Aware Neural Extractive Text Summarization. In Proc. 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5021-5031.
- [20]. Liu Y., Lapata M. Text Summarization with Pretrained Encoders. In Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3730-3740.
- [21]. Cheng X., Luo D., Chen X., Liu L., Zhao D., Yan R. Lift yourself up: Retrieval-augmented text generation with self-memory. *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [22]. Liu Y., Liu P., Radev D., Neubig G. BRIO: Bringing order to abstractive summarization. arXiv preprint arXiv:2203.16804, 2022.
- [23]. Liu Y., Liu P. SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization. arXiv preprint arXiv:2106.01890, 2021.
- [24]. Dou Z.-Y., Liu P., Hayashi H., Jiang Z., Neubig G. GSum: A General Framework for Guided Neural Abstractive Summarization. In Proc. 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 4830-4842.
- [25]. Qi W., Yan Y., Gong Y., Liu D., Duan N., Chen J., Zhang R., Zhou M. ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training. In Proc. 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 2020, pp. 2401-2410.
- [26]. Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., Cistac P., Rault T., Louf R., Funtowicz M., Davison J., Shleifer S., von Platen P., Ma C., Jernite Y., Plu J., Xu C., Le Scao T., Gugger S., Drame M., Lhoest Q., Rush A. M. Transformers: State-of-the-Art Natural Language Processing. In Proc. 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 2020, pp. 38-45.

Информация об авторах / Information about authors

Павел Александрович ТИХОНОВ – магистр компьютерных наук, аспирант Сколтеха, научный сотрудник AIRI. Сфера научных интересов: обработка естественного языка, генеративные модели, интерпретируемость.

Pavel Alexandrovich TIKHONOV – Master Sci. (Comp. Sci.), postgraduate student at Skoltech, researcher at AIRI. Research interests: Natural Language Processing, Generation Models, Interpretability.

Анастасия ЯНИНА – руководитель направления LLM в Wildberries. Сфера научных интересов: языковые модели, эмбединги для обработки текста, оптимизация поисковых систем маркетплейсов.

Anastasia YANINA – Head of LLM Department at Wildberries. Research interests: Language Models, Text Embeddings, Marketplace Search Optimization.

Валентин Андреевич МАЛЫХ – кандидат технических наук, руководитель исследований в области обработки естественного языка в MTS AI. Сфера научных интересов: генерация естественного языка, глубокое обучение.

Valentin Andreevich MALYKH – Cand. Sci. (Tech.), NLP Research Head at MTS AI. Research interests: Natural Language Generation, Deep Learning.



Обзор методов контроля перегрузки с использованием машинного обучения

^{1,2} И.А. Степанов, ORCID: 0009-0003-1964-5001 <ivan_mipt@ispras.ru>

^{1,4} М.В. Попов, ORCID: 0009-0007-7774-2220 <mpopov@ispras.ru>

^{1,2,3,4} А.И. Гетьман, ORCID: 0000-0002-6562-9008 <ever@ispras.ru>

¹ М.К. Иконникова, ORCID: 0000-0003-1530-5133 <mikonnikova@ispras.ru>

^{1,4} А.А. Белеванцев, ORCID: 0000-0003-2817-0397 <abel@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.

² Московский физико-технический институт,
141700, Россия, Московская область, г. Долгопрудный, Институтский пер., 9.

³ Национальный исследовательский университет «Высшая школа экономики»,
101978, Россия, г. Москва, ул. Мясницкая, д. 20.

⁴ Московский государственный университет имени М.В. Ломоносова,
119991, Россия, г. Москва, Ленинские горы, д. 1.

Аннотация. Контроль перегрузки является ключевым аспектом современных сетей. Первые алгоритмы контроля перегрузки, такие как TCP Tahoe и TCP Reno, были разработаны в конце XX века, и их основные идеи остаются актуальными до сих пор. С развитием высокоскоростных сетей для них были созданы специализированные алгоритмы, например, TCP BIC и TCP CUBIC. Однако классические алгоритмы, основанные на определённых правилах, не всегда оказываются эффективными во всех сетевых условиях, и с развитием 4G, 5G и спутниковой связи задача контроля перегрузки стала более актуальной. Это привело к появлению решений этой задачи на основе машинного обучения и обучения с подкреплением, в частности таких, которые способны адаптироваться к динамически изменяющимся условиям сети. В статье представлены и рассмотрены как классические алгоритмы контроля перегрузки, так и наиболее популярные и новые алгоритмы, основанные на машинном обучении, а также некоторые реализации с использованием технологии multipath. Кроме того, выделены наиболее значимые проблемы алгоритмов на основе машинного обучения и обсуждены потенциальные направления будущих исследований в данной области.

Ключевые слова: контроль перегрузки в сети; обучение с подкреплением.

Для цитирования: Степанов И.А., Попов М.В., Гетьман А.И., Иконникова М.К., Белеванцев А.А. Обзор методов контроля перегрузки с использованием машинного обучения. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 251–276. DOI: 10.15514/ISPRAS-2025-37(3)-18.

Machine Learning Dased Congestion Control Methods: a Survey

^{1,2} I.A. Stepanov, ORCID: 0009-0003-1964-5001 <ivan_mipt@ispras.ru>

^{1,4} M.V. Popov, ORCID: 0009-0007-7774-2220 <mpopov@ispras.ru>

^{1,2,3,4} A.I. Getman, ORCID: 0000-0002-6562-9008 <ever@ispras.ru>

¹ M.K. Ikonnikova, ORCID: 0000-0003-1530-5133 <mikonnikova@ispras.ru >

^{1,4} A.A. Belevantsev, ORCID: 0000-0003-2817-0397 <abel@ispras.ru>

¹ *Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

² *Moscow Institute of Physics and Technology (National Research University)
9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.*

³ *National Research University «Higher School of Economics»
20 Myasnitskaya ulitsa, Moscow 101000 Russia.*

⁴ *Lomonosov Moscow State University
1 Leninskie Gory, Moscow, 119991 Russia.*

Abstract. Congestion control is a key aspect of modern networks. The first congestion control algorithms, such as TCP Tahoe and TCP Reno, were developed in the late 20th century, and their core concepts remain relevant to this day. With the development of high-speed networks, specialized algorithms such as TCP BIC and TCP CUBIC were created, which are adapted to these conditions. However, classical algorithms with predefined rules are not always effective in all network environments, and with the rise of 4G, 5G, and satellite communications, the congestion control issue has become increasingly relevant. This has led to the emergence of numerous works on machine learning-based congestion control algorithms, particularly reinforcement learning, which can adapt to dynamically changing network conditions. This paper presents and reviews both classical congestion control algorithms and the most popular and recent machine learning-based algorithms, along with some implementations using multipath. Additionally, it highlights the most significant challenges of machine learning-based algorithms and discusses potential directions for future research in this field.

Keywords: network congestion control; reinforcement learning.

For citation: Stepanov I.A., Popov M.V., Getman A.I., Ikonnikova M.K., Belevantsev A.A. Overview of congestion control methods using machine learning. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 251-276 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-18.

1. Введение

Алгоритмы контроля перегрузки являются необходимым компонентом, обеспечивающим возможность функционирования компьютерных сетей в настоящее время. Перегрузка в сети возникает, когда объём передаваемых данных превышает пропускную способность маршрутов, по которым они передаются, что приводит к невозможности сети эффективно обрабатывать трафик. Это вызывает деградацию качества обслуживания, увеличенные задержки, потерю пакетов и снижение общей производительности сети.

В 1988 году Ван Джейкобсон в своей работе подробно описал проблему сетевой перегрузки, что привело к разработке первого алгоритма контроля перегрузки – TCP Tahoe. Этот алгоритм стал основой для последующих исследований и разработок в области контроля перегрузок, заложив фундамент для новых подходов.

Со временем алгоритмы контроля перегрузки развивались. Изначальные методы (Reno, Cubic) регулировали скорость передачи данных на основе обнаружения потерь. Позже появились подходы, учитывающие задержки в сети (Veno, Vegas), которые стремились предсказать перегрузку на основе изменения времени приема-передачи (round trip time, RTT). Эти алгоритмы позволяли в некоторых случаях более эффективно управлять передачей данных, предотвращая потери ещё до их возникновения. Позднее появились гибридные

алгоритмы (BBR, Сора), комбинирующие различные метрики, такие как пропускная способность и задержка.

С развитием технологий и усложнением сетевых сценариев традиционные алгоритмы управления перегрузкой сталкиваются с серьёзными ограничениями, так как они зачастую разрабатывались для конкретных условий и не обладают достаточной адаптивностью для работы в динамично изменяющихся сетях.

В последнее время появились новые технологии и сети, такие как WiFi, 4G, 5G и спутниковая связь, что привело к большому разнообразию сетевых сценариев. Классические алгоритмы контроля перегрузки не могут эффективно работать во всех существующих сценариях, к примеру, в средах, содержащих как проводные, так и беспроводные соединения. Кроме того, трудности создают среды, динамически меняющие свои характеристики.

Для того, чтобы решить представленные выше проблемы, разрабатываются алгоритмы на основе машинного обучения и, в частности, на основе обучения с подкреплением (RL). Главное отличие алгоритмов контроля перегрузки на основе RL от классических заключается в способности принимать решения на основе данных о сети в реальном времени, что позволяет им лучше адаптироваться к различным сетевым структурам и изменению состояния сети. Однако у алгоритмов, основанных на машинном обучении, есть ряд недостатков, о которых будет сказано позднее.

Особое место в задаче контроля перегрузки занимает технология multipath, позволяющая клиенту (отправителю) посылать и принимать данные одновременно по нескольким сетевым маршрутам. Существуют две основные причины использования данной технологии. Первая заключается в объединении (агрегации) ресурсов нескольких путей для передачи данных по одному логическому соединению. Это важно для передачи больших объемов данных на устройствах, например, смартфонах. Вторая причина заключается в обеспечении устойчивости к неполадкам в сети: при невозможности передать данные по одному из путей данные могут быть переданы по другим. Задача контроля перегрузки в данном случае несколько усложняется в силу наличия нескольких путей, передачу данных по которым нужно эффективно оптимизировать, при этом критерий оптимизации может быть различным. Поведение классических алгоритмов контроля перегрузки в настоящий момент подробно изучено и, кроме того, существуют работы, описывающие область алгоритмов контроля перегрузки с помощью обучения с подкреплением [1, 2]. В данной работе, помимо обзора данных алгоритмов, уделена значительная часть технологии multipath, которая также связана с контролем перегрузки, и детально разобраны возможные недостатки алгоритмов, основанных на машинном обучении.

Далее статья структурирована следующим образом. В разделе II представлена постановка задачи и даны основные определения по теме. В разделе III описаны классические алгоритмы контроля перегрузки без использования машинного обучения и приведено их сравнение. Раздел IV содержит описание популярных методов обучения с подкреплением. В разделе V представлен анализ и описание работ по контролю перегрузки с помощью машинного обучения и приведено их сравнение в рамках рассматриваемой задачи. Раздел VI содержит анализ алгоритмов, реализующих технологию multipath. Раздел VII описывает проблемы и направление будущих исследований алгоритмов, основанных на машинном обучении в задаче контроля перегрузки.

2. Описание задачи

Существует широкое разнообразие формулировок и определений, описывающих одни и те же величины, связанные с контролем перегрузки. Поэтому необходимо ввести чёткие определения и описать формулировку задачи, исходя из них, что и будет сделано в данном разделе.

2.1 Определения

Рассмотрим некоторые базовые определения по нашей теме, необходимые для постановки задачи.

Ширина канала (bandwidth) – предельный объем данных, который может проходить через заданный канал связи.

Задержка (delay) – время, необходимое для передачи данных от отправителя к получателю.

Время приема-передачи (Round trip time, RTT) – время, необходимое для передачи данных от отправителя к получателю и получения подтверждения, что данные были получены.

BDP (Bandwidth-delay product) – произведение ширины канала на время приема-передачи (рис. 1). Этот коэффициент встречается во многих работах по данной теме и является максимальным объемом данных, которые могут находиться в сети в некоторый момент времени.

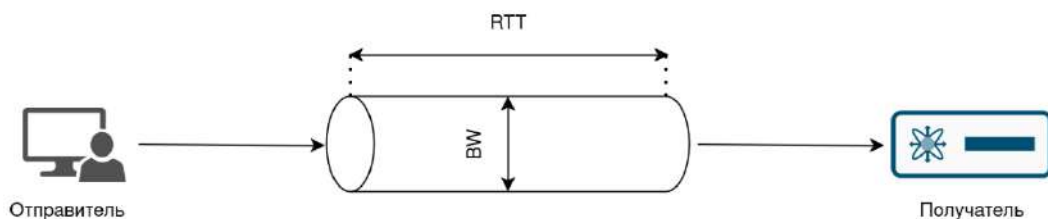


Рис. 1. Коэффициент BDP.
Fig. 1. The BDP coefficient.

Взвешенное скользящее среднее или усредненная круговая задержка (SRTT): $SRTT = \alpha \cdot SRTT + (1 - \alpha) \cdot RTT$.

Отклонение RTT: $RTTVAR = \beta \cdot RTTVAR + (1 - \beta) \cdot |SRTT - RTT|$.

Интервал ожидания: $RTO = SRTT + 4 \cdot RTTVAR$.

V (объем передаваемых данных) – число байт, переданных отправителем получателю за некоторое время.

Пропускная способность (throughput) – отношение объема данных, проходящих через сеть (канал) за заданный интервал времени, к этому интервалу. Пропускная способность является изменяемой величиной и показывает скорость отправки данных, в то время как ширина канала – это величина неизменная и является характеристикой канала.

CWND (окно перегрузки) – число байт, которые могут находиться в сети от каждого отправителя во время работы алгоритма контроля перегрузки.

Bytes in flight (байты в «полёте») – количество отправленных неподтвержденных байт.

Loss (потери) пакетов – число потерянных пакетов за некоторый интервал времени.

Loss_rate (скорость потери пакетов) – отношение числа потерянных пакетов за заданный интервал времени к этому интервалу.

MSS (Maximum segment size) – максимальный размер полезного блока данных в байтах для TCP-пакета.

Capacity (мера свободы канала) – отношение пропускной способности к ширине канала в данный момент времени.

2.2 Постановка задачи контроля перегрузки

Классическая задача контроля перегрузки выглядит следующим образом (рис. 2): существует отправитель и получатель, которые обмениваются данными по некоторому пути, состоящему из разных каналов. Каналы сети имеют некоторые характеристики, главной из которых является ширина канала. Алгоритм контроля перегрузки должен стремиться передавать

данные таким образом, чтобы общий объем данных был равен значению BDP или близко к нему. С другой стороны, алгоритм должен избегать ситуаций, когда данных так много, что задержка становится критичной.

Таким образом, задачу контроля перегрузки можно сформулировать так:

$$V \rightarrow BDP$$

$$\text{delay} \rightarrow \min$$

Однако стоит понимать, что данная задача имеет ряд трудностей: маршрут от отправителя к получателю может содержать большое число путей с разной пропускной способностью, тип связи может быть различен (беспроводная связь, оптоволокно и т.д.), у отправителя может быть несколько каналов передачи данных (multipath) и т.д.

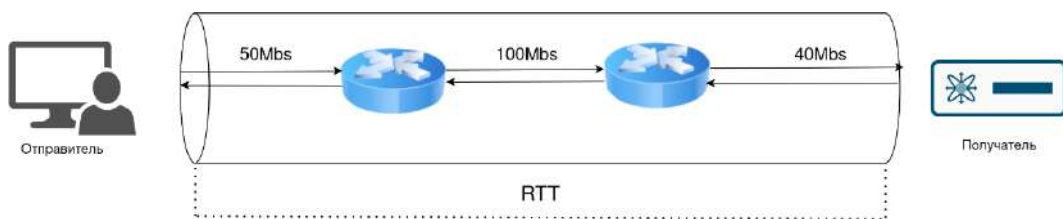


Рис. 2. Постановка задачи контроля перегрузки.
Fig. 2. Setting the task of overload control.

Более детально, различные алгоритмы контроля перегрузки решают схожие задачи, которые можно сформулировать следующим образом:

Максимизация пропускной способности (throughput). Высокая пропускная способность означает высокую степень использования канала, то есть высокую эффективность передачи данных. Алгоритмы контроля перегрузки направлены на решение данной задачи. Однако высокая пропускная способность может привести к созданию длинных очередей, что может значительно увеличить задержку и время приема-передачи (RTT).

Минимизация времени приема-передачи (RTT) и задержки. Алгоритмы контроля перегрузки стараются уменьшить время приема-передачи, чтобы повысить эффективность передачи данных. Зачастую задача минимизации этого времени является взаимоисключающей к задаче максимизации пропускной способности, что приводит к поиску оптимального значения между решениями двух данных задач.

Минимизация скорости потерь (loss_rate). Высокая скорость потерь пакетов приводит к нестабильной работе сети и, соответственно, к уменьшению скорости передачи данных и увеличению времени RTT. Поэтому алгоритмы контроля перегрузки должны уменьшать данную скорость потерь.

Быстрота реагирования на перегрузку. Высокая скорость реагирования позволяет оперативно решить проблему перегрузки, не уменьшая эффективность передачи данных. Однако стоит понимать, что высокая скорость реагирования достигается путём частой смены значений *cwnd*, что может быть ресурсозатратно. Именно поэтому алгоритмы контроля перегрузки стараются соблюдать баланс, выбирая оптимальную частоту обновления размеров окна *cwnd*.

Справедливость. Высокая справедливость требует, чтобы алгоритмы контроля перегрузки одинаково распределяли ресурсы между различными получателями и отправителями, использующими одну и ту же часть пути для передачи данных.

Основной способ управлять перегрузкой – это увеличение или уменьшение размера окна *cwnd*, в то время как перегрузку можно определять разными способами. Классические алгоритмы контроля перегрузки делятся на три группы: основанные на потерях, основанные на задержке и гибридные.

Алгоритмы, **основанные на потерях**, определяют факт перегрузки после её возникновения, что в некоторых случаях может быть недостатком. При этом индикатором перегрузки является истечение интервала ожидания RTO (повторной передачи, retransmission) до получения подтверждения или получение трех повторяющихся подтверждений. Стоит заметить, что в случае истёкшего интервала поведение алгоритма будет отличаться от случая трёх подтверждений.

Алгоритмы, **основанные на задержках**, определяют перегрузку до её возникновения, полагаясь на время приема-передачи RTT. Однако измерить это время в сложных сетях бывает проблематично, что может влиять на корректность работы алгоритма. Во-вторых, данный класс алгоритмов в силу своей природы использует доступную ширину канала не столь эффективно, как алгоритмы, основанные на потерях.

Гибридные алгоритмы стараются достичь баланса между двумя классами, представленными ранее. Их основная идея заключается в следующем: значительное увеличение скорости передачи данных в том случае, если полоса пропускания загружена незначительно и более мягкое поведение в ситуациях, когда канал почти заполнен. К главным недостаткам данного класса стоит отнести плохую совместимость с другими алгоритмами контроля перегрузки, о чём будет сказано в дальнейшем.

3. Классические алгоритмы контроля перегрузки

В этом разделе рассмотрены наиболее популярные алгоритмы контроля перегрузки без использования машинного обучения. В конце раздела представлена сравнительная таблица описанных алгоритмов.

3.1 Алгоритмы, основанные на потерях

TCP Tahoe (1988) [3] был первым алгоритмом контроля перегрузки в сети. Tahoe использует два ключевых механизма для управления перегрузкой: медленный старт и предотвращение перегрузки. Медленный старт начинает свою работу с минимальным значением размера окна. Значение cwnd увеличивается экспоненциально при подтверждении сегмента, то есть через каждый интервал приема-передачи RTT. Этот процесс продолжается до тех пор, пока размер окна cwnd не достигнет порогового значения (sssthresh), после чего начинается фаза предотвращения перегрузки, или пока не произойдет событие потери, то есть истечение интервала ожидания RTO, после которого пороговое значение sssthresh устанавливается равным половине текущего cwnd, размер окна принимается равным минимальному значению, и медленный старт начинается заново.

Предотвращение перегрузки линейно увеличивает размер cwnd за один интервал RTT. При обнаружении потери пакетов Tahoe уменьшает sssthresh в половину от текущего значения, резко снижает cwnd до минимального значения и вновь начинает процесс с фазы медленного старта. Этот механизм интегрирует принципы подхода AIMD, обеспечивая аддитивное увеличение (additive increase) размера окна до момента обнаружения перегрузки, после чего происходит мультипликативное уменьшение (multiplicative decrease).

Основной недостаток данного алгоритма заключается в том, что после обнаружения потери пакета Tahoe сбрасывает размер окна cwnd до минимального значения и начинает процесс увеличения заново, что приводит к снижению общей производительности сети.

TCP Reno (1990) [4] усовершенствует процесс обнаружения потерь, вводя механизмы быстрой повторной передачи и быстрого восстановления. Вместо того, чтобы инициировать повторную передачу потерянного сегмента после истечения интервала ожидания, как это делает Tahoe, Reno запускает повторную передачу быстрее, реагируя на получение трех дубликатных подтверждений (ACK) для одного и того же сегмента данных. Это позволяет быстрее восстанавливаться после потерь, не дожидаясь истечения таймера повторной

передачи. После того, как отправитель получает три дубликатных ACK для одного сегмента, TCP Reno входит в фазу быстрого восстановления.

В этой фазе Reno уменьшает порог $ssthresh$ вдвое от текущего размера окна перегрузки и устанавливает окно $cwnd$ равным $ssthresh + 3 \cdot MSS$. После уменьшения размера окна перегрузки Reno продолжает отправку данных, используя новый размер окна. За каждый последующий дубликатный ACK, полученный в этой фазе, значение $cwnd$ увеличивается на размер одного блока MSS, позволяя отправлять дополнительный сегмент данных за каждое подтверждение, которое предположительно указывает на то, что другой сегмент был успешно доставлен.

Фаза быстрого восстановления завершается, когда получено подтверждение о доставке нового сегмента данных, и алгоритм переходит в фазу предотвращения перегрузки – или когда происходит тайм-аут, и алгоритм переходит в фазу медленного старта.

TCP NewReno (1999) [5] вводит улучшение по сравнению с алгоритмом Reno, заключающееся в обработке частичных подтверждений. В отличие от Reno, NewReno остается в фазе быстрого восстановления, пока не получит подтверждения всех оставшихся пакетов. Если приходит частичный ACK, который подтверждает только часть пакетов, алгоритм рассматривает это как признак потери и инициирует повторную передачу. Это позволяет NewReno эффективно обрабатывать множественные потери без необходимости выхода из фазы быстрого восстановления, а также избежать многократного уменьшения $cwnd$. Поэтому по сравнению с Reno алгоритм NewReno более эффективен при обработке множественных потерь пакетов.

Scalable TCP (2003) [6] базируется на подходе MIMD (Multiplicative Increase, Multiplicative Decrease). При обнаружении потери пакетов Scalable TCP уменьшает размер окна перегрузки мультипликативно, то есть на определенный процент от текущего размера окна. Scalable TCP гораздо более эффективен, чем Reno, в сетях с высокой задержкой. Быстрее и агрессивней восстанавливается после потери пакета. Может проявлять несправедливость по отношению к другим потокам данных, использующим стандартный TCP, в смешанных сетевых средах. Его агрессивное увеличение окна перегрузки может привести к тому, что потоки Scalable TCP будут занимать большую часть доступной пропускной способности за счет других потоков.

Highspeed TCP (2003) [7] – это вариант протокола TCP, разработанный для улучшения производительности в сетях с высоким BDP и с высокой задержкой, таких как дальнемагистральные и широкополосные сети. В HSTCP изменение размера окна при наличии или отсутствии перегрузки сети зависит от определяющих размер изменения функций $a(cwnd)$ и $b(cwnd)$, которые в свою очередь зависят от текущего размера окна. Когда перегрузка не обнаружена, размер окна $cwnd$ увеличивается через каждое RTT на величину $a(cwnd)/cwnd$. При обнаружении перегрузки размер окна $cwnd$ уменьшается на $b(cwnd) \cdot cwnd$. В сетях с высоким BDP HSTCP ведет себя эффективно и агрессивно, однако может привести к несправедливому распределению сетевых ресурсов между потоками.

TCP BIC (2004) [8] решает проблему несправедливости, характерную для HSTCP. Помимо медленного старта и аддитивного увеличения окна, BIC использует механизм бинарного поиска. Бинарный поиск позволяет найти максимальный размер $cwnd$, при котором сеть может работать без перегрузок. BIC TCP быстро увеличивает размер окна до тех пор, пока не достигнет порога. После достижения порога алгоритм переключается на бинарный поиск, чтобы найти новое оптимальное значение окна перегрузки. Если потери не обнаруживаются, BIC продолжает увеличивать размер окна, сужая диапазон поиска между текущим минимальным размером окна (последним известным стабильным значением) и размером окна перед последней потерей. Алгоритм устанавливает $cwnd$ как среднее между данными значениями и проверяет наличие потерь, постепенно уменьшая разницу между ними до тех пор, пока не будет найдено оптимальное значение. Это помогает более плавно адаптироваться к состояниям перегрузки. Таким образом, алгоритм может эффективно

работать в высокоскоростных сетях с большой задержкой. При возникновении перегрузки в сети BIC использует мультипликативное уменьшение для коррекции cwnd.

CUBIC TCP (2008) [9] управляет окном перегрузки, заменяя сложные алгоритмы BIC на кубическую функцию. При каждом подтверждении (ACK) окно изменяется следующим образом:

$$W(t) = W_{max} + C(t - K)^3$$

$$K = \sqrt[3]{\frac{W_{max} \cdot \beta}{C}}$$

- ⑩ t – время с момента последней потери,
- ⑩ C – параметр, определяющий скорость увеличения окна,
- ⑩ β – начальная константа,
- ⑩ W_{max} – размер окна перед последней потерей,
- ⑩ W – текущий размер окна перегрузки.

При обнаружении потери пакета:

$$W_{max} = W$$
$$W = W \cdot (1 - \beta)$$

Кубическая функция позволяет резко увеличивать размер окна сразу после потери и после длительного времени работы без потерь. В то же время окно перегрузки меняется незначительно в состоянии, близком к предыдущей потере, что обеспечивает более стабильную работу алгоритма в сетях с высокой шириной канала.

CUBIC на данный момент является одним из самых широко используемых алгоритмов управления перегрузкой.

3.2 Алгоритмы, основанные на задержке

Как уже говорилось ранее, данный класс кардинально отличается от алгоритмов, основанных на потерях с точки зрения определения перегрузки.

TCP Vegas (1994) [10] основан на использовании информации о RTT. Данный алгоритм постоянно измеряет RTT для текущего соединения. Vegas вычисляет ожидаемую пропускную способность как отношение текущего размера окна к базовому RTT (BASERTT), который является минимальным измеренным RTT для соединения, и предполагает, что это отношение будет поддерживаться, если в сети нет перегрузки.

Фактическая пропускная способность вычисляется как отношение размера окна к текущему RTT. Если разница между ожидаемой и фактической пропускной способностью (Differ) меньше заданного порога, то Vegas увеличивает размер окна на один MSS. Если разница больше некоторого другого порога, то Vegas уменьшает размер окна. Уменьшение размера окна в Vegas происходит более мягко, чем мультипликативное уменьшение в других алгоритмах.

Основываясь на идеях Vegas, исследователи разработали ряд его модификаций [11-12], которые в основном отличались способами измерения и предсказания RTT для прогнозирования перегрузки. Однако данный алгоритм не стал популярным в высокоскоростных сетях, так как зачастую использует доступную полосу пропускания не полностью.

TCP Veno (2003) [13] комбинирует методы Reno и Vegas, используя элементы контроля перегрузки Reno и механизм оценки состояния сети на основе задержек из Vegas. В любой момент времени, если

$BASERTT \cdot Differ \geq \beta$ – перегрузка;

$BASERTT \cdot Differ < \beta$ – перегрузки нет.

Если обнаружена потеря пакета, когда соединение находится в состоянии перегрузки, то TCP Veno использует стандартную схему AIMD TCP Reno для уменьшения $cwnd$ в режиме предотвращения перегрузки. В отличие от Reno, при отсутствии перегрузки Veno уменьшает окно $cwnd$ на 20% вместо стандартных 50%.

3.3 Гибридные алгоритмы

TCP Westwood (2001) [14] был разработан для улучшения производительности передачи данных в сетях с высокой пропускной способностью и высокими скоростями потерь. Алгоритм постоянно оценивает пропускную способность и измеряет значения RTT при отсутствии нагрузки.

Эти оценки затем используются для адаптации размера окна перегрузки $cwnd$ и порога медленного старта $ssthresh$ после обнаружения потери пакетов, что позволяет более точно и гибко реагировать на изменения в сетевых условиях.

Вместо фиксированного сброса, как в TCP Reno, Westwood минимизирует снижение производительности, пропорционально снижая размер окна с учетом текущей оценки пропускной способности и RTT. В случае отсутствия потерь Westwood увеличивает окно перегрузки аналогично традиционным алгоритмам TCP через фазы медленного старта и предотвращения перегрузки.

BBR (2016) [15] использует модель, основанную на оценке ширины канала $BtlBw$ и минимальной задержке RTT_{prop} для определения оптимального размера окна. Этот размер окна соответствует идеальному объему данных, который может быть передан по сети без перегрузки и без необходимости дожидаться потерь пакетов для корректировки скорости передачи. Поэтому BBR стремится поддерживать окно передачи на уровне оцененного BDP, используя канал максимально эффективно и с минимальной задержкой, не полагаясь на обнаружение потерь пакетов.

Алгоритм BBR регулярно измеряет сеть, чтобы адаптировать скорость передачи и выявлять изменения в пропускной способности или задержке. Для этого BBR использует механизм *gain cycling*, который чередует фазы изменения коэффициента передачи, позволяя эффективно использовать ширину канала.

Однако BBR может не обеспечивать идеальное справедливое распределение пропускной способности с алгоритмами контроля перегрузки, основанными на потерях. Данная проблема отчасти была решена в более поздних версиях BBR [16, 17].

Сора (2018) [18] превосходит BBRv1 по способности поддерживать высокую пропускную способность с более низкой задержкой. Целевая скорость передачи данных определяется как $\frac{1}{(\delta \cdot d_q)}$, где d_q – измеренная задержка в очереди, а δ – параметр, который регулирует баланс между задержкой и пропускной способностью: чем выше δ , тем выше акцент на снижении задержки, а при меньших значениях – на увеличении пропускной способности. Это позволяет гибко адаптировать скорость передачи в зависимости от условий сети. Алгоритм Сора динамически адаптирует $cwnd$: увеличивает его при низкой скорости и уменьшает при превышении целевой. При конкуренции с другими потоками алгоритм изменяет δ по принципу AIMD для поддержания пропускной способности и стабильности соединения.

Краткое описание представленных выше алгоритмов дано в табл. 1.

Табл.1. Классические алгоритмы контроля перегрузки.
Table 1. Classical congestion control algorithms.

№	Год	Название	Тип алгоритма	Особенности
3	1988	TCP Tahoe	На потерях	Первый алгоритм контроля перегрузки. Использует медленный старт и предотвращение перегрузки.
4	1990	TCP Reno	На потерях	Объединяет идеи TCP Tahoe и механизмов быстрой повторной передачи и быстрого восстановления.
5	1999	TCP New Reno	На потерях	Улучшение TCP Reno с обработкой частичных подтверждений.
6	2003	Scalable TCP	На потерях	Увеличивает cwnd относительно текущего размера окна.
7	2003	Highspeed TCP	На потерях	Предназначен для сетей с высоким BDP, агрессивно увеличивает окно.
8	2004	TCP BIC	На потерях	Поиск оптимального размера окна с помощью бинарного поиска.
9	2008	TCP CUBIC	На потерях	Использует кубическую функцию для изменения окна.
10	1994	TCP Vegas	На задержках	Определяет перегрузку, основываясь на измерении разницы RTT.
13	2003	TCP VenO	На задержках	Предсказывает перегрузку, используя идеи Vegas. Изменяет размеры окна, используя идеи Reno.
14	2001	TCP Westwood	Гибридный	Адаптирует окно в зависимости от оценок RTT и пропускной способности.
15	2016	BBR	Гибридный	Стремится использовать максимально доступную пропускную способность узких мест без перегрузок.
18	2018	Cora	Гибридный	Изменяет cwnd в соответствии с целевой скоростью, с параметром δ для регулировки приоритетов задержки и пропускной способности.

4. Обучение с подкреплением

Алгоритмы обучения с подкреплением можно разделить на два класса: на основе ценностей и на основе политик. В контексте контроля перегрузки наиболее рассматриваемыми алгоритмами являются:

- Q-обучение (на основе ценностей);
- глубокое Q-обучение (на основе ценностей);
- Actor-Critic (на основе ценностей и политики);
- Proximal Policy Optimization (PPO) (на основе ценностей и политики).

Основная схема работы обучения с подкреплением представлена на рис. 3. Агент выбирает некоторое действие a_t из своей стратегии. Среда реагирует на это и выдаёт следующие состояние и награду s_{t+1}, r_{t+1} . Основываясь на полученных результатах, агент корректирует свою стратегию. В задаче контроля перегрузки состояниями обычно являются: скользящее среднее (SRTT), число подтверждённых пакетов и т.д. Наградой выступают величины, которые мы хотим оптимизировать (пропускная способность, RTT и т.д). Действиями – уменьшение или увеличение окна перегрузки. Выбор состояний, действий и функции награды в данной задаче будет представлен в следующих разделах.

4.1 Q-обучение

Q-обучение (Q-learning) – это один из методов обучения с подкреплением [19], который обучает агента с помощью функции ценности действия (Q-функции). Формально, Q-функция

– ценность действия s в состоянии a – определяет ожидаемое вознаграждение следующим образом:

$$Q(s_t, a_t) = E[R_t \vee s_t = s, a_t = a]$$

- R_t – общее вознаграждение, полученное от времени t до конца эпизода;
- a_t – действие, выбранное в состоянии;
- E – математическое ожидание.

Для каждого t выбирается действие, среда генерирует награду и следующее состояние. Затем происходит обновление Q-функции:

$$Q(s_t, a_t) \rightarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t))$$

- $Q(s_t, a_t)$ – текущее значение Q-функции для заданного состояния и действия;
- r_t – вознаграждение, полученное после выполнения действия a_t ;
- α – скорость обучения (learning rate);
- γ – коэффициент дисконтирования, который определяет влияние будущих вознаграждений.

Одним из преимуществ данного метода является отсутствие необходимости предварительного знания о динамике среды, что может быть очень полезно в случае контроля перегрузки в сети с динамическими параметрами.

Однако, с другой стороны, Q-обучение может быть чувствительным к шуму и нестабильным параметрам в динамических средах, где вознаграждения могут существенно изменяться от одного эпизода к другому, что может приводить к низкой эффективности.

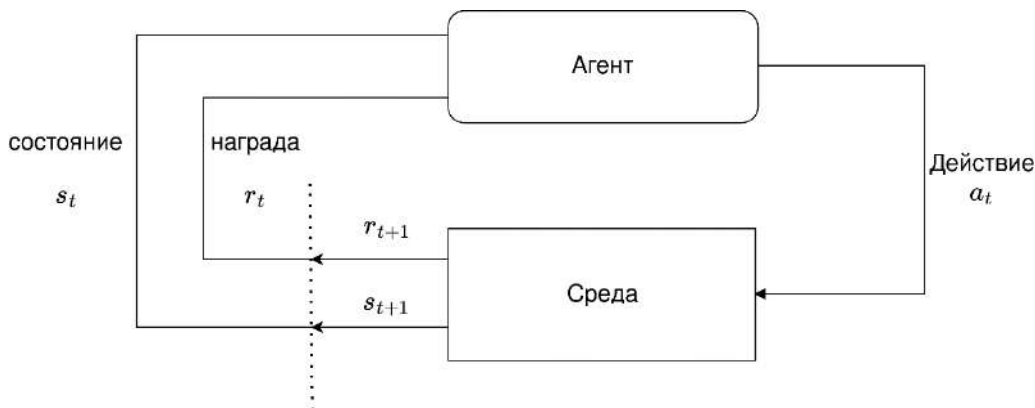


Рис. 3. Обучение с подкреплением.

Fig. 3. Reinforcement learning.

4.2 Глубокое Q-обучение

Для проведения глубокого Q-обучения (DQL) Q-функция представляется в виде нейронной сети [20]. В процессе работы алгоритма происходит обучение нейронной сети на основе истории о состояниях, действиях и награды. Как правило, награда зависит от цели алгоритма: достижения высокой пропускной способности и уменьшения RTT. В контексте задачи контроля перегрузки входом нейронной сети являются состояния, а выходом – действия (изменения размера окна $cwnd$).

DQL способен работать с высокоразмерными пространствами, то есть с объектами, содержащими большое число признаков, что может быть полезно в случае контроля перегрузки при использовании в качестве признаков последовательностей RTT, $cwnd$ и др.

Другим преимуществом является высокая обобщающая способность нейронных сетей, которая позволяет улучшить обучение в новых, ранее незнакомых состояниях, то есть сетевых средах.

Однако стоит понимать, что использование нейронных сетей требует значительно большего объема вычислений и памяти, что может быть критично в том случае, если алгоритм контроля перегрузки должен с высокой частотой выдавать решение об изменении размера окна *cwnd*. Кроме того, для надежного обучения требуется большое количество собранных данных, что может увеличить время обучения и привести к некорректной работе алгоритма, особенно в начале.

4.3 Метод Actor-Critic

Метод Actor-Critic (AC) сочетает в себе идеи как из алгоритмов на основе ценностей, так и из алгоритмов на основе политик [21]. Актор отвечает за выбор действий. Он представляет собой политику, которая определяет, какое действие выполнять в зависимости от текущего состояния. Политика может быть детерминированной или стохастической. Актор обновляет свои параметры, используя информацию о вознаграждении, полученную от критика.

Критик, в свою очередь, оценивает действие, выбранное Актором, и предоставляет обратную связь в виде оценки (обычно функции ценности Q). Основной задачей критика является регрессия к целевому значению, что позволяет Актору получать качественную оценку своей текущей политики.

Помимо классического алгоритма AC существуют Advantage AC (A2C) [22] и Asynchronous AC (A3C). Данные методы основаны на функции преимущества:

$$A(s_t, a_t) = r_t + \gamma V(s_{t+1}, \theta_v) - V(s_t, \theta_v)$$

- V – функция ценности;
- r_t – вознаграждение, полученное после выполнения действия a_t ;
- θ_v – параметры политики;
- γ – коэффициент дисконтирования, который определяет влияние будущих вознаграждений.

A3C использует несколько копий одного и того же агента для одновременного и независимого изучения среды. Поэтому такой метод позволяет быстрее исследовать среду и следовательно, даёт возможность быстрее сходиться к оптимальному значению, что очень важно в рамках задачи контроля перегрузки.

4.4 Proximal Policy Optimization

Proximal Policy Optimization (PPO) является методом, основанным на градиентах политик [23], и использует подход актор-критик, обладая при этом некоторыми усовершенствованиями и ограничивая диапазон целевой функции. В данном случае задача выглядит следующим образом:

$$L^{clip} = E \min(r(\theta) \cdot A^\pi(s, a), \text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon) \cdot A^\pi(s, a))$$

- clip – функция, обрезающая значения, ограничивая его в заданных границах;
- ϵ – некоторое положительное, близкое к нулю число.

Данный метод может быть эффективен в силу своей устойчивости к средам, выходящим за рамки его обучения, что крайне полезно в случае контроля перегрузки, когда характеристики сетевой среды могут постоянно меняться.

5. Машинное обучение для контроля перегрузки

Большинство работ, посвящённых контролю перегрузки на основе машинного обучения, можно разделить на две группы: алгоритмы контроля перегрузки с помощью обучения с учителем и алгоритмы контроля перегрузки с помощью обучения с подкреплением.

5.1 Обучение с учителем

Обучение с учителем предполагает собранный набор данных, на котором и учится алгоритм машинного обучения. Существует две популярные задачи контроля перегрузки, которые могут быть решены с помощью обучения с учителем: классификация типа потерь и прогнозирование задержки или RTT.

Алгоритмы контроля перегрузки, основанные на потерях, указывают лишь на факт перегрузки, но ничего не говорят о её типе, что бывает полезно в некоторых случаях. Поэтому имеет смысл использовать алгоритмы машинного обучения для определения типа перегрузки. Данная задача, с точки зрения машинного обучения с учителем, является задачей классификации на n классов, где признаками могут являться некоторые характеристики сети, а классом – тип перегрузки.

Так, в [24] авторы, используя в качестве признаков одностороннюю задержку и время между передачами пакетов, классифицируют потери на два класса (потеря связи или перегрузка) в сценарии беспроводных сетей. В качестве алгоритма классификации используется градиентный бустинг решающих деревьев – алгоритм машинного обучения, последовательно строящий ансамбль деревьев, каждое из которых корректирует ошибки предыдущих. В качестве признаков – задержки пакета до потери и три задержки трех пакетов после потери. Обучающий набор классификатора представлял собой набор данных из 25 000 событий потерь, а встроенный классификатор использовался авторами в TCP NewReno.

Более полное исследование с точки зрения числа алгоритмов машинного обучения было представлено в [25]. Сценарий был тем же – сценарий беспроводных сетей; классами являлись потеря связи или перегрузка сети. В качестве признаков выступали различные статистические значения интервалов между пакетами, а в качестве алгоритмов – модели, основанные на решающих деревьях, и полносвязные нейронные сети. В работе авторы показали, что классическое решающее дерево даёт оптимальный результат при небольших вычислительных затратах.

В [26] авторы используют скрытые марковские модели (НММ) для классификации потерь на два класса (потеря связи или перегрузка) в случае проводных сетей. В качестве признаков использовались количества подтверждённых пакетов между двумя потерями.

Как уже отмечалось ранее, алгоритмы, основанные на задержках, прогнозируют перегрузку, оценивая время RTT. Именно поэтому в некоторых случаях может быть полезным прогноз времени приёма-передачи на основе набора признаков. Кроме того, алгоритмы, основанные на потерях, косвенно используют время RTT, так как сигналом перегрузки служит истечение таймера RTO. Следовательно, задача прогнозирования RTT в ряде случаев является актуальной. С точки зрения машинного обучения – это задача регрессии, где предсказываемой величиной является следующие значения RTT.

Так, в [27] авторы используют ансамбль для предсказания времени RTT в сценарии сетей, чувствительных к задержкам. Авторы подтверждают полученные результаты экспериментами в реальном времени, реализовав предложенный алгоритм в ядре Linux. Данные эксперименты показывают более высокую точность оценки RTT с помощью машинного обучения: по сравнению со стандартным протоколом TCP улучшение более чем на 40%. В качестве признаков используется линейная комбинация максимальных и минимальных предыдущих значений RTT.

С внедрением видеоприложений реального времени и беспроводных сетей появилась актуальность контроля перегрузки в этом сценарии. Одной из первых работ в данной области был алгоритм [28], основанный на теореме Байеса.

Однако классические алгоритмы машинного обучения могут быть не столь эффективны в случае предсказания ряда RTT, в отличие от рекуррентных нейронных сетей (RNN). В [29] авторы используют последовательность из 6 интервалов между пакетами для предсказания нейронной сетью 4 следующих значений RTT в сценарии проводной сети. В качестве оценки работы используется метрика SNR, и её значение на тестовом наборе данных равно 0.13.

Чуть позже появились исследования с большим числом объектов в наборе данных. Так, в [30] авторы в качестве набора данных использовали 2160 объектов. Признаками выступали средние значения RTT за минуту, и на их основе рекуррентных нейронная сеть предсказывала от одного до 30 значений следующих RTT. MSE при этом составила на тестовом наборе данных от 0.004 до 0.01 в зависимости от числа предсказанных значений RTT.

Однако средние значения RTT за длительный интервал могут быть не информативны. В [31] авторы, используя каждое RTT, с помощью k-NN выделяли ряд признаков (10-15), которые затем подавались на вход рекуррентной нейронной сети. Удалось предсказать большое число следующих значений RTT: от 320 до 960. Размер набора данных составлял 10000, и метрика NRMSE на тестовом наборе данных составила 0.26-0.32 в зависимости от числа предсказанных значений RTT.

Предсказание RTT в мобильной сети (4G) представлено в [32]. В силу особенности сценария авторами выбраны в качестве признаков не только предыдущие значения RTT, но и уровень принимаемого сигнала, густонаселенность района сигнала и день недели, так как это может влиять на нагрузку сети. Кроме того, данная работа характерна предсказанием не конкретного значения RTT, а лишь его серьёзного изменения, что на практике может быть более полезно для контроля перегрузки.

Краткий обзор алгоритмов обучения с учителем для предсказания типа перегрузки и RTT представлен в табл. 2.

5.2 Обучение с подкреплением

С точки зрения контроля перегрузки алгоритмы обучения с подкреплением основаны на следующей идее: агент использует состояния, наблюдаемые из окружающей сетевой среды, и на основе этих состояний принимает решения об увеличении скорости отправки или размера окна перегрузки.

QTCP [33] был одной из первых работ по контролю перегрузки с помощью обучения с подкреплением. В качестве метода обучения выступало Q-обучение, функция награды зависела от пропускной способности и RTT. Состоянием была следующая тройка: средний интервал между отправкой двух пакетов, средний интервал между получением двух последовательных подтверждений, среднее RTT. На основе состояний и функций награды представленный алгоритм совершал всего три действия: увеличивал окно на 10 байт, уменьшал окно на один байт, оставлял окно без изменений. Авторы сравнивали представленный алгоритм лишь с NewReno, но по отношению к нему получили значительное улучшение, как с точки зрения скорости передачи, так и с точки зрения минимизации RTT.

В **Aurora [34]** функция награды зависела от пропускной способности, задержки, RTT и числа потерь. Агент DRL обучался с помощью Proximal Policy Optimization (PPO). Состоянием была скорость передачи данных и доля потерь, а действиями – уменьшение или увеличение скорости отправки. Авторы изучали работу алгоритма в сети с каналами с шириной от 1,2 до 6 Мб/с и задержками от 50 до 500 миллисекунд. Предложенный алгоритм показал высокую эффективность в такой сетевой среде по сравнению с TCP CUBIC, но возникает вопрос работоспособности алгоритма в средах с высокой пропускной способностью.

Табл. 2. Сравнение различных алгоритмов обучения с учителем.

Table 2. Comparing different learning algorithms with a teacher.

№	Год	Сценарий	Признаки	Модель	Задача
24	2005	Беспроводная сеть	Задержки	Градиентный бустинг	Потеря связи или перегрузка (классификация)
25	2004	Беспроводная сеть	Интервалы между пакетами	Решающие дерево	Потеря связи или перегрузка (классификация)
26	2008	Проводная сеть	Подтверждённые пакеты между потерями	НММ	Потеря пакетов или перегрузка (классификация)
27	2014	Сети, чувствительные к задержкам	Время приема-передачи RTT	Ансамбль алгоритмов	Предсказание времени RTT для контроля перегрузки (регрессия)
28	2018	Беспроводная сеть (видео)	Задержки	теорема Байеса	Предсказание времени RTT для контроля перегрузки (регрессия)
29	2002	Проводная сеть	RTT	RNN	Предсказание следующего значения RTT (регрессия)
30	2007	Проводная сеть	RTT	RNN	Предсказание следующих значений RTT (регрессия)
31	2009	Проводная сеть	Разложения RTT	RNN	Предсказание следующих значений RTT (регрессия)
32	2021	Мобильная сеть (4G)	RTT и косвенные характеристики сети	Градиентный бустинг	Предсказание увеличения RTT (классификация)

R3Net [35] ориентирован на потоковое видео и приложения для видеоконференцсвязи в реальном времени. Функция награды зависела от пропускной способности, задержки, RTT и числа потерь. Агент обучается с помощью Proximal Policy Optimization (PPO). В состояние входили скорость передачи, средний интервал между пакетами, потери пакетов и среднее значение RTT, действиями были уменьшение или увеличение скорости отправки. Авторы исследовали работу алгоритма в сети с низкой задержкой, сильно отличающейся от реальных сетей. Кроме того, в работе не были представлены сравнения с другими классическими алгоритмами.

Orca [36] в отличие от двух предыдущих алгоритмов регулировал не скорость отправки, а размер окна перегрузки. При этом диапазон действий был достаточно широк: умножение размера окна на 2^a , где a может принимать значения от -2 до 2. Функция награды равнялась отношению текущей пропускной способности к ширине канала, а пространство состояний было шире, чем у многих других алгоритмов: пропускная способность, потери, задержка пакетов, количество подтверждений, SRTT, окно перегрузки cwnd, максимальная скорость передачи, минимальная задержка пакетов. Алгоритм обучения, используемый Orca, представляет собой глубокий детерминированный градиент политики (DDPG), который сочетает в себе элементы Q-обучения и метода актор-критик, что также отличает его от двух предыдущих работ.

MVFS-RL [37] был разработан для контроля перегрузки в случае протокола QUIC, то есть для потокового видео. Состояние – это вектор из 20 значений (последнее значение времени приема-передачи RTT, минимальное значение RTT в течении эпизода, усредненная круговая задержка SRTT, дисперсия RTT, количество байт, отправленных с момента последнего подтверждения, количество байт, полученных момента последнего подтверждения и др.). Пространство действий – это 5 вариантов: уменьшить cwnd на 10, увеличить cwnd на 10, оставить cwnd без изменений, увеличить или уменьшить cwnd в 2 раза. Стоит отметить, что данные изменения cwnd (в несколько раз) достаточно агрессивны и используются не часто, так как, с одной стороны, способны увеличить скорость передачи за короткое время, но при этом существует большой риск вызвать перегрузку. Награда же представлена классической

комбинацией пропускной способности и задержки, но без учёта потерь. В 2023 году появилась схожая работа [38] по контролю перегрузки в протоколе QUIC. Функция награды имеет классическую зависимость от пропускной способности, RTT и потерь пакетов, а действиями являются уменьшение или увеличение $cwnd$ в некоторое количество раз, наподобие двоичного поиска в алгоритме BIC. Кроме того, отличительной особенностью является использование алгоритма BBR для вычисления некоторых характеристик, которые уже затем используются для обучения.

Spine [39]. Функция награды не всегда является простой комбинацией пропускной способности, задержки и потерь. К примеру, в Spine авторы используют более сложную конструкцию, учитывая в функции награды только высокие значения задержки. Вектор состояний выбран довольно классический: пропускная способность, задержка, $cwnd$ и др. (всего 8 показателей). Также в данной работе представлены сравнения с другими классическими алгоритмами контроля перегрузки и алгоритмами, основанными на машинном обучении (Aurora, Orca, Cubic, BBR). Результаты показывают, что в ряде случаев Spine работает несколько лучше с характеристиками сети из диапазона bandwidth (40-200 Mbps), delay (10-100ms) и loss (0-1%).

TCP-PPO2. В [40] была предложена ещё одна реализация алгоритма. В состояние входили: время с момента начала работы, текущий размер $cwnd$, количество отправленных, но ещё не подтверждённых байт, количество подтверждённых пакетов, RTT, скорость передачи данных, число потерянных пакетов. $Cwnd$ менялось согласно следующей формуле: $cwnd = cwnd + n * MSS$, где высокие значения n должны быть использованы при низкой заполненности канала, $n = 1$ при почти заполненном канале, n является отрицательным значением в случае перегрузки. В качестве моделирования компьютерной сети авторы используют Mininet, являющийся популярным эмулятором компьютерных сетей, встречающийся во многих работах. Результаты показывают, что политика TCP-PPO2 снижает задержку на 11,7–87,5% по сравнению с классическим CUBIC.

С развитием IoT (Интернета вещей) появилась необходимость контроля перегрузки в данных сетях, где устройства пользователей в жилых домах решают конкретные задачи. В 2024 году появилась работа [41], в которой механизм контроля перегрузки адаптирован для приложений умных сетей электроснабжения (smart grid), использующих ненадежные транспортные протоколы. Примером такого протокола является UDP, который, в отличие от TCP, не имеет встроенного контроля перегрузки, что может значительно снижать производительность сети. В силу нестандартности задачи авторы используют для состояния всего два неклассических показателя: долю пакетов, успешно принятых принимающим узлом, по отношению к общему количеству пакетов, отправленных со всех узлов сети; долю пакетов, успешно принятых принимающим узлом, по отношению к общему количеству пакетов, отправленных к данному узлу. Действием является не изменение значения $cwnd$, а решение о том, передавать или не передавать данные.

Проанализировав популярные алгоритмы контроля перегрузки с помощью обучения с подкреплением, можно выделить следующие закономерности (рис. 4).

- Функция награды зависит от пропускной способности, RTT и потерь. При этом чем выше скорость передачи, тем награда выше и, наоборот, чем выше RTT или потери, тем награда меньше.
- Пространство состояний разнообразно от работы к работе и задаётся различными характеристиками (от 2 до 20).
- Пространство действий также отличается от работы к работе, однако существуют две общие идеи: увеличение или уменьшение $cwnd$ на некоторое фиксированное число байт или же увеличение или уменьшение в 2^a , где a – некоторое число.

Краткий обзор алгоритмов обучения с подкреплением для контроля перегрузки представлен в табл. 3 и табл. 4. В табл. 4 представлены награды данных алгоритмов.

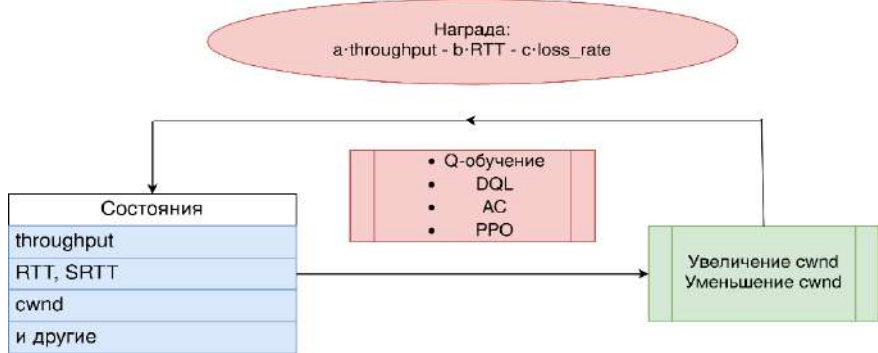


Рис. 4. Контроль перегрузки с помощью обучения с подкреплением.

Fig. 4. Congestion control using reinforcement learning.

Табл. 3. Сравнение различных алгоритмов обучения с подкреплением.

Table 3. Comparison of various reinforcement learning algorithms.

№	Год	Сценарий	Состояния	Действия
33	2018	Проводная сеть	1. интервалы между пакетами 2. RTT	1. $\text{cwnd} = \text{cwnd} + 2 \text{ MSS}$ 2. $\text{cwnd} = \text{cwnd} - \text{MSS}$ 3. $\text{cwnd} = \text{cwnd}$
34	2019	Сеть с низким bw	1. пропускная способность 2. число потерь	Уменьшение или увеличение скорости отправки
35	2019	Потоковое видео	1. пропускная способность 2. интервал между пакетами 3. потери пакетов 4. среднее значение RTT	Уменьшение или увеличение скорости отправки
36	2020	Потоковое видео	8 различных состояний (cwnd, RTT, ...)	Умножение cwnd на 2^a , а от -2 до 2
37	2019	Потоковое видео	20 различных состояний (cwnd, RTT и т.д.)	1. $\text{cwnd} = \text{cwnd} \pm 10 \text{ MSS}$ 2. $\text{cwnd} = \text{cwnd}/2$ 3. $\text{cwnd} = 2 \text{ cwnd}$ 4. $\text{cwnd} = \text{cwnd}$
38	2023	Потоковое видео	–	Умножение cwnd на 2^a , а от -1 до 1
39	2022	Проводная сеть	8 различных состояний (cwnd, RTT, ...)	Умножение cwnd на 2^a , а от -1 до 1
40	2023	Проводная сеть	8 различных состояний (cwnd, RTT, время с начала работы и т.д.)	1. $\text{cwnd} = \text{cwnd} + n \cdot \text{MSS}$ 2. $\text{cwnd} = \text{cwnd} + \text{MSS}$ 3. $\text{cwnd} = \text{cwnd} - n \cdot \text{MSS}$
41	2024	IoT	Доля успешно принятых пакетов	1. передавать данные 2. не передавать данные

Табл. 4. Функции награды.

Table 4. Reward function.

№	Функция награды	Алгоритм
33	$a \cdot \log(\text{throughput}) - b \cdot \log(\text{RTT})$	Q-обучение
34	$10 \cdot \text{throughput} - 1000 \cdot \text{delay} - 2000 \cdot \text{loss}$	PPO
35	$0.6 \cdot \ln(4 \cdot \text{throughput}) - \text{RTT} - 10 \cdot \text{loss}$	PPO
36	$\text{throughput} - b \cdot \text{delay}$	DDPG
37	$a \cdot \text{throughput} - b \cdot \max(\text{delay})$	AC
38	$a \cdot \text{throughput} - b \cdot \text{loss}$	PPO
40	$(a \cdot \frac{\text{throughput}}{\text{throughput}_{\max}}) - (1 - a) \cdot \frac{\text{delay}_{\min}}{\text{delay}}$	PPO

6. Алгоритмы контроля перегрузок в технологии multipath

Как уже отмечалось ранее, технология multipath позволяет клиенту (отправителю) посылать и принимать данные одновременно по нескольким сетевым маршрутам для увеличения эффективности работы алгоритма контроля перегрузки. Помимо стабильной работы технологии multipath, можно выделить два главных требования к этой технологии:

- соединению, состоящему из нескольких путей, необходимо обеспечивать пропускную способность не меньше, какую оно могло бы получить при использовании классического протокола TCP на лучшем пути (с самой высокой шириной канала);
- необходимо обеспечивать меру свободы канала (capacity) таким образом, чтобы ее значение для каждого пути был не больше, чем при использовании классического протокола TCP на лучшем пути.

Классической реализацией технологии multipath для TCP является протокол MRTCP [42-44]. Решение о выборе пути, по которому отправить пакет, принимает планировщик, выбирая путь с минимальным значением RTT. Однако как отмечают некоторые исследователи [45], протокол MRTCP в определённых сценариях обеспечивает пропускную способность ниже, чем при использовании классического TCP. Причиной такого поведения является Head-of-line blocking (HOL-блокировка) [46] – явление, ограничивающее пропускную способность (производительность), которое возникает, когда пакеты задерживаются первым пакетом в очереди. Некоторые улучшения, решающие данную проблему, представлены в работе [42].

Помимо реализации технологии multipath для TCP, существует несколько реализаций для протокола QUIC [47, 48]. Протокол MPQUIC реализован на прикладном уровне, а не на уровне ядра, что сильно упрощает его обновления. Кроме того, в данном случае не наблюдается проблема HOL-блокировки, характерной для протокола MRTCP.

Перечисленные выше алгоритмы технологии multipath имеют в своей основе планировщики, основанные на определённых наборах правил. Однако некоторый набор правил может быть эффективен для работы в одной среде, но менее эффективен в другой. С другой стороны, планировщики, основанные на обучении с подкреплением, могут учитывать особенности сетевой среды, в которой работают [49]. Так, в [50] авторами был предложен алгоритм multipath для QUIC на основе обучения с подкреплением. В качестве награды выступает число подтверждённых пакетов, состояниями являются cwnd, SRTT и другие классические параметры. Действия же направлены на выбор пути – выбор пути с некоторой вероятностью или выбор пути с минимальным RTT.

Существуют реализации технологии multipath с помощью обучения с подкреплением и для TCP [51]. Характеристики состояния – cwnd каждого из путей, а действия – отправление следующего пакета по каналу Wi-Fi или каналу сотовой связи. В [52] представлена реализация технологии multipath для 5G, что в последние годы является популярным направлением. Используя идеи контроля перегрузки с помощью обучения с подкреплением в случае одного пути, авторы [53] реализуют данную технологию для сценария беспроводных сетей. Характеристики состояния – это классические cwnd, SRTT, а в качестве действий выбрано регулирование окна каждого пути, что отличает данную работу от других работ по рассматриваемой теме, где действиями служат выбор конкретного пути для отправки конкретного пакета.

Спутниковая связь имеет высокую задержку и высокую скорость передачи данных. Авторы в [54] предлагают алгоритм multipath без использования машинного обучения в данном сценарии и представляют результаты, которые показывают, что в случае использования нескольких путей с высокой задержкой обеспечивается время загрузки веб-страницы на таком же уровне, как и при использовании одного пути с низкой задержкой. В [55, 56] представлен алгоритм в похожем сценарии, но уже с использованием обучения с подкреплением.

Исследования в данной области продолжаются и сейчас. В [57] был предложен алгоритм для протокола QUIC в сценарии 5G, главным отличием которого является использование метода fuzzy Q-learning [58] вместо классического Q-learning или DQL.

Проанализировав популярные алгоритмы технологии multipath с помощью обучения с подкреплением, можно выделить следующие закономерности:

- функция награды зависит от скорости передачи, RTT и потерь. При этом чем выше скорость передачи, тем награда выше и, наоборот, чем выше RTT или потери, тем награда меньше;
- пространство состояний зачастую представлено только cwnd и SRTT, в отличие от задач, рассмотренных в табл. 3, где существует широкое разнообразие состояний;
- пространство действий включает либо выбор пути для отправки пакета, либо изменение cwnd конкретного пути.

Краткий обзор алгоритмов multipath, реализованных с помощью обучения с подкреплением, представлен в табл. 5 и табл. 6.

Табл. 5. Сравнение различных алгоритмов multipath.

Table 5. Comparison of different multipath algorithms.

№	год	сценарий	состояния	действия
50	2022	Беспроводная сеть	1. cwnd 2. SRTT 3. Другие состояния	1. Выбор пути с некоторой вероятностью 2. Выбор пути с минимальным RTT
51	2019	Беспроводная сеть	cwnd каждого из путей	1. Выбор пути Wi-Fi 2. Выбор пути мобильной связи
52	2019	Мобильная связь 5G	1. cwnd 2. SRTT 3. Байты в «полёте»	1. Выбор первого пути 2. Выбор второго пути
53	2021	Беспроводная сеть	1. cwnd 2. SRTT	Регулировка cwnd каждого пути
49	2017	Беспроводная сеть	Каждый канал находится в «хорошем» или «плохом состоянии».	Отправка пакета в определённый путь
55	2019	Спутниковая связь	1. cwnd 2. SRTT 3. число подтверждений 4. сумма throughput всех каналов	Регулировка cwnd каждого пути
56	2023	Спутниковая связь	—	Регулировка cwnd каждого пути
57	2024	Мобильная связь 5G	1. cwnd 2. RTT 3. $\ln(\text{loss\_rate})$	1. Регулировка cwnd каждого пути

Табл. 6. Сравнение различных алгоритмов multipath.

Table 6. Comparison of different multipath algorithms.

№	Функция награды	Алгоритм
50	$throughput$	Глубокое Q-обучение
51	$packet_{ack} - packet_{loss}$	Глубокое Q-обучение
52	$throughput$	Глубокое Q-обучение
53	$a \cdot throughput - b \cdot delay - c \cdot loss_{rate}$	Глубокое Q-обучение
55	$a \cdot cwnd - b \cdot RTT - c \cdot packet_{retransmissions}$	DDPG

Сравнительная схема наиболее популярных алгоритмов, представленных в данной работе изображена на рис. 5. Таким образом, все алгоритмы и технологии для контроля перегрузки можно разбить на следующие группы:

- классические алгоритмы контроля перегрузки, основанные на потерях, на задержках и гибридные;
- алгоритмы контроля перегрузки, основанные на обучении с учителем;
- алгоритмы контроля перегрузки, основанные на обучении с подкреплением;
- алгоритмы, реализующие технологию multipath без использования машинного обучения;
- алгоритмы, реализующие технологию multipath с использованием обучения с подкреплением.

На потерях	На задержках	Обучение с учителем	Multipath
TCP Tahoe [3]	TCP Vegas [10]	LCC [28]	MPTCP [42]
TCP Reno [4]	TCP Vegas-A [11]	PCC Vivace [60]	MPTCP-LA [43]
TCP New Reno [5]	TCP Veno [13]	PCC Allegro [61]	MPTCP [44]
Scalable TCP [6]	TCP Vegas-V [59]	Обучение с подкреплением	MPQUIC [47]
Highspeed TCP [7]	Гибридные	QTCP [33]	MPQUIC [48]
TCP BIC [8]	TCP Westwood [14]	Aurora [34]	Multipath RL
TCP CUBIC [9]	BBR [15]	R3Net [35]	MPQUIC [50]
	BBR2 [16]	Orca [36]	MPTCP [51]
	Copa [18]	MVFS-RL [37]	MPQUIC [52]
		PBQ-Enhanced QUIC[38]	CMT-DRL [53]
		Spine [39]	MPTCP in Satellites [55]
		TCP-PPO2 [40]	mobile MPQUIC [57]

Рис. 5. Алгоритмы, представленные в работе.
Fig. 5. The algorithms presented in the work.

7. Заключение: проблемы алгоритмов, основанных на обучении с подкреплением

Сравнение эффективности различных алгоритмов. Как уже отмечалось ранее, сетевые среды могут быть очень различны с точки зрения ширины каналов, задержек и потерь. В большинстве работ авторы сравнивают эффективность реализованных алгоритмов лишь в узком диапазоне характеристик сети. При этом сравнения в широком диапазоне характеристик зачастую не осуществляются в силу их высокой трудоёмкости.

Кроме того, существует различные способы эмуляции сетей: как с помощью mininet [62], так и с помощью NS3 [63]. Данный факт также вносит неоднородность в результаты различных экспериментов и не позволяет достоверно сравнить эффективность работы различных алгоритмов.

Сбор набора данных. Данная проблема более характерна для алгоритмов обучения с учителем. Существует несколько готовых решений для записи трафика и получения его характеристик [64, 65] для дальнейшего обучения модели на собранном наборе данных. Однако собранные с помощью них признаки не всегда могут быть достаточно точны, чтобы использовать их для классификации типа потерь или предсказания RTT.

Вычислительная сложность. Алгоритмы, в основе которых лежат нейронные сети, могут обладать высокой точностью, однако их использование приводит к дополнительным

вычислительным затратам на определение оптимального действия. Таким образом, может произойти задержка в выполнении действий по контролю перегрузки сети, что значительно снижает эффективность работы алгоритмов. Кроме того, высокие затраты на обучение могут сделать алгоритм непригодным в силу большого времени отклика.

Низкая скорость обучения. Алгоритмы RL, в основе которых лежат глубокие нейронные сети, могут не сразу достигать оптимальных действий. Это может привести к непредвиденной работе алгоритма контроля перегрузки, особенно в начале.

Выбор состояний, действий и награды. Состояния, действительно влияющие на контроль перегрузки, могут быть различными в силу разнородности сетевых сред. Выбор состояний является нетривиальной задачей, поэтому в представленных работах можно заметить их широкий набор. Действия также могут быть различны и направлены как на резкое увеличение $swnd$, так и на более плавное. Награда же обычно зависит от пропускной способности, RTT и потерь, так как они являются главными показателями эффективности работы алгоритма. Таким образом, можно утверждать, что состояния и действия, оптимальные для одной среды, могут не быть таковыми для другой, что резко уменьшает преимущество алгоритмов обучения с подкреплением – эффективно работать в различных средах.

Выбор гиперпараметров. Проблема, характерная для многих алгоритмов RL, проявляет себя и в данной задаче. Чтобы получить нужный результат, зачастую требуется очень тонкая настройка гиперпараметров модели, что приводит к долгим поискам этих параметров.

Совместимость с другими алгоритмами. На данный момент плохо изучен вопрос совместимости представленных алгоритмов контроля перегрузки на основе машинного обучения с классическими алгоритмами контроля перегрузки (NewReno, CUBIC, BBR и др.). Однако, как известно, алгоритмы с агрессивным поведением (BBR) могут быть несправедливы при работе с другими алгоритмами (NewReno), занимая большую часть доступной полосы пропускания. Именно поэтому задачи исследования совместимости новых RL алгоритмов является актуальной.

Внедрение алгоритмов. Контроль перегрузок в классическом варианте находится на транспортном уровне и в случае TCP является частью ядра операционной системы (New Reno, CUBIC). Алгоритмы, основанные на RL, может быть проблематично встраивать в ядро операционной системы, что не способствует их широкому внедрению.

Большое разнообразие реализаций протокола QUIC. В последние годы появилось множество реализаций протокола QUIC, каждый из которых имеет особенности в работе. К примеру, в [66] было выделено 11 популярных реализаций QUIC и проанализировано их поведение в классических алгоритмах контроля перегрузки (CUBIC, Reno и т.д.). Результаты показывают, что некоторые реализации значительно отличаются. Данные отличия могут создать определённые проблемы при реализации алгоритма контроля перегрузки с помощью обучения с подкреплением для конкретного протокола QUIC, так как для каждой реализации QUIC придётся создавать свой алгоритм RL, который учитывает ее особенности.

Список литературы / References

- [1]. Jiang H. et al. When machine learning meets congestion control: A survey and comparison //Computer Networks. – 2021. – Т. 192. – С. 108033.
- [2]. Wei W., Gu H., Li B. Congestion control: A renaissance with machine learning //IEEE network. – 2021. – Т. 35. – №. 4. – С. 262-269.
- [3]. Jacobson V. Congestion avoidance and control //ACM SIGCOMM computer communication review. – 1988. – Т. 18. – №. 4. – С. 314-329.
- [4]. Mo J. et al. Analysis and comparison of TCP Reno and Vegas //IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320). – IEEE, 1999. – Т. 3. – С. 1556-1563.

- [5]. Floyd S., Henderson T. The NewReno modification to TCP's fast recovery algorithm. – 1999. – №. rfc2582.
- [6]. Kelly T. Scalable TCP: Improving performance in highspeed wide area networks //ACM SIGCOMM computer communication Review. – 2003. – Т. 33. – №. 2. – С. 83-91.
- [7]. Floyd S. HighSpeed TCP for large congestion windows. – 2003. – №. rfc3649.
- [8]. Xu L., Harfoush K., Rhee I. Binary increase congestion control (BIC) for fast long-distance networks //IEEE INFOCOM 2004. – IEEE, 2004. – Т. 4. – С. 2514-2524.
- [9]. Ha S., Rhee I., Xu L. CUBIC: a new TCP-friendly high-speed TCP variant //ACM SIGOPS operating systems review. – 2008. – Т. 42. – №. 5. – С. 64-74.
- [10]. Brakmo L. S., O'malley S. W., Peterson L. L. TCP Vegas: New techniques for congestion detection and avoidance //Proceedings of the conference on Communications architectures, protocols and applications. – 1994. – С. 24-35.
- [11]. Srijith K. N., Jacob L., Ananda A. L. TCP Vegas-A: Solving the fairness and rerouting issues of TCP Vegas //Conference Proceedings of the 2003 IEEE International Performance, Computing, and Communications Conference, 2003. – IEEE, 2003. – С. 309-316.
- [12]. Srijith K. N., Jacob L., Ananda A. L. TCP Vegas-A: Improving the performance of TCP Vegas //Computer communications. – 2005. – Т. 28. – №. 4. – С. 429-440.
- [13]. Fu C. P., Liew S. C. TCP Veno: TCP enhancement for transmission over wireless access networks //IEEE Journal on selected areas in communications. – 2003. – Т. 21. – №. 2. – С. 216-228.
- [14]. Mascolo S. et al. TCP Westwood: Bandwidth estimation for enhanced transport over wireless links //Proceedings of the 7th annual international conference on Mobile computing and networking. – 2001. – С. 287-297.
- [15]. Cardwell N. et al. Bbr: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time //Queue. – 2016. – Т. 14. – №. 5. – С. 20-53.
- [16]. Cardwell N. et al. BBRv2: A model-based congestion control performance optimization //Proc. IETF 106th Meeting. – 2019. – С. 1-32.
- [17]. Zhang Y., Cui L., Tso F. P. Modest BBR: Enabling better fairness for BBR congestion control //2018 IEEE symposium on computers and communications (ISCC). – IEEE, 2018. – С. 00646-00651.
- [18]. Arun V., Balakrishnan H. Copa: Practical {Delay-Based} congestion control for the internet //15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18). – 2018. – С. 329-342.
- [19]. Watkins C. J. C. H., Dayan P. Q-learning //Machine learning. – 1992. – Т. 8. – С. 279-292
- [20]. Gu S. et al. Continuous deep q-learning with model-based acceleration //International conference on machine learning. – PMLR, 2016. – С. 2829-2838.
- [21]. Konda V., Tsitsiklis J. Actor-critic algorithms //Advances in neural information processing systems. – 1999. – Т. 12.
- [22]. Labao A. B., Martija M. A. M., Naval P. C. A3C-GS: Adaptive moment gradient sharing with locks for asynchronous actor-critic agents //IEEE Transactions on Neural Networks and Learning Systems. – 2020. – Т. 32. – №. 3. – С. 1162-1176.
- [23]. Schulman J. et al. Proximal policy optimization algorithms //arXiv preprint arXiv:1707.06347. – 2017.
- [24]. El Khayat I., Geurts P., Leduc G. Improving TCP in wireless networks with an adaptive machine-learned classifier of packet loss causes //NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems: 4th International IFIP-TC6 Networking Conference, Waterloo, Canada, May 2-6, 2005. Proceedings 4. – Springer Berlin Heidelberg, 2005. – С. 549-560.
- [25]. Geurts P., El Khayat I., Leduc G. A machine learning approach to improve congestion control over wireless computer networks //Fourth IEEE International Conference on Data Mining (ICDM'04). – IEEE, 2004. – С. 383-386.
- [26]. Jayaraj A., Venkatesh T., Murthy C. S. R. Loss classification in optical burst switching networks using machine learning techniques: improving the performance of tcp //IEEE Journal on Selected Areas in Communications. – 2008. – Т. 26. – №. 6. – С. 45-54.
- [27]. Arouche Nunes B. A. et al. A machine learning framework for TCP round-trip time estimation //EURASIP Journal on Wireless Communications and Networking. – 2014. – Т. 2014. – С. 1-22.
- [28]. Dai T., Zhang X., Guo Z. Learning-based congestion control for internet video communication over wireless networks //2018 IEEE International Symposium on Circuits and Systems (ISCAS). – IEEE, 2018. – С. 1-5.

- [29]. Parlos A. G. Identification of the internet end-to-end delay dynamics using multi-step neuro-predictors //Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290). – IEEE, 2002. – Т. 3. – С. 2460-2465.
- [30]. Bui V. et al. Long horizon end-to-end delay forecasts: A multi-step-ahead hybrid approach //2007 12th IEEE Symposium on Computers and Communications. – IEEE, 2007. – С. 825-832.
- [31]. Belhaj S., Tagina M. Modeling and Prediction of the Internet End-to-end Delay using Recurrent Neural Networks //J. Networks. – 2009. – Т. 4. – №. 6. – С. 528-535.
- [32]. Ahmed A. H. et al. Predicting high delays in mobile broadband networks //IEEE Access. – 2021. – Т. 9. – С. 168999-169013.
- [33]. Li W. et al. QTCP: Adaptive congestion control with reinforcement learning //IEEE Transactions on Network Science and Engineering. – 2018. – Т. 6. – №. 3. – С. 445-458.
- [34]. Jay N. et al. A deep reinforcement learning perspective on internet congestion control //International Conference on Machine Learning. – PMLR, 2019. – С. 3050-3059.
- [35]. Fang J. et al. Reinforcement learning for bandwidth estimation and congestion control in real-time communications //arXiv preprint arXiv:1912.02222. – 2019.
- [36]. Abbasloo S., Yen C. Y., Chao H. J. Classic meets modern: A pragmatic learning-based congestion control for the internet //Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication. – 2020. – С. 632-647.
- [37]. Sivakumar V. et al. Mvfst-rl: An asynchronous rl framework for congestion control with delayed actions //arXiv preprint arXiv:1910.04054. – 2019.
- [38]. Zhang Z. et al. PBQ-Enhanced QUIC: QUIC with Deep Reinforcement Learning Congestion Control Mechanism //Entropy. – 2023. – Т. 25. – №. 2. – С. 294.
- [39]. Tian H. et al. Spine: An efficient DRL-based congestion control with ultra-low overhead //Proceedings of the 18th International Conference on emerging Networking Experiments and Technologies. – 2022. – С. 261-275.
- [40]. Shi H., Wang J. Intelligent TCP congestion control policy optimization //Applied Sciences. – 2023. – Т. 13. – №. 11. – С. 6644.
- [41]. Andrade-Zambrano A. R. et al. A Reinforcement Learning Congestion Control Algorithm for Smart Grid Networks //IEEE Access. – 2024.
- [42]. Wischik D. et al. Design, implementation and evaluation of congestion control for multipath {TCP} //8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11). – 2011.
- [43]. Nguyen K. et al. An approach to reinforce multipath TCP with path-aware information //Sensors. – 2019. – Т. 19. – №. 3. – С. 476.
- [44]. Chao L. et al. A brief review of multipath tcp for vehicular networks //Sensors. – 2021. – Т. 21. – №. 8. – С. 2793.
- [45]. Deng S. et al. WiFi, LTE, or both? Measuring multi-homed wireless internet performance //Proceedings of the 2014 Conference on Internet Measurement Conference. – 2014. – С. 181-194.
- [46]. Scharf M., Kiesel S. NXG03-5: Head-of-line Blocking in TCP and SCTP: Analysis and Measurements //IEEE Globecom 2006. – IEEE, 2006. – С. 1-5.
- [47]. De Coninck Q., Bonaventure O. Multipath quic: Design and evaluation //Proceedings of the 13th international conference on emerging networking experiments and technologies. – 2017. – С. 160-166.
- [48]. Viernickel T. et al. Multipath QUIC: A deployable multipath transport protocol //2018 IEEE International Conference on Communications (ICC). – IEEE, 2018. – С. 1-7.
- [49]. Wang S. et al. Deep reinforcement learning for dynamic multichannel access //International Conference on Computing, Networking and Communications (ICNC). – 2017. – С. 257-265.
- [50]. Lee S., Yoo J. Reinforcement learning based multipath QUIC scheduler for multimedia streaming //Sensors. – 2022. – Т. 22. – №. 17. – С. 6333.
- [51]. Luo J., Su X., Liu B. A reinforcement learning approach for multipath TCP data scheduling //2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). – IEEE, 2019. – С. 0276-0280.
- [52]. Roselló M. M. Multi-path scheduling with deep reinforcement learning //2019 European Conference on Networks and Communications (EuCNC). – IEEE, 2019. – С. 400-405.
- [53]. Yu C. et al. Reliable cybertwin-driven concurrent multipath transfer with deep reinforcement learning //IEEE Internet of Things Journal. – 2021. – Т. 8. – №. 22. – С. 16207-16218.
- [54]. Deutschmann J., Hielscher K. S. J., German R. An ns-3 model for multipath communication with terrestrial and satellite links //Measurement, Modelling and Evaluation of Computing Systems: 20th International

- GI/ITG Conference, MMB 2020, Saarbrücken, Germany, March 16–18, 2020, Proceedings 20. – Springer International Publishing, 2020. – C. 65-81.
- [55]. Mai T. et al. Self-learning congestion control of MPTCP in satellites communications //2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). – IEEE, 2019. – C. 775-780.
- [56]. Xing Z. et al. A multipath routing algorithm for satellite networks based on service demand and traffic awareness //Frontiers of Information Technology & Electronic Engineering. – 2023. – T. 24. – №. 6. – C. 844-858.
- [57]. Lv G. et al. Chorus: Coordinating Mobile Multipath Scheduling and Adaptive Video Streaming //Proceedings of the 30th Annual International Conference on Mobile Computing and Networking. – 2024. – C. 246-262.
- [58]. Berenji H. R. Fuzzy Q-learning: a new approach for fuzzy dynamic programming //Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference. – IEEE, 1994. – C. 486-491.
- [59]. Zhou W. et al. TCP Vegas-V: Improving the performance of TCP Vegas //International Conference on Automatic Control and Artificial Intelligence (ACAI 2012). – IET, 2012. – C. 2034-2039.
- [60]. Dong M. et al. {PCC} vivace: {Online-Learning} congestion control //15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18). – 2018. – C. 343-356.
- [61]. Dong M. et al. {PCC}: Re-architecting congestion control for consistent high performance //12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15). – 2015. – C. 395-408.
- [62]. De Oliveira R. L. S. et al. Using mininet for emulation and prototyping software-defined networks //2014 IEEE Colombian conference on communications and computing (COLCOM). – Ieee, 2014. – C. 1-6.
- [63]. Riley G. F., Henderson T. R. The ns-3 network simulator //Modeling and tools for network simulation. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. – C. 15-34.
- [64]. Beale J., Orebaugh A., Ramirez G. Wireshark & Ethereal network protocol analyzer toolkit. – Elsevier, 2006.
- [65]. Deri L. et al. ndpi: Open-source high-speed deep packet inspection //2014 International Wireless Communications and Mobile Computing Conference (IWCMC). – IEEE, 2014. – C. 617-622.
- [66]. Mishra A., Leong B. Containing the Cambrian Explosion in QUIC Congestion Control //Proceedings of the 2023 ACM on Internet Measurement Conference. – 2023. – C. 526-539.

Информация об авторах / Information about authors

Иван Александрович СТЕПАНОВ – аспирант ИСП РАН, ассистент кафедры информатики и вычислительной математики МФТИ. Сфера научных интересов: анализ сетевого трафика с помощью машинного обучения.

Ivan Alexandrovich STEPANOV – postgraduate student of the ISP RAS, an assistant at the Department of Computer Science and Computational Mathematics at MIPT. Research interests: network traffic analysis using machine learning.

Максим Владимирович ПОПОВ – старший лаборант ИСП РАН, студент магистратуры факультета ВМК МГУ. Сфера научных интересов: анализ сетевого трафика, алгоритмы контроля перегрузок.

Maxim Vladimirovich POPOV – graduate student at the Faculty of Computational Mathematics and Cybernetics of Moscow State University. Research interests: network traffic analysis, congestion control algorithms.

Александр Игоревич ГЕТЬМАН – кандидат физико-математических наук, старший научный сотрудник ИСП РАН, ассистент ВМК МГУ и МФТИ, доцент ВШЭ. Сфера научных интересов: анализ бинарного кода, восстановление форматов данных, анализ и классификация сетевого трафика.

Aleksandr Igorevich GETMAN – Cand. Sci. (Phys.-Math.), senior researcher at ISP RAS, assistant at CMC MSU and MIPT, associate professor at HSE. Research interests: binary code analysis, data format recovery, network traffic analysis and classification.

Мария Кирилловна ИКОННИКОВА – младший научный сотрудник ИСП РАН. Сфера научных интересов: анализ сетевого трафика, машинное обучение.

Maria Kirillovna IKONNIKOVA is a junior researcher at the ISP RAS. Research interests: network traffic analysis, machine learning.

Андрей Андреевич БЕЛЕВАНЦЕВ – доктор физико-математических наук, ведущий научный сотрудник ИСП РАН, профессор МГУ. Сфера научных интересов: статический анализ программ, оптимизация программ, параллельное программирование.

Andrey Andreevich BELEVANTSEV – Dr. Sci. (Phys.-Math.), Prof., leading researcher at ISP RAS, Professor at MSU. Research interests: static analysis, program optimization, parallel programming.



Методы динамической верификации промышленных средств защиты информации на основе формальных моделей управления доступом.

^{1,2,3} А.К. Петренко, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>

⁴ П.Н. Девянин, ORCID: 0000-0003-2561-794X <pdevyanin@astralinux.ru>

¹ Д.В. Ефремов, ORCID: 0000-0002-9916-056X <efremov@ispras.ru>

^{1,3} А.А. Карнов, ORCID: 0000-0002-2066-9946 <karnov@ispras.ru>

^{1,2} Е.В. Корныхин, ORCID: 0000-0001-9303-3132 <kornevgen@ispras.ru>

^{1,2,3} В.В. Кулямин, ORCID: 0000-0003-3439-9534 <kuliamin@ispras.ru>

^{1,2,3,5} А.В. Хорoshiлов, ORCID: 0000-0002-6512-4632 <khoroshilov@ispras.ru>

¹ Институт системного программирования имени В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.

² Московский государственный университет имени М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1.

³ НИУ Высшая школа экономики,
101978, Россия, г. Москва, ул. Мясницкая, д. 20.

⁴ ООО «РусБИТех-Астра»
117105, Россия, Москва, Варшавское шоссе, д. 26, стр. 11.

⁵ Московский физико-технический институт,
141701, Россия, Московская область, г. Долгопрудный, Институтский пер., 9.

Аннотация. В статье рассматриваются методы динамической верификации программных систем, представляющих собой средства защиты информации (СЗИ) или включающих такие средства в свой состав. Для обеспечения высокого уровня доверия и защищенности программных систем необходимо применять разные методы и технологии верификации, при этом важны не только потенциальная мощность метода, но возможность использовать его в реальных условиях промышленной разработки крупных и сложных программных систем. Строгость и точность верификации обеспечивают формальные методы, однако использование классических формальных методов диктует особые, крайне высокие требования к персоналу и влечёт за собой дополнительные трудозатраты. Статья предлагает технологию динамической верификации СЗИ, которая, с одной стороны, близка к техникам тестирования, поэтому проще для освоения инженерами-тестировщиками, и, с другой стороны, в качестве базы использует формальные модели управления доступом и спецификации внешних интерфейсов СЗИ, которые уже появляются у разработчиков ОС и СУБД, чья продукция должна соответствовать требованиям нового национального стандарта ГОСТ Р 59453.4-2025 «Защита информации. Формальная модель управления доступом. Часть 4. Рекомендации по верификации средства защиты информации, реализующего политики управления доступом, на основе формализованных описаний модели управления доступом». Данный стандарт также представлен в статье.

Ключевые слова: динамическая верификация; мониторинг; тестирование на основе моделей; формальная модель управления доступом; функциональная спецификация.

Для цитирования: Петренко А.К., Девианин П.Н., Ефремов Д.В., Карнов А.А., Корныхин Е.В., Кулямин В.В., Хорошилов А.В. Методы динамической верификации промышленных средств защиты информации на основе формальных моделей управления доступом. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 277–290. DOI: 10.15514/ISPRAS–2025–37(3)–19.

Methods of Runtime Verification of Industrial Information Security Tools Based on Formal Access Control Models.

- ^{1,2,3} A.K. Petrenko, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>
⁴ P.N. Devyanin, ORCID: 0000-0003-2561-794X <pdevyanin@astralinux.ru>
¹ D.V. Efremov, ORCID: 0000-0002-9916-056X <efremov@ispras.ru>
^{1,3} A.A. Karnov, ORCID: 0000-0002-2066-9946 <karnov@ispras.ru>
^{1,2} E.V. Kornukhin, ORCID: 0000-0001-9303-3132 <kornevgen@ispras.ru>
^{1,2,3} V.V. Kuliain, ORCID: 0000-0003-3439-9534 <kuliain@ispras.ru>
^{1,2,3,5} A.V. Khoroshilov, ORCID: 0000-0002-6512-4632 <khoroshilov@ispras.ru>
¹ *Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*
² *Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia.*
³ *National Research University, Higher School of Economics
20, Myasnitskaya ulitsa, Moscow, 101978, Russia.*
⁴ *RusBITech-Astra,
26, Varshavskoe, Moscow, 117105, Russia.*
⁵ *Moscow Institute of Physics and Technology (MIPT),
9, Campus per., Dolgoprudny, Moscow region, 114701, Russia.*

Abstract. The paper discusses methods of runtime verification of software systems that are security protection mechanisms (PM) or include such mechanisms in their design. To ensure a high level of trust and security of software systems, it is necessary to use different verification methods and technologies. In this case, not only the potential power of the method is important, but also the possibility of using it in real conditions of industrial development of large and complex software systems. The rigor and accuracy of verification are ensured by formal methods; however, the use of classical formal methods dictates special, extremely high requirements for personnel and entails additional labor costs. The article proposes a technology for runtime verification of PM, which, on the one hand, is close to testing techniques, therefore it is easier for test engineers to master, and, on the other hand, uses formal access control models and specifications of external PM interfaces as a basis, which are already appearing among OS and DBMS developers, whose products must meet the requirements of the new national standard GOST R 59453.4-2025 "Information Security. Formal access control model. Part 4. Recommendations for verification of information security tools implementing access control policies based on formalized descriptions of the access control model. This standard is also presented in the article.

Keywords: runtime verification; monitoring; Model Based Testing; formal access control model; functional specification.

For citation: Petrenko A.K., Devyanin P.N., Efremov D.V., Karnov A.A., Kornukhin E.V., Kuliain V.V., Khoroshilov A.V. Methods of runtime verification of industrial information security tools based on formal access control models. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 277-290 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-19.

1. Введение

Теме обеспечения защиты программных и программно-аппаратных систем уделяется все большее внимание. Особенно важно выполнение требований информационной безопасности к самим средствам защиты информации (СЗИ). Спектр СЗИ достаточно широк, но в данной статье мы фокусируемся на СЗИ, которые входят в состав и, по сути, даже выполняют часть функций по защите информации таких компонентов базового программного обеспечения (ПО) как операционные системы (ОС) и системы управления базами данных (СУБД). Такие СЗИ обеспечивают контроль за обеспечением корректности предоставления (или запрета) доступа тех или иных программ/процессов (субъектов доступа) к тем или иным программным и программно-аппаратным сущностям (объектам доступа, файлам, каналам ввода-вывода, процессам и т.д.). Роль ОС и СУБД является основополагающей в обеспечении информационной безопасности, этим и объясняется выбранный в данной статье фокус рассмотрения, хотя рассматриваемый метод и технологии подобного рода, конечно, могут применяться и для верификации СЗИ других видов.

Поводом к появлению данной публикации стали завершение разработки и утверждение нового национального стандарта ГОСТ Р 59453.4-2025 «Защита информации. Формальная модель управления доступом. Часть 4. Рекомендации по верификации средства защиты информации, реализующего политики управления доступом, на основе формализованных описаний модели управления доступом» [1]. Данный ГОСТ продолжает серию стандартов ГОСТ Р 59453:

- ГОСТ Р 59453.1-2021 «Защита информации. Формальная модель управления доступом. Часть 1. Общие положения» [2];
- ГОСТ Р 59453.2-2021 «Защита информации. Формальная модель управления доступом. Часть 2. Рекомендации по верификации формальной модели управления доступом» [3];
- ГОСТ Р 59453.3-2025 «Защита информации. Формальная модель управления доступом. Часть 3. Рекомендации по разработке» [4].

Стандарты были разработаны Федеральной службой по техническому и экспортному контролю (ФСТЭК России), Институтом системного программирования им. В.П. Иванникова Российской академии наук (ИСП РАН), Обществом с ограниченной ответственностью «РусБИТех-Астра» (ООО «РусБИТех-Астра») и Федеральным автономным учреждением «Государственный научно-исследовательский испытательный институт проблем технической защиты информации Федеральной службы по техническому и экспортному контролю» (ФАУ «ГНИИИ ПТЗИ ФСТЭК России»).

Первый из этих стандартов содержит общие положения и критерии, которым должны соответствовать формальные модели управления доступом (ФМД). Второй стандарт дает рекомендации по использованию методов верификации и определяет требования к инструментам верификации формальных моделей управления доступом. Третий стандарт описывает расширенные рекомендации по разработке формальных моделей в условиях создания и сертификации крупных программных систем, в частности, отечественных ОС.

Далее в статье представлены краткие сведения о стандарте, описывающем рекомендации по динамической верификации средства защиты информации, реализующего политики управления доступом [1], общая схема верификации СЗИ, методы и технологии, которые могут использоваться в рамках схемы, и требования к их выбору, обусловленные типовыми сценариями верификации и сертификации СЗИ.

2. ГОСТ Р 59453.4-2025. «Защита информации. Формальная модель управления доступом. Часть 4»

Стандарт представляет собой рекомендации по верификации средств защиты информации, реализующих политики управления доступом, на основе формализованного описания модели управления доступом. Он предназначен для разработчиков средств защиты информации, а также для органов по сертификации и испытательных лабораторий при проведении сертификации СЗИ.

В контексте развития и расширения использования перспективных технологий данный стандарт открывает возможности одного из современных подходов к задачам повышения надежности и защищенности программных и программно-аппаратных систем, который известен как динамическая верификация (Runtime Verification – RV) или тестирование на основе формальных моделей (Model Based Testing – MBT¹). Наиболее известным международным стандартом, в котором RV и MBT представляется как один из рекомендуемых элементов технологической цепочки создания программных систем ответственного назначения, является DO-178C [5] и его дополнения DO-331 (Model-Based Development and Verification Supplement to DO-178C and DO-278A) и DO-333 (Formal Methods Supplement to DO-178C and DO-278A). В Российской Федерации действует аналог DO-178C, который известен как КТ-178C [6].

Семейство стандартов DO-178 широко используется в области аэрокосмической техники. Область распространения DO-331 и DO-333 пока еще не велика, но она активно расширяется. Главным фактором, который способствует применению формальных методов в разработке и сертификации программных систем ответственного назначения, является концентрация внимания основных участников программного проекта: аналитиков, архитекторов, инженеров по разработке требований, разработчиков и верификаторов - на задаче четкого определения функциональных требований к интерфейсам системы с внешним миром и к внутренним интерфейсам. Тщательная проработка интерфейсов и поддержка их спецификаций в актуальном и консистентном состоянии позволяет на строгой и систематической основе контролировать точность и корректность реализации проектных решений, на ранних стадиях проектирования и разработки выявлять отклонения от них и принимать меры по их устранению. То есть главный позитивный вклад MBT (тестирования на основе формальных моделей) состоит не столько в автоматизации собственно тестирования, сколько во внедрении в практику разработки ответственных программных систем (СЗИ очевидный пример таких систем) технологий и процессов построения строгих спецификаций функциональных требований как на уровне внешних интерфейсов, так и на уровне отдельных подсистем и модулей. Если спецификации записываются на однозначно понимаемом языке (а еще лучше на машиночитаемом языке), то открывается возможность реализовать единообразный и инструментально поддерживаемый подход к анализу полноты и корректности результатов разработки.

Первые три части стандарта [2-4] сосредотачиваются не на вопросах тестирования, а на вопросах разработки и верификации формальных моделей, которые являются основой для тестирования СЗИ. В них рассматриваются вопросы строгого описания требований к отдельным функциям (действиям) СЗИ и спецификации интегральных требований обеспечения информационной безопасности, так называемых условий безопасности. Область действия четвертой части стандарта ГОСТ Р 59453.4-2025 покрывает задачи проверки того, что все требования к СЗИ, которые представлены в формальной модели управления доступом и формальной функциональной спецификации программных интерфейсов (ФСП) СЗИ, выполнены в полной мере и корректно.

¹ Синонимом MBT является сокращение SBT – Specification Based Testing, если речь идет о формальных спецификациях.

В стандарте рассмотрено два сценария тестирования. Первый сценарий предлагает тестировать объект оценки на системном уровне, то есть не погружаться на уровень отдельных модулей и подсистем, которые образуют ОС или СУБД. Например, в случае ОС интерфейсом системного уровня являются системные вызовы ее ядра. Такое тестирование целесообразно проводить на основе формальных спецификаций системных вызовов, то есть ФСП. При этом предварительно надо доказать, что спецификация системных вызовов соответствует требованиям ФМД.

Второй сценарий предполагает наличие возможности выделения и определения интерфейсов модуля информационной безопасности (в ОС Linux такой модуль обычно называется Linux Security Module – LSM) и тестирования этого модуля либо по схеме модульного тестирования (то есть в изолированной среде), либо по некоторой комбинированной схеме. В последнем случае либо тестовые воздействия подаются на модуль непосредственно от тестовой системы, минуя некоторые службы ядра ОС, либо результаты работы модуля передаются непосредственно в тестирующую систему и анализируются ею.

Использование второй схемы не отменяет использование первой. Второй сценарий, как и любое модульное тестирование, позволяет добиться более полного тестового покрытия, но технически он существенно сложнее первого, так как требуется создание еще одного слоя формальных спецификаций интерфейсов модуля безопасности и тестового окружения этого модуля. В рамках данной статьи мы остановимся только на первом сценарии, опыт, полученный в эксперименте по реализации второго сценария, описан авторами и их коллегами в [7], где рассмотрен подход к тестированию механизмов управления доступом ОС Astra Linux [8] на основе описания на языке спецификаций (нотации) Event-B [9, 10] мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в ОС семейства Linux (МРОСЛ ДП-модели) [11].

В приложениях к стандарту приведен пример верификации системного вызова открытия файла *open* и рекомендации по оценке тестового покрытия на основе формальной спецификации системного вызова.

Пример представляет сначала описание операции *open* на уровне ФМД, потом приводится формальная спецификация системного вызова *open*. Обе формальные спецификации написаны на языке Event-B. Далее приводится пример таблицы, представляющей описание полученного тестового покрытия

3. Обзор работ в исследуемой области

Данную работу можно рассматривать как один из подходов к тестированию на основе формальных моделей для задач оценки и повышения степени защищенности программных систем, то есть мы говорим о Model Based Testing (MBT) в контексте тестирования безопасности (security testing). Этой теме посвящено много публикаций. Для сужения области рассмотрения уточним, что в качестве формальных моделей, которые мы используем для извлечения тестов, оценки результата их выполнения и полноты тестового покрытия, мы используем формальные модели безопасности, а еще точнее формальные модели управления доступом.

Анализу работ в этой области посвящен фундаментальный обзор М. Фельдерера и др. [12]. Авторы различают три вида моделей безопасности и, соответственно, три группы целей тестирования на основе таких моделей:

- Общие модели безопасности. Цель – тестирование базовых составляющих информационной безопасности, а именно, обеспечение конфиденциальности, целостности, доступности, аутентификации, авторизации или неотказуемости.
- Модели уязвимостей. Цель – поиск уязвимостей, приводящих к определенным классам нарушений безопасности.

- Модели безопасности или моделей управления доступом. Цель – тестирование механизмов безопасности.

В свете такой классификации описанный в данной работе подход соответствует третьему виду моделей и целей тестирования, а именно тестированию функций СЗИ.

К первому виду моделей можно отнести модели сценариев воздействия на СЗИ. Широко известны работы группы Ины Шифердеккер (Schieferdecker, Ina) из Fraunhofer FOKUS по моделям тестовых сценариев и их использованию для тестирования [13]. В основе этого подхода лежат язык TTCN-3 и UML Testing Profile [14].

Ко второму виду моделей можно отнести различного рода статические и динамические паттерны уязвимостей, которые часто приводят к нарушениям информационной безопасности.

Работ по формализации спецификаций моделей безопасности и их верификации в свое время было достаточно много. Однако большая часть работ рассматривала верификацию моделей безопасности в отрыве от процесса разработки и верификации реализации СЗИ, поэтому в них не уделялось должного внимания интеграции различных процессов жизненного цикла и выбора технологий, удобных для применения не в обстановке академических исследований, а в промышленных проектах.

Ближе всего к задаче описания полного цикла работ, связанных с разработкой и верификацией моделей безопасности и с процессами создания/генерации тестов на построенной модели, подошли авторы работы [15]. В качестве языка описания модели безопасности ими был выбран В-метод [16]. Основой для генерации тестов были паттерны тестовых последовательностей, которые, по сути, описывали темпоральные свойства тестовых сценариев. Из соображений упрощения технологии авторы использовали не темпоральные логики и нотации на их основе, а регулярные выражения, которые хорошо известны в практике программирования. Роль модели безопасности фактически была сведена к роли тестового оракула, при помощи которого тест мог автоматически детектировать отклонения тестируемой системы от требований к управлению доступом. Основным фокусом данной публикации являлась задача генерации тестовых последовательностей, в частности методам мутации тестовых сценариев. Соответственно, задачам интеграции технологий поддержки моделей, совместного использования артефактов на разных языках моделирования и программирования, что важно в промышленном применении таких методов, внимания не уделялось.

Важный методологический результат описывается в работе европейской команды [17]. В этой работе предлагается общий подход на основе идей Model Driven Engineering (MDE). Рамки подхода охватывают разработку моделей безопасности, координацию работ по проектированию системы и встраиваемых в нее функций и компонентов СЗИ, развертывание системы и ее тестирование. В качестве базовой технологии для представления политик безопасности предлагается использовать XACML [18]. Интеграция функционального программного обеспечения (ПО) и компонентов СЗИ обеспечивается при помощи технологий аспектно-ориентированного программирования. Описание собственно предложенной методологии представлено в форме метамодели, которая уточняется при развертывании целевого ПО на конкретной программно-аппаратной конфигурации. Конкретных техник моделирования, верификации моделей и генерации тестов в данной работе не описано, но работа интересна целостным взглядом на проблему создания и верификации СЗИ в индустриальном окружении с привлечением перспективных методов проектирования и реализации программных систем ответственного назначения.

Подводя итог краткому анализу методов и технологий создания и верификации промышленных СЗИ на основе формальных моделей безопасности или управления доступом можно заключить, что в доступной литературе хорошо проработаны вопросы решения

отдельных задач жизненного цикла, но пока нет достаточно эффективного способа бесшовной интеграции таких решений в единую технологическую цепочку. При этом помимо собственно задач интеграции в промышленном контексте важно найти компромисс, который обеспечивает эффективное решение отдельных задач и не требует сверхвысокой квалификации разработчиков и инженеров-верификаторов или владения сложными в освоении инструментами и технологиями.

Предложенная ниже схема верификации СЗИ разрабатывалась с целью обеспечить такой компромисс и при этом достичь достаточно высокой степени интеграции процессов и технологий в общей цепочке работ жизненного цикла разработки безопасного ПО.

4. Схема верификации СЗИ на основе формальной модели управления доступом

Общая схема верификации СЗИ на основе формальной модели управления доступом описана в монографии [19]. Если не рассматривать задачи по спецификации и верификации модуля безопасности ОС Linux (Linux Security Module - LSM), а ограничиться задачами

- разработки и верификации ФМД,
- построения ФСП, отвечающей требованиям модели, и верификации реализации СЗИ,
- проверки выполнения требований ФСП,

то схема верификации будет выглядеть так, как показано на рис. 1.

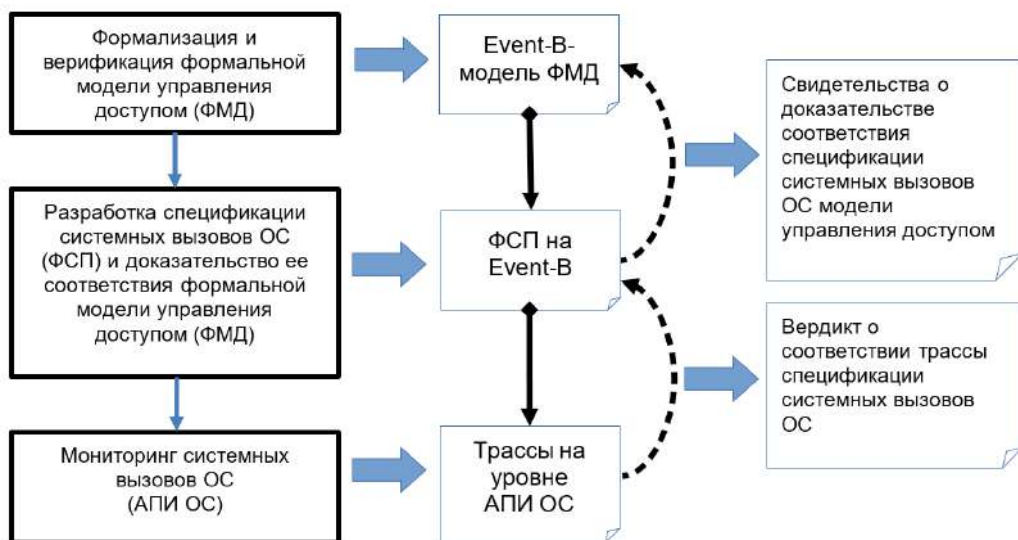


Рис. 1. Общая схема верификации СЗИ ОС на основе формальной модели управления доступом.
Fig. 1. General scheme of verification of OS security protection mechanism based on formal access control model.

Процесс начинается с разработки формальной модели управления доступом. В настоящее время известны две нотации, которые успешно используются для этой задачи: Event-B [9, 10] и TLA+ [20, 21]. В качестве примера спецификационной нотации во всех частях стандарта группы ГОСТ Р 59453 выбрана нотация Event-B и инструменты, которые с ней работают [22]. Это объясняется тем, что она базируется на привычной парадигме процедурного программирования и оснащена инструментами дедуктивной верификации в отличие от

TLA+, где парадигмой является темпоральная логика, а зрелые и удобные инструменты принадлежат к классу верификаторов моделей (model checker), что означает поддержку не строгого доказательства корректности, а лишь проверку перебором в рамках ограниченного числа экземпляров моделей поведения.

Вторая задача – это разработка и верификация формальной функциональной спецификации системных вызовов (ФСП). Здесь также используется нотация Event-B и инструменты дедуктивной верификации. В случае ФСП реальной альтернативой Event-B мог бы быть B-Method и инструменты, которые его поддерживают, но в России опыта их использования пока нет.

В данной статье мы не рассматриваем первую задачу верификации – как проверить, что формальная спецификация системных вызовов операционной системы отвечает требованиям формальной модели управления доступом. Хотя в монографии [19] приводится один из способов доказательства такого соответствия, применять его на практике затруднительно – создание эффективных методов и инструментов такой верификации пока еще не закончено. Тем не менее мы исходим из того, что ФСП соответствует модели и поэтому, если реализация СЗИ отвечает требованиям ФСП, то она соответствует и требованиям формальной модели управления доступом.

Третья задача – собственно динамическая верификация, которая выполняется либо в режиме пассивного тестирования, когда под реальной нагрузкой осуществляется мониторинг, либо в режиме выполнения специально подготовленного тестового набора с мониторингом. Мониторинг сводится к сбору трассы, в которой фиксируются данные о системных вызовах, связанных с работой СЗИ, и данные о результатах их выполнения. По завершении тестирования или в реальном времени выполняется анализ собранной трассы. Она должна соответствовать поведению СЗИ, представленному в спецификации ФСП, и в этом случае выносится положительный вердикт. В противном случае тест обнаруживает ошибку – отклонение реализации СЗИ от требований ФСП.

В ГОСТ Р 59453.4-2025 особое внимание уделяется контролю за полнотой тестового покрытия, где основной метрикой для оценки служат структуры формальной спецификации интерфейсов СЗИ. Рекомендации по оценке тестового покрытия приводятся в Приложении Б данного стандарта. Ниже приводится сокращенный вариант рекомендаций, который позволяет понять суть предлагаемого подхода.

...Отдельный подход необходим, если условие *(в спецификации некоторой операции)* представляет собой дизъюнкцию из нескольких выражений-дизъюнктов. То же верно для импликации, при этом импликация может быть преобразована в дизъюнкцию по правилу $(A \Rightarrow B) \equiv (\neg A \vee B)$. В этом случае для создания ситуации, в которой условие принимает значение FALSE, необходимо, чтобы все входящие в дизъюнкцию выражения приняли значение FALSE. Для покрытия ситуаций, в которых полное условие принимает значение TRUE, рекомендуется создать, как минимум, набор ситуаций, в которых каждое отдельное входящее в дизъюнкцию выражение принимает значение TRUE, а остальные — FALSE. Если ситуация, в которой только одно из выражений выполнено, невозможна, рекомендуется создавать возможные ситуации, в которых выполнено минимальное множество выражений-дизъюнктов, включающее рассматриваемое. Эта рекомендация применяется, когда необходимо выполнение условия-дизъюнкции при выполнении остальных охранных условий. Если одно из других охранных условий нарушается, обеспечение значения TRUE для дизъюнкции не имеет особого значения. *(В приведенном ниже примере используется нотация Event-B. В квадратные скобки заключены метки-комментарии, которые нужны для ссылок на составные части условий, входящих в спецификации).*

Пример.

Допустим, мы пытаемся покрыть различные ситуации, связанные с работой операции получения доступа субъекта к объекту $GetAccess(subj, obj, akind)$. Пусть условие успешного выполнения этого события имеет следующий вид (числа в квадратных скобках нумеруют отдельные условия и дизъюнкты).

$subj \in Subjects$ [1] /* *subj является субъектом доступа* */
 $\wedge obj \in Objects$ [2] /* *obj является объектом доступа* */
 $\wedge akind \in AccessKind$ [3] /* *akind является видом доступа* */
 $\wedge subj \in ActiveSubjects$ [4] /* *пусть в системе есть субъекты, которые могут получать доступ к объектам, и есть неактивные, которые доступ получать не могут, но могут выполнять какие-то другие операции* */
 $\wedge (subj = Admin$ [5.1] $\vee (obj, akind) \in AccessRights(subj)$ [5.2]) [5]
 /* *получение доступа успешно, либо если субъект является привилегированным (администратором), либо если у него есть право на указанный вид доступа к указанному объекту* */

Получение минимального покрывающего набора тестовых ситуаций в соответствии с представленными выше рекомендациями выполняется следующим образом.

Условия [1], [2], [3], по сути, представляют собой типовые ограничения на параметры и не могут быть нарушены при любой попытке исполнения данной операции. Они должны иметь значение TRUE.

Условия [4] и [5] могут быть нарушены. При этом условие [5] состоит из дизъюнктов [5.1] и [5.2], и его выполнение может быть обеспечено обращением в TRUE любого из этих дизъюнктов.

Рекомендуемый для создания в тестах минимальный набор ситуаций для данного примера такой:

1. Условия успешного исполнения операции выполнены, дизъюнкция [5] выполнена за счет дизъюнкта [5.1]
 $subj \in Subjects$
 $\wedge obj \in Objects$
 $\wedge akind \in AccessKind$
 $\wedge subj \in ActiveSubjects$
 $\wedge (subj = Admin \wedge (obj, akind) \notin AccessRights(subj))$ [нарушено 5.2]
2. Условия успешного исполнения операции выполнены, дизъюнкция [5] выполнена за счет дизъюнкта [5.2]
 $subj \in Subjects$
 $\wedge obj \in Objects$
 $\wedge akind \in AccessKind$
 $\wedge subj \in ActiveSubjects$
 $\wedge (subj \neq Admin \wedge (obj, akind) \in AccessRights(subj))$
3. Условия успешного исполнения нарушены за счет нарушения условия [4]
 $subj \in Subjects$
 $\wedge obj \in Objects$
 $\wedge akind \in AccessKind$
 $\wedge subj \notin ActiveSubjects$ [нарушено 4]

$\wedge (subj \neq Admin [нарушено\ 5.1, \text{выполнить его при нарушении 4 может быть невозможно}] \wedge (obj, akind) \in AccessRights(subj) [выполнено\ 5.2])$

4. Условия успешного исполнения нарушены за счет нарушения условия [5]

$subj \in Subjects$

$\wedge obj \in Objects$

$\wedge akind \in AccessKind$

$\wedge subj \in ActiveSubjects$

$\wedge (subj \neq Admin \wedge (obj, akind) \notin AccessRights(subj)) [нарушены оба дизъюнкта\ 5.1 \text{ и } 5.2]$

5. Особенности динамической верификации промышленных средств защиты информации

Опыт разработки инструмента динамической верификации СЗИ операционных систем АНИС [23] помог обнаружить некоторые подводные камни ее применения к такой сложной СЗИ, как ОС семейства Linux. Чтобы остановиться на вопросах, которые потребовали серьезного внимания, предварительно опишем общую схему работ по верификации СЗИ при помощи АНИС.

В соответствии со стандартами ГОСТ Р 59453 сначала разрабатывается формальная модель управления доступом – ФМД, потом спецификация СЗИ – ФСП. Разработка и верификация модели и спецификации может проводиться, например, при помощи инструмента дедуктивной верификации Rodin [22]. Далее инженер-верификатор переходит к собственно задачам тестирования, и, если модель и спецификация разрабатывается не с нуля, то инженеру-верификатору на этой фазе жизненного цикла удобнее работать не с инструментом дедуктивной верификации, а сразу с инструментом, который предназначен для разработки и отладки тестов. То есть функциональность инструмента для разработки тестов на основе формальной модели должна позволять редактировать, проводить легковесную проверку ФСП и, возможно, даже отлаживать ФСП.

Динамическая верификация на основе формальной модели накладывает некоторые ограничения на язык и стиль спецификации. Так использование кванторных выражений или имплицитных деклараций либо требует существенно больших ресурсов памяти и времени верификации, либо вообще делает невозможным точную интерпретацию модели в ходе тестирования. Это особенно важно, когда мониторинг выполняется под рабочей нагрузкой системы. По этой причине, во-первых, следует четко очертить подмножество языка, на котором разрабатывается спецификация, и, во-вторых, реализовать в инструменте функцию проверки того, что ФСП отвечает требованиям данного подмножества.

Далее важно отметить, что при тестировании промышленных систем на основе формальных спецификаций практически всегда приходится работать со спецификациями разного уровня абстракции (детализации), спецификация интерфейсов системы или модулей реализации также по уровню абстракции отличается от спецификаций, написанных на языках спецификации. В случае тестирования СЗИ в ОС для сопоставления требований из спецификации ФСП с результатами выполнения системных вызовов, которые представляются данными в формате, описанном на языке Си, необходимо создать слой адаптеров. Кроме того, в задачу сопоставления входит и анализ корректности последовательностей событий на уровне модели и событий на уровне системных вызовов, что тоже является нетривиальной задачей. Более подробная схема реализации и сценарии использования АНИС представлена в статье [23].

7. Заключение

В работе представлен новый стандарт, в котором описаны рекомендации по верификации СЗИ и метод тестирования на основе формальных моделей управления доступом. Данный подход позволяет одновременно решить несколько задач важных для повышения уровня доверия к СЗИ, в которых управление доступом является неотъемлемой частью функциональности объекта оценки. Важными примерами таких систем являются ОС и СУБД. Тестирование на основе формальных моделей отличается от других подходов тем, что оно одновременно позволяет:

- опираться на строгие спецификации требований информационной безопасности;
- обеспечивать высокую степень масштабируемости вплоть до систем размером в десятки миллионов строк кода;
- обеспечивать объективную базу для оценки полноты тестирования как на основе спецификации требований, так и на основе структуры реализации;
- давать основу для трассируемости всех артефактов процессов разработки и тестирования на основе единой системы спецификационных сущностей;
- интегрировать на единой концептуальной и технологической базе различные техники верификации, включая дедуктивную верификацию, верификацию моделей (model checking) и тестирование, в том числе тестирование на основе традиционных, привычных инженерам методов.

Опыт разработки и общественного обсуждения нового стандарта позволили обобщить требования к инструментам для динамической верификации СЗИ промышленных программных систем. В частности, важной возможностью таких инструментов должны быть средства, которые позволяют не только генерировать и выполнять тесты, но и работать с формальными моделями и спецификациями разных уровней абстракции, поддерживать сбор и анализ трассовой информации и соотносить ее с требованиями спецификаций, уровень абстракции которых может существенно отличаться от собранной трассовой информации. Кроме того, изложенные в стандарте рекомендации прошли апробацию при тестировании на соответствии МРОСЛ ДП-модели механизмов управления доступом отечественной ОС Astra Linux.

Работа над группой стандартов ГОСТ Р 59453 будет продолжена, соответственно будут продолжены работы по расширению инструментальной поддержки процессов и технологий, которые рекомендуются данными стандартами. В направлении динамической верификации СЗИ инструмент АНИС будет развиваться для обеспечения возможности мониторинга работы ОС под рабочей нагрузкой. Сложной задачей является верификация ФСП на соответствие требованиям ФМД. Как уже отмечалось в монографии [19] приводится один из способов доказательств такого соответствия, но применять его на практике затруднительно. В настоящее время разрабатывается альтернативный вариант техники такой верификации. Запланирована разработка соответствующего стандарта ГОСТ Р 59453. Часть 5. Также ведутся работы по созданию инструментальной поддержки предлагаемой техники. Еще одним направлением работ по верификации СЗИ на основе формальных спецификаций является создание набора инструментов на основе широко известных фреймворков тестирования, например, PyTest. Это позволит построить удобную комбинацию привычных средств разработки тестов и системного подхода к анализу полноты тестов на основе строгих функциональных требований ФМД и ФСП.

Благодарности. Авторы выражают благодарность всем, принявшим участие в обсуждении и улучшении редакций разработанных стандартов. Особый вклад в работу внесла Ольга Валентиновна Кулагина ФАУ «ГНИИИ ПТЗИ ФСТЭК России».

Список литературы / References

- [1]. ГОСТ Р 59453.4-2025 «Защита информации. Формальная модель управления доступом. Часть 4. Рекомендации по верификации средства защиты информации, реализующего политики управления доступом, на основе формализованных описаний модели управления доступом». М.: Стандартинформ, 2025. 20 с. / GOST R 59453.4-2025 «Information protection. Formal access control model. Part 4. Recommendations for verification of information security features that implement access control policies based on formal descriptions of the access control model», 2025 (in Russian).
- [2]. ГОСТ Р 59453.1-2021 «Защита информации. Формальная модель управления доступом. Часть 1. Общие положения». М.: Стандартинформ, 2021. 16 с. / GOST R 59453.1-2021 «Information protection. Formal access control model. Part 1. General principles», 2021 (in Russian).
- [3]. ГОСТ Р 59453.2-2021 «Защита информации. Формальная модель управления доступом. Часть 2. Рекомендации по верификация формальной модели управления доступом». М.: Стандартинформ, 2021. 12 с./ GOST R 59453.2-2021 «Information protection. Formal access control model. Part 2. Recommendations on verification of formal access control model», 2021 (in Russian).
- [4]. ГОСТ Р 59453.3-2025 «Защита информации. Формальная модель управления доступом. Часть 3. Рекомендации по верификации средства защиты информации, реализующего политики управления доступом, на основе формализованных описаний модели управления доступом». М.: Стандартинформ, 2025. 20 с. / GOST R 59453.3-2025 «Information protection. Formal access control model. Part 3. Recommendations on development», 2025 (in Russian).
- [5]. DO-178C - Software Considerations in Airborne Systems and Equipment Certification Software Considerations in Airborne Systems and Equipment Certification (RTCA DO178C), 2011 https://my.rtca.org/nc_store?search=do-178c
- [6]. КТ-178С – Квалификационные требования. Часть 178 «Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники».
- [7]. П.Н. Девянин, С.С. Жиликов, А.И. Смирнов. Тестирование подсистемы безопасности ОС Astra Linux на основе формализованного описания модели управления доступом, том 37, вып. 4, 2025. / Devyanin P.N., S.S. Zhiliakov, A.I. Smirnov. Testing the Astra Linux OS security subsystem based on a formalized description of the access control model. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 4, 2025 (in Russian).
- [8]. Операционная система специального назначения Astra Linux Special Edition. Доступно по ссылке: <https://astra.ru/software-services/os/>, 17.04.2025. / Astra Linux Special Edition operating system. Available at: <https://astra.ru/software-services/os/>, accessed 17.04.2025.
- [9]. Abrial, Jean-Raymond. *Modeling in Event-B: system and software engineering*.- Cambridge University Press, 2010.
- [10]. Девянин П.Н., Леонова М.А. Приемы по доработке описания модели управления доступом ОССН Astra Linux Special Edition на формализованном языке метода Event-B для обеспечения ее автоматизированной верификации с применением инструментов Rodin и ProB // Прикладная дискретная математика. 2021. № 52. С. 83-96. / P. N. Devyanin, M. A. Leonova, “The techniques of formalization of OS Astra Linux Special Edition access control model using Event-B formal method for verification using Rodin and ProB”, *Prikl. Diskr. Mat.*, 2021, no. 52, pp. 83–96 (In Russian).
- [11]. Девянин П.Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Учебное пособие для вузов. 3-е изд., перераб. и доп. М.: Горячая линия – Телеком, 2020. 352 с.: ил. / P.N. Devyanin. *Security models of computer systems. Control for access and information flows*. Hotline-Telecom, 2020, 352 p. (in Russian).
- [12]. Michael Felderer, Philipp Zech, Ruth Breu, Matthias Büchler, Alexander Pretschner. Model-Based Security Testing: A Taxonomy and Systematic Classification.- *Software Testing Verification and Reliability* 00(2), July 2015:1-29, DOI:10.1002/stvr.1580.
- [13]. Schieferdecker, Ina & Großmann, Jürgen & Schneider, Martin. (2012). Model-Based Security Testing. *Electronic Proceedings in Theoretical Computer Science*. A.K. Petrenko, H. Schlingloff (Eds.): Workshop on Model-Based Testing 2012 (MBT 2012), EPTCS 80, 2012, <https://doi.org/10.4204/EPTCS.80>
- [14]. Krishnan, P., Pari-Salas, P. (2009). Model-Based Testing and the UML Testing Profile. In: Palsberg, J. (eds) *Semantics and Algebraic Specification*. Lecture Notes in Computer Science, vol 5700. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04164-8_16/
- [15]. Pierre-Alain Masson, Jacques Julliand, Jean-Christophe Plessis, Eddie Jaffuel, Georges Debois. Automatic generation of model based tests for a class of security properties.- *A-MOST '07: Proceedings of the 3rd*

- international workshop on Advances in model-based testing, Pages 12-22, <https://doi.org/10.1145/1291535.1291537>
- [16]. The B-Book: Assigning Programs to Meanings, Jean-Raymond Abrial, Cambridge University Press, 1996. ISBN 0-521-49619-5.
- [17]. Tejeddine Mouelhi, Franck Fleurey, Benoit Baudry, Yves Le Traon. A Model-Based Framework for Security Policy Specification, Deployment and Testing. - K. Czarnecki et al. (Eds.): MoDELS 2008, LNCS 5301, pp. 537–552
- [18]. OASIS eXtensible Access Control Markup Language (XACML) TC | OASIS. www.oasis-open.org.
- [19]. Девянин П.Н., Ефремов Д.В., Кулямин В.В., Петренко А.К., Хорошилов А.В., Щепетков И.В. Моделирование и верификация политик безопасности управления доступом в операционных системах. М.: Горячая линия – Телеком, 2019. 214 с.: ил. / P.N. Devyanin, D.V. Efremov, V.V. Kuliamin, A.K. Petrenko, A.V. Khoroshilov. Modeling and verification of access control access policies in operating systems. Hotline-Telecom, 2019, 214 p. (in Russian).
- [20]. L. Lamport. Specifying Systems, The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley, 2002.
- [21]. Козачок В.И., Козачок А.В., Кочетков Е.В. Многоуровневая модель политики безопасности управления доступом операционных систем семейства Windows.- Вопросы кибербезопасности, 2021, № 1(41), DOI: 10.21681/2311-3456-2021-1-41-56.
- [22]. Event-B and the Rodin Platform, <https://www.event-b.org/>.
- [23]. Е.В. Корныхин, Д.В. Ефремов, А.В. Карнов, А.К. Петренко. Инструмент АНИС - поддержка процессов верификации средств защиты информации на основе формальных моделей управления доступом.- Труды ИСП РАН, 2025, том 37, вып. 4.

Информация об авторах / Information about authors

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

Alexander Konstantinovich PETRENKO – Dr. Sci. (Phys.-Math.), Prof., Head of Software Engineering Department of the Ivannikov Institute for System Programming, Russian Academy of Sciences, Professor of the Department of System Programming, Faculty of Computational Mathematics and Cybernetics, Moscow State University and the Faculty of Computer Science, National Research University Higher School of Economics. His research interests include formal methods of software engineering, specification and modeling languages, and their use in software development and verification.

Петр Николаевич ДЕВЯНИН – член-корреспондент Академии криптографии России, доктор технических наук, профессор, научный руководитель ООО "РусБИТех-Астра" («Группа Астра»). Область интересов: теория информационной безопасности, формальные модели безопасности компьютерных систем, разработка безопасного программного обеспечения, операционные системы семейства Linux.

Petr Nikolaevich DEVYANIN – Dr. Sci. (Tech.), corresponding member of Russian Academy of Cryptography, professor, scientific director in RusBITech-Astra (Astra Linux). Field of Interest: information security theory, formal security models of computer systems, secure software development, operating systems of Linux family.

Денис Валентинович ЕФРЕМОВ – научный сотрудник. Сфера научных интересов: формальная верификация, статический и динамический анализ.

Denis Valentinovich EFREMOV – researcher. Research interests: formal verification, static and dynamic analysis.

Алексей Александрович КАРНОВ – научный сотрудник. Научные интересы: формальные спецификации, верификация и тестирование, статический и динамический анализ.

Aleksei Aleksandrovich KARNOV – researcher. Research interests: formal specifications, verification and testing, static and dynamic analysis.

Евгений Валерьевич КОРНЫХИН – кандидат физико-математических наук, доцент кафедры системного программирования МГУ, старший научный сотрудник ИСП РАН. Область интересов: формальная дедуктивная верификация моделей, тестирование на основе моделей.

Eugeny Valerievich KORNYKHIN – Cand. Sci. (Phys.-Math.), Associate Professor of system programming departments at MSU and Senior Researcher at ISP RAS. Fields of Interest: formal deductive verification, model-based testing.

Виктор Вячеславович КУЛЯМИН – кандидат физико-математических наук, ведущий научный сотрудник ИСП РАН, доцент кафедр системного программирования МГУ и ВШЭ. Область интересов: формальная дедуктивная верификация моделей, тестирование на основе моделей.

Viktor Vyacheslavovich KULIAMIN – Cand. Sci. (Phys.-Math.), leading researcher at ISP RAS, associate professor of system programming departments at MSU and the HSE. Fields of Interest: formal deductive model verification, model-based testing.

Алексей Владимирович ХОРОШИЛОВ – кандидат физико-математических наук, ведущий научный сотрудник, директор Центра верификации ОС Linux в ИСП РАН, доцент кафедр системного программирования МГУ, ВШЭ и МФТИ. Основные научные интересы: методы проектирования и разработки ответственных систем, формальные методы программной инженерии, методы верификации и валидации, тестирование на основе моделей, методы анализа требований, операционная система Linux.

Alexey Vladimirovich KHOROSHILOV – Cand. Sci. (Phys.-Math.), Leading Researcher, Director of the Linux OS Verification Center at ISP RAS, Associate Professor of System Programming Departments at MSU, HSE, and MIPT. Main research interests: design and development methods for critical systems, formal methods of software engineering, verification and validation methods, model-based testing, requirements analysis methods, Linux operating system.



Поиск утечек памяти и ресурсов в статическом анализаторе Svace

^{1,2} Н.Е. Малышев, ORCID: 0000-0002-5245-4915 <neket@ispras.ru>

¹ А.Е. Бородин, ORCID: 0000-0003-3183-9821 <alexey.borodin@ispras.ru>

^{1,2} А.А. Белеванцев, ORCID: 0000-0003-2817-0397 <abel@ispras.ru>

^{1,3} В.А. Семенов, ORCID: 0000-0002-8766-8454 <sem@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.

² Московский государственный университет имени М.В. Ломоносова,
119991, Россия, г. Москва, Ленинские горы, д. 1.

³ Московский физико-технический институт,
141701, Россия, Московская область, г. Долгопрудный, Институтский пер., 9.

Аннотация. В статье представлен метод поиска утечек памяти и других схожих ресурсов в статическом анализаторе Svace. Формулируются требования к статическому анализатору, представляющие стоящую за Svace философию, кратко описывается основная инфраструктура анализа на основе межпроцедурного символьного выполнения с объединением состояний и показывается, как ее можно применить для поиска утечек. Описываются атрибуты, которые необходимо вычислить в ходе анализа, демонстрируется, как они вычисляются, как используются спецификации функций выделения и освобождения памяти, а также как учитываются участки памяти, выходящие из-под контроля анализатора. Предлагается способ учета отсутствующих в исходном коде ограничений на выделение и освобождение памяти с помощью создания искусственных функций. Представляются экспериментальные результаты работы метода на наборе тестов Juliet и открытом пакете Binutils, показывающие жизнеспособность предлагаемых идей.

Ключевые слова: статический анализ; символьное выполнение; утечки памяти; чувствительность к путям; анализ потока данных; анализ указателей.

Для цитирования: Малышев Н.Е., Бородин А.Е., Белеванцев А.А., Семенов В.А. Поиск утечек памяти и ресурсов в статическом анализаторе Svace. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 291–302. DOI: 10.15514/ISPRAS-2025-37(3)-20.

Detecting Memory and Resource Leaks in the Svace Static Analyzer

^{1,2} N.E. Malyshev, ORCID: 0000-0002-5245-4915 <neket@ispras.ru>

¹ A.E. Borodin, ORCID: 0000-0003-3183-9821 <alexey.borodin@ispras.ru>

^{1,2} A.A. Belevantsev, ORCID: 0000-0003-2817-0397 <abel@ispras.ru>

^{1,3} V.A. Semenov, ORCID: 0000-0002-8766-8454 <sem@ispras.ru>

¹ *Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

² *Lomonosov Moscow State University
1 Leninskie Gory, Moscow, 119991 Russia.*

³ *Moscow Institute of Physics and Technology (National Research University)
9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.*

Abstract. The paper presents an approach to detecting memory and other resource leaks in the Svace static analyzer. We lay down a set of static analysis requirements that show the philosophy behind Svace, briefly describe the main analysis infrastructure based on an interprocedural symbolic execution with state merging, and show how this infrastructure can be applied to leak detection. We list attributes that are computed during analysis and present how this computation is performed, how allocation and release functions are modeled via specifications, how escaped memory is accounted for. We propose a way to provide external information not present in the source code to the analyzer via creating artificial functions. Experimental results on the Juliet test suite and the Binutils package show the viability of our ideas.

Keywords: static analysis; symbolic execution; memory leak; path sensitivity; dataflow analysis; points-to analysis.

For citation: Malyshev N.E., Borodin A.E., Belevantsev A.A., Semenov V.A. Detecting Memory and Resource Leaks in the Svace static analyzer. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025. pp. 291-302 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-20.

1. Введение

Утечки памяти – это вид ошибки, часто встречающийся в программах на языках с ручным управлением памятью. Из-за того, что при компиляции утечки не приводят к ошибкам, а при выполнении – к немедленному аварийному завершению программы, они нередко остаются незамеченными и присутствуют в современных программах на C/C++. При этом в языках со сборкой мусора актуальность приобретает другой тип утечек – утечки ресурсов: файлов, сетевых соединений, графических дескрипторов и пр.

В связи с этим для статических анализаторов утечки памяти и ресурсов – неизменная тема исследований. Традиционным подходом к поиску утечек является анализ потока данных, обычно выраженный в виде анализа распространения информации между источниками и стоками (source-sink). Именно это направление, как правило, и развивается в настоящее время; из исключений можно назвать инструменты сепарационной и некорректной логики [1, 2]. Среди важных инструментов можно упомянуть работы [3-7]. Их особенность, однако, заключается в том, что они предлагают специфичные алгоритмы поиска утечек, не применимые к другим анализам.

Несмотря на научную ценность таких алгоритмов, их применение неизменно влечет за собой следующие сложности: во-первых, для реализации предложенных алгоритмов в рамках уже развивающейся системы многоязыкового, промышленного статического анализа требуется адаптация, часто непростая; во-вторых, сами по себе такие инструменты могут показывать результаты хуже, чем поиск утечек, пусть несложный, но сделанный на инфраструктуре развитого анализатора – на практике часто именно качество инфраструктуры вносит определяющее значение в результаты. Как следствие, бывает тяжело сравнить

опубликованные алгоритмы между собой, т.к. честное сравнение предполагает их реализацию в одной и той же базовой инфраструктуре.

В статье мы описываем подход к поиску утечек ресурсов, применяющийся в статическом анализаторе Svace [8, 9]. На протяжении последних десяти лет развивались и претерпевали изменения как инфраструктура анализатора по организации чувствительного к путям анализа методами символического выполнения, так и методы моделирования памяти. В разделе 2 мы коротко опишем постановку задачи и ожидаемые свойства статического анализатора, которые, по нашему мнению, позволят успешно применить его для поиска утечек. Мы также прокомментируем наличие или отсутствие этих свойств на примере нескольких открытых анализаторов. В разделе 3 изложим алгоритм поиска утечек, начав с краткого изложения организации используемых им анализов в инструменте Svace. За подробным разбором необходимой для нас архитектуры Svace мы отсылаем читателя к предыдущей статье [10]. Здесь же сконцентрируемся на описании алгоритма поиска утечек и полученных экспериментальных результатов, которые представлены в разделе 4. Раздел 5 содержит заключение и планируемые в дальнейшем исследования.

2. Утечки ресурсов и статический анализ

Утечкой ресурсов называется ситуация, при которой некоторый ресурс ограниченного объема (распределяемый через некую систему управления доступом) был получен, но не был возвращен системе обратно. Самым частым примером такого ресурса является память (динамическая); другие примеры – файлы, сетевые соединения и соединения к базам данных, графические объекты. Система управления предполагает наличие явных точек (вызовов), в которых ресурс выделяется и освобождается. В противном случае утечки не возникает (например, в языках со сборкой мусора не бывает утечек памяти).

Обычно ресурс управляется через некоторый дескриптор (указатель на память, файл и пр.). Тогда утечка регистрируется в тот момент, когда все дескрипторы (ссылки на ресурс) вышли из области действия, или остались только рекурсивные ссылки без внешних ссылок на цикл, при этом ресурс не был освобожден.

Для успешной разработки и реализации алгоритмов статического анализа, ищущих утечки памяти и ресурсов, необходим анализатор с определенной инфраструктурой. Требования к этой инфраструктуре формируются, исходя из особенностей современных программ, в которых можно выделить три основных момента: как написаны программы (размеры, используемые свойства языков); как они разрабатываются (особенности современных систем непрерывной интеграции); какая информация доступна анализатору (весь ли исходный код или его часть, некоторая внешняя информация о программе). Все они так или иначе освещались в литературе, поэтому здесь перечислим их кратко с пояснениями в самых важных моментах.

- 1. Чувствительность анализа к потоку, полям, путям и контексту.** Необходимость чувствительности к потоку не требует пояснений. Чувствительность к полям, как правило, не требует существенных затрат ресурсов, но в нее часто включается и чувствительность к элементам массива, которой можно добиться, например, путем отслеживания нескольких первых и последних элементов, а для циклов выражая свойства элементов массива через аффинные итерационные функции.

Масштабируемая чувствительность к контексту достигается за счет рассмотрения ограниченного количества контекстов вызова некоторой функции за счет либо группировки контекстов (по стеку вызовов, типу объектов) либо создания резюме функции (summary), хранящего релевантные результаты анализа для группы контекстов вызова (в идеале – параметризованного таким образом, что единственное резюме хранит результаты для всех возможных контекстов). Искусство выстраивания межпроцедурного анализа заключается в определении компромисса

между размером резюме, сложностью его применения и количеством сохраняемых для функции данных.

Чувствительность к путям выполнения реализуется анализатором за счет либо интерпретации некоторого фиксированного количества путей в каждой функции, либо с помощью символического выполнения с объединением состояний. В первом случае есть риск пропустить ошибку, если исключить из рассмотрения путь, на котором она возникает, зато сложность анализа растет линейно относительно длины пути; во втором случае с ростом длины путей растет и сложность формул, а их упрощение также неизбежно приводит к потере ошибок.

2. **Точность и полнота анализа.** Если классические исследования, в частности, алгоритмов абстрактной интерпретации, предполагали консервативные анализы, в которых полнота (нахождение всех возможных ошибок) предпочиталась точности (отсутствию ложных срабатываний), то на практике, по нашему мнению, такой компромисс оправдан лишь для критических с точки зрения безопасности областей (авионики, атомной промышленности и пр.). Для больших программных систем общего назначения нормой стало допускать уменьшение полноты с целью увеличения точности, т.е. повышать процент истинных срабатываний в том числе за счет пропуска реальных ошибок, оговаривая контексты, в которых таковой пропуск возможен [11].
3. **Размеры обрабатываемых программ и циклы разработки в системах непрерывной интеграции.** Регулярное применение статического анализа в жизненном цикле разработки к программам в миллионы строк кода требует выстраивания упомянутых выше в разделе алгоритмов и свойств анализа таким образом, чтобы их масштабируемость позволяла анализировать программы за ночную сборку. Также важно иметь возможность организовывать инкрементальный анализ изменившихся частей программы для быстрого анализа каждого изменения (этот вопрос выходит за рамки статьи).
4. **Объяснение пользователю выдаваемого предупреждения об ошибке.** Программист или инженер безопасности, анализирующий результаты инструмента, должен справляться с определением истинности предупреждений с учетом, во-первых, упомянутых выше существенных размеров программ и соответствующего количества предупреждений, а во-вторых, того, что сам он может не являться экспертом в используемых языках программирования или нюансах кибербезопасности. При выдаче предупреждений разработчики статического анализатора не должны рассчитывать на схожий уровень знаний пользователя [12].
5. **Неявные контракты (предусловия и постусловия).** Практически постоянно анализируемая программа используется в неизвестных анализатору контекстах – по причине неполноты имеющегося кода, неточности анализа на большом объеме этого кода, либо тогда, когда информация о контекстах не содержится в самом коде. В этом случае анализатор должен предоставлять способы описать эту дополнительную информацию, либо в виде *спецификаций* функций или переменных, либо в виде конфигурируемых пользователем эвристических алгоритмов.

Последние три соображения, которые связаны больше со стратегией разработки анализатора, чем с самими алгоритмами анализа, приводят к следующим важным выводам:

- Анализатор может **не выдавать** даже истинные с точки зрения модели программы, памяти и алгоритмов анализа предупреждения, если 1) такие предупреждения не получается внятно объяснить пользователю на основе собранных данных или 2) вероятно существование неизвестных ограничений на контексты вызова, которые сделают эти предупреждения бесполезными для программиста.

- Анализатору стоит **выдавать** предупреждения не только об истинных с точки зрения его информации ошибках, но и о местах *потенциальных* ошибок, которые нереализуемы с учетом только известного кода программы, но в будущем в ходе разработки могут стать реальными ошибками (в частности, именно поэтому наш анализатор выдает ошибки в том числе в не используемом на данный момент коде либо в случаях, когда участок кода с местами возникновения и проявления ошибки полностью заключен в недостижимой области кода).

2.1 Схожие работы

Отсылая читателя к недавней работе по сравнению инструментов поиска утечек [13], кратко рассмотрим ряд известных открытых инструментов с учетом сказанного выше в разделе и нашей цели построить алгоритм, пригодный для использования в уже имеющемся промышленном статическом анализаторе.

Самым известным открытым анализатором языков C/C++ является Clang Static Analyzer [14], в котором реализовано символьное выполнение (однако без объединения состояний), межпроцедурный анализ (на основе встраивания); в отдельном компоненте есть вариант межмодульного анализа, также с ограниченной масштабируемостью. Тем самым анализатор удовлетворяет части выдвинутых требований по алгоритмам анализа, однако имеет сложности с масштабируемостью и, как следствие, применимостью для больших программ в регулярном цикле разработки.

Инструмент Smoke [6] находит исключительно утечки, причем масштабируемость чувствительного к путям анализа достигается за счет разреженных графовых представлений сущностей программы, которые используются для хранения лишь необходимой информации для поиска потенциальных утечек, и дальнейшей проверки совместности условий вдоль найденного пути с помощью SMT-решателя. Разреженные представления помогают добиться масштабируемости, однако плохо подходят для универсального статического анализатора, который поддерживает сотни детекторов и тем самым вынужден рассматривать с той или иной целью все инструкции программы.

Анализатор Sparrow [7] демонстрирует хорошую масштабируемость за счет использования резюме для сохранения результатов внутрипроцедурного анализа (параметризуются пути к точкам выделения, использования и освобождения памяти), однако его точность страдает из-за отсутствия чувствительности к путям выполнения.

Инструмент Infer [15] является широко известным примером использования сепарационной логики [1] для моделирования памяти. Infer выдавал предупреждения об ошибках тогда, когда не мог доказать их отсутствие, тогда как применение в дальнейшем некорректной логики [2] позволило перейти к выдаче ошибок в том случае, когда имеются основания подозревать их присутствие. Такая стратегия используется в Pulse, следующей версии Infer, поиск утечек памяти в которой присутствует, однако официально еще не выпущен.

3. Поиск утечек ресурсов

Предлагаемый алгоритм поиска утечек опирается на существующие возможности анализатора Svace, за подробным описанием которых мы отсылаем читателя к статье [10]. Здесь же представим очень краткое резюме всей необходимой инфраструктуры.

Анализатор Svace осуществляет межпроцедурный контекстно-чувствительный анализ на основе резюме, а внутрипроцедурный анализ – на основе символьного выполнения с объединением состояний. Внутреннее представление Svace IR содержит обычные арифметические и логические операции, инструкции cast, load/store, инструкции split и join для отображения потока управления, инструкции call и annotate для вызова и создания резюме, инструкцию assume для записи предусловий данного пути. Символьное выполнение

осуществляется над атрибутами – базовой единицей информации анализа, представляющей из себя, по сути, пару «ключ-значение» (примеры атрибутов – интервал значений, двоичный или троичный флаг, условие выполнения некоторого факта). Таким образом, символическое хранилище содержит соответствие не переменных и символьных выражений, а ключей и значений атрибутов. Кроме того, значения одних атрибутов могут выступать в качестве ключей других атрибутов. Объединение состояний происходит при обработке инструкции `join`, для корректной обработки которой каждый тип значений атрибутов должен определять соответствующую операцию слияния. Таким образом, степень чувствительности к путям и полноты получаемых формул может контролироваться на этом этапе через специфику операций слияния. Циклы анализируются путем раскрытия нескольких первых, абстрактной «средней» и последней итераций (в будущем можно использовать резюме циклов [16] для уточнения анализа).

Пример символического выполнения над атрибутами показан на рис. 1. Слева на рисунке классическое символическое выполнение, связывающее символические выражения с переменными `a` и `x`; после разделения потока управления появляются инструкции `assume`, и в переменную `a` записываются результаты вызова `B()` или `C()`. После объединения состояний значение, содержащееся по указателю `a`, выражается через тернарный оператор (дизъюнкцию). Справа ссылки, соответствующие переменным, связываются с атрибутами, обновляющими свои значения после разделения потока; из резюме функций `B()` и `C()` получаем условия выделения памяти под `b` и `c` ($x > 2$ и $x < 3$). При слиянии состояний эти условия пересекаются с предикатом пути и объединяются, давая финальную формулу выделения памяти, записанной в `a`.

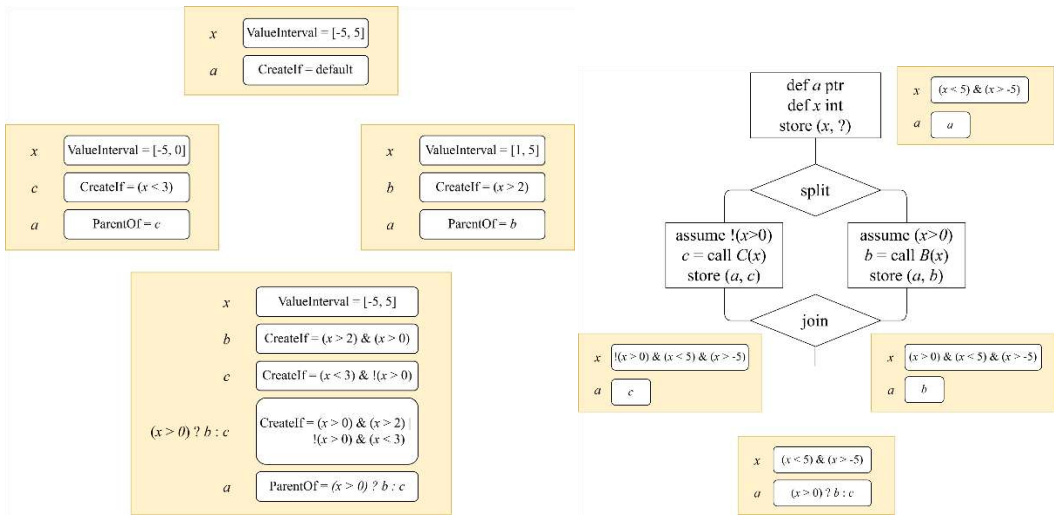


Рис. 1. Символическое выполнение над атрибутами.

Fig. 1. Symbolic execution with attributes.

Резюме функции, служащее передаточной функцией при обработке инструкции вызова, представляет из себя часть символического хранилища, содержащего вычисленные атрибуты для аргументов функции, глобальных переменных и возвращаемого значения. На этом этапе формулы упрощаются и обрезаются для обеспечения масштабируемости. При применении резюме в формулы вместо формальных подставляются фактические параметры. Резюме неизвестных анализатору функций конструируются в ходе обработки спецификаций функций, в которых на анализируемом языке с помощью интерфейсов Svace описывается необходимая анализу семантика.

Выдача *предупреждений* происходит после того, как при обработке инструкций детектор проверяет условие ошибки и убеждается в его выполнимости. При необходимости условие транслируется в запрос стандарта SMT-LIB2 и проверяется на совместность SMT-решателем. Эффективное решение специфичных для статического анализа запросов исследовалось нами в работе [17].

3.1 Алгоритм анализа для обнаружения утечек

Вспомнив определение утечек из раздела 1, прежде всего заметим, что Svace не оперирует специальными сущностями для представления ячеек памяти, как на ранних этапах развития инструмента; вместо этого каждой отслеживаемой ячейке памяти соответствует два выражения над значениями, обозначающих ссылку и ее содержимое. Отношение между ссылкой и содержимым также может рассматриваться как атрибут и является объектом для символического выполнения. Срез графа атрибутов, содержащий только подобные отношения между значениями, будем для краткости называть *графом значений*. Для сложных структур данных – записей и массивов – хранится базовая ссылка и смещение; в случае больших массивов, как и для циклов, моделируется несколько первых и последних элементов, а также абстрактный «средний».

Утечка определяется в тот момент, когда при выходе ссылки (дескриптора) из области действия не осталось других достижимых ссылок на указываемый ресурс. Так происходит либо при выходе из функции, либо при записи по ссылке нового значения инструкциями store или call, поэтому утечки проверяются при обработке детектором этих инструкций. Достижимые ссылки определяются обходом графа значений, содержащего в том числе указательные отношения (points-to).

Для тех ссылок, что оказались недостижимы, необходимо определить, был ли выделен и впоследствии освобожден ресурс. Мы вводим два атрибута, CreateIf и ReleaseIf, которые соответствуют условиям выделения и освобождения. Предупреждение выдается, если формула, состоящая из конъюнкции CreateIf, отрицания ReleaseIf, предиката пути и некоторых глобальных условий, оказывается совместной. SMT-решатель в таком случае предоставляет модель, используя которую, строится пример утечки. Из основной инфраструктуры Svace используется и добавляется в глобальные условия атрибут ValueInterval, позволяющий точнее учесть арифметические операции в условиях пути, а также результаты анализа недостижимого кода. В случае, когда анализ обладает неполной информацией о программе (из-за обрезания формул или резюме, или из-за специфичных свойств языка), вычисляется отдельный атрибут EscapeValue, и при его наличии предупреждение не выдается, чтобы избежать дополнительных ложных срабатываний.

Опишем более подробно устройство необходимых анализу атрибутов:

- **Анализ указательных отношений** выполняется через граф значений, где ключом служат выражения для ссылок, а значением – выражения для значений в памяти. Слияние выполняется через дизъюнкцию (классическое объединение состояний). В начале функции создаются ссылки для параметров, при обработке вызова – для возвращаемого значения и изменившихся параметров; в инструкции load связывается ссылка и указываемое значение, а в инструкции store – обновляется связь ссылки с новым значением.
- Атрибут **CreateIf** отвечает за анализ выделения памяти и связывает указательные значения с условиями, при выполнении которых для них выделена память. При обработке вызова, если произошло выделение новой памяти для возвращаемого значения или аргументов, значение атрибута переносится из резюме в контекст вызывающей функции. Спецификации функций выделения памяти (malloc, new и пр.) используют специальный вызов интерфейса анализатора sf_new (т.н. спецфункцию), которая устанавливает атрибут в истину.

- Атрибут **ReleaseIf** отвечает за анализ освобождения памяти и связывает указательные значения с условиями, при выполнении которых память для этих значений освобождается. Аналогично, при обработке вызова при освобождении памяти аргументов или возвращаемого значения значение атрибута переносится из резюме в контекст вызывающей функции. Спецификации функций освобождения памяти используют спецификацию `sf_delete`, чтобы установить атрибут в истину.
- Атрибут **ValueInterval** из основной инфраструктуры Svace связывает символьные выражения с целочисленными интервалами. Значения атрибута сливаются и обрабатываются передаточными функциями арифметических и битовых операций обычным для интервального домена способом [18]; соответственно, возможна потеря точности при слиянии из-за того, что результат операции должен быть представлен единственным интервалом. Тем не менее, для значения, являющегося результатом слияния, в графе значений можно найти составные части и при необходимости восстановить точность. При обработке инструкции `assume` интервал модифицируется с учетом условия из инструкции, а при обработке вызова переносится в контекст вызывающей функции.
- Атрибут **EscapeValue** устанавливается для тех выражений, которые «выходят» из-под контроля анализа, и становится невозможно или не нужно определять, были ли они освобождены. В частности, при обработке вызова он устанавливается для возвращаемых значений и выражений, на которые указывают аргументы, в случае, если вызываемая функция неизвестна анализу. Также может устанавливаться спецификацией `sf_escape`, что используется в ряде библиотечных функций (например, `realloc`).

Отдельно нужно упомянуть метод, используемый для анализа утечек в случае ограничений, не следующих напрямую из исходного кода. Например, в языке C++ деструктор должен освобождать выделенную в конструкторе память, и если этого не происходит, мы сочтем такую ситуацию ошибочной, даже если в видимом анализаторе участке программы он не реализуется (вспомним требования к анализу и выводы, обсуждавшиеся в разделе 2). Общий метод представления таких ограничений и соответствующего анализа является предметом отдельной статьи; здесь же заметим, что в контексте утечек мы создаем искусственные функции, задающиеся некоторой парой функций: сначала в искусственную функцию вставляется тело первой функции пары, затем – вызов второй функции пары. Мы создаем функции для пар «конструктор-деструктор», «конструктор-оператор присваивания», «оператор присваивания-оператор присваивания» и «оператор присваивания-деструктор». Эти искусственные функции анализируются обычным образом в конце анализа, т.к. они не вызываются и не требуют создания резюме. Если в ходе анализа обнаруживаются ошибки, мы сообщаем об утечке памяти в программе.

4. Экспериментальные результаты

Предложенный алгоритм поиска утечек реализован в анализаторе Svace для утечек памяти и ресурсов. Мы протестировали его на тестах из набора Juliet [19] для класса CWE-401 (Missing Release of Memory after Effective Lifetime), определив число истинных и ложных срабатываний. Для открытых проектов мы оценили результаты на операционной системе Tizen и пакете Binutils версии 2.24.

На тестах Juliet для языка C мы определили 868 функций как имеющие ошибку, из них успешно выдали 560 ошибок, а из безошибочных функций (которых всего 2266) выдали ложное срабатывание в 32-х (см. табл. 1). Уровень ложных срабатываний (FPR) вычислили как $FP / (FP + TN)$, а уровень пропусков (FNR) – как $FN / (FN + TP)$. Аналогично вычислили данные для языка C++ за тем исключением, что после ручного просмотра часть выданных

срабатываний были исключены из рассмотрения, так как несмотря на то, что они были выданы в тестах, помеченных как «хорошие» с точки зрения набора, на самом деле эти тесты содержали релевантную ошибку в другом месте (а именно, в парах конструктор «конструктор-оператор присваивания», описанных нами выше).

Для пакета Binutils мы оценивали результаты отдельно по каждому типу детектора утечки памяти, который реализован в рамках анализатора. Все варианты детектора используют одну и ту же инфраструктуру, описанную нами выше, но разнятся в стратегии выдачи и показываемых ситуациях – такое разделение упрощает работу с результатами детектора. Общее название всех вариантов – MEMORY_LEAK, среди них базовый детектор (без суффикса в имени) содержит наиболее надежные срабатывания, которые были дополнительно отфильтрованы набором эвристик. Детектор MEMORY_LEAK.EX содержит те срабатывания, для которых составленные формулы были признаны совместными, однако дополнительной фильтрации эвристиками предпринято не было. Варианты .PAIR и .CTOR служат для предупреждений, выданных при генерации искусственных функций, как описано в конце раздела 3. Варианты .STRUCT и .STRDUP содержат подмножества срабатываний основного детектора, выделенные для удобства работы и содержащие ситуации соответственно со структурами и функцией strdup.

Результаты разметки представлены в табл. 2. Выделены истинные и ложные срабатывания, а также срабатывания “won’t fix”, которые являются истинными, но, как правило, разработчики их не исправляют. Как видно, процент истинных срабатываний превысил 80%, что по сравнению с более ранними результатами статьи [10] объясняется улучшениями чувствительного к путям анализа в части отслеживания условий отдельных веток при слиянии имеющих отношение к утечкам атрибутов.

Табл. 1. Результаты на пакете Juliet CWE-401.

Table 1. Juliet CWE-401 results.

Язык	TP	TN	FP	FN	FPR	FNR
C	560	2234	32	308	1,4%	35,5%
C++	511	1929	35	279	1,8%	35,3%

Табл. 2. Результаты на пакете Binutils.

Table 2. Binutils results.

Детектор	Всего	TP	FP	WF
MEMORY_LEAK	20	16	4	0
MEMORY_LEAK.EX	50	45	5	0
MEMORY_LEAK.STRUCT	18	10	8	0
MEMORY_LEAK.STRDUP	0	0	0	0
MEMORY_LEAK.CTOR	0	0	0	0
MEMORY_LEAK.PAIR	0	0	0	0
Всего	88	71	17	0

На рис. 2 представлены два примера срабатываний на пакете Binutils. Они связаны с обработкой нескольких выделений памяти (проверяется лишь одно, код на рисунке справа) и с ошибочной организацией потока управления (код на рисунке слева). Остальные ошибки в этом пакете чаще всего связаны с отсутствием освобождения ранее выделенной памяти при

возникновении неожиданных ситуаций в функциях, при которых они не может продолжать свою работу.

Тестирование на исходном коде операционной системе Tizen [18], содержащей более сотни написанных на С и С++ пакетов, показало для предыдущей версии алгоритма утечек около 67% истинных срабатываний (включая “won’t fix”) [10]; разметка результатов последней версии реализации пока не проведена.

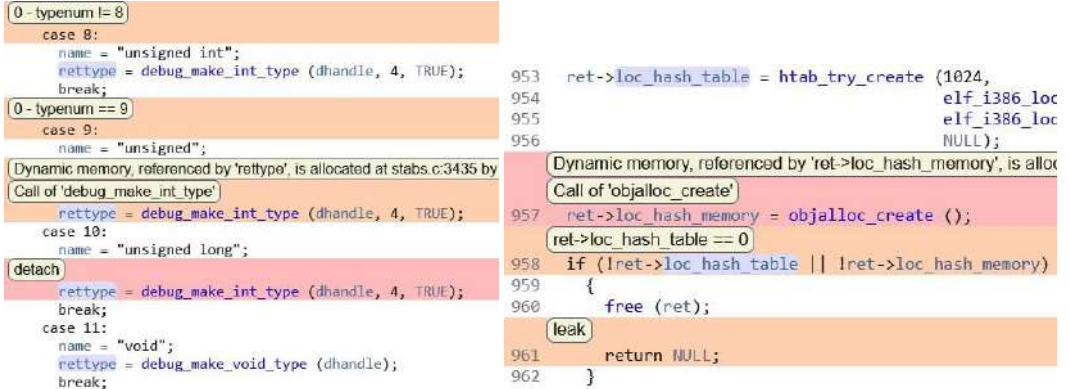


Рис. 2. Примеры ошибок из пакета Binutils.

Fig. 2. Binutils package error examples.

5. Заключение

В статье представлен подход к поиску утечек памяти и ресурсов на основе инфраструктуры промышленного статического анализатора Svace. Мы описали требования, которым, по нашему мнению, должен удовлетворять инструмент для успешного поиска ошибок в больших программах, и показали, как на его основе построить анализ для обнаружения утечек. Результаты тестирования нашей реализации на открытых пакетах показывают, что сформулированные в разделе 2 требования проходят проверку практикой. В дальнейшем мы планируем развивать подход к представлению в анализе внешней для программы информации, как показано на примере искусственных функций в разделе 3, и проводить эксперименты с резюме вложенных циклов и SMT-решателями, в том числе специализированными.

Список литературы / References

- [1]. J. C. Reynolds, "Separation logic: a logic for shared mutable data structures," Proceedings 17th Annual IEEE Symposium on Logic in Computer Science, Copenhagen, Denmark, 2002, pp. 55-74, doi: 10.1109/LICS.2002.1029817.
- [2]. Peter W. O'Hearn. Incorrectness logic. Proc. ACM Program. Lang. 4, POPL, Article 10 (January 2020), 32 pages. <https://doi.org/10.1145/3371078>
- [3]. Wen Li, Haipeng Cai, Yulei Sui, and David Manz. PCA: memory leak detection using partial call-path analysis. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020). Association for Computing Machinery, New York, NY, USA, 1621–1625. <https://doi.org/10.1145/3368089.3417923>
- [4]. Yulei Sui and Jingling Xue. SVF: interprocedural static value-flow analysis in LLVM. In Proceedings of the 25th International Conference on Compiler Construction (CC '16). Association for Computing Machinery, New York, NY, USA, 265–266.
- [5]. Yungbum Jung and Kwangkeun Yi. 2008. Practical memory leak detector based on parameterized procedural summaries. In Proceedings of the 7th international symposium on Memory management (ISMM '08). Association for Computing Machinery, New York, NY, USA, 131–140. <https://doi.org/10.1145/1375634.1375653> <https://doi.org/10.1145/2892208.2892235>

- [6]. G. Fan, R. Wu, Q. Shi, X. Xiao, J. Zhou and C. Zhang, "SMOKE: Scalable Path-Sensitive Memory Leak Detection for Millions of Lines of Code," 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), Montreal, QC, Canada, 2019, pp. 72-82, doi: 10.1109/ICSE.2019.00025.
- [7]. X. Sun et al., "A Projection-Based Approach for Memory Leak Detection," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 2018, pp. 430-435, doi: 10.1109/COMPSAC.2018.10271.
- [8]. A. Belevantsev, A. Borodin, I. Dudina, V. Ignatiev, A. Izbyshev, S. Polyakov, D. Zhurikhin. Design and development of Svsace static analyzers. In 2018 Ivannikov Memorial Workshop (IVMEM), Yerevan, Armenia, 2018, pp. 3-9, doi: 10.1109/IVMEM.2018.00008.
- [9]. Бородин А.Е., Белеванцев А.А. Статический анализатор Svsace как коллекция анализаторов разных уровней сложности. Труды Института системного программирования РАН. 2015;27(6):111-134.
- [10]. N. Malyshev, A. Borodin and A. Belevantsev, "A Comprehensive Approach to Finding Resource Leaks via Static Analysis," 2024 Ivannikov Ispras Open Conference (ISPRAS), Moscow, Russian Federation, 2024, pp. 1-9, doi: 10.1109/ISPRAS64596.2024.10899157.
- [11]. Benjamin Livshits, Manu Sridharan, Yannis Smaragdakis, Ondřej Lhoták, J. Nelson Amaral, Bor-Yuh Evan Chang, Samuel Z. Guyer, Uday P. Khedker, Anders Möller, and Dimitrios Vardoulakis. In defense of soundness: a manifesto. Commun. ACM 58, 2 (February 2015), 44–46. <https://doi.org/10.1145/2644805>
- [12]. Al Bessey, Ken Block, Ben Chelf, Andy Chou, Bryan Fulton, Seth Hallem, Charles Henri-Gros, Asya Kamsky, Scott McPeak, and Dawson Engler. A few billion lines of code later: using static analysis to find bugs in the real world. Commun. ACM 53, 2 (February 2010), 66–75. <https://doi.org/10.1145/1646353.1646374>
- [13]. H. Aslanyan, Z. Gevorgyan, R. Mkoyan, H. Movsisyan, V. Sahakyan and S. Sargsyan, "Static Analysis Methods For Memory Leak Detection: A Survey," 2022 Ivannikov Memorial Workshop (IVMEM), Moscow, Russian Federation, 2022, pp. 1-6, doi: 10.1109/IVMEM57067.2022.9983955.
- [14]. "Clang Static Analyzer." [Online]. Available: <https://clanganalyzer.lvm.org/>
- [15]. Calcagno, C. et al. (2015). Moving Fast with Software Verification. In: Havelund, K., Holzmann, G., Joshi, R. (eds) NASA Formal Methods. NFM 2015. Lecture Notes in Computer Science(), vol 9058. Springer, Cham. https://doi.org/10.1007/978-3-319-17524-9_1
- [16]. C. Scherb, L. B. Heitz, H. Grieder, and O. Mattmann, "Divide, conquer and verify: Improving symbolic execution performance," arXiv preprint arXiv:2310.03598, 2023.
- [17]. N. Malyshev, I. Dudina, D. Kutz, A. Novikov and S. Vartanov, "SMT Solvers in Application to Static and Dynamic Symbolic Execution: A Case Study," 2019 Ivannikov Ispras Open Conference (ISPRAS), Moscow, Russia, 2019, pp. 9-15, doi: 10.1109/ISPRAS47671.2019.00008.
- [18]. Операционная система Tizen. [Online]. Available: <https://docs.tizen.org/platform/get-started/open-source-project/>

Информация об авторах / Information about authors

Никита Евгеньевич МАЛЫШЕВ – младший научный сотрудник ИСП РАН, ассистент кафедры системного программирования ВМК МГУ. Сфера научных интересов: статический анализ, SMT-решатели, оптимизации программ.

Nikita Evgenyevich MALYSHEV – researcher at ISP RAS, assistant at Chair for System Programming of Department of Computational Mathematics and Cybernetics at MSU. Research interests: static analysis, SMT solvers, program optimization.

Алексей Евгеньевич БОРОДИН – кандидат физико-математических наук, старший научный сотрудник ИСП РАН. Сфера научных интересов: статический анализ исходного кода программ для поиска ошибок.

Alexey Evgenevich BORODIN – Cand. Sci. (Phys.-Math.), researcher. Research interests: static analysis for finding errors in source code.

Андрей Андреевич БЕЛЕВАНЦЕВ – доктор физико-математических наук, ведущий научный сотрудник ИСП РАН, профессор кафедры системного программирования ВМК МГУ. Сфера

научных интересов: статический анализ программ, оптимизация программ, параллельное программирование.

Andrey Andreevich BELEVANTSEV – Dr. Sci. (Phys.-Math.), Prof., leading researcher at ISP RAS, Professor at MSU. Research interests: static analysis, program optimization, parallel programming.

Виталий Адольфович СЕМЕНОВ – доктор физико-математических наук, профессор, заведующий отделом системной интеграции и прикладных программных комплексов ИСП РАН с 2015 года. Сфера научных интересов: модельно-ориентированные методологии и инструменты программной инженерии для создания цифровых платформ и мультидисциплинарных программных комплексов, визуализация и компьютерная графика, технологии информационного моделирования в архитектуре и строительстве, проектное управление и календарно-сетевое планирование.

Vitaly Adolfovich SEMENOV – Dr. Sci. (Phys.-Math.), Prof., Head of the Department of System Integration and Multi-disciplinary Applied Systems of the ISP RAS since 2015. Research interests: model-driven methodologies and CASE toolkits for creating digital platforms and advanced applied systems, visualization and computer graphics, building information modeling, project management and scheduling.



Тестирование в жизненном цикле автоматизированных систем

Б.А. Позин, ORCID: 0000-0002-0012-2230 <bpozin@ec-leasing.ru>

*Институт системного программирования РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.*

*Научно-исследовательский университет Высшая школа экономики,
Россия, 101978, Мясницкая ул., д.20.*

ЗАО ЕС-лизинг, Россия, 117405, Москва, Варшавское ш. д.125.

Аннотация. Тестирование ПО автоматизированных систем на разных стадиях их жизненного цикла отличается по целям, решаемым задачам, объектам, методам и результатам тестирования, несмотря на отличие свойств этих объектов. При этом в научно-технической литературе используются одни и те же термины для описания разнотипных объектов, несмотря на различие их свойств и методов работы с ними в процессе тестирования. Целью работы являются рассмотрение используемого в жизненном цикле ПО автоматизированных систем комплекса понятий в области тестирования и изменение смыслового содержания этих понятий в зависимости от того, какими свойствами должно обладать ПО АС на текущем этапе ЖЦ. Соответственно изменяются характеристики тестируемого ПО, объемы тестов, степень соответствия объекта тестирования и необходимых для конкретного вида тестирования ресурсов для его применения. Понимание этих отличий существенно влияет на используемые виды и методы тестирования, а также на требования к средствам автоматизации тестирования.

Ключевые слова: тестирование программного обеспечения; виды тестирования; функциональное тестирование; нагрузочное тестирование; тестирование безопасности; статический анализ; композиционный анализ; динамический анализ; фаззинг-тестирование.

Для цитирования: Позин Б.А. Тестирование в жизненном цикле автоматизированных систем. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 303–310. DOI: 10.15514/ISPRAS-2025-37(3)-21.

Благодарности: Автор признателен А.К. Петренко и В.В. Кулямину за плодотворные дискуссии.

Testing in Life Cycle of Automated Systems

B.A. Pozin, ORCID: 0000-0002-0012-2230 <bpozin@ec-leasing.ru>

*Institute for System Programming of the Russian Academy of Sciences, 25,
Alexander Solzhenitsyn st., Moscow, 109004, Russia
National Research University, Higher School of Economics,
20, Myasnitskaya ulitsa, Moscow, 101978, Russia
«EC-leasing» Co., 125 Varshavskoye shosse, Moscow, 117405, Russia*

Abstract. Software testing of automated systems at different stages of their life cycle (LC) differs in terms of goals, tasks, objects, methods, and test results, despite the difference in the properties of these objects. At the same time, the same terms are used in the scientific and technical literature to describe different types of objects, despite the difference in their properties and methods of working with them during testing. The purpose of the work is to consider the complex of concepts used in the software lifecycle of automated systems in the field of testing and to change the semantic content of these concepts depending on what properties the automated system software should have at the current stage of the LC. Accordingly, the characteristics of the software under test, the volume of tests, the degree of compliance of the test object and the resources necessary for a specific type of testing for its application change. Understanding these differences significantly affects the types and methods of testing used, as well as the requirements for test automation tools.

Keywords: software testing; types of testing; functional testing; load testing; security testing; static analysis; compositional analysis; dynamic analysis; fuzz testing.

For citation: Pozin B.A. Testing in the life cycle of automated systems. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025, pp. 303-310 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-21.

Acknowledgments: The author is grateful to A.K. Petrenko and V.V. Kulyamin for fruitful discussions.

1. Введение

Программное обеспечение (ПО) может быть как самостоятельным продуктом (операционная система, сервис), так и представлять собой составную часть системы, реализующую ее функции в интересах решения некоторой задачи автоматизации (управления бизнесом, транспортом, финансовой, коммерческой деятельностью и т.п.).

Для контроля качества человеко-машинных программно-аппаратных систем (далее – систем) одним из наиболее широко применяемых на практике является метод тестирования. Тестирование представляет собой проверку соответствия реального поведения системы требованиям к ней, выполняемую на выбранном определенным образом наборе сценариев работы этой системы.

Суть тестирования состоит в том, чтобы в результате прогона (выполнения) тестируемой программы обнаружить существующий в ней дефект (успешное тестирование), то есть несоответствие объекта тестирования требованиям к нему. Тестирование – это прежде всего обнаружение ошибки. Однако этот метод можно использовать и для контроля качества. Разделяют понятия тестирование и отладка. Отладка включает *обнаружение* дефектов (собственно тестирование); *локализацию* мест, в которых дефекты/ошибки расположены в тестируемой программе; *диагностики ошибок* (определение типов ошибок и целей их устранения); *исправление ошибок* – установление фактов, подтверждающих исправление ошибок, и того, что ПО соответствует требованиям к нему.

Следует особенно отметить, что процесс тестирования осуществляется на всех стадиях жизненного цикла системы: при разработке, эксплуатации, сопровождении и развитии ПО систем.

2. Требования к системе и ПО как эталоны для тестирования

Требования, которые предъявляются к системе в целом и к ее ПО, формулируются в документах, чаще всего в техническом задании (ТЗ) на систему и в конкретизирующих его частных технических заданиях (ЧТЗ) или спецификациях [1, 2], согласованных Заказчиком. При проектировании, разработке ПО и системы в целом эти требования обязательны к выполнению при разработке. Требования технического задания являются требованиями высокого уровня: функциональными и нефункциональными и используются при тестировании. При тестировании ПО автоматизированных систем (АС) они являются эталоном для проверки соответствия ПО и системы (объекта тестирования) предъявляемым требованиям (эталону для тестирования).

Функциональные требования представляют собой описание того, какие функции должны быть выполнены, и иногда описывают исходные данные и ожидаемые результаты выполнения системой или ПО функций. Нефункциональные требования описывают интересующие заказчика системы нефункциональные свойства, такие как производительность, надежность, безопасность и другие, зачатую с указанием уровня достижимых значений этих характеристик. Конкретизация значений нефункциональных требований и возможных свойств реализуемых функций в ходе разработки системы производится в техническом проекте или формализованных соглашениях (протоколах и т.п.) сторон, которые также можно интерпретировать как требования.

Совокупность требований можно проверять поэтапно, по мере реализации программных компонент на этапах разработки. При декомпозиции технического задания на стадии проектирования обычно выделяют функциональные подсистемы, каждая из которых конкретизирует требования к одной системной функции, специфицированной в ТЗ. Именно эти требования являются затем эталонными при тестировании этой системной функции.

Ключевыми понятиями (функционального) тестирования являются понятия тестовой ситуации и теста. Выбранный сценарий для проверки некоторой функции (функционального требования) называют тестовой ситуацией.

На каждом этапе разработки производят тестирование того ПО, которое реализует каждую из подсистем и соответственно проводят функциональное тестирование подсистемы, как объекта разработки, для которого существует набор функциональных требований как эталон для тестирования.

Тестированию может подвергаться как отдельная компонента программного обеспечения, так и несколько компонент. В последнем случае используют термин *интеграционное тестирование*, то есть проверку того, что несколько компонент функционируют согласованно, совместно реализуя некоторую системную функцию, предусмотренную ТЗ или ЧТЗ. Степень полноты проверки оценивается критерием полноты, который может устанавливаться как для программного обеспечения в целом, так и для проверок отдельных компонент. Во многих случаях используют понятие критерий покрытия, который изначально применялся только при тестировании структуры отдельных программных модулей, чтобы оценить, какая доля логики программ проверена тестовым набором (существовал даже стандарт BBC Великобритании по критериям покрытия логики программ [3]).

Под *системным тестированием* в процессе разработки понимают тестирование ПО системы в статике для проверки функционирования системы в определенных режимах функционирования, например, подготовка к выполнению запланированного процесса обработки данных и т.п.

При тестировании ПО на стадии разработки, начиная с интеграционного тестирования, объем тестов становится довольно большим, трудоемкость их хранения, внесения изменений и сопровождения становится значительной.

3. Виды тестирования

Существует довольно много видов тестирования, то есть подходов к построению тестовых наборов разного назначения – с учетом разных типов объектов тестирования, информации для выбора тестовых ситуаций и возможности подачи тестов на вход объекта тестирования. По возможности использования структуры ПО в критериях покрытия различают виды тестирования, основанные на методах:

- черного ящика, когда разработчик тестов практически не имеет сведений о структуре тестируемого ПО или его компонент;
- белого ящика, когда разработчик знает внутреннюю структуру объекта тестирования и может ее использовать при разработке теста и оценке степени покрытия;
- серого ящика (смешанное), когда внутренняя структура объекта тестирования частично может быть использована.

Эффективность применения разных видов тестирования зависит от стадии жизненного цикла и используемого метода функционального тестирования. По видам тестирования на стадии ввода в действие [5, 6] различают (по отечественной практике) три вида *приемочного тестирования*:

- *предварительные испытания* для верификации соответствия ПО функциональным требованиям;
- *опытная эксплуатация (валидация)*, целью которой является оценка соответствия системы /ПО потребностям заказчика на период проверки;
- *приемочные испытания (acceptance testing)*.

3.1 Регрессионное тестирование

После внесения изменений необходимо убедиться в том, что изменения совместимы с ранее проверенным ПО, то есть после их внесения в этом ПО не проявляются новые ошибки – ошибки регрессии. Для целей такой проверки после внесения изменений осуществляют проверку ПО на тех же тестах (тестовых наборах), на которых ПО уже проверялось ранее и подтвердило работоспособность. Такой вид тестирования называется регрессионным тестированием и имеет целью обнаружение ошибок регрессии. Регрессионное тестирование в технологически зрелых организациях проводится систематически, повторно при изменении ранее отработанного ПО – с какой бы целью эти изменения не производились. Регрессионное тестирование применяется на стадии разработки при системном тестировании для контроля совместимости обновлений с уже проверенной частью ПО и, как правило, проводится систематически (например, после каждой пересборки релиза ПО), на стадии сопровождения при проверке корректности вносимых изменений в действующую программную систему и т. п.

3.2 Методы тестирования нефункциональных требований

Нефункциональные требования предъявляются к общим свойствам системы, то есть к тому, какой должна быть система в целом: производительность, надежность, доступность, мобильность, масштабируемость и др. Объектом тестирования является система в целом. В силу этого все методы тестирования нефункциональных требований оказываются весьма дорогими при реализации, требуют тщательного планирования. Тестирование качества реализации этих требований осуществляется разными методами. Цели тестирования нефункциональных требований обычно устанавливаются для каждого эксперимента отдельно с учетом того, какие свойства системы подвергаются тестированию и как предполагается использовать результаты. По результатам тестирования нефункциональных требований оцениваются эксплуатационные характеристики системы (на момент тестирования).

Предметами тестирования являются конкретные свойства системы: при тестировании переносимости – переносимость между какими аппаратными платформами проверяется и оценивается; возможность переноса программного обеспечения из одной операционной среды в другую (конкретные среды) и т. п. При тестировании совместимости – совместимость в части конкретных протоколов и интерфейсов системы со смежными системами. При тестировании надежности – аспекты надежности, которые требуется оценить, например, тестирование времени восстановления работоспособности системы после отказа.

3.3 Нагрузочное тестирование системы

Наиболее подробно исследованным является, пожалуй, нагрузочное тестирование (performance testing), поскольку оно отвечает на важнейший вопрос: какова производительность системы в терминах количества работ (которые она автоматизирует) в единицу времени. Например, количество банковских транзакций в течение операционного дня или количество проданных билетов в час и т.п.

При нагрузочном тестировании исследуются различные свойства производительности: собственно производительность системы при плановой нагрузке; время отклика системы на запрос (среднее или по видам запросов); нагрузочная способность при резких изменениях нагрузки (soak testing); способность системы долго выдерживать повышенную нагрузку (spike testing) и др. Обычно рассматривают три вида характеристик производительности: реактивность, продуктивность и использование.

Нагрузочное тестирование – это всегда проведение нагрузочного эксперимента, тщательное планирование которого позволяет получить корректные результаты и ограничить затраты на его проведение. Для планирования нагрузочного эксперимента необходимо совместно с владельцем системы сформулировать цель проведения нагрузочного тестирования и сценарий нагрузочного эксперимента, исходя из потребностей владельца по оценке или прогнозированию эксплуатационных характеристик тестируемой системы и соответствующих нефункциональных требований, а также исходя из потребностей периодического контроля деградации эксплуатационных характеристик тестируемой системы. Все характеристики системы оцениваются методами теории вероятностей и математической статистики по результатам эксперимента, обеспечивающего по возможности достаточное количество измерений для достоверной оценки этих характеристик. В зависимости от целей нагрузочного тестирования можно выделить следующие его виды и назначение [6]:

- *Оценочное* – оценка характеристик производительности в однократном нагрузочном эксперименте;
- *Аналитическое* – выявление зависимостей (например, производительности от вычислительных ресурсов) в серии нагрузочных экспериментов;
- *Настроечное* – настройка и оптимизация нагрузочных характеристик системы или ее составной части;
- *Регрессионное* – многократное периодическое нагрузочное тестирование при неизменных условиях для выявления признаков деградации тестируемой системы.

4. Тестирование с целью поиска уязвимостей в программах

В последние годы в связи с активным использованием в разрабатываемом ПО компонент из Open Source библиотек и соответствующем возрастании риска наличия уязвимостей в ПО АС, в промышленности применяются новые методы поиска уязвимостей. Это далеко не всегда методы, похожие на классическое тестирование. В настоящее время активно используются методы статического анализа программ [8, 9], методы композиционного анализа, методы динамического анализа [10, 13].

При использовании метода *статического анализа* [5] осуществляется сопоставление конструкций анализируемой (тестируемой) программы с эталоном, каковым являются компоненты классификаций уязвимостей: OWASP, CWE, CVE, БДУ ФСТЭК. Механизм этих методов довольно понятный: проводится анализ исходных текстов ПО АС на используемом языке программирования, в процессе которого выделяется конструкция с признаками наличия уязвимости, которая сопоставляется с элементами вышеприведенных классификаций. В случае совпадения такая конструкция оценивается как уязвимость, в противном случае – как правильная конструкция[11]. Отчеты о результатах статического анализа предоставляются разработчикам ПО для доработки/внесения изменений в ПО, а также уполномоченным сотрудникам службы информационной безопасности для оценки качества разработанного ПО с позиций безопасности, а также для поддержки принятия решения о возможности промышленного использования ПО, имеющего в своем составе уязвимости того уровня критичности, который установлен по результатам статического анализа. Инструментальные средства статического анализа записывают в файл SARIF сведения о найденных уязвимостях, их уровне серьезности и рекомендации по исправлению. Подробнее о SARIF см. ниже в этом разделе.

При использовании методов композиционного анализа объектом тестирования является состав ПО АС, которое планируется поставлять в составе АС. Объект тестирования состоит из множества компонент, часть из которых написана разработчиками ПО АС для реализации функциональных задач АС, а другая часть (по имеющимся данным до 70-90%) заимствуется из библиотек Open Source. В результате возникает коллизия, состоящая в том, что авторы ПО зачастую точно не знают, какие компоненты входят в состав разработанного ими ПО, а также не до конца понимают, каков уровень безопасности заимствованных компонент и ПО в целом, то есть какие уязвимости присутствуют в составе ПО, а также в процессе разработки и отладки ПО АС- какие уязвимости исправлены, а какие нет. В том числе авторы заимствованных компонент могут и сами исправлять уязвимости, о чем в библиотеках хранится информация. Количество компонент при этом может достигать несколько сотен и более. Средства автоматизации композиционного анализа проводят инвентаризацию компонент, входящих в состав ПО и формируют каталог этих компонент с учетом того, из какой библиотеки заимствована компонента, на каком языке написана и с какими другими связана. Эта информация предназначена прежде всего для руководителей разработки ПО в каталоге/перечне, который называется SBoM (Software Bill of Materials), содержатся также сведения руководителей разработки ПО, представителя заказчика и сотрудника информационной безопасности, контролирующего уровень безопасности. SBoM содержит метаданные о компонентах, сторонних сервисах, зависимостях, уязвимостях ПО. На основе этих данных можно спланировать изменения, связанные с устранением уязвимостей, выявить наиболее подверженные атакам части, оценить уровень качества и безопасности ПО и его составных частей. Средства композиционного анализа формируют отчет о проведенном анализе в виде файла SARIF (Static Analysis Results Interchange Format), в котором содержатся результаты статического и композиционного анализа, которые позволяют оценить количество и уровень остаточных уязвимостей в ПО АС. Регулярное проведение обоих видов анализа обеспечивает возможность мониторинга/контроля качества и уровня безопасности, разрабатываемого АС.

Метод динамического анализа ПО АС основан на выполнении программы, однако, в отличие от функционального тестирования при динамическом анализе осуществляется не проверка выполнения функционального требования к тестируемой программе, а проверка наличия конкретных видов известных уязвимостей, систематизированных в классификациях OWASP, CWE, CVE, БДУ ФСТЭК. Тестирование серверной части клиент-серверных систем сопряжено с необходимостью анализа весьма большого количества маршрутов выполнения ПО АС. В этой связи применение метода динамического анализа оказывается не очень удобным в силу высокой трудоемкости ручных операций при

подготовке тестов и анализе результатов. С этой целью для тестирования безопасности серверного ПО чаще всего применяется фаззинг-тестирование.

Систематическое применение описанных средств и методов позволяет осуществлять мониторинг того, как устраняются уязвимости в ПО АС в ходе разработки и какие критические уязвимости могут попасть в ПО АС, передаваемого в постоянную эксплуатацию.

5. Заключение

Состав используемых методов тестирования зависит от размера ПО АС (количество строк кода) и нефункциональных требований к нему (наличие требований к режиму реального времени и требований по информационной безопасности).

1. Для небольших и средних по размеру комплексов ПО (несколько десятков тысяч строк), разрабатываемых по схеме DevOps, функциональное тестирование является основным видом тестирования и может проводиться поэтапно (в случае реализации ПО в виде нескольких микросервисов: каждый микросервис как отдельная сущность и интеграционное тестирование микросервисов на единой тестовой системе).
2. Для больших по размеру комплексов ПО (миллион и более строк кода), разработанных в виде набора микросервисов, которые разрабатываются каждый по схеме DevOps, а комплекс ПО интегрируется на отдельной инфраструктуре, определяемой потребностями заказчика и его требованиями к инфраструктуре эксплуатации АС. Тестовая система должна включать защищенные области (контура), доступ к которым осуществляется по политикам безопасности, разрабатываемым заказчиком для АС в целом [12].
3. Функциональное тестирование производится в основном для микросервисов, как объектов тестирования. При этом в рамках релизных циклов разработка микросервисов осуществляется по технологии DevSecOps. Повышается роль интеграционного тестирования, а системное тестирование становится обязательным. При этом объекты интеграционного и системного тестирования становятся также объектами тестирования безопасности, а проведение выявления и устранения уязвимостей на регулярной основе становятся обязательными перед вводом ПО АС в эксплуатацию.
4. Содержанием тестирования безопасности является использование статического и композиционного анализа в течение релизных циклов, а также при необходимости (для многозвенных клиент-серверных систем) дополнительно - использование средств динамического и фаззинг-тестирования. В настоящее время складывается технология использования динамического тестирования прежде всего для тестирования клиентской части ПО АС на регулярной основе, в том числе и в процессе эксплуатации системы – на комплексном стенде или резервной инфраструктуре параллельно с эксплуатацией ПО АС.
5. В организации, эксплуатирующей АС, как правило, устанавливают регламент использования системы для того, чтобы обеспечивать требования бизнеса по срокам сбора данных о функционировании организации, использующей АС, сроках формирования отчетности, выполнения временных требований по реализации бизнес-операций и т.п. Для оценки степени выполнения этого бизнес-регламента при текущем состоянии эксплуатационных характеристик АС, а также при перспективных требованиях бизнеса, оказывающих влияние на эксплуатационные характеристики АС, должно производиться нагрузочное тестирование на специализированном стенде или резервном сервере системы по программе и методике испытаний, специально разрабатываемой для каждого нагрузочного эксперимента. Такие эксперименты, как показывает опыт крупнейших

пользователей подобных корпоративных автоматизированных систем (банки, операторы связи, крупные энергетические компании и т.д.) целесообразно проводить периодически, но на плановой, постоянной основе. В этом случае должна быть разработана тестовая система для согласованных с заказчиком сценариев функционирования, критичных с точки зрения оцениваемых (паспортизуемых) эксплуатационных и нефункциональных характеристик АС. При этом должна быть разработана тестовая система для НТ, в которой учтены требования к чувствительным данным.

Список литературы / References

- [1]. ГОСТ 34.602 – 2020. Информационные технологии/ КОМПЛЕКС СТАНДАРТОВ НА АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ Техническое задание на создание автоматизированной системы. М.: 9 с.
- [2]. Батоврин В. К., Позин Б. А. Инженерия требований на современном промышленном предприятии // Программная инженерия. 2019. Т. 10. № 3. С. 114-124.
- [3]. Липаев В.В. Тестирование крупных комплексов программ на соответствие требованиям, М.ИПЦ «Глобус», 2008, 376 с.
- [4]. ГОСТ 34.603. Информационные технологии. Виды испытаний автоматизированных систем, 1992.
- [5]. ГОСТ Р 71207-2024. Защита информации. Разработка безопасного программного обеспечения. Статический анализ программного обеспечения. Общие требования.
- [6]. Позин Б.А. Ввод в действие информационных систем и сопровождение их программного обеспечения. - М.: Новые технологии, 2010. - 32 с. - (Прил. к журн. "Информационные технологии"; N 4/2010)
- [7]. B. Pozin, I. Galakhov, Experience in automated functional and load testing in the life cycle of the mission-critical system. *Baltic J. Modern Computing*, Vol. 8 (2020), No. 2, 241-258, DOI: 10.22364/bjmc.2020.8.2.03.
- [8]. Иванников В.П., Белеванцев А.А., Бородин А.Е., Игнатьев В.Н., Журихин Д.М., Аветисян А.И., Леонов М.И. Статический анализатор Svase для поиска дефектов в исходном коде программ. Труды Института системного программирования РАН. 2014;26(1): с.231-250.
- [9]. Белеванцев А., Аветисян А. Многоуровневый статический анализ для поиска закономерностей ошибок и дефектов в исходном коде. В: Петренко А., Воронков А. (ред.) Перспективы системной информатики. PSI 2017. Конспекты лекций по информатике, том 10742, стр. 28–42.
- [10]. Белеванцев А. (ИСП РАН) для форума "Russia DevOps Report - 2023". [Электронный ресурс]. – 2023 – URL: <https://russiadevopsreport.ru/>. – (Дата обращения: 13.10.2024).
- [11]. Позин Б.А., Бородушкина П.А., Коротков Д.А., Федоров М.А., Муратов А.Ф. Методика поиска уязвимостей в ПО, написанном на нескольких языках программирования. Труды ИСП РАН, том 37, вып. 1, 2025 г., стр. 121–132. DOI: 10.15514/ISPRAS–2025–37(1)–7.
- [12]. Позин Б.А. Автоматизация и экономика для обеспечения жизненного цикла безопасного ПО, Информационная безопасность, № 4, 2024, с. 60.
- [13]. Кулямин В.В. Обзор методов динамического анализа программного обеспечения. Труды Института системного программирования РАН. 2023; 35(4):7-44. DOI: 10.15514/ISPRAS-2023-35(4)-1.

Информация об авторах / Information about authors

Борис Аронович ПОЗИН – доктор технических наук, профессор, главный научный сотрудник ИСП РАН, профессор базовой кафедры ЗАО ЕС-лизинг в МИЭМ НИУ ВШЭ, технический директор ЗАО ЕС-лизинг. Сфера научных интересов: программная инженерия, системы обеспечения жизненного цикла доверенного программного обеспечения, автоматизированное тестирование программ.

Boris Aronovich POZIN – Dr. Sci. (Tech.), Professor, Chief Researcher at the ISP RAS, Professor of the Basic Department of CJSC EC-Leasing at the Higher School of Economics, Technical Director of CJSC EC-Leasing. Research interests: software engineering, life cycle ensuring systems for trusted software, automated software testing.



Оптимизации генерации иерархических уровней детализации для масштабных полигональных сцен

¹ В.Н. Шуткин, ORCID: 0000-0002-9238-5029 <v451ly@ispras.ru>

¹ Н.К. Морозкин, ORCID: 0000-0003-3949-7731 <nmzik@ispras.ru>

^{1,2} В.А. Семенов, ORCID: 0000-0002-8766-8454 <sem@ispras.ru>

^{1,2,3} О.А. Тарлапан, ORCID: 0000-0001-5559-2013 <oleg@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

² Московский физико-технический институт,
Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., 9.

³ Московский государственный университет имени М.В. Ломоносова,
Россия, 119991, Москва, Ленинские горы, д. 1.

Аннотация. Альтернативные уровни детализации (LOD) являются одним из наиболее перспективных подходов к эффективному рендерингу сложных пространственно-трехмерных сцен. Подход получил развитие в методах иерархических уровней детализации (HLOD) и иерархических динамических уровней детализации (HDLOD), которые в настоящее время хорошо проработаны и успешно применяются для консервативного и интерактивного рендеринга больших динамических сцен. Вместе с тем, вопросам эффективной генерации уровней детализации не уделялось должного внимания, а они оказываются критичными в ряде приложений, связанных с визуальным моделированием сложных промышленных проектов и масштабных инфраструктурных программ. В работе рассматриваются перспективные техники ускорения генерации HLOD и HDLOD. Также обсуждается возможность быстрого обновления иерархических уровней детализации с учетом перманентных локальных изменений в трёхмерной модели, характерных для приложений совместной работы.

Ключевые слова: иерархические уровни детализации; иерархические динамические уровни детализации; кластеризация; параллельные вычисления; упрощение геометрии; инкрементальное обновление; удаление невидимой геометрии; интерактивный рендеринг.

Для цитирования: Шуткин В.Н., Морозкин Н.К., Семенов В.А., Тарлапан О.А. Оптимизации генерации иерархических уровней детализации для масштабных полигональных сцен. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 311–324. DOI: 10.15514/ISPRAS-2025-37(3)-22.

Optimizations for Hierarchical Levels of Detail Generation on Large-Scale Polygonal Scenes

¹ V.N. Shutkin, ORCID: 0000-0002-9238-5029 <v451ly@ispras.ru>

¹ N.K. Morozkin, ORCID: 0000-0003-3949-7731 <nmzik@ispras.ru>

^{1, 2} V.A. Semenov, ORCID: 0000-0002-8766-8454 <sem@ispras.ru>

^{1, 2, 3} O.A. Tarlapan, ORCID: 0000-0001-5559-2013 <oleg@ispras.ru>

¹ *Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

² *Moscow Institute of Physics and Technology,
9, Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.*

³ *Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia.*

Abstract. Alternative levels of detail (LOD) are one of the most promising approaches to effective rendering of complex spatial 3D scenes. The approach has been realized in the hierarchical levels of detail (HLOD) and hierarchical dynamic levels of detail (HDLOD) methods, which are currently well studied and successfully applied for conservative and interactive rendering of large dynamic scenes. At the same time, the issues of efficient generation of levels of detail, which are critical in a number of applications related to visual modeling of complex industrial projects and large-scale infrastructure programs, have not received due attention. The paper considers emerging techniques for accelerating HLOD and HDLOD generation. It also discusses the possibility of quickly updating hierarchical levels of detail, taking into account permanent local changes in the three-dimensional model, typical for collaborative applications.

Keywords: hierarchical levels of detail; hierarchical dynamic levels of detail; clustering; parallel computing; geometry simplification; incremental update; hidden geometry removal; interactive rendering.

For citation: Shutkin V.N., Morozkin N.K., Semenov V.A., Tarlapan O.A. Optimizations for hierarchical levels of detail generation on large-scale polygonal scenes. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025, pp. 311-324 (in Russian). DOI: 10.15514/ISPRAS-2025-37(3)-22.

1. Введение

Альтернативные уровни детализации (англ. levels of detail, LOD) получили широкое применение в приложениях компьютерной графики, оперирующих масштабными полигональными сценами и требующих эффективных средств рендеринга. Благодаря заранее подготовленным, упрощенным представлениям объектов сцены и их контекстному применению в зависимости от положения камеры и требований точности удается обеспечить высокую скорость рендеринга с приемлемым визуальным качеством.

Впервые идея использования альтернативных представлений объектов при рендеринге изображений была высказана ещё в 1976 году в работе [1]. Однако термин «уровни детализации» активно начал использоваться только в 1990-е годы, когда с развитием персональных компьютеров трёхмерная графика «шагнула в массы». Идея создания нескольких уровней детализации и выбора подходящего уровня в зависимости от расстояния до объекта (или размера объекта на экране) была высказана в работах [2, 3]. Использование более упрощённых представлений для более дальних объектов позволяет существенно увеличить производительность рендеринга. Однако при этом общее число объектов в сцене не сокращается. Чтобы сократить число обрабатываемых объектов, и для дальних участков сцены оперировать не индивидуальными объектами, а группами объектов, в 2001 году были разработаны иерархические уровни детализации (hierarchical levels of detail, HLOD) [4]. Идея заключается в построении иерархии уровней детализации, в которой каждый узел представляет собой упрощённое представление для целой группы объектов. Объекты иерархически группируются во всё большие и большие группы с возрастающей степенью

упрощения. В 2000-е годы основным направлением исследований в этой области стал рендеринг сверхбольших сцен, не помещающихся в оперативную память, с использованием иерархических уровней детализации [5, 6] (рендеринг во внешней памяти). В 2020-е годы уровни детализации получили дальнейшее развитие и были обобщены на случай динамических сцен. Иерархические динамические уровни детализации (hierarchical dynamic levels of detail, HDLOD) успешно применяются для консервативного (гарантирующего заданную пространственную и временную точность) и интерактивного (стремящегося обеспечить максимально возможную частоту генерации изображений при приемлемом уровне реализма) режимов отображения больших псевдо-динамических сцен [7-9]. Многовариантный метод HDLOD [10] обеспечивает одновременное отображение альтернативных сценариев без дополнительных вычислительных затрат на пересчет уровней детализации, характерных для основного метода.

Параллельно с развитием методов уровней детализации развивались методы упрощения геометрических представлений. Наиболее распространённым способом задания геометрического представления является полигональная сетка — набор полигонов, аппроксимирующих форму поверхности объекта. Основные подходы к упрощению полигональных сеток: стягивание рёбер [11], удаление вершины [12], кластеризация вершин [2, 13]. Подробный обзор методов упрощения полигональных представлений и уровней детализации представлен в книге [14]. В последнее время можно выделить три направления исследований по данной тематике: улучшение частных аспектов методов, адаптация методов для конкретных приложений и обобщение методов на новые типы геометрии. Работы [15, 16] предлагают улучшенные алгоритмы упрощения сеток на основе метода стягивания рёбер. Работа [17] развивает идею упрощения на основе изображений классической работы [18] с использованием дифференцируемого рендеринга и современных алгоритмов нелинейной оптимизации. Работа [19] посвящена вопросу предварительной загрузки данных при рендеринге во внешней памяти, а [20] предлагает схему размещения данных на диске для быстрого поиска. В работе [21] предлагается алгоритм упрощения сетки на графическом процессоре, а также подход к внешнему рендерингу, минимизирующий передачу данных между центральным и графическим процессором, а в следующей работе от тех же авторов [22] приводится подход, учитывающий количество видеопамати.

Ряд работ посвящён генерации уровней детализации для моделей городов: в работах [23-25] приводятся алгоритмы упрощения, оптимизированные для обработки моделей зданий, работа [26] предлагает улучшения качества упрощения, размера текстур и производительности рендеринга, в работе [27] предлагается использовать гибридные представления уровней детализации (точки, линии, сплэты) для моделей городов, работа [28] посвящена веб-рендерингу больших моделей городов. Следующие работы описывают использование уровней детализации с другими типами геометрии: NeRF [29], 3D Gaussian [30], облако точек [31], воксельное представление [32], гибридное представление полигональная сетка + воксель [33].

Важно отметить, что в большинстве работ внимание исследователей сосредоточено на производительности рендеринга. Анализ времени подготовки уровней детализации проводится только для получения некоторой референсной информации. Вопросы же эффективной генерации уровней детализации не уделяется должного внимания, а они становятся критичными для приложений, связанных с визуализацией сложных промышленных проектов и масштабных инфраструктурных программ на основе цифровых моделей. В этих приложениях время подготовки уровней детализации исчисляется часами и сутками процессорного времени на компьютерах типовой конфигурации, что является неприемлемым для формирования и визуального анализа цифровых моделей в условиях перманентных изменений, обусловленных, в частности групповой работой.

В статье рассматривается зарекомендовавшая себя вычислительная стратегия генерации иерархических уровней детализации с кластеризацией снизу-вверх. Для нее обсуждаются три

фактора, которые существенно влияют на производительность генерации уровней детализации и которые следует принимать во внимание в программных реализациях. Первый фактор – это методы и средства полигонального упрощения, которые имеют разную вычислительную сложность и которые следует рационально и избирательно применять в зависимости от контекста рендеринга, в частности, с учетом положения камеры, видимости объектов сцены и требуемого визуального качества. Второй важный фактор – это эффективная программная реализация методов и, прежде всего, возможность распараллеливания основной процедуры формирования индивидуальных уровней детализации и упрощения их полигонального представления. Наконец, третьим обсуждаемым фактором является возможность инкрементального обновления уровней детализации при локальных изменениях представления сцены вместо повторной генерации полной иерархии уровней детализации.

2. Генерация иерархических уровней детализации

Целью данной работы является исследование способов ускорения генерации иерархических уровней детализации с кластеризацией снизу-вверх. Данная вычислительная стратегия хорошо себя зарекомендовала как для статических, так детерминированных псевдодинамических сцен при использовании иерархических уровней детализации HLOD и иерархических динамических уровней детализации HDLOD соответственно [7-10]. Поэтому основные результаты работы в равной степени применимы к обоим случаям, а мы для краткости применяем аббревиатуру первого метода.

Пусть сцена S определена в трёхмерном евклидовом пространстве E^3 и представлена как набор объектов $s(g_s, \iota_s) \in S$. Каждый объект имеет фиксированное геометрическое представление g_s , а также уникальный идентификатор ι_s . В рамках данной работы будем считать, что g_s задаётся в виде полигональной сетки треугольников, а ι_s может быть задан любым способом: строка, номер, GUID или UUID.

Будем рассматривать генерацию HLOD как построение дерева кластеров. Каждый кластер $c(g_c, b_c, \varepsilon_c)$ имеет фиксированное геометрическое представление g_c , ограничивающий объём b_c и геометрическую погрешность ε_c . Для кластеров определено отношение агломерации $<$. Запись $c' < c$ означает, что кластер c' является ребёнком кластера c в дереве, а также то, что кластер c' вошёл в кластер c во время кластеризации. Пусть у кластера c имеется n детей $\forall i = 1, \dots, n \ c_i < c$. Это означает, что геометрическое представление g_c получено путём объединения и упрощения представлений всех дочерних кластеров, а именно $g_c = \bigcup_{i=1, \dots, n} g_{c_i}$, после чего g_c упрощено с геометрической погрешностью ε_c . Листовые кластеры получаются путём объединения объектов, а кластеры последующих уровней, путём объединения кластеров (и, возможно, объектов). Для листового кластера, его геометрическое представление получается путём объединения представлений вошедших в него объектов без упрощения. Если в листовой кластер c вошли n объектов $s_1, \dots, s_n$, то $g_c = \bigcup_{i=1, \dots, n} g_{s_i}$.

Геометрическое представление кластера должно сохранять идентификаторы объектов для своих частей. Идентификатор объекта ι_s должен храниться для каждого треугольника (или вершины). Это необходимо, чтобы во время рендеринга иметь возможность связать часть геометрии кластера с оригинальным объектом сцены. Например, чтобы реализовать выделение оригинальных объектов по клику пользователя в 3D окне, чтобы применять цвета и стили к оригинальным объектам при отображении HLOD представления, или для реализации динамики в виде появления-исчезновения объектов (сокрытие/показ частей геометрии кластера). Поэтому во время генерации HLOD важно сохранять связь треугольников с идентификаторами объектов, в том числе не терять идентификацию треугольников в процессе упрощения полигональных сеток. Можно было бы представить геометрию кластера как набор полигональных сеток с идентификаторами, однако практика показывает, что для эффективности рендеринга необходимо минимизировать число

обрабатываемых сущностей, сокращать число вызовов отрисовки и т.д. Поэтому оптимальным представляется использование одной сетки на кластер с сокрытием или покраской её частей в шейдере.

Ограничивающий объём кластера всегда охватывает ограничивающие объёмы дочерних кластеров: $\forall c' < c \ b_{c'} \subseteq b_c$. Это свойство позволяет использовать HLOD как иерархию ограничивающих объёмов для эффективных проверок видимости во время рендеринга. А именно, если кластер невидим, то не видны и все его дети. С другой стороны, если b_c кластера полностью попадает в область видимости, то все дочерние кластеры тоже будут попадать в область видимости и проверки для них можно уже не проводить.

Геометрическая погрешность дочернего кластера не может превышать геометрическую погрешность родительского кластера: $\forall c' < c \ \varepsilon_{c'} \leq \varepsilon_c$. Это свойство необходимо во время рендеринга, чтобы выбор более детальных представлений осуществлялся как обход вглубь дерева.

Листовые кластеры содержат точную (неупрощённую) геометрию: $\forall c: \exists c' < c \ \varepsilon_{c'} = 0$. Это свойство необходимо, чтобы всегда иметь возможность отобразить точную (оригинальную) геометрию объекта.

На рис. 1 (а) приведён пример дерева кластеров: объекты сцены объединяются в листовые кластеры без упрощения, после чего кластеры объединяются во всё большие и более упрощённые кластеры пока не процесс не сойдётся к одному корневому кластеру, который будет являться наиболее упрощённым представлением для всей сцены целиком.

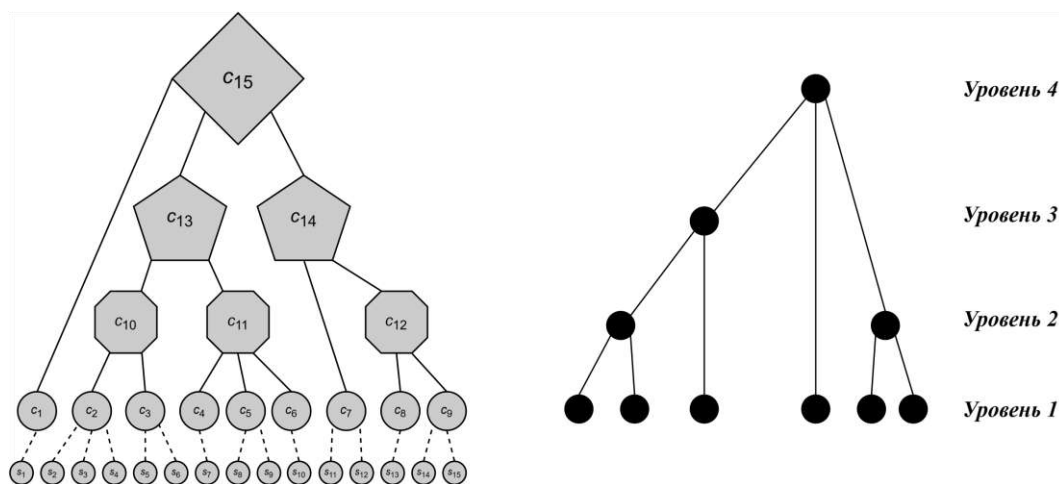


Рис. 1. а) пример дерева кластеров; б) нумерация уровней в дереве кластеров.
Fig. 1. a) an example of a cluster tree; b) numbering of levels in a cluster tree.

Опишем общую стратегию построения HLOD. Первым этапом является построение дерева кластеров. На данном этапе используется только информация об ограничивающих объёмах объектов сцены. Поскольку ограничивающий объём представляет собой достаточно компактную структуру данных, можно не беспокоиться об ограничениях по оперативной памяти и загрузить ограничивающие объёмы всех объектов сцены в оперативную память даже для очень больших сцен. Дерево кластеров строится снизу вверх по шагам. На первом шаге запускается процедура кластеризации, которая группирует объекты сцены в кластеры. Эти кластеры становятся листовыми узлами дерева. На втором шаге та же процедура кластеризации используется уже для группировки этих кластеров в новые (родительские) кластеры. И так далее, пока процесс не сойдётся к единому корневому кластеру.

После того, как дерево построено, запускается генерация кластеров, а именно, формирование их геометрических представлений. При этом происходит объединение полигональных сеток

дочерних кластеров и упрощение полученной сетки до целевой геометрической погрешности или требуемой сложности, выражаемой количеством вершин и граней в результирующей сетке. Данный этап является самым вычислительно затратным. Поскольку полигональные сетки занимают много места в памяти, необходимо подгружать их в оперативную память только по мере необходимости, и освобождать память, когда они становятся не нужны. Также, этап генерации кластеров может быть распараллелен, так как кластеры одного уровня не зависят друг от друга, а зависят только от дочерних кластеров. Алгоритм упрощения полигональной сетки также заслуживает отдельного внимания, поскольку качество и время генерации в основном зависят именно от него.

Зафиксируем эту общую вычислительную стратегию и в рамках неё исследуем возможности для ускорения генерации.

Описанная выше стратегия генерации HLOD допускает три ключевых способа оптимизации:

1. ускорение процедур упрощения;
2. распараллеливание;
3. инкрементальные обновления.

В разделе 3 приводится краткий обзор методов полигонального упрощения, которые в сочетании с методами кластеризации являются ключевыми компонентами формирования иерархических уровней детализации HLOD, HDLOD. Важное внимание при этом уделяется возможностям идентификации и удаления внутренних элементов кластера, которые будучи невидимыми при внешнем расположении камеры, занимают дополнительный объем RAM оперативной памяти (видеопамяти) и расходуют значительное время GPU графического процессора при рендеринге. В разделе 4 описывается схема крупноблочного распараллеливания процедуры кластеризации и упрощения полигональных представлений, которая допускает эффективную реализацию на мульти-ядерных, мультипроцессорных архитектурах и вычислительных кластерах. Раздел 5 посвящен частной, но практически важной возможности оптимизации вычислений при повторном формировании уровней детализации.

3. Ускорение упрощения геометрии

Существует множество алгоритмов упрощения полигональных сеток, различающихся по скорости и качеству. Ограничимся двумя популярными алгоритмами.

Стягивание рёбер с квадратичной метрикой погрешности [11] является одним из наиболее широко используемых алгоритмов упрощения. Алгоритм заключается в итеративном стягивании рёбер сетки пока не будет достигнуто целевое число треугольников и кратко может быть описан следующим псевдокодом [15]

- оценить кривизну поверхности в каждой из исходных вершин x_i ;
- для каждого ребра (x_1, x_2) вычислить оптимальную результирующую вершину x_α и «стоимость» стягивания;
- отсортировать все рёбра по стоимости от минимальной к максимальной;
- последовательно стягивать рёбра и обновлять стоимости.

Стоимость — это мера того, насколько сильную погрешность внесёт стягивание ребра. Таким образом, сначала стягиваются рёбра, которые наименьшим образом исказят форму сетки. При стягивании ребра некоторые треугольники вырождаются, что приводит к сокращению числа треугольников в сетке. x_α может быть как новой вершиной (лучшее качество упрощения), так и одной из вершин x_1, x_2 (для экономии памяти при хранении нескольких уровней детализации для одной сетки). При добавлении ряда проверок можно гарантировать сохранение топологии. Также, есть вариация алгоритма, в которой рассматриваются не

только рёбра, но и любые пары вершин, что позволяет лучше упрощать несвязные сетки (с нарушением топологии).

Кластеризация вершин [2, 13] — это один из самых быстрых и простых в реализации алгоритмов упрощения. Алгоритм использует регулярную сетку. Все вершины, попавшие в одну ячейку сетки, объединяются в одну вершину. Таким образом сокращается число вершин (в каждой ячейке остаётся одна вершина), что приводит к вырождению части треугольников в точку или линию, что в свою очередь приводит к сокращению общего числа треугольников. Данный алгоритм нарушает топологию. Какие-то треугольники могут увеличиться в размерах, а какие-то уменьшиться, но размер ячейки ε гарантирует, что никакая вершина не сдвинется более, чем на $\varepsilon\sqrt{3}$ (диагональ куба), что позволяет контролировать погрешность.

От выбранного алгоритма существенным образом будут зависеть скорость генерации уровней детализации и их качество. Если требуется, чтобы все кластеры имели приблизительно одинаковую сложность и размер, необходим алгоритм упрощения до целевого числа треугольников. Для этой цели подходит, например, алгоритм стягивания рёбер с квадратичной метрикой погрешности. Сгенерированные уровни детализации будут удобны с точки зрения предсказуемости объёмов данных, кэшируемых, загружаемых в основную память и передаваемых по сети, что особенно важно для рендеринга больших сцен веб-приложениями. С другой стороны, при таком способе генерации иерархических уровней детализации не контролируется геометрическая погрешность кластеров, которая может влиять на эффективность рендеринга в ходе навигации по сцене из-за необходимого замещения кластеров с надлежащей точностью. Для этой цели можно порекомендовать алгоритм кластеризации вершин с целевой геометрической погрешностью, заданной относительно габаритов кластера и учитывающей его иерархический уровень.

Радикальной техникой упрощения, сочетаемой с основными алгоритмами, является **удаление внутренних невидимых граней**. Это особенно важно при подготовке HLOD для цифровых промышленных моделей, содержащих большое число конструктивных и инженерных элементов, перекрывающих друг друга и не видимых снаружи генерируемых кластеров. При этом важно заметить, что упрощённые представления используются только тогда, когда камера располагается достаточно далеко от объекта, то есть кластеры HLOD всегда просматриваются снаружи. Исключение могут составлять только два случая: интерактивный внешний рендеринг, когда необходимое детальное представление ещё не успело загрузиться из внешней памяти, и поэтому используется упрощённое представление, и прозрачные объекты (или возможность изменять прозрачность объектов в ходе визуализации). Случай с интерактивным рендерингом не является столь критичным, если навигация по сцене осуществляется плавно, и геометрия успевает загружаться. При этом удаление внутренних граней способно не только уменьшить объем данных, но и существенно повысить скорость генерации и рендеринга HLOD за счет сокращения общего числа обрабатываемых и растеризуемых полигонов, а также ускорения времени подгрузки данных из внешней памяти.

Удаление внутренних граней может быть осуществлено одним из нескольких способов:

- алгоритм наподобие затенения фоновое освещения (англ. ambient occlusion), когда из грани выпускаются лучи, и если хотя бы один луч достиг «неба» (не встретил на пути геометрию), то эту грань можно считать видимой снаружи. Различие с затенением заключается в том, что там вычисляется доля лучей, достигших неба, для расчёта степени освещённости, а в нашем случае достаточно одного луча для вынесения вердикта о видимости грани;
- выпускание лучей снаружи к центру геометрии (равномерно по сфере) с целью обнаружения внешних граней;

- рендеринг геометрии с разных ракурсов, когда в качестве цвета пикселя выводится идентификатор грани; идентификаторы, попавшие во фрейм буфер – это обнаруженные внешние грани.

4. Распараллеливание

После того, как дерево кластеров построено, начинается процесс генерации кластеров, который заключается в создании и упрощении геометрических представлений. Поскольку геометрические представления занимают много памяти, они хранятся во внешней памяти и загружаются в оперативную память в процессе генерации по мере необходимости. Дерево кластеров хранится в оперативной памяти, поскольку оно содержит лишь информацию об ограничивающих объёмах кластеров, их идентификаторах и связях родитель-ребёнок, и поэтому не занимает много места.

Генерация кластеров осуществляется по уровням дерева. Сначала обрабатывается первый уровень, при этом происходит генерация геометрических представлений кластеров из представлений оригинальных объектов. Затем обрабатывается второй уровень, при этом генерируются геометрические представления кластеров второго уровня из представлений кластеров первого уровня. Затем обрабатывается третий уровень, при этом генерируются геометрические представления кластеров третьего уровня из представлений кластеров второго и, возможно, первого уровней (рис. 1 (б)). И так далее, пока не будет обработан последний уровень, содержащий корневой кластер.

Обработка кластеров происходит по уровням, поскольку это позволяет произвести распараллеливание. А именно, кластеры одного уровня не зависят друг от друга, а только от кластеров нижних уровней, поэтому обработка кластеров одного уровня может осуществляться в нескольких параллельных потоках. Обработка одного кластера заключается в следующем:

- полигональные сетки дочерних кластеров загружаются в оперативную память;
- полигональные сетки объединяются в одну новую сетку;
- освобождается оперативная память, выделенная под полигональные сетки дочерних кластеров;
- новая сетка упрощается;
- полученная полигональная сетка кластера сохраняется во внешней памяти;
- освобождается оперативная память, выделенная под полигональную сетку.

При этом между кластерами одного уровня нет зависимости по данным, что позволяет произвести распараллеливание. Следует заметить, что, степень параллелизма падает с ростом уровня в дереве, поскольку на каждом следующем уровне становится меньше кластеров. Также, параллельное исполнение будет увеличивать рабочий расход оперативной памяти. Фактором, который может препятствовать распараллеливанию, является способ хранения геометрии оригинальных объектов сцены и кластеров. Например, если для хранения используется база данных, не предусматривающая параллельное исполнение запросов.

5. Инкрементальные обновления

Для широкого ряда приложений, например, для приложений просмотра 3D моделей, может быть достаточно однократной генерации HLOD с последующим многократным использованием этих данных. В таких приложениях сцена может быть подготовлена один раз (в момент публикации модели) и затем она может многократно просматриваться множеством пользователей на различных клиентских устройствах. Однако, в приложениях для редактирования 3D моделей возникает необходимость быстро обновлять HLOD

представления, чтобы не заставлять пользователя проводить время в ожидании увидеть результат внесённых им изменений.

Если изменения в сцене *глобальные* и меняется кластеризация объектов (дерево кластеров), то осуществить инкрементальные обновления не представляется возможным и требуется заново проводить кластеризацию и построение дерева кластеров, после чего генерировать представления всех кластеров. Если же изменения в сцене *локальные*, например, изменились геометрические представления некоторых объектов сцены, но их положения и ограничивающие объёмы остались близки к прежним, то становится возможным не проводить повторную кластеризацию, а использовать прежнее дерево кластеров. В нём необходимо найти кластеры, в которые попали изменившиеся объекты, и произвести регенерацию представлений только для этих кластеров. Вообще говоря, оценка степени изменений возможна лишь в контексте *ограничений* на HLOD и *метрики качества* HLOD. Так, если HLOD нельзя обновить без нарушения ограничений, требуется полная регенерация. С другой стороны, если обновление HLOD повлечёт ухудшение качества HLOD, это негативно скажется на скорости рендеринга, что может стать поводом провести полную регенерацию.

Набор ограничений зависит от конкретной реализации, но может включать следующие ограничения:

- ограничение на размер кластера в зависимости от уровня, то есть размер ограничивающего объёма кластера на уровне l не должен превышать некоторого порогового значения $w(l)$;
- ограничение на число детей кластера (кластер может иметь не более X детей);
- ограничение на число треугольников в кластере (геометрическое представление кластера может содержать не более X треугольников);

Оценка качества HLOD — это более нетривиальный вопрос. Можно выделить несколько показателей:

- плотность кластеров (минимизация ограничивающего объёма кластера при максимизации числа объектов, вошедших в кластер);
- качество упрощения кластера (минимизация числа треугольников при незначительном увеличении геометрической погрешности);
- лучше иметь небольшое число крупных кластеров, чем много мелких.

Общая стратегия реакции на изменения может быть описана следующим образом:

1. необходимо произвести попытку обновить дерево кластеров (пока без пересчёта геометрии);
2. произвести проверку ограничений для обновлённого дерева;
3. оценить качество обновлённого дерева;
4. если обновлённое дерево удовлетворяет ограничениям и требованиям к качеству, изменения можно считать локальными, в противном случае изменения считаются глобальными.

При *локальных* изменениях осуществляется поиск «устаревших» кластеров. После чего запускается процедура, которая генерирует геометрию только для этих кластеров, не затрагивая остальные. При *глобальных* изменениях осуществляется удаление всех прежних HLOD данных и запускается новая генерация HLOD.

Рассмотрим возможные реакции на следующие типы изменений: добавление новых объектов, удаление объектов, изменение геометрии существующих объектов.

Добавление объектов. Пусть ранее имелось N объектов, которые вошли в K листовых кластеров. Пусть теперь добавилось M объектов. Реакция: если эти M объектов можно

добавить в К кластеров без нарушения ограничений, такие изменения можно считать локальными. В противном случае изменения считаются глобальными.

Удаление объектов. Как правило, удаление объектов не способно нарушить ограничений, поскольку оно ослабляет, а не усиливает характеристики кластеров (уменьшается ограничивающий объём, число треугольников и т.д.). Какие-то кластеры могут опустеть и потребуются их удаление, однако это не является жёстким требованием для повторной кластеризации. Тем не менее, кластеризация может стать менее оптимальной в терминах плотности кластеров (минимизация ограничивающего объёма при максимизации числа детей), числа треугольников в кластере (минимизация общего числа кластеров) и т.д. Число детей кластера может сократиться до одного, что является нежелательным, так как кластер перестаёт быть агрегированным представлением и иерархические уровни детализации вырождаются до обычных уровней детализации. Реакция на изменения: удалить вырожденные кластеры, вычислить новые параметры кластеров (ограничивающий объём, число треугольников, число детей и т.д.) и на основе них вычислить метрики качества HLOD. Если метрики качества ниже допустимых значений, изменения считаются глобальными, в противном случае локальными.

Изменение геометрии существующих объектов. Если изменения не нарушают ограничения, то их можно считать локальными. В противном случае (например, объект передвинулся, что влечёт существенное увеличение ограничивающего объёма кластера, и теперь было бы оптимальней включить этот объект в другой кластер, или же в геометрическом представлении объекта увеличилось число треугольников, что приводит к нарушению ограничения на число треугольников в кластере) изменения считаются глобальными. Если объект передвинулся, можно произвести попытку удалить его кластера и добавить в некоторый другой кластер, с целью не допустить нарушения ограничений и ухудшение качества. Если это удастся сделать для всех передвинувшихся объектов, то изменения можно считать локальными.

Для реализации инкрементальных обновлений HLOD также необходимо уметь отслеживать изменения каждого объекта. Для этого будем считать, что у каждого объекта $s \in S$ имеется временная метка t_s , хранящая дату и время последней модификации объекта. Далее, у каждого HLOD кластера c тоже должна быть временная метка t_c , чтобы можно было понять, что кластер «устарел» и требуется его обновление. В дереве кластеров листовые кластеры формируются из оригинальных объектов сцены, а внутренние кластеры формируются из дочерних кластеров. Таким образом, временные метки нужны только в листовых кластерах, чтобы путём их сравнения с временными метками объектов, вошедших в кластер, определить, «устарел» ли кластер, или нет. Для внутренних кластеров их признак «устарелости» можно определить следующим образом: если хотя бы один из дочерних кластеров является «устаревшим», то этот кластер тоже является «устаревшим». Временную метку кластера можно вычислить как максимум (наиболее позднюю) из временных меток вошедших в него объектов:

$$t_c = \max_{s < c} t_s.$$

Она присваивается каждому листовому кластеру при генерации или обновлении. В начале обновления, при поиске кластеров, требующих обновления, кластер считается «устаревшим», если: $\forall s < c \ t_s > t_c$ (для листовых кластеров) или $\exists c' < c$, такой что c' является устаревшим (для внутренних кластеров). Используя эти правила, признак «устарелости» может быть вычислен для кластеров итеративно по уровням дерева. Сначала он вычисляется для листовых кластеров, затем для их родителей, затем для родителей этих родителей и т.д.

Перегенерация «устаревших» кластеров осуществляется итеративно по уровням дерева. На первой итерации запускается генерация листовых кластеров. При этом те кластеры, у которых признак «устарелости» не выставлен, игнорируются. На следующей итерации

обрабатывается следующий уровень дерева (родители листовых кластеров) и так далее. В целом процесс отличается от обычной генерации кластеров только наличием проверки признака «устарелости» кластера, поэтому в программной реализации первичная генерация кластеров и регенерация «устаревших» кластеров могут быть реализованы одной процедурой. Если кластеры хранятся на диске, каждый кластер в отдельном файле, то обновление кластеров реализуется естественным образом. «Устаревшие» кластеры генерируются и их файлы перезаписываются, в то время как остальные файлы не затрагиваются, но читаются при генерации родительских кластеров.

6. Заключение

Таким образом, предложены и описаны перспективные техники ускорения генерации иерархических уровней детализации HLOD и HDLOD в приложениях компьютерной графики, связанных с консервативным и интерактивным рендерингом больших динамических сцен и нуждающихся в оперативном вычислении и обновлении уровней детализации. В частности, подобные критические требования возникают в приложениях визуального моделирования сложных индустриальных проектов и масштабных инфраструктурных программ. Главное внимание было уделено зарекомендовавшей себя вычислительной стратегии генерации уровней детализации с кластеризацией снизу-вверх, которая имеет потенциал для многочисленных оптимизаций. Для нее предложены и детально описаны техники распараллеливания вычислений кластеров на разных уровнях и инкрементального обновления кластеров при локальных изменениях в трехмерной модели. Также проведен анализ возможностей ускорения вычислений за счет грамотного выбора основного алгоритма упрощения полигональной геометрии, в том числе, с использованием техник удаления невидимых внутренних граней. Данные оптимизации могут применяться изолированно или совместно, а их эффективность может варьироваться в зависимости от специфики приложения и среды исполнения. Параллельная обработка кластеров на отдельных уровнях иерархии позволяет эффективно масштабировать процедуры упрощения геометрии и удаления невидимых граней, выполняя их в параллельных потоках на современных многоядерных процессорах. В приложениях для совместной работы с трехмерными моделями ключевым фактором ускорения может стать применение техники инкрементальных обновлений, позволяющей отследить локальные изменения в кластерах и не тратить вычислительные ресурсы на повторное вычисление всей иерархии уровней детализации. В дальнейших планах авторов программная реализация предложенных техник оптимизации и проведение серий вычислительных экспериментов для подтверждения их релевантности требованиям эффективности и масштабируемости. Эксперименты позволяют также определить области применимости техник оптимизации и выработать рекомендации по их совместному использованию.

Список литературы / References

- [1]. Clark J. H. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*. – 1976. – Т. 19. – №. 10. – С. 547-554.
- [2]. Rossignac J., Borrel P. Multi-resolution 3D approximations for rendering complex scenes. *Modeling in computer graphics: methods and applications*. – Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. – С. 455-465.
- [3]. Heckbert P., Garland M. Multiresolution modeling for fast rendering. *Graphics Interface*. – Canadian Information Processing Society, 1994. – С. 43-43.
- [4]. Erikson C., Manocha D., Baxter III W. V. HLODs for faster display of large static and dynamic environments. *Proceedings of the 2001 symposium on Interactive 3D graphics*. – 2001. – С. 111-120.
- [5]. Varadhan G., Manocha D. Out-of-core rendering of massive geometric environments. *IEEE Visualization, 2002. VIS 2002*. – IEEE, 2002. – С. 69-76.

- [6]. Lakhia A. Efficient interactive rendering of detailed models with hierarchical levels of detail. Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. – IEEE, 2004. – C. 275-282.
- [7]. Семёнов В.А., Шуткин В.Н., Золотов В.А., Морозов С.В. Расширение метода иерархических уровней детализации для динамических сцен с детерминированным характером событий. Труды конференции ГрафиКон-2019, стр. 37-41, DOI: 10.30987/graphicon-2019-1-37-41.
- [8]. Semenov V.A., Shutkin V.N., Zolotov V.A., Morozov S.V., Gonakhchyan V.I. Visualization of Large Scenes with Deterministic Dynamics. *Programming and Computer Software*, 2020, 46(3), pp. 223–232.
- [9]. V. Semenov, V. Shutkin, V. Zolotov. Conservative Out-of-Core Rendering of Large Dynamic Scenes Using HDLODs. Proceedings of the 31st International Conference on Computer Graphics and Vision (GraphiCon 2021), 2021, Nizhny Novgorod, Russia, pp. 105-115, DOI:10.20948/graphicon-2021-3027-105-115.
- [10]. Shutkin V., Semenov V., Zolotov V., Morozkin N. Alternative HDLOD method for large scenes with multiple dynamic behaviors. Proceedings of 17th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, Porto, Portugal 16 – 18 July 2023, pp. 141-148.
- [11]. Garland M., Heckbert P. S. Surface simplification using quadric error metrics. Proceedings of the 24th annual conference on Computer graphics and interactive techniques. – 1997. – C. 209-216.
- [12]. Schroeder W. J., Zarge J. A., Lorensen W. E. Decimation of triangle meshes. Proceedings of the 19th annual conference on Computer graphics and interactive techniques. – 1992. – C. 65-70.
- [13]. Low K. L., Tan T. S. Model simplification using vertex-clustering. Proceedings of the 1997 symposium on Interactive 3D graphics. – 1997. – C. 75-ff.
- [14]. Luebke D. et al. Level of detail for 3D graphics. – Elsevier, 2002.
- [15]. Michaud C., Mellado N., Paulin M. Mesh simplification with curvature error metric. *Eurographics 2017*. – 2017.
- [16]. Liang Y., He F., Zeng X. 3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution. *Integrated Computer-Aided Engineering*. – 2020. – T. 27. – №. 4. – C. 417-435.
- [17]. Hasselgren J. et al. Appearance-Driven Automatic 3D Model Simplification. *EGSR (DL)*. – 2021. – C. 85 -97.
- [18]. Lindstrom P., Turk G. Image-driven simplification. *ACM Transactions on Graphics (ToG)*. – 2000. – T. 19. – №. 3. – C. 204-241.
- [19]. Zhou Z., Chen K., Zhang J. Efficient 3-D scene prefetching from learning user access patterns. *IEEE Transactions on Multimedia*. – 2015. – T. 17. – №. 7. – C. 1081-1095.
- [20]. Chen J. et al. Optimally redundant, seek-time minimizing data layout for interactive rendering. *The Visual Computer*. – 2017. – T. 33. – C. 139-149.
- [21]. Peng C., Cao Y. A GPU-based approach for massive model rendering with frame-to-frame coherence. *Computer Graphics Forum*. – Oxford, UK : Blackwell Publishing Ltd, 2012. – T. 31. – №. 2pt2. – C. 393- 402.
- [22]. Peng C., Cao Y. Parallel LOD for CAD model rendering with effective GPU memory usage. *Computer-Aided Design and Applications*. – 2016. – T. 13. – №. 2. – C. 173-183.
- [23]. Zhao T., Jiang J., Guo X. A Novel Quadratic Error Metric Mesh Simplification Algorithm For 3D Building Models Based On ‘Local-Vertex’ Texture Features. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. – 2022. – T. 48. – C. 109-115.
- [24]. Salinas D., Lafarge F., Alliez P. Structure-aware mesh decimation. *Computer Graphics Forum*. – 2015. – T. 34. – №. 6. – C. 211-227.
- [25]. Li M., Nan L. Feature-preserving 3D mesh simplification for urban buildings. *ISPRS Journal of Photogrammetry and Remote Sensing*. – 2021. – T. 173. – C. 135-150.
- [26]. Ge Y. et al. A Novel LOD Rendering Method with Multi-level Structure Keeping Mesh Simplification and Fast Texture Alignment for Realistic 3D Models. *IEEE Transactions on Geoscience and Remote Sensing*. – 2024.
- [27]. Zhou S. et al. A hybrid level-of-detail representation for large-scale urban scenes rendering. *Computer Animation and Virtual Worlds*. – 2014. – T. 25. – №. 3-4. – C. 243-253.
- [28]. Zhang L. et al. Web-based visualization of large 3D urban building models. *International Journal of Digital Earth*. – 2014. – T. 7. – №. 1. – C. 53-67.
- [29]. Liang J. et al. InfNeRF: Towards Infinite Scale NeRF Rendering with O (log n) Space Complexity. *SIGGRAPH Asia 2024 Conference Papers*. – 2024. – C. 1-11.

- [30]. Kerbl B. et al. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*. – 2024. – Т. 43. – №. 4. – С. 1-15.
- [31]. Pečnik S., Žalik B. Real-time visualization using GPU-accelerated filtering of lidar data. *World Acad. Sci., Eng. Technol., Int. J. Comput., Elect., Automat., Control Inf. Eng.* – 2014. – Т. 8. – №. 12. – С. 2108- 2112.
- [32]. Zhou Y. et al. Efficient Scene Appearance Aggregation for Level-of-Detail Rendering. *arXiv preprint arXiv:2409.03761*. – 2024.
- [33]. Loubet G., Neyret F. Hybrid mesh-volume LoDs for all-scale pre-filtering of complex 3D assets. *Computer Graphics Forum*. – 2017. – Т. 36. – №. 2. – С. 431-442.

Информация об авторах / Information about authors

Василий Николаевич ШУТКИН – инженер ИСП РАН. Сфера научных интересов: компьютерная графика, рендеринг трехмерных сцен, полигональные упрощения, информационное моделирование в области архитектуры и строительства.

Vasily Nikolayevich SHUTKIN – engineer at ISP RAS. Research interests: computer graphics, rendering of 3D scenes, polygonal geometry simplification, building information modeling.

Никита Константинович МОРОЗКИН – инженер ИСП РАН. Сфера научных интересов: компьютерная графика, рендеринг трехмерных сцен, игровые движки, технологии виртуальной и смешанной реальности.

Nikita Konstantinovich MOROZKIN – engineer at ISP RAS. Research interests: computer graphics, rendering of 3D scenes, game engines, virtual and mixed reality.

Виталий Адольфович СЕМЕНОВ – доктор физико-математических наук, профессор, заведующий отделом системной интеграции и прикладных программных комплексов Института системного программирования им. В.П. Иванникова РАН. Сфера научных интересов: модельно-ориентированные технологии программной инженерии, системная интеграция, визуализация и компьютерная графика, вычислительная геометрия, информационное моделирование в области архитектуры и строительства, проектное управление и календарно-сетевое планирование.

Vitaly Adolfovich SEMENOV – Dr. Sci. (Phys.-Math.), Prof., Head of the Department of System Integration and Multi-disciplinary Applied Systems of the Ivannikov Institute for System Programming of the RAS. Research interests: model-driven methodologies and CASE toolkits, system integration, visualization and computer graphics, computational geometry, building information modeling, project management and scheduling.

Олег Анатольевич ТАРЛАПАН – кандидат физико-математических наук, ведущий научный сотрудник ИСП РАН, доцент кафедры системного программирования МГУ им. М.В. Ломоносова. Сфера научных интересов: компьютерная графика и научная визуализация, системы управления базами данных, информационное моделирование в области архитектуры и строительства.

Oleg Anatolevich TARLAPAN – Cand. Sci. (Phys.-Math.), leading researcher of ISP RAS, assistant professor of the department of system programming of Lomonosov Moscow State University. Scientific interests: computer graphics and scientific visualization, database management systems, building information modeling.

DOI: 10.15514/ISPRAS-2025-37(3)-23



Обзор механизмов усиления защищенности операционных систем и пользовательских приложений

¹ Д.В. Ефремов, ORCID: 0000-0002-9916-056X <efremov@ispras.ru>

^{1,2,3} А.К. Петренко, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>

^{1,3,4} Б.А. Позин, ORCID: 0000-0002-0012-2230 <bpozin@ec-leasing.ru>

^{1,5} В.А. Семенов, ORCID: 0000-0002-8766-8454 <sem@ispras.ru>

¹ Институт системного программирования РАН,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

² Московский государственный университет имени М.В. Ломоносова,
Россия, 119991, г. Москва, Ленинские горы, д. 1.

³ НИУ Высшая школа экономики,
Россия, 101978, г. Москва, ул. Мясницкая, д. 20.

⁴ ЗАО ЕС-лизинг, Россия, 117405, Москва, Варшавское шоссе, д. 125.

⁵ Московский физико-технический институт,
Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., 9.

Аннотация. В данной работе представлен систематический обзор механизмов усиления защищенности (hardening) операционных систем и пользовательских приложений. Рассматриваются различные типы защитных механизмов, включая механизмы защиты памяти, защиты аппаратного стека, защиты динамической памяти, рандомизация адресного пространства, защита потока управления и защита целостности системы. Детально анализируются принципы работы данных механизмов, их эффективность и влияние на производительность систем. Особое внимание уделяется реализации защитных механизмов в современных операционных системах, в частности, в ядре Linux. Работа предназначена для специалистов в области информационной безопасности, разработчиков операционных систем и исследователей, занимающихся вопросами защиты информации.

Ключевые слова: защита информации; операционные системы; безопасность приложений; защита памяти; контроль целостности; изоляция; усиление защищенности

Для цитирования: Ефремов Д.В., Петренко А.К., Позин Б.А., Семенов В.А. Обзор механизмов усиления защищенности операционных систем и пользовательских приложений. Труды ИСП РАН, том 37, вып. 3, 2025 г., стр. 325-354. DOI: 10.15514/ISPRAS-2023-37(3)-23.

Благодарности: Работа поддержана компанией “Лаборатория Касперского” в рамках проекта “Анализ мирового уровня техники по архитектурным средствам обеспечения доверия”.

Overview of Hardening Mechanisms in Operating Systems and User Applications

¹ D.V. Efremov, ORCID: 0000-0002-9916-056X <efremov@ispras.ru>

^{1,2,3} A.K. Petrenko, ORCID: 0000-0001-7411-3831 <petrenko@ispras.ru>

^{1,3,4} B.A. Pozin, ORCID: 0000-0002-0012-2230 <bpozin@ec-leasing.ru>

^{1,5} V.A. Semenov, ORCID: 0000-0002-8766-8454 <sem@ispras.ru>

¹ Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

² Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia.

³ National Research University, Higher School of Economics,
20, Myasnitskaya ulitsa, Moscow, 101978, Russia.

⁴ “EC-leasing” Co., 125 Varshavskoye shosse, Moscow, 117405, Russia.

⁵ Moscow Institute of Physics and Technology,
9, Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.

Abstract. This paper presents a systematic review of hardening mechanisms for operating systems and user applications. Various types of protection mechanisms are discussed, including memory protection mechanisms, hardware stack protection, dynamic memory protection, address space randomization, control flow protection, and system integrity protection. The principles of these mechanisms, their effectiveness, and their impact on system performance are analyzed in detail. Special attention is given to the implementation of protective mechanisms in modern operating systems, particularly in the Linux kernel. This work is intended for information security specialists, operating system developers, and researchers working on information security issues.

Keywords: information security; operating systems; application security; memory protection; integrity control; isolation; security hardening; hardening.

For citation: Efremov D.V., Petrenko A.K., Pozin B.A., Semenov V.A. Overview of Hardening Mechanisms in Operating Systems and User Applications. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 3, 2025., pp. 325-354. DOI: 10.15514/ISPRAS-2023-37(3)-23.

Acknowledgements. The work was supported by Kaspersky Lab as a part of the project “Analysis of world-class technology in architectural means of ensuring trust”.

1. Введение

Механизмы усиления защищенности (hardening) представляют собой комплекс технологий и методов, направленных на минимизацию рисков эксплуатации уязвимостей в программных системах. В условиях постоянно растущего числа и сложности кибератак, внедрение таких механизмов становится критически важным. Объектом внимания мероприятий, нацеленных на усиление защищенности, является как программный код, так и описания конфигураций собственно программных компонентов системы и сетевой составляющей. Иногда в тематику защищенности включают и вопросы контроля за полномочиями персонала. В данной статье мы ограничиваемся вопросами усиления защиты программного кода.

Усиление защищенности применяется к коду, который потенциально может подвергаться попыткам компрометации – коду, работающему с ограниченными ресурсами или имеющему доступ к конфиденциальной информации. Часто это низкоуровневый код операционных систем (ОС) или системных программ, от безопасности которых зависит защищенность всей вычислительной системы.

Современные кибератаки характеризуются значительным усложнением техник эксплуатации, что требует применения комплексных подходов к защите. Эффективное противодействие таким атакам возможно только при взаимодействии исследователей безопасности, разработчиков ОС, компиляторов и производителей аппаратного обеспечения. То есть усиление защищенности как система подразумевает рассмотрение разных аспектов конструирования программной системы и видов взаимодействия различных компонентов (например, межпроцессорные взаимодействия, связи в многопоточных программах, сетевые взаимодействия). Совокупность прямых и опосредованных связей в современной компьютерной системе крайне сложна, поэтому высока роль анализа архитектуры системы на различных уровнях детализации и привлечение методов архитектурного моделирования и анализа на всех фазах жизненного цикла вычислительной системы.

Комплексный подход к усилению защищенности методически существенно отличается от практик, которые ранее так или иначе использовались специалистами по информационной безопасности и администраторами компьютерных систем. Если старые методы в большей степени придерживались тактики реагирования на вновь обнаруженные уязвимости, то новая методология рассматривает усиление защищенности как планомерный процесс, в котором всегда предусматривается возможность появления проблем информационной безопасности и в проактивной форме реализуются меры по их нейтрализации. Важным инструментом планирования такой проактивной деятельности также является архитектурный анализ. Отметим, что следуя принципам конструктивной безопасности, заботиться о защищенности системы нужно начиная с фазы ее замысла, однако при этом тогда, когда система уже создана и эксплуатируется, тоже необходимо, возвращаться к проблемам, которые рассматривались в ходе ее проектирования и разработки, и иногда проводить ретроспективный анализ вплоть до фазы замысла. Усиление защищенности, один из ориентиров, которые помогает выстроить систематический процесс анализа и постоянного улучшения в соответствии с принципами конструктивной информационной безопасности.

Мы можем увидеть результаты планомерного архитектурного подхода к усилению защиты комплексного программного обеспечения на примере проектов OpenBSD [1], Linux Kernel Self-Protection Project (KSPP) [2], Qubes OS [3], GrSecurity [4], Pax Project [5], GrapheneOS [6] и других.

В данной работе рассматриваются основные категории механизмов усиления защищенности, их принципы работы и реализация в современном программном обеспечении, а также перспективы их развития. Мы намеренно ограничиваемся обзором механизмов усиления, которые уже внедрены в большие открытые проекты и применимы в продуктовых сборках, а не только в качестве средства отладки. Существует большое количество исследовательских работ по механизмам усиления. Однако, их реальное внедрение в проект может быть сильно осложнено проблемами совместимости с существующим кодом, сложностью самого внедрения и ее последующей поддержкой, а также иными моментами, упущенными из рассмотрения в исходных исследовательских работах. Мы рассматриваем факты внедрения защиты в открытые проекты и одобрение со стороны сообщества разработчиков как фильтр, доказывающий на практике высокую ценность технологии и её зрелость.

2. Систематизация механизмов усиления защищенности

В современной парадигме информационной безопасности механизмы усиления защищенности (hardening) представляют собой фундаментальный комплекс мер, направленных на минимизацию поверхности атаки и усиления защитных свойств как ОС, так и пользовательских приложений. Данные механизмы формируют многоуровневую систему защиты, охватывающую все аспекты функционирования программно-аппаратных комплексов: от архитектурного проектирования до этапа эксплуатации. Систематизация данных механизмов позволяет не только структурировать существующие подходы к

обеспечению безопасности, но и выявить их взаимосвязи, что важно при формировании целостной стратегии защиты информационных систем.

2.1 Концептуальные основы усиления защищенности

Усиление защищенности как методология базируется на принципе многоуровневой защиты (defense-in-depth), предполагающем выстраивание последовательных барьеров, преодоление каждого из которых требует от потенциального злоумышленника дополнительных ресурсов и компетенций.

Усиление защиты можно рассматривать в контексте разницы между стоимостью её внедрения, использования и увеличением стоимости компроментации программы. Хорошим можно считать такое усиление, которое кратно увеличивает стоимость компроментации программы по отношению к стоимости его внедрения и использования, где под стоимостью следует понимать не только прямое денежное измерение задачи, но также временные затраты и уровень специалистов, необходимых для решения задачи. Так, если простое использование специальных опций сборки программы, автоматом закрывает некоторый широко распространённый класс уязвимостей, то его внедрение требует минимальных усилий и компетенций, а практическим результат выражается в повышенных требованиях к квалификации взломщика, его временных затрат и возможности стабильного взлома программы.

В основе данного подхода лежит понимание того, что абсолютная безопасность недостижима, а эффективная защита должна основываться на комплексном применении разнородных механизмов, компенсирующих недостатки друг друга. Современные механизмы усиления защищенности можно классифицировать на основе разных признаков.

Например, на стадии жизненного цикла программного обеспечения можно выделить механизмы проектирования безопасности (архитектурный анализ, моделирование угроз), разработки безопасного кода (безопасные библиотеки, статический анализ кода), безопасной компиляции и сборки (защиты компилятора), защиты при эксплуатации (динамические средства защиты).

По уровню реализации механизмы усиления можно выделить аппаратные механизмы (NX-бит, защита целостности потока управления на уровне процессора), механизмы на уровне гипервизора/ядра ОС (KASLR, защита стека ядра), механизмы уровня пользовательского пространства (ASLR), прикладные механизмы (защита на уровне приложений). Многие из защит не специфичны для какого-то уровня и могут быть применены на каждом из уровней в не зависимости друг от друга.

Также можно классифицировать виды защиты по типу предотвращаемых угроз. Например, защита от переполнения буфера, защита от инъекций кода, утечек данных, повышения привилегий и др. Классификация самих угроз достаточно обширна и до сих пор пополняется, так как появляются новые вектора атак. Наиболее известная из таких классификаций Common Weakness Enumeration (CWE) [7] представляет собой реестр, созданный организацией MITRE. Реестр представляет собой многоуровневую древовидную структуру, состоящую из типов и определений различных дефектов и недостатков безопасности программного обеспечения, возникающих на стадиях проектирования, разработки, интеграции и эксплуатации информационных систем.

Хорошим примером классификации усилений защиты по типу предотвращаемых угроз может служить карта защитных механизмов ядра Linux [8].

2.2 Архитектурный анализ и моделирование угроз

Архитектурный анализ представляет собой этап обеспечения безопасности информационных систем, предшествующий непосредственной реализации защитных механизмов. Данный процесс включает в себя декомпозицию системы на компоненты, анализ их взаимодействия

и выявление потенциальных уязвимостей, возникающих на границах взаимодействия. Результаты архитектурного анализа служат основой для последующего моделирования угроз.

Моделирование угроз [9,10] является методологическим подходом к систематическому выявлению, оценке и приоритизации потенциальных угроз безопасности системы. Наиболее распространенными методологиями моделирования угроз являются Microsoft STRIDE, P.A.S.T.A (Process for Attack Simulation and Threat Analysis) и OCTAVE. Каждая из этих методологий предлагает структурированный подход к идентификации угроз, классифицируя их по различным категориям и связывая с конкретными компонентами анализируемой системы. Процесс моделирования угроз включает следующие этапы:

- Идентификация “активов”, подлежащих защите;
- Документирование архитектуры системы;
- Декомпозиция системы на компоненты;
- Идентификация угроз для каждого компонента;
- Ранжирование угроз по уровню риска;
- Определение стратегий минимизации рисков.

Результаты моделирования угроз напрямую определяют выбор и конфигурацию механизмов усиления защищенности, обеспечивая целенаправленное укрепление наиболее уязвимых аспектов системы.

2.3 Анализ рисков информационной безопасности

Анализ рисков представляет собой процесс оценки вероятности реализации угроз и потенциального ущерба от них, что позволяет оптимизировать распределение ресурсов при внедрении механизмов защиты. Современные методологии анализа рисков включают количественные и качественные подходы, такие как FAIR (Factor Analysis of Information Risk) [11], NIST Risk Management Framework (RMF) [12] и ISO 27005 [13]. Ключевыми компонентами анализа рисков являются:

- *Идентификация рисков* – выявление потенциальных сценариев реализации угроз;
- *Оценка вероятности* – определение вероятности реализации каждого сценария;
- *Оценка воздействия* – анализ потенциального ущерба при реализации угроз;
- *Приоритизация рисков* – ранжирование рисков на основе их вероятности и воздействия;
- *Выбор стратегии управления рисками* – принятие, передача, снижение или избежание рисков.

Результаты анализа рисков служат основой для формирования стратегии усиления защищенности, определяя приоритетные направления и оптимальный уровень инвестиций в механизмы защиты.

Архитектурный анализ информационных систем, в комбинации с методологиями моделирования угроз и формализованными фреймворками анализа рисков, формирует комплексный, многоаспектный подход к идентификации уязвимостей, что непосредственно влияет на реализацию мер по усилению безопасности. В частности, архитектурный анализ создает фундаментальное понимание структуры системы, включая точки взаимодействия компонентов, каналы передачи данных и границы доверия, в то время как моделирование угроз (STRIDE) обеспечивает систематическую идентификацию потенциальных угроз. Данные, полученные в результате этих процессов, подвергаются количественной и качественной оценке посредством методологий анализа рисков, где FAIR предлагает экономическую модель оценки потенциальных потерь, NIST RMF фокусируется на соответствии нормативным требованиям, а ISO 27005 предоставляет структурированный подход к управлению информационными рисками.

Усиление безопасности преобразует выявленные риски и уязвимости в конкретные технические и организационные меры защиты, приоритезированные согласно их потенциальному воздействию на критические активы. Данный интегративный подход позволяет организациям разрабатывать стратегии усиления безопасности, который не только противодействует идентифицированным угрозам через целенаправленное укрепление уязвимых компонентов архитектуры, но и оптимизируют распределение ресурсов безопасности с учетом экономической эффективности защитных мер относительно стоимости потенциальных инцидентов. Синтез архитектурного анализа, моделирования угроз и оценки рисков создает научно обоснованный фундамент для разработки дифференцированных, контекстуально соответствующих стратегий усиления безопасности, повышая общую устойчивость информационных систем к киберугрозам. Таким образом, hardening переходит из области лучших практик в дисциплинированную реализацию мер, направленных на минимизацию конкретных, формально выявленных и оценённых рисков.

Рассмотрев общий контекст, в который вписывается усиление защищенности приложений, а также с какой точки зрения они могут и должны оцениваться, перейдем к рассмотрению самих технологий. Мы выстраиваем наш обзор различных технологий усиления защищенности ОС и пользовательских приложений от типа защищаемого ресурса, степени зрелости и уровня внедрения технологии. При этом, мы намеренно ограничиваемся защитами на уровне кода самого приложения (как исходного, так и бинарного), оставляя такие темы как защита от зловредных периферийных устройств, системы контроля доступа, антивирусные технологии, системы предотвращения утечек информации, системы обнаружения вторжений (IDS), защиты конечных точек (EDR), и многие другие за пределами нашего обзора.

Рассматриваемые технологии реализуются на уровне загрузчиков, ядер ОС, в системных библиотеках, средах выполнения (runtime), программах пользовательского пространства и компиляторах. Меры усиления построены на принципах рандомизации, проверки границ, изоляции и ограничения времени жизни и доступа, контроля целостности, дублирования вычислений. Архитектура фон Неймана, подразумевает принцип однородности памяти: хранения данных и инструкций в одном разделе памяти. Поэтому мы начинаем наш обзор с механизмов защиты памяти. Защита кода основывается на механизмах защиты памяти и дополняет их, поэтому она продолжает этот обзор. В отдельную категорию мы выносим контроль целостности. Несмотря на то, что сам принцип лежит в основе множества защит, рассматриваемых в прошлых разделах, мы считаем необходимым поместить рассматриваемые технологии в отдельный раздел, так как они не могут быть применены к программам автоматически и планирование их внедрения происходит на архитектурном уровне. По тому же принципу мы выделяем в отдельный раздел механизмы самоограничения процессов.

3. Механизмы защиты памяти

Механизмы защиты памяти обеспечивают контроль над использованием памяти программой и предотвращают эксплуатацию уязвимостей, связанных с нарушением целостности данных или несанкционированным доступом.

В программах можно выделить различные регионы памяти по типу их предназначения и тому, как они используются. Такие как: стек вызовов, динамическая память (куча, heap), кеш и регистры процессора, память для хранения данных и кода программы, регионы памяти для взаимодействия с аппаратурой (MMIO, PMIO, DMA). Атаки на память в общем виде сводятся к несанкционированному чтению из неё и записи в неё. Механизмы защиты в общем виде можно свести к выставлению разрешений на доступ к памяти, проверке границ доступа к памяти, к разделению и изоляции регионов памяти друг от друга, к рандомизации адресов

памяти и смещений в ней, к принудительной инициализации и затиранию регионов памяти, к проверке целостности данных и согласованности структур данных, хранящихся в памяти.

3.1 Механизмы защиты памяти

Рассмотрим наиболее известные и распространенные механизмы защиты памяти. Специализированные механизмы для конкретного типа памяти учитывают специфику использования данного типа памяти и векторов атак на него. Как правило, за счет этого они накладывают меньшие накладные расходы на производительность чем общие механизмы защиты. Ограничимся рассмотрением механизмов, которые применяются на практике в ОС со страничной организацией памяти, работающих на вычислительных машинах с архитектурой фон Неймана.

Рандомизация размещения адресного пространства

Рандомизация размещения адресного пространства Address Space Layout Randomization (ASLR) [14, 15] представляет собой механизм обеспечения безопасности, основной целью которого является усложнение эксплуатации уязвимостей программного обеспечения, основанных на знании конкретных адресов в памяти. Суть данного подхода заключается в случайном изменении расположения ключевых областей данных и кода в виртуальном адресном пространстве процесса при каждом его запуске.

В контексте ядра ОС этот механизм известен как Kernel Address Space Layout Randomization (KASLR) [16] в Linux. KASLR предназначен для рандомизации адресов, по которым загружается код самого ядра и размещаются его структуры данных. KASLR иницируется на этапе загрузки системы. Этот процесс включает генерацию случайного значения смещения в пределах допустимого диапазона адресов, последующее перемещение сегментов ядра в соответствии с этим смещением и обновление системных структур управления памятью, таких как таблицы страниц и дескрипторы памяти, для отражения нового, рандомизированного расположения. При успешной реализации KASLR увеличивает сложность выполнения атак, требующих точного знания адресов в памяти ядра, тем самым затрудняя обход защитных механизмов или использование уязвимостей, зависящих от статических адресов.

Для эффективного применения принципов ASLR используется технология создания исполняемых файлов, независимых от позиции загрузки (Position Independent Executables, PIE). PIE позволяет ОС загружать код и данные программы в произвольные участки виртуальной памяти. Это достигается за счет того, что исполняемый файл генерируется с использованием относительных адресов вместо абсолютных. При загрузке такой программы ОС, применяя механизмы ASLR, определяет случайные базовые адреса для ее сегментов (кода, данных, стека, кучи). Все внутренние ссылки в программе затем пересчитываются во время выполнения относительно этих динамически определенных базовых адресов.

Слабым местом защиты является то, что смещения в структурах между полями, между функциями, между сегментами памяти является константным. Имея собранную программу, злоумышленник может вычислить эти смещения для использования в эксплойте. Если в ядре или программе не применяется дополнительных техник противодействия утечкам информации, которые мы рассмотрим далее, то злоумышленнику достаточно одной такой ошибки, чтобы узнать адрес и имея предварительно вычисленные смещения обойти защиту ASLR.

В OpenBSD для решения проблемы постоянства смещений, упомянутой выше, реализована защита Kernel Address Randomized Link (KARL) [17]. В отличие от KASLR, который рандомизирует только базовый адрес загрузки ядра, KARL при каждой загрузке системы заново компоует ядро. В процессе этой перекомпоновки объектные файлы, составляющие ядро, связываются в случайном порядке. Это приводит к тому, что относительные смещения между различными функциями и структурами данных внутри самого ядра становятся непредсказуемыми от загрузки к загрузке. Защита KARL значительно усложняет

эксплуатацию уязвимостей, основанных на знании фиксированных смещений внутри ядра. Это делает эксплойты, полагающиеся на такие смещения, непереносимыми между перезагрузками одной и той же системы.

Политика W^X (Write XOR Execute)

Политика “запись, ЛИБО выполнение (W^X)” или запрета выполнения данных (DEP) воплощает принцип разделения данных и кода. Согласно этой политике, страница памяти может быть либо доступна для записи, либо для выполнения, но никогда одновременно и для того, и для другого. Этот подход усложняет эксплуатацию уязвимостей, поскольку даже если атакующему удастся записать вредоносный код в память, он не сможет его выполнить, и если он найдет способ выполнить код из какой-либо области, он не сможет ее модифицировать.

Реализация опирается на аппаратную поддержку, предоставляемую современными процессорами (например, бит NX – No-eXecute у компании AMD, или XD – Execute Disable у компании Intel), и ядро ОС, которое соответствующим образом настраивает атрибуты страниц памяти [18-20].

Защита W^X применяется не только к программам пользовательского пространства. Подобная защита может быть реализована и в ядре для защиты его кода [21-22]. Этот же принцип реализуется в механизме NX-physmap [23], который представляет собой усиление безопасности ядра ОС, направленную на предотвращение исполнения кода из области прямого отображения физической памяти (physmap).

Запрет отображение памяти в нижней части адресного пространства процесса

Механизм MMAP_MIN_ADDR [24] предотвращает отображения памяти в наиболее низких адресах виртуального адресного пространства процесса. Ключевая задача данного механизма – исключить возможность для процессов создавать отображения, начинающиеся с нулевого адреса, что исторически являлось вектором атак на ядро Linux.

Уязвимость, которую устраняет MMAP_MIN_ADDR, заключалась в том, что разыменования нулевого указателя (NULL pointer dereference) в коде ядра могла быть эксплуатирована злонамеренным процессом. Такой процесс мог предвительно отобразить контролируемую им страницу памяти по нулевому адресу, разместить на ней вредоносный код и затем спровоцировать ошибку в ядре. Это приводило к тому, что ядро, пытаясь обратиться по нулевому адресу, фактически передавало управление вредоносному коду, который исполнялся с привилегиями ядра, компрометируя всю систему.

Для противодействия описанной угрозе параметр ядра MMAP_MIN_ADDR устанавливает минимально допустимый адрес, с которого процессу разрешено запрашивать выделение памяти посредством системного вызова `mmap()`. Любые попытки отобразить память по адресу ниже этого установленного порога отклоняются ядром.

Аппаратные защиты SMAP и SMEP разграничения адресных пространств пользователя и ядра

Механизмы предотвращения доступа в режиме супервизора (Supervisor Mode Access Prevention, SMAP) [25] и предотвращения исполнения в режиме супервизора (Supervisor Mode Execution Prevention, SMEP) [26] представляют собой аппаратные функции безопасности, реализованные в современных процессорах, предназначенные для усиления изоляции между пространством ядра (режим супервизора) и пространством пользователя. Эти механизмы направлены на предотвращение эксплуатации уязвимостей ядра, которые могли бы позволить атакующему получить контроль над системой.

SMEP предотвращает исполнение кода, находящегося на страницах памяти, принадлежащих пространству пользователя, когда процессор работает в режиме ядра (режим супервизора). Если ядро, работая с наивысшими привилегиями, пытается выполнить инструкцию, расположенную в памяти, помеченной как пользовательская (например, из-за уязвимости, позволяющей атакующему контролировать указатель инструкций ядра), процессор генерирует исключение. Это значительно усложняет классические атаки, при которых

злоумышленник размещает вредоносный код в пользовательском пространстве и затем обманом заставляет ядро перейти на его исполнение. Без SMEP, если атакующий мог перенаправить поток выполнения ядра на адрес в пользовательском пространстве, то он выполнить произвольный код с привилегиями ядра. SMEP блокирует такую возможность на аппаратном уровне, требуя, чтобы код, исполняемый ядром, находился исключительно в памяти, принадлежащей ядру [27].

SMAP расширяет защиту, предоставляемую SMEP, предотвращая произвольный доступ (чтение или запись) ядра к данным, находящимся на страницах памяти пространства пользователя, если такой доступ не разрешен явно. По умолчанию, любая попытка ядра прочитать или записать данные в пользовательском пространстве (за исключением стека инструкций, который покрывается SMEP) приведет к генерации исключения. Для легитимного обмена данными между ядром и пространством пользователя (например, при системных вызовах с использованием функций `copy_to_user` или `copy_from_user` в Linux) существуют специальные механизмы, позволяющие временно и контролируемо обойти ограничения SMAP [28].

SMEP и SMAP совместно создают барьер, затрудняющий эксплуатацию многих типов ошибок в ядрах ОС. SMEP защищает от прямого исполнения пользовательского кода ядром, а SMAP – от несанкционированного доступа ядра к пользовательским данным. Эти механизмы значительно повышают сложность взлома для атакующих. Включение SMEP и SMAP является стандартной практикой в современных ОС для усиления их безопасности и снижения поверхности атаки на ядро.

Рандомизация полей структур при сборке

Механизм рандомизации полей структур, известный как RANDSTRUCT [29,30], представляет собой технику усиления защиты ядра ОС, направленную на усложнение эксплуатации уязвимостей, связанных с повреждением памяти и утечкой информации. Основная цель RANDSTRUCT заключается в том, чтобы сделать непредсказуемым внутреннее расположение (смещения) полей в критически важных структурах данных ядра.

Принцип действия RANDSTRUCT основан на изменении порядка следования полей внутри структур ядра на этапе сборки. Это достигается с помощью специального плагина для компилятора GCC, который случайным образом переставляет поля в структурах, помеченных для рандомизации, при каждой сборке ядра. Таким образом, смещения полей относительно начала структуры становятся уникальными для каждой конкретной сборки ядра, что делает невозможным создание эксплойтов, полагающихся на фиксированное расположение полей.

Внедрение RANDSTRUCT затрудняет атаки, основанные на знании точного расположения полей в структурах ядра. Злоумышленники часто полагаются на фиксированные смещения для чтения или перезаписи определенных данных внутри структур с целью повышения привилегий или обхода защитных механизмов. Рандомизация делает такие атаки ненадежными, поскольку смещения полей становятся неизвестными заранее. Также RANDSTRUCT снижает вероятность успешной эксплуатации уязвимостей, приводящих к утечке информации, так как атакующему сложнее определить, какие именно данные он считывает из памяти. Этот механизм эффективно дополняет другие техники усиления защиты, такие как KASLR.

Необходимо отметить, что RANDSTRUCT является защитой времени сборки. Если злоумышленнику удастся получить доступ к конкретному собранному образу ядра, он сможет определить актуальное расположение полей. Эта защита не имеет смысла, если собранные образы ядра выкладываются публично. Однако, RANDSTRUCT значительно повышает барьер для разработки универсальных эксплойтов, работающих на различных сборках ядра, и усложняет анализ ядра для поиска уязвимостей.

Санитайзеры (ASan, KASan, UBSan) и KFENCE

Санитайзеры представляют собой набор инструментов динамического анализа, предназначенных для обнаружения различных типов ошибок в программах во время их выполнения. К наиболее известным относятся AddressSanitizer (ASan) и UndefinedBehaviorSanitizer (UBSan) [31].

AddressSanitizer (ASan) разработан для выявления ошибок работы с памятью в приложениях пользовательского пространства, таких как использование памяти после освобождения (use-after-free), переполнения буферов на куче или стеке, а также утечки памяти. ASan характеризуется существенными накладными расходами на производительность и увеличением потребления памяти, что обычно исключает его применение в стандартных производственных сборках. KernelAddressSanitizer (KASan) [32] функционирует аналогично ASan, но предназначен для ядра ОС, помогая обнаруживать схожие ошибки доступа к памяти в коде ядра. Как и ASan, KASan значительно снижает производительность и увеличивает потребления памяти, что делает его непригодным для повсеместного использования в производственных версиях ядер. Его основное назначение – отладка и тестирование ядра разработчиками. Однако, падение производительности может быть сведено до приемлемого уровня в случае использования аппаратных механизмов тегирования памяти, что делает использование ASan актуальным для усиления защищенности.

UndefinedBehaviorSanitizer (UBSan) нацелен на обнаружение различных видов неопределенного поведения в программах, например, переполнения знаковых целых чисел, некорректных операций сдвига, использования невыровненных указателей и других подобных ошибок неопределённого поведения. UBSan обладает значительно меньшими накладными расходами по сравнению с ASan и KASan. Некоторые из его проверок могут быть достаточно легковесными для использования в производственных сборках, особенно для критически важных участков кода.

Kernel Electric Fence (KFENCE) [33] представляет собой механизм обнаружения ошибок памяти в ядре Linux, специально спроектированный для использования в производственных системах. В отличие от KASan, KFENCE отличается низкими накладными расходами. Его принцип работы заключается в выделении небольшого процента объектов кучи на специально защищенных страницах памяти. Такой подход позволяет с высокой вероятностью обнаруживать ошибки типа выхода за границы буфера и использование после освобождения для этих выборочных аллокаций. Благодаря своей низкой ресурсоемкости, KFENCE может быть активирован в производственных ядрах для выявления трудноуловимых ошибок памяти без существенного негативного влияния на общую производительность системы.

Memory Tagging Extension (MTE) [34] является аппаратной функцией, реализованной в некоторых современных процессорах архитектуры ARM, и предоставляет механизм для обнаружения ошибок памяти с низкими накладными расходами. MTE позволяет ассоциировать теги как с указателями, так и с областями памяти. При каждом доступе к памяти процессор проверяет соответствие тега указателя тегу области памяти. Несовпадение тегов указывает на потенциальную ошибку, такую как выход за границы выделенной памяти или использование памяти после ее освобождения. Инструменты-санитайзеры, например, Hardware-assisted AddressSanitizer (HWASan) [35], могут использовать MTE для эффективного обнаружения ошибок памяти в производственных условиях, оказывая значительно меньшее влияние на производительность по сравнению с чисто программными реализацией ASan.

Очистка регистров, используемых вызываемой функцией (Call-used register wiping)

Механизм очистки регистров, используемых вызываемой функцией (call-used register wiping) [36,37], представляет собой меру по усилению безопасности, направленную на предотвращение утечек конфиденциальной информации через регистры процессора. Основная цель данной техники заключается в минимизации риска того, что данные,

оставшиеся в регистрах после выполнения одной функции, могут быть доступны другой, потенциально контролируемой злоумышленником, функции.

Принцип действия механизма основан на принудительной очистке содержимого регистров, которые, согласно соглашению о вызовах конкретной архитектуры, считаются “используемыми вызываемой функцией”. Это означает, что вызывающая функция не может полагаться на сохранение их значений после вызова другой функции. Очистка обычно происходит либо непосредственно перед возвратом из функции, либо при входе в новую функцию, особенно в критически важных компонентах системы, таких как ядро ОС. Защита реализуется на уровне компилятора.

Внедрение очистки регистров, используемых вызываемой функцией, способствует повышению общей безопасности системы. Во-первых, это значительно снижает поверхность атаки для утечек данных, поскольку чувствительная информация, временно хранящаяся в регистрах, не остается доступной после завершения операции, для которой она была необходима. Во-вторых, данный механизм усложняет эксплуатацию некоторых типов уязвимостей, так как злоумышленник, получивший контроль над потоком выполнения, не сможет легко извлечь полезные данные из регистров, если они были предварительно очищены. Эта техника дополняет другие меры по усилению защиты, такие как рандомизация адресного пространства и защита стека, создавая более устойчивую к атакам среду. Хотя очистка регистров может вносить некоторые накладные расходы на производительность из-за дополнительных инструкций, ее применение оправдано в системах с высокими требованиями к безопасности.

Проверка границ при вызове стандартных функций

Механизм FORTIFY_SOURCE [38] представляет собой меру по усилению безопасности, реализованную в компиляторах и предназначенную для обнаружения и предотвращения некоторых распространенных типов уязвимостей, связанных с переполнением буфера, при работе со стандартными функциями библиотеки языка C.

Основной принцип действия FORTIFY_SOURCE заключается в замене небезопасных версий стандартных библиотечных функций (например, `memcpy()`, `strcpy()`, `sprintf()`) на их более защищенные аналоги. Эти аналоги включают дополнительные проверки во время компиляции и во время выполнения, чтобы убедиться, что операции с памятью не выходят за пределы выделенного буфера назначения. Компилятор использует информацию, полученную в ходе оптимизационных проходов, для определения размеров буферов.

Когда FORTIFY_SOURCE обнаруживает потенциальное переполнение буфера во время компиляции, он выдает ошибку. Если проверка происходит во время выполнения и обнаруживается попытка записи за пределы буфера, программа обычно аварийно завершается, что предотвращает эксплуатацию уязвимости, которая могла бы привести к выполнению произвольного кода или повреждению данных.

Преимущества использования FORTIFY_SOURCE заключаются в повышении устойчивости программного обеспечения к атакам, эксплуатирующим переполнения буфера в стандартных функциях. Это достигается без необходимости модификации исходного кода приложения, так как защита встраивается компилятором.

Для функций ядра, осуществляющих копирование из регионов памяти пользовательского пространства и в них могут быть реализована аналогичная защита. Например, для ядра Linux была реализована защита PAX_USERCOPY и включена в основную ветвь разработки под названием HARDENED_USERCOPY [39].

Пометка флагом “только для чтения” записей в таблице перемещений

Технология RELRO (Relocation Read-Only) [40] направлена на усиление безопасности исполняемых файлов и разделяемых библиотек путем установления атрибута “только для чтения” для определенных участков памяти после их первоначальной инициализации динамическим компоновщиком. Ключевой задачей является защита внутренних структур

данных, используемых для разрешения адресов символов во время выполнения, в частности Global Offset Table (GOT), от несанкционированной модификации злоумышленником.

Применение RELRO обеспечивает существенное повышение уровня безопасности, поскольку полностью исключает возможность модификации таблицы GOT после завершения процесса загрузки программы. Это эффективно противодействует атакам, основанным на перезаписи указателей в GOT с целью перенаправления вызовов легитимных функций на вредоносный код. Подобные атаки часто применяются для обхода других защитных механизмов, таких как W^X.

В качестве компромисса за повышенную безопасность выступает некоторое увеличение времени запуска программы при использовании RELRO. Это обусловлено необходимостью разрешения всех символов до начала выполнения основного кода, а не по мере их использования. Для большинства приложений данное увеличение времени запуска является незначительным, однако оно может быть более заметным для программ, имеющих большое количество динамических зависимостей.

3.2 Механизмы защиты аппаратного стека

Стек в программе – это область памяти, которая используется для хранения локальных переменных функций, адресов возврата и других данных, необходимых для корректной работы программы. Он работает по принципу LIFO (Last In, First Out – “последний пришёл, первый ушёл”), что означает, что последний элемент, добавленный в стек, будет первым извлечён из него. Стек является уязвимым местом, которое может быть использовано злоумышленниками в худшем случае для выполнения произвольного кода и изменения поведения программы. Однако, на нем также могут храниться чувствительные данные (пароли, ключи шифрования), несанкционированное считывание которых приводит к утечке информации.

Одна из наиболее распространённых уязвимостей связана с переполнением буфера в стеке. Если программа не проверяет границы массива или не контролирует количество данных, записываемых в буфер, злоумышленник может записать данные за пределы выделенной области памяти и изменить содержимое стека.

Стековые канарейки

Самым известным механизмом защиты стека является StackGuard [41,42]/Stack Smashing Protector [43]. Он основан на концепции “канареек” (stack canaries). Этот метод защиты получил своё название по аналогии с канарейками, которых шахтёры брали с собой в шахты для раннего обнаружения опасных газов.

Механизм работает путем размещения специального контрольного значения (канарейки) между локальными переменными функции и служебной информацией стека, такой как адрес возврата. При входе в функцию это значение записывается в стек, а при выходе проверяется. Если значение изменилось, это означает, что произошло переполнение буфера, которое могло повредить важные данные стека.

Существует три основных типа канареечных значений, каждый со своими особенностями защиты: терминаторные канарейки, использующие символы-терминаторы (NULL, CR, LF, EOF, ...) для предотвращения переполнения при использовании строковых функций; случайные канарейки; и случайные XOR-канарейки, дополнительно комбинируемые с нестатическим значением (обычно регистр %rbp), что в современных ОС с рандомизацией адресного пространства обеспечивает дополнительную защиту. В x86-64 Linux с glibc, где используется 64-битное случайное значение с обнуленными последними байтами.

Метод встроен в современные компиляторы, такие как GCC, Clang и включается по умолчанию. Это позволяет осуществлять сборку программного обеспечения со встроенной защитой. Существуют разные степени глубины внедрения проверок, от проверки каждой функции и до полного отключения защиты, что позволяет найти баланс между

производительностью и безопасностью. Сбалансированным подходом выступает эвристика оценки степени задействованности стека функцией. Если функция активно использует стек, хранит на нем значений больше определенного порога, вызывает сторонние функции, работающие со значениями на стеке, то срабатывает эвристика и компилятор добавляет проверку контрольного значения со стека в функцию. Помимо этого, компиляторы GCC и Clang для повышения эффективности защиты размещают переменные на стеке специальным образом: ближе всего к канарейке располагаются большие массивы и структуры с ними, затем малые массивы и их структуры, за ними следуют переменные, на которые ссылаются, и в последнюю очередь – все остальные переменные. Это максимизирует вероятность срабатывания защиты при переполнении.

Защита имеет свои недостатки. Она лучше всего срабатывает, когда эксплуатируется ошибка переполнения стека с линейной последовательной перезаписью памяти. Если атакующий контролирует смещение, по которому производится перезапись памяти на стеке, то он может обойти эту защиту. Также, эксплуатируя другую ошибку, приводящую к утечке информации со стека, возможно считать или угадать значение канарейки.

Анализ современных настольных дистрибутивов показывает, что большинство пакетов уже собирается со встроенной защитой канарейки. Такие дистрибутивы как Fedora, Ubuntu, Arch Linux, OpenBSD, Alt Linux и другие включили защиту при сборке всех пакетов более 10 лет назад. Так, в работе [44] было показано что 85% пакетной базы Debian собрано с данной защитой. Однако, анализ прошивок встроенных устройств в той же работе показывает, что уровень внедрения защиты в них на 2022 сильно отстает и составляет в среднем 29.7% (и опускается до 3.28% в случае исключения прошивок двух производителей из общего анализа).

Не смотря на результаты приведенного анализа можно утверждать, что данная технология защиты является устоявшейся. В исследованиях [45,46] поднимается вопрос эффективности данной защиты с точки зрения предотвращения эксплуатации ошибки переполнения стека и накладных расходов на производительность, по сравнению с более новыми методами защиты SafeStack и “теневым стеком” (shadow stack).

Страничная защита стека (Stack Guard Gap+Stack Clash Protection)

В современных ОС для предотвращения эксплуатации уязвимостей, связанных с переполнением стека и конфликтом областей памяти (stack clash) [47-49], применяются механизмы Stack Guard Gap на уровне ядра ОС и Stack Clash Protection, встроенная в компиляторы. Эти технологии обеспечивают изоляцию стека пользовательского пространства от других регионов памяти за счёт размещения специальных неотображаемых (guard) страниц и “зондирование” (probing) памяти на стеке при каждом её выделении.

Механизм атаки Stack Clash позволяет злоумышленнику добиться перекрытия областей стека и, например, кучи. Используя различные техники (выделение больших регионов памяти на стеке, переменные окружения или рекурсивные вызовы функций), злоумышленник сближает области стека и кучи, используя знания об относительном размещении областей. После этого стек и куча “сталкиваются”, что позволяет атакующему манипулировать данными в куче через операции со стеком и наоборот.

Основная идея защиты заключается в создании “защитного промежутка” (guard gap) между стеком и другими областями памяти. Этот промежуток реализуется как неотображаемый регион памяти. При попытке выхода за границы стека (например, из-за переполнения) процесс обращается к одной из защитных страниц, что приводит к генерации исключения и аварийному завершению программы, предотвращая потенциальную эксплуатацию уязвимости. Исследования Qualys [49] и более ранние [47] показали, что использования одной страницы памяти (guard page) для изоляции стека недостаточно [50] и должен использоваться больший промежуток [51] (guard gap).

Увеличение размера защитной области между стеком и кучей является только частичным решением проблемы, поскольку крупные аллокации памяти на стеке все еще могут “перепрыгнуть” даже через увеличенную защитную область. Для полноценной защиты необходимо обеспечить “зондирование” (probing) памяти на стеке при каждом выделении. Для этого в компиляторах GCC и Clang была реализована опция `-fstack-clash-protection` [52,53].

Для защиты стеков самого ядра ОС применяется схожий механизм изоляции. Рассмотрим защиту на примере стеков самого ядра Linux. В основной ветви разработки было внедрено виртуальное отображение стека ядра `VMAP_STACK` [54]. Вместо использования непрерывного адресного пространства каждый стек ядра размещается в отдельном виртуальном регионе, окружённом защитными страницами. При выходе за границы стека происходит обращение к защитной странице, что вызывает ошибку памяти. Дополнительно, в ядре Linux с версии v4.20 на уровне исходного кода было запрещено использование массивов переменной длины (Variable-Length Arrays, VLA) [55]. С точки зрения безопасности, использование VLA значительно затрудняет статическую оценку использования стека программой, что может привести к выходу за его пределы, особенно в контексте ядра ОС.

Страничная защита стека является важным механизмом безопасности, который предотвращает атаки типа Stack Clash, направленные на обход защиты Stack Guard Gap. Комбинация защиты на уровне ядра и на уровне компилятора обеспечивает хороший уровень защиты от данного типа атак.

Проверка целостности конца стека

Также, существуют механизмы дополнительные проверки целостности конца стека. В конец стека вставляется канарейка, и при каждом переключении контекста происходит проверка целостности стека. Это актуально, когда программа сама управляет своим стеком. Например, в ядре Linux используется механизм `SCHED_STACK_END_CHECK` [56].

В отличие от механизма стековых канареек Stack Protector, который размещает канарейку перед локальными переменными в каждом фрейме стека, механизм `SCHED_STACK_END_CHECK` делает это только в конце всего стека потока. У каждого потока ядра есть свой стек, и при каждом вызове планировщика ядра происходит проверка целостности конца стека путем сверки значения канарейки. Stack Protector же осуществляет проверку при выходе из функции. Stack Protector защищает от переполнения буфера в рамках функции. `SCHED_STACK_END_CHECK` - обнаруживает общее переполнение всего стека потока. В большей степени данная опция является отладочной, однако в виду легковесности проверки, её часто также рассматривают как механизм усиления.

Предотвращение утечек информации через стек ядра (StackLeak)

StackLeak [57,58] является механизмом защиты в ядре Linux, который нацелен на предотвращение утечек конфиденциальной информации через неинициализированные участки стека ядра. Проблема, которую он решает, заключается в том, что функции ядра при завершении своей работы могут оставлять на стеке чувствительные данные. Эти данные, если их не очистить, могут быть прочитаны другими функциями, которые впоследствии будут использовать ту же область стека, создавая тем самым уязвимость утечки информации [59]. Это особенно важно в случаях, когда злоумышленник может выполнять последовательные системные вызовы и анализировать содержимое стека.

StackLeak работает на двух уровнях: во время компиляции и во время выполнения. На этапе компиляции специальный плагин компилятора GCC добавляет в функции ядра instrumentation для отслеживания максимальной глубины использования стека. Во время выполнения эта информация используется для определения области стека, которую необходимо очистить. При выходе из системного вызова StackLeak определяет, насколько глубоко был использован стек, и очищает все области стека выше этой отметки. Это

предотвращает утечку информации через неинициализированные области стека при последующих системных вызовах.

Таким образом, StackLeak включает в себя несколько ключевых компонентов. Первый компонент – это плагин компилятора, который добавляет необходимую инструментацию в код ядра. Он анализирует каждую функцию и вставляет код для отслеживания использования стека. Второй компонент – это механизм очистки стека, который активируется при возврате из системных вызовов. Третий компонент – это система мониторинга использования стека, которая помогает обнаруживать потенциальные проблемы с глубиной стека на ранних стадиях.

За счет использования оптимизированных инструкции процессора для очистки памяти, и ограничения инструментации кода до функций, которые действительно могут привести к утечке информации, разработчикам удалось минимизировать накладные расходы.

Принудительная инициализация переменных на стеке

Такие руководства по безопасному коду как MISRA C [60] не просто содержат требования к инициализации переменных. Использование неинициализированных переменных может привести уязвимостям утечки информации. Однако на практике, проконтролировать инициализацию локальных переменных в больших проектах с длительной историей разработки, которые изначально не придерживались стандартов безопасного написания кода, имеют много контрибьютеров с разными взглядами на оформление кода, представляется сложным. Статические анализаторы могут выявлять такого рода ошибки, но не дают гарантий их полного отсутствия в проекте, а также могут сообщать ложноположительные срабатывания. Динамические анализаторы, выявляют подобные ошибки, но также не дают гарантий их полного отсутствия в проекте, накладывают существенные накладные расходы при работе и имеют ложноотрицательные срабатывания. К тому же, в больших открытых проектах, не все изменения кода проходят статический и/или динамический анализ перед тем, как попасть в релиз. Это означает, что ошибка может быть в проекте достаточно длительное время, до того как исправление будет принято, попадет в новый релиз и релизы с исправлениями.

В качестве ограниченного решения проблемы неинициализированных переменных, которые могут приводить к утечкам информации в пространство пользователя при обработке системных вызовов в ядре Linux, команда PaX предложила плагин для GCC STRUCTLEAK [61]. Плагин добавляет принудительную инициализацию полей структур при сборке ядра, которые могут копироваться в пространство пользователя при обработке системных вызовов в ядре Linux. Позднее этот плагин был добавлен в основную ветку разработки ядра Linux [62]. Однако, плагин завязан на специфику ядра Linux и защищает только от утечек информации из ядра в пространство пользователя, накладывая минимальные накладные расходы.

Разработчики компиляторов GCC и Clang реализовали более общий вариант защиты и добавили механизм принудительной инициализации неинициализированных переменных на стеке trivial-auto-var-init [63, 64]. Он работает для всех переменных, не завязан на специфику какого-либо из проектов, и превентивно защищает от утечек информации через неинициализированные переменные на стеке. Позднее, разработчики ядра Linux, также включили этот механизм в ядро в виде INIT_STACK_ALL_ZERO [65].

Рандомизация смещения стека ядра при каждом системном вызове

Рандомизация смещения стека ядра Linux (PaX RANDKSTACK [66]/RANDOMIZE_KSTACK_OFFSET [67]) – это механизм защиты, повышающий безопасность системы за счет внесения случайности (энтропии) в расположение стека ядра при обработке системных вызовов. Это значительно затрудняет злоумышленникам предсказание адресов в стеке и, следовательно, эксплуатацию уязвимостей, связанных со стеком.

Изначально защита была реализована PaX Team в виде механизма RANDKSTACK, но в основную ветвь разработки ядра Linux она попала позднее в виде переработанного механизма RANDOMIZE_KSTACK_OFFSET. Рассмотрим последний подробнее.

Основная идея заключается в том, что, поскольку смещение стека рандомизируется при каждом системном вызове, злоумышленнику становится сложнее надежно попасть в определенное место в стеке, даже при утечках адресов, так как база стека будет меняться при следующем системном вызове.

Традиционно ядро выделяет новый стек для процесса или потока, выравнивая его по фиксированному смещению относительно базового адреса области стека. RANDOMIZE_KSTACK_OFFSET изменяет этот подход, добавляя к указателю стека случайное смещение при каждом системном вызове. Это смещение выбирается из определенного диапазона, что позволяет соблюсти архитектурные требования к выравниванию стека и одновременно делает его точное положение непредсказуемым.

При каждом системном вызове, после сохранения регистров процессора, но до начала активного использования стека, ядро генерирует это случайное смещение. Оно применяется к указателю стека, сдвигая его фактическое начало.

С точки зрения безопасности, RANDOMIZE_KSTACK_OFFSET затрудняет атаки методом возвратно-ориентированного программирования (ROP) [68], так как адреса гаджетов в стеке становятся менее предсказуемыми. Помогает защититься от атак на переполнение стека, поскольку неопределенность в расположении стека мешает злоумышленникам точно разместить свою полезную нагрузку. Препятствует использованию утечек адресов стека: адрес, полученный в одном системном вызове, становится неактуальным для следующего, так как база стека изменится.

Влияние RANDOMIZE_KSTACK_OFFSET на производительность, как правило, минимально, но измеримо и составляет порядка 1%. RANDOMIZE_KSTACK_OFFSET эффективно дополняет другие механизмы безопасности ядра. Он усиливает рандомизацию адресного пространства ядра (KASLR), внося дополнительный уровень непредсказуемости именно для адресов в стеке. Механизм также хорошо сочетается со стековыми канарейками (stack canaries) и другими техниками защиты стека (например, STACKLEAK), формируя многоуровневую систему защиты от атак на стек.

3.3 Механизмы защиты динамической памяти

Для усиления безопасности динамического выделения памяти применяется комплекс механизмов. Основная цель данных механизмов – противодействие широкому спектру уязвимостей, связанных с повреждением памяти, таких как использование после освобождения (use-after-free), переполнения буфера кучи (heap overflows), двойное освобождение (double-free) и утечки конфиденциальных данных. Эти защиты могут охватывать различные компоненты подсистемы управления памятью, включая аллокаторы кучи (например, SLAB/SLUB в ядре Linux) и страничные аллокаторы. Для достижения этой цели применяются разнообразные техники, включая рандомизацию расположения объектов и списков свободной памяти, принудительную инициализацию и очищение выделяемой/освобождаемой памяти, кодирование указателей и изоляцию критически важных структур. Совокупность этих мер значительно повышает сложность и снижает надежность эксплуатации уязвимостей, основанных на манипуляциях с динамической памятью.

Стоит отметить, что описанные подходы к защите динамической памяти не являются уникальными для какой-либо одной системы или только для пространства ядра. Аналогичные принципы и механизмы активно применяются и в пользовательском пространстве, в частности, в реализациях стандартных библиотечных аллокаторов памяти, таких как glibc malloc, jemalloc или tcmalloc. Эти аллокаторы также могут включать

в себя рандомизацию, защиту метаданных, “красные зоны” (red zones) и другие техники для противодействия атакам на кучу.

Мы рассмотрим реализацию этих механизмов защиты динамической памяти на примере ядра Linux.

Рандомизация списков свободной памяти аллокатора

Механизм SLAB_FREELIST_RANDOM [69] представляет собой средство усиления защиты, направленное на рандомизацию порядка следования освобожденных объектов в списках свободных блоков распределителя памяти SLAB. Принцип его действия заключается в том, что в момент освобождения блока памяти он помещается не в предопределенную позицию списка свободных элементов (например, в его начало или конец), а на случайным образом выбранное место. Данная рандомизация осуществляется на этапе инициализации каждого отдельного SLAB-кэша, который представляет собой пул для хранения объектов определенного размера. В результате формируется уникальный, непредсказуемый паттерн организации списка свободных блоков, остающийся неизменным на протяжении всего жизненного цикла соответствующего кэша.

Такой подход к управлению свободными блоками памяти значительно затрудняет эксплуатацию определенных классов уязвимостей. В частности, усложняется реализация атак типа использование после освобождения, поскольку предсказать, какой именно ранее освобожденный объект будет выделен повторно, становится значительно труднее. Также снижается эффективность атак методом заполнения кучи, целью которых является размещение контролируемых злоумышленником данных по предсказуемым адресам. В более общем смысле, любые атаки, основанные на повреждении памяти и использующие предсказуемость расположения объектов, затрудняются подобной защитой.

Кодирование указателей списка свободной памяти аллокатора

Механизм SLAB_FREELIST_HARDENED [70] представляет собой средство усиления защиты в системе управления памятью ядра Linux. Его функционирование основано на преобразовании указателей, содержащихся в списках свободных объектов памяти. Суть преобразования заключается в том, что каждый указатель в таком списке подвергается побитовой операции XOR с двумя величинами: со случайным значением, которое является уникальным для каждого отдельного SLAB-кэша, и с адресом той ячейки памяти, где непосредственно хранится сам преобразуемый указатель.

Такой подход к кодированию указателей обеспечивает несколько важных аспектов защиты. Прежде всего, он эффективно противодействует прямой манипуляции указателями, поскольку злоумышленник не может просто заменить существующий указатель своим собственным значением: оно не будет корректно закодировано. Кроме того, система защищена от предсказания. Даже если атакующему удастся узнать один закодированный указатель, это не позволит ему узнать значения других указателей из-за случайного значения и зависимости от адреса хранения. Наконец, реализуется принцип локальности: закодированный указатель сохраняет свою валидность исключительно в том конкретном месте памяти, для которого он был сформирован, что ограничивает его возможное неправомерное использование в других контекстах.

Разделение корзин аллокации для объектов ядра и пользовательского пространства

Механизм SLAB_BUCKETS [71] представляет собой средство усиления безопасности управления памятью, позволяющее изолировать потенциально опасные аллокации памяти в отдельные, специализированные наборы корзин. Ключевым отличием от традиционного подхода, при котором все объекты одного размера могли размещаться в общих корзинах аллокатора, является целенаправленное разделение объектов, например по типу объектов, между пользовательским пространством и пространством ядра.

Такое разделение корзин аллокации имеет ряд преимуществ с точки зрения безопасности. Оно обеспечивает физическое разграничение между объектами, которые может

контролировать пользователей (например, через системные вызовы) и объектами ядра. Это затрудняет атаки, стремящиеся повлиять на структуры ядра через манипуляции с пользовательскими данными. Кроме того, усложняется проведение атак типа заполнения кучи и снижается общая предсказуемость размещения объектов в памяти, что является важным фактором противодействия попыткам эксплуатации уязвимостей. Данный механизм также формирует дополнительный уровень изоляции для критических структур ядра, оберегая их от потенциального повреждения.

Принудительная инициализация выделяемой и очищение освобожденной памяти

Механизм `INIT_ON_ALLOC` [72] обеспечивает принудительное затирание нулевыми значениями всей выделяемой памяти. Это позволяет полностью устранить уязвимости, связанные с использованием неинициализированной памяти кучи, предотвратить утечки конфиденциальных данных и обеспечить детерминированное начальное состояние выделенной памяти.

Механизм `INIT_ON_FREE` [72], в свою очередь, принудительно затирает нулевыми значениями всю освобождаемую память. Данный подход способствует сокращению времени жизни данных в памяти, противодействует криминалистическому анализу памяти, обеспечивает защиту от атак с холодной загрузкой (*cold boot attacks*) и предоставляет дополнительный уровень защиты от утечек данных. Сокращение времени жизни данных затрудняет такие атаки типа заполнения кучи.

Влияние этих механизмов на производительность различно. Использование `INIT_ON_ALLOC` приводит к снижению производительности примерно на 1% в реальных системах, тогда как `INIT_ON_FREE` может вызвать более заметное снижение, составляющее приблизительно 3-5% в реальных условиях эксплуатации.

Рандомизация кэшей аллокации

Механизм `RANDOM_KMALLOC_CACHES` [73] является средством усиления защиты ядра Linux, вносящим элемент случайности в систему управления динамической памятью. В отличие от традиционного подхода, где выделения памяти определенного размера и типа обслуживаются одним и тем же кэшем, `RANDOM_KMALLOC_CACHES` предусматривает создание нескольких идентичных наборов кэшей. При каждом запросе на выделение памяти система производит случайный выбор одного из этих наборов для его обслуживания. Такое решение существенно снижает предсказуемость размещения объектов в памяти для потенциального злоумышленника.

Функционирование данного механизма основано на двух ключевых аспектах. Во-первых, на этапе инициализации системы создается несколько идентичных наборов кэшей. Во-вторых, при каждом выделении памяти осуществляется случайный выбор одного из этих наборов.

Внедрение `RANDOM_KMALLOC_CACHES` значительно усиливает защиту ядра от атак, основанных на предсказании адресов памяти. Невозможность для атакующего точно определить, в каком из кэшей будет размещен целевой объект, существенно затрудняет проведение атак типа заполнения кучи, эксплуатацию уязвимостей использования после освобождения, а также иные манипуляции, связанные с предсказуемым расположением объектов в памяти. Таким образом, механизм усложняет эксплуатацию уязвимостей, вынуждая злоумышленника учитывать множество вероятных сценариев размещения объектов, что требует разработки более сложных и менее надежных эксплойтов.

Данный механизм эффективно взаимодействует с другими средствами защиты. Например, если `SLAB_FREELIST_RANDOM` обеспечивает рандомизацию порядка объектов внутри отдельного кэша, то `RANDOM_KMALLOC_CACHES` вносит дополнительный уровень случайности путем выбора самого кэша, формируя таким образом многоуровневую систему защиты. Аналогичным образом, при совместном использовании с `SLAB_BUCKETS`, рандомизация выбора кэшей еще более затрудняет атаки, основанные на предсказуемом размещении объектов в памяти.

В заключение, `RANDOM_KMALLOC_CACHES` представляет собой действенный инструмент повышения безопасности ядра ОС за счет привнесения непредсказуемости в процесс управления памятью. Сочетание незначительных накладных расходов с ощутимым усилением защиты делает его ценным элементом комплексной стратегии обеспечения безопасности ядра Linux.

4. Защита потока управления

4.1 Контроль целостности потока управления (CFI)

Контроль целостности потока управления (Control-Flow Integrity, CFI) [74,75] – это технология защиты, разработанная для предотвращения атак, направленных на несанкционированное изменение потока выполнения программы. Основной принцип CFI заключается в обеспечении того, чтобы выполнение программы строго соответствовало заранее определенному графу потока управления (Control-Flow Graph, CFG). Этот граф представляет собой набор всех допустимых путей выполнения, который обычно определяется на этапе компиляции путем статического анализа исходного кода или бинарного файла, либо во время выполнения посредством динамического анализа (профилирования).

CFI предназначен для противодействия эксплуатации уязвимостей, связанных с повреждением памяти (например, переполнение буфера, использование после освобождения). Подобные уязвимости могут позволить атакующему получить контроль над указателем инструкций (instruction pointer, IP) и перенаправить выполнение программы на произвольный или контролируемый им код. Часто для этого используются такие техники, как возвратно-ориентированное программирование (Return-Oriented Programming, ROP), переходо-ориентированное программирование (Jump-Oriented Programming, JOP), а также их вариации (Call-Oriented Programming, Loop-Oriented Programming). Продвинутое программирование зачастую способно обойти механизмы защиты предотвращения исполнения данных, рандомизация размещения адресного пространства и защиты от переполнения стека. CFI существенно усложняет или делает практически невозможным использование этих уязвимостей, принуждая поток управления оставаться в пределах доверенного CFG.

Механизм CFI основан на динамической проверке инструкций косвенной передачи управления во время выполнения программы. К таким инструкциям относятся косвенные вызовы функций, косвенные переходы и возвраты из функций, чьи адреса назначения определяются в процессе работы программы. Механизмы CFI призваны защищать как “прямые ребра” графа потока управления (forward edges), соответствующие косвенным вызовам и переходам, так и “обратные ребра” (backward edges), соответствующие инструкциям возврата из функций. Защита обратных ребер часто реализуется с помощью теневого стека вызовов (Shadow Call Stack, SCS), который будет рассмотрен далее.

Процесс реализации CFI обычно включает следующие ключевые этапы:

1. Построение графа потока управления (CFG): На этом этапе генерируется CFG, который определяет все легитимные точки назначения для каждой инструкции косвенной передачи управления в программе.
2. Инструментация кода: Каждой допустимой точке назначения косвенного перехода (например, началу функции или определенному блоку кода) присваивается уникальный идентификатор (метка, ID или тип). Перед каждой инструкцией косвенной передачи управления вставляется специальный код (проверка).
3. Проверка во время выполнения: Непосредственно перед осуществлением косвенной передачи управления выполняется ранее вставленный проверочный код. Эта проверка заключается в том, чтобы убедиться, что целевой адрес перехода

соответствует ожидаемому идентификатору (или типу) для данной точки перехода согласно CFG.

4. Обработка нарушений: Если проверка не проходит (т.е. обнаружено отклонение от разрешенного CFG), это сигнализирует о потенциальной атаке. В этом случае выполнение программы обычно прерывается.

Реализации CFI могут значительно различаться по нескольким ключевым характеристикам, которые влияют на их строгость, производительность и практическую применимость [75]. Различают два основных подхода к точности построения CFG: fine-grained CFI и coarse-grained CFI. Fine-grained CFI стремится к построению максимально точного и строгого CFG, разрешая переходы потока управления только по точно определенным допустимым ребрам, при этом каждому ребру CFG могут назначаться уникальные метки, которые проверяются при передаче управления. В отличие от этого, coarse-grained CFI ослабляет строгость проверок, используя более общие и простые правила, которые могут не требовать полного и точного CFG. Например, в этом случае возвраты могут быть разрешены к любой инструкции, непосредственно следующей за инструкцией вызова, а косвенные переходы – к любой точке входа в функцию. Такой подход выступает компромиссным между уровнем обеспечиваемой безопасности и накладными расходами.

CFG может быть построен с использованием статического анализа, динамического анализа или анализа на основе типов. Статический анализ вычисляет CFG на основе исходного кода или бинарных файлов; его ограничения (например, невозможность точно разрешить все косвенные переходы) могут привести к аппроксимации реальных путей выполнения, включая несуществующие (но теоретически возможные с точки зрения анализатора) ребра в CFG, что потенциально снижает безопасность. Динамический анализ строит или уточняет CFG во время исполнения программы на основе собранных трасс; при этом существует риск ложных срабатываний CFI, если не все легитимные участки кода были пройдены при анализе. Анализ на основе типов (Type-based CFI) генерирует CFG или его часть на основе информации о типах, например, сигнатур функций, для проверки корректности вызовов.

Современные процессоры [76] начинают включать аппаратные функции для CFI, такие как Intel Control-flow Enforcement Technology (CET) [77] и ARM Pointer Authentication Code (PAC) [78]. Они направлены на обеспечение практичного и эффективного применения CFI с низкими накладными расходами.

Хотя CFI значительно усложняет атаки, нацеленные на перехват потока управления, различные исследования продемонстрировали ограничения и уязвимости в существующих реализациях.

Программные проверки CFI могут вызывать значительные накладные расходы на производительность, что выступает значительным барьером для их широкого внедрения. Аппаратная поддержка CFI призвана снизить эти издержки. Несмотря на эти проблемы, исследования продолжают улучшать техники CFI [79,80], фокусируясь на повышении точности анализа CFG, снижении накладных расходов, улучшении формальных гарантий безопасности и более эффективном использовании аппаратной поддержки.

4.2 Теневой стек (Shadow Call Stack)

Механизм теневого стека (Shadow Call Stack, SCS) представляет собой средство защиты, предназначенное для хранения копий адресов возврата функций в обособленной, защищенной области памяти. Дублирование позволяет эффективно обнаруживать и предотвращать атаки, нацеленные на несанкционированное изменение адресов возврата, сохраненных в основном стеке вызовов. Данный механизм является одной из реализаций концепции контроля целостности потока управления, он используется для защиты обратных ребер (Backward-Edge Protection) в CFI.

Принцип функционирования теневого стека основан на том, что в момент вызова функции ее адрес возврата записывается одновременно как в основной, так и в теневой стек. Впоследствии, при выполнении инструкций возврата из функции, адрес, извлекаемый из основного стека, подвергается сравнению с соответствующим адресом из теневого стека. В случае выявления каких-либо расхождений между этими адресами, что указывает на потенциальную модификацию адреса возврата в основном стеке, выполнение программы немедленно прерывается. Это действие направлено на предотвращение эксплуатации уязвимости и захвата контроля над потоком выполнения программы.

Существуют аппаратные реализации теневого стека. Например, буфер стека возвратов (Return Stack Buffer/Intel RSB) или теневой стек (AMD Shadow Stack), предназначенные для нейтрализации ROP-атак, которые осуществляются путем модификации адресов возврата в основном стеке вызовов. Они создают изолированную и защищенную на аппаратном уровне область памяти, предназначенную исключительно для хранения копий адресов возврата.

Теневой стек обеспечивает более высокий уровень защиты по сравнению с традиционными методами, такими как стековые канарейки. Это достигается за счет того, что адреса возврата не просто обрамляются контрольными значениями, а полностью дублируются в специально выделенной и защищенной области памяти. При корректной реализации данный подход делает практически невозможным успешное осуществление атак класса Return-Oriented Programming (ROP), которые полагаются на перезапись адресов возврата.

5. Защита целостности системы

Защита целостности системы направлена на обеспечение того, чтобы программное обеспечение и данные не были несанкционированно изменены. Это важно для поддержания доверия к системе и предотвращения выполнения вредоносного кода или использования искаженных данных. Механизмы контроля целостности проверяют, что компоненты системы, такие как загрузчик, ядро операционной системы, системные файлы и конфигурации, соответствуют ожидаемому (доверенному) состоянию.

5.1 Безопасная загрузка (Secure Boot)

Безопасная загрузка (Secure Boot) представляет собой механизм обеспечения целостности программного обеспечения на ранних этапах инициализации вычислительной системы. Данная технология, являющаяся неотъемлемой частью стандарта UEFI (Unified Extensible Firmware Interface) [81] предназначена для предотвращения выполнения неавторизованного или модифицированного кода, такого как загрузочные вирусы (буткиты) и руткиты, до полной загрузки ОС.

Принцип действия Secure Boot основан на концепции криптографически заверенной “цепочки доверия” (chain of trust). Начиная с первоначального загрузочного кода прошивки, который предполагается доверенным, каждый последующий загружаемый компонент – от прошивки до загрузчика ОС, ядра ОС и его модулей – подвергается проверке цифровой подписи. Эта верификация осуществляется с использованием набора доверенных сертификатов (ключей), хранящихся в защищенной энергонезависимой памяти платформы, обычно в переменных UEFI, таких как Platform Key (PK), Key Exchange Keys (KEKs). В случае, если цифровая подпись компонента недействительна процесс загрузки прерывается или система переходит в режим ограниченной функциональности, предотвращая выполнение потенциально скомпрометированного кода.

Основным преимуществом Secure Boot является значительное усиление защиты от вредоносного программного обеспечения, активного на этапе загрузки. Путем обеспечения того, что только подписанный и доверенный код может быть выполнен до загрузки операционной системы, Secure Boot эффективно противодействует многим классам атак, направленных на раннюю компрометацию системы и обхода защитных механизмов ОС [82].

Secure Boot формирует основу для построения доверенной вычислительной среды и повышает общую целостность системы.

5.2 Контроль целостности

Архитектура измерения целостности (Integrity Measurement Architecture, IMA) [83] представляет собой компонент подсистемы безопасности ядра Linux, предназначенный для динамической проверки целостности файлов и конфигураций. Основная задача IMA заключается в обнаружении несанкционированных изменений в файловой системе путем вычисления контрольных сумм файлов перед их использованием и сравнения их с эталонными значениями.

Принцип работы IMA включает несколько этапов. Во-первых, это измерение: при каждом доступе к файлу (например, при открытии на чтение или выполнение) вычисляется его хэш. Во-вторых, это оценка: вычисленный хэш может сравниваться с ранее сохраненным эталонным значением, которое считается доверенным. В случае несоответствия система может предпринять определенные действия, такие как генерация оповещения или блокировка доступа к файлу. В-третьих, это аудит: все события измерения и результаты проверок могут протоколироваться для последующего анализа. Важной особенностью IMA является ее интеграция с доверенным платформенным модулем (TPM). Хэши измеряемых файлов могут сохраняться в регистрах конфигурации платформы (PCR) TPM, что обеспечивает защищенное хранение истории измерений и позволяет проводить удаленную аттестацию целостности системы.

Для усиления защиты, особенно в отношении метаданных файлов, IMA часто используется совместно с модулем расширенной проверки (Extended Verification Module, EVM) [84]. EVM нацелен на защиту расширенных атрибутов файлов (xattrs), включая те, в которых IMA может хранить эталонные хэши, а также других важных атрибутов безопасности, таких как права доступа. EVM вычисляет криптографическую подпись для набора защищаемых метаданных файла, используя ключ, который может быть защищен TPM. При каждом доступе к файлу EVM проверяет целостность этих метаданных, предотвращая атаки, направленные на их модификацию с целью обхода механизмов IMA.

Преимущества совместного использования IMA и EVM заключаются в обеспечении комплексной защиты как содержимого файлов, так и их важных метаданных от несанкционированных модификаций. Интеграция с TPM значительно повышает надежность системы, так как ключи и эталонные измерения защищены аппаратными средствами. Это позволяет строить доверенные вычислительные среды и проводить достоверную аттестацию состояния системы.

Однако применение данных механизмов сопряжено с определенными ограничениями. К ним относятся потенциально высокие накладные расходы на производительность, особенно при интенсивной работе с файловой системой. Кроме того, требуется тщательная настройка политик измерения и оценки для предотвращения ложных срабатываний и обеспечения корректной работы системы. Управление эталонными значениями хэшей и ключами EVM также представляет собой нетривиальную задачу.

5.3 Блокировка ядра (Kernel Lockdown)

Механизм блокировки ядра (Kernel Lockdown) [85] представляет собой модуль безопасности, реализованный в ядре Linux и предназначенный для усиления границы между пользовательским пространством и ядром. Основная цель данного механизма заключается в предотвращении как случайного, так и намеренного изменения кода ядра или его критических данных со стороны процессов, обладающих правами суперпользователя, что могло бы привести к компрометации ядра ОС. То есть, Lockdown выступает логическим продолжением механизмов Secure Boot, контроля целостности и в некоторой степени обеспечивает контроль целостности самого ядра.

Принципы работы механизма блокировки ядра основаны на ограничении доступа к определенным функциям и интерфейсам ядра, которые могут быть использованы для обхода существующих механизмов безопасности или для выполнения произвольного кода в контексте ядра, даже если процесс уже обладает привилегиями. Многие атаки нацелены на первоначальное получение прав суперпользователя с последующим использованием этих прав для загрузки вредоносных модулей ядра, модификации его памяти или иного вмешательства в его работу. Функциональность Lockdown включает запрет на загрузку неподписанных модулей ядра, особенно если система загружена в режиме UEFI Secure Boot. Также накладываются ограничения на доступ к таким интерфейсам, как `/dev/mem`, `/dev/kmem` и `/dev/port`, с целью предотвращения прямого чтения или записи в память ядра и управления портами ввода-вывода. Кроме того, ограничивается использование некоторых отладочных интерфейсов, которые могут быть неправомерно использованы для модификации поведения ядра, и запрещается изменение некоторых критически важных параметров ядра через файловые системы `/proc` или `/sys` после активации режима блокировки.

В ядре Linux механизм Lockdown может функционировать в нескольких режимах, в частности режимы целостности и конфиденциальности. Режим целостности направлен на предотвращение любых модификаций ядра, способных ее нарушить. Например, в этом режиме запрещается загрузка модулей ядра, если они не подписаны доверенным цифровым ключом, или использование интерфейсов, позволяющих изменять код работающего ядра. Режим конфиденциальности, в свою очередь, включает все ограничения режима целостности и дополнительно вводит меры, направленные на предотвращение извлечения конфиденциальной информации из памяти ядра. Это включает ограничение доступа к информации, которая могла бы раскрыть адреса в памяти ядра, структуру данных ядра или другие чувствительные сведения.

Применение механизма блокировки ядра предоставляет ряд существенных преимуществ для безопасности системы. Во-первых, он значительно сокращает поверхность атаки на ядро со стороны процессов, уже обладающих правами суперпользователя, ограничивая возможности для злонамеренных действий даже после получения привилегий. Во-вторых, усиливается защита от руткитов и других типов вредоносных программ, которые пытаются модифицировать ядро для сокрытия своего присутствия или для получения полного контроля над системой. В-третьих, повышается общий уровень безопасности системы, что особенно актуально для сред, где предъявляются высокие требования к доверию, например, при использовании технологии Secure Boot. Связка UEFI Secure Boot, IMA/EVM и Lockdown обеспечивает формирование непрерывной цепочки доверия от загрузчика до полностью функционирующего ядра.

Несмотря на очевидные преимущества, использование механизма блокировки ядра сопряжено с некоторыми ограничениями и требует учета определенных соображений. Существует вероятность нарушения нормальной работы приложений или специализированных системных инструментов, которым для выполнения своих функций необходим глубокий доступ к внутренним компонентам ядра. К таким инструментам могут относиться, например, отладчики, специфические драйверы устройств или программное обеспечение для виртуализации. Блокировка ядра не является универсальным решением всех проблем безопасности и должна применяться как один из элементов комплексной, многоуровневой стратегии защиты.

6. Самоограничение привилегий

Принцип минимальных привилегий (Principle of Least Privilege, PoLP) – это концепция в области информационной безопасности, согласно которой каждому модулю системы (например, процессу, пользователю или программе) должны быть предоставлены только те

привилегии, которые необходимы для выполнения его непосредственных задач, и не более того. Этот подход направлен на ограничение потенциального ущерба, который может быть нанесен в случае компрометации или некорректной работы компонента системы.

Одним из ключевых методов реализации принципа минимальных привилегий является практика сброса привилегий (*privilege dropping*). Суть этого метода заключается в том, что процесс, изначально запущенный с повышенными привилегиями (например, правами суперпользователя, необходимыми для инициализации, такой как открытие привилегированных портов или доступ к защищенным файлам), после выполнения этих начальных операций добровольно и необратимо отказывается от избыточных прав. Таким образом, основную часть своей работы процесс выполняет с минимально необходимым набором полномочий. Это значительно снижает риски: если такой процесс будет скомпрометирован злоумышленником, то атакующий получит доступ только к ограниченному набору привилегий, что существенно сужает его возможности для дальнейшего развития атаки и нанесения вреда системе.

Рассмотрим два механизма в ядре Linux, когда программы сами могут ограничивать себя через специальные системные вызовы в исполнении системных вызовов и доступе к ресурсам.

6.1 Ограничение системных вызовов *Seccomp*

Seccomp [86] – это механизм ядра Linux, позволяющий процессу необратимо перейти в состояние ограниченного выполнения, в котором он может использовать только строго определённый набор системных вызовов. Основная цель *Seccomp* – минимизировать поверхность атаки ядра, доступную из пользовательского пространства. Если процесс скомпрометирован после наложения ограничений, *Seccomp* ограничивает набор действий, которые злоумышленник может выполнить через этот процесс.

Изначально *Seccomp* был представлен в очень строгом режиме [87], который разрешал процессу использовать только системные вызовы `read()`, `write()`, `_exit()` и `sigreturn()`. Любая попытка вызова другого системного вызова приводила к завершению процесса.

Позднее был добавлен более гибкий режим, который использует Berkeley Packet Filter (BPF) для определения пользовательских политик фильтрации системных вызовов. С помощью BPF-программы процесс может указать, какие системные вызовы разрешены, какие запрещены, а также проверять аргументы системных вызовов. Это позволяет создавать детальные политики безопасности, адаптированные под конкретные нужды приложения. Например, веб-серверу может быть запрещен доступ к сетевым вызовам, кроме тех, что необходимы для его работы, или приложению для обработки изображений может быть запрещен доступ к файлам вне определенной директории.

Ограничение доступных системных вызовов значительно уменьшает количество векторов атак на ядро через скомпрометированный процесс. Помогает изолировать процессы, особенно те, которые обрабатывают недоверенные данные, предотвращая их злонамеренное использование для атаки на другие части системы. *Seccomp* широко используется в различных системах для повышения безопасности, например, в контейнерных технологиях (Docker, Kubernetes), веб-браузерах (Google Chrome, Firefox) и других приложениях, где важна изоляция и ограничение привилегий.

6.2 Ограничение доступа к файлам *Landlock*

LandLock [88, 89] – это относительно новый механизм безопасности ядра Linux, который позволяет процессам создавать собственные политики контроля доступа к файловой системе без необходимости обладания привилегиями суперпользователя. *LandLock* также реализует принцип наименьших привилегий. Однако делает это на уровне доступа к файлам. *LandLock* позволяет процессу определить набор правил, которые ограничивают его доступ к

определенным иерархиям файлов и каталогов. Эти правила применяются к самому процессу и ко всем его дочерним процессам, созданным после активации политики LandLock. Политика определяет, какие типы доступа (например, чтение, запись, выполнение, создание файлов) разрешены для указанных путей в файловой системе.

Он обеспечивает гранулярность, позволяя тонко настраивать права доступа к конкретным файлам и каталогам. Также предлагает проактивную защиту, ограничивая потенциальный ущерб от уязвимостей в приложении, так как даже скомпрометированное приложение не сможет получить доступ к файлам за пределами разрешенной ему области. Кроме того, LandLock может использоваться совместно с другими механизмами безопасности Linux, такими как Sesscomp, пространства имен (namespaces) и другими системами контроля доступа, для создания многоуровневой защиты.

LandLock представляет собой перспективный инструмент для усиления безопасности приложений, особенно тех, которые работают с файлами и могут быть подвержены атакам, направленным на несанкционированный доступ к данным. Он позволяет разработчикам встраивать механизмы самоограничения непосредственно в код приложений, повышая их устойчивость к эксплуатации уязвимостей.

7. Заключение

Механизмы усиления защищенности представляют собой критически важный компонент современной информационной безопасности, поскольку они значительно повышают сложность успешной эксплуатации уязвимостей, а в некоторых случаях делают ее практически невозможной. Из проведенного обзора следует, что для обеспечения надежной защиты необходим комплексный подход, охватывающий все уровни системы – от аппаратного до прикладного, так как ни один отдельный механизм не может гарантировать полную безопасность. Большинство современных методов усиления сфокусированы на противодействии определенным классам уязвимостей, что подчеркивает важность применения комбинации различных защитных стратегий для создания многоуровневой и всесторонней защиты, обеспечивающей в том числе целостность системы и изоляцию ее компонентов. При внедрении таких механизмов необходимо тщательно взвешивать компромисс между повышением уровня безопасности и потенциальным снижением производительности системы, поскольку некоторые техники, например, полная инициализация освобождаемой памяти (INIT_ON_FREE) или контроль потока управления (CFI), могут оказывать заметное влияние на производительность.

Интеграции средств защиты с аппаратным обеспечением позволяет снизить накладные расходы и повысить эффективность защит. Кроме того, проактивное планирование мероприятий по усилению защиты требует проведения архитектурного анализа на всех этапах жизненного цикла программного обеспечения, что помогает сделать экономически обоснованный выбор между объектами защиты и применяемыми средствами.

Таким образом, эффективное применение механизмов усиления защищенности требует от специалистов по информационной безопасности глубокого понимания принципов их работы и особенностей реализации в конкретных операционных системах и приложениях. Только всесторонний, многоуровневый подход, учитывающий специфику защищаемой системы и актуальный ландшафт угроз, способен обеспечить достаточный уровень защиты в постоянно меняющихся условиях.

Список литературы / References

- [1]. OpenBSD. Доступно по ссылке: <https://www.openbsd.org/>, 19.05.2025.
- [2]. Linux Kernel Self-Protection Projection (KSPP). Доступно по ссылке: <https://kspp.github.io/>, 19.05.2025.
- [3]. Rutkowska J., Wojtczuk R. Qubes OS architecture. Invisible Things Lab Tech Rep, 2010.

- [4]. Grsecurity. Доступно по ссылке: <https://grsecurity.net/>, 19.05.2025.
- [5]. PaX project. Доступно по ссылке: <https://pax.grsecurity.net/docs/pax.txt>, 19.05.2025.
- [6]. GrapheneOS. Доступно по ссылке: <https://grapheneos.org/features>, 19.05.2025.
- [7]. Common Weakness Enumeration: A community-developed list of SW & HW weaknesses that can become. Доступно по ссылке: <https://cwe.mitre.org/index.html>, 19.05.2025.
- [8]. Попов А. Карта средств защиты ядра Linux. Системный Администратор, 2022, Выпуск №3 (232), Доступно по ссылке: <https://samag.ru/archive/article/4535>, 19.05.2025.
- [9]. Xiong W., Lagerström R. Threat modeling – A systematic literature review. *Comput. Secur.*, vol. 84, no. C, pp. 53–69, Jul. 2019, DOI: 10.1016/j.cose.2019.03.010.
- [10]. Shevchenko N., Chick T. A., O’Riordan P., Scanlon T. P., Woody C. Threat Modeling: A Summary of Available Methods. 2018.
- [11]. Freund J., Jones J. Measuring and Managing Information Risk: A FAIR Approach. Amsterdam: Butterworth-Heinemann, 2015, doi: <https://doi.org/10.1016/B978-0-12-420231-3.00001-4>.
- [12]. Joint Task Force Transformation Initiative Risk management framework for information systems and organizations: a system life cycle approach for security and privacy. National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-37r2, Dec. 2018. doi: 10.6028/NIST.SP.800-37r2.
- [13]. International Organization for Standardization ISO/IEC 27005:2022 Information security, cybersecurity and privacy protection - Guidance on managing information security risks. 2022.
- [14]. De Raadt T. Exploit Mitigation Techniques in OpenBSD. OpenCON, 2005, Доступно по ссылке: <https://www.openbsd.org/papers/ven05-deraadt/index.html>, 19.05.2025.
- [15]. Pax Team Address Space Layout Randomization. 2001, Доступно по ссылке: <https://pax.grsecurity.net/docs/aslr.txt>, 19.05.2025.
- [16]. Molnar I. x86: Enable KASLR by default. 2017, Доступно по ссылке: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=6807c84652b0b7e2e198e50a9ad47ef41b236e59>, 19.05.2025.
- [17]. De Raadt T. KARL - kernel address randomized link. 2017, Доступно по ссылке: <https://marc.info/?l=openbsd-tech&m=149732026405941&w=2>, 19.05.2025.
- [18]. Molnar I. NX (No eXecute) support for x86. 2004, Доступно по ссылке: <https://git.kernel.org/pub/scm/linux/kernel/git/history/history.git/commit/?id=36bc33bac78f6bc08282c622138f4e432b62e7be>, 19.05.2025.
- [19]. Pax Team PaX pageexec. 2000, Доступно по ссылке: <https://pax.grsecurity.net/docs/pageexec.txt>, 19.05.2025.
- [20]. Pax Team PaX mprotect(). 2000, Доступно по ссылке: <https://pax.grsecurity.net/docs/mprotect.txt>, 19.05.2025.
- [21]. De Raadt T. amd64 kernel W^X. 2015, Доступно по ссылке: <https://marc.info/?l=openbsd-tech&m=142120787308107&w=2>, 19.05.2025.
- [22]. Zijlstra P. module: Harden STRICT_MODULE_RWX. 2020, Доступно по ссылке: <https://lore.kernel.org/all/20200403171303.GK20760@hirez.programming.kicks-ass.net/>, 19.05.2025.
- [23]. Kemerlis V., Polychronakis M., Keromytis A. ret2dir: Rethinking kernel isolation. In 23rd USENIX Security Symposium (USENIX Security 14), pages 957–972, 2014.
- [24]. Paris E. Allow Kconfig to set default mmap_min_addr protection. 2007, Доступно по ссылке: <https://www.mail-archive.com/linux-security-module@vger.kernel.org/msg02399.html>, 19.05.2025.
- [25]. Intel Corporation INTEL64 AND IA-32 ARCHITECTURES SOFTWARE DEVELOPER’S MANUAL. Instruction Set Extensions Programming Reference. 2013.
- [26]. George V., Piazza T., Jiang H.: Technology Insight: *Intel Next Generation Microarchitecture*. Codename Ivy Bridge, IDF 2011.
- [27]. Yu F. Enable/Disable Supervisor Mode Execution Protection. 2011, Доступно по ссылке: <http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=commit;h=de5397ad5b9ad22e2401c4dacdf1bb3b19c05679>, 19.05.2025.
- [28]. Corbet, J. Supervisor mode access prevention. 2012. Доступно по ссылке: <http://lwn.net/Articles/517475/>, 19.05.2025.
- [29]. Spengler B. SSTIC 2016 Keynote. Rennes, France, 2016, Доступно по ссылке: <https://grsecurity.net/SSTIC2016.pdf#page=3>, 19.05.2025.
- [30]. LWN Randomizing structure layout. 2017, Доступно по ссылке: <https://lwn.net/Articles/722293/>, 19.05.2025.
- [31]. Serebryany, K., Bruening, D., Potapenko, A., & Vyukov, D. AddressSanitizer: A Fast Address Sanity Checker. USENIX ATC, 2012.

- [32]. The Kernel Address Sanitizer (KASAN). Доступно по ссылке: www.kernel.org/doc/html/v4.10/dev-tools/kasan.html, 19.05.2025.
- [33]. Elver M. KFENCE: A low-overhead sampling-based memory safety error detector. 2020, Доступно по ссылке: <https://lore.kernel.org/all/20201103175841.3495947-1-elver@google.com/>, 19.05.2025.
- [34]. ARM, ARMv8.5-A Memory Tagging Extension.
- [35]. Konovalov A. kasan: add hardware tag-based mode for arm64. 2020, Доступно по ссылке: <https://lwn.net/Articles/838211/>, 19.05.2025.
- [36]. Zhao Q. Security Improvements in GCC. Linux Plumbers Conference, 2021, Доступно по ссылке: <https://lpc.events/event/11/contributions/1001/>, 19.05.2025.
- [37]. Cook K. hardening: Introduce CONFIG_ZERO_CALL_USED_REGS. Доступно по ссылке: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=a82adfd5c7cb>, 19.05.2025.
- [38]. Jelinek J. Object size checking to prevent (some) buffer overflows. 2004, Доступно по ссылке: <https://gcc.gnu.org/legacy-ml/gcc-patches/2004-09/msg02055.html>, 19.05.2025.
- [39]. Cook K. mm: Hardened usercopy. 2016, Доступно по ссылке: <https://lwn.net/Articles/691012/>, 19.05.2025.
- [40]. Sidhpurwala H. Hardening ELF binaries using Relocation Read-Only (RELRO). Доступно по ссылке: <https://www.redhat.com/en/blog/hardening-elf-binaries-using-relocation-read-only-relro>, 19.05.2025.
- [41]. Cowan C., Pu C., Maier D., Hintony H., Walpole J., Bakke P., Beattie S., Grier A., Wagle P., Zhang Q. StackGuard: automatic adaptive detection and prevention of buffer-overflow attacks. Proceedings of the 7th conference on USENIX Security Symposium, 1998.
- [42]. Wagle P., Cowan C. StackGuard: Simple Stack Smash Protection for GCC. 2003.
- [43]. Etoh H., Yoda K. Propolice: Protecting from stack-smashing attacks. Technical Report, IBM Research Division, Tokyo Research Laboratory, 2000.
- [44]. Yu R., Nin F. D., Zhang Y., Huang S., Kaliyar P., Zakto S., Conti M., Portokalidis G., Xu J. Building Embedded Systems Like It's 1996. Proceedings 2022 Network and Distributed System Security Symposium, DOI: 10.14722/ndss.2022.24031.
- [45]. Bierbaumer, B., Kirsch, J., Kittel, T., Francillon, A., Zarras, A. Smashing the Stack Protector for Fun and Profit. SEC 2018, IFIP Advances in Information and Communication Technology, vol 529.2 2018 Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-99828-2_21.
- [46]. Depuydt H., Gülmez M., Nyman T., Mühlberg J. T. Do we still need canaries in the coal mine? Measuring shadow stack effectiveness in countering stack smashing. 2024, DOI: 10.48550/arXiv.2412.16343.
- [47]. Delalleau G. Large memory management vulnerabilities. CanSecWest, 2005. Доступно по ссылке: https://cdn.atraining.ru/docs/memory_vulns_delalleau.pdf, 19.05.2025.
- [48]. Wojtczuk R. Exploiting large memory management vulnerabilities in Xorg server running on Linux. 2010. Доступно по ссылке: <https://invisiblethingslab.com/resources/misc-2010/xorg-large-memory-attacks.pdf>, 19.05.2025.
- [49]. Qualys Security Advisory: The Stack Clash. 2017. Доступно по ссылке: <https://www.qualys.com/2017/06/19/stack-clash/stack-clash.txt>, 19.05.2025.
- [50]. Torvalds L. mm: keep a guard page below a grow-down stack segment. 2010. Доступно по ссылке: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=320b2b8de12698082609ebbc1a17165727f4c893>, 19.05.2025.
- [51]. Dickins H. mm: larger stack guard gap, between vmas. 2017. Доступно по ссылке: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=1be7107f18eed3e319a6c3e83c78254b693acb>, 19.05.2025.
- [52]. Law J. Stack clash mitigation, 2017. Доступно по ссылке: <https://gcc.gnu.org/legacy-ml/gcc-patches/2017-07/msg00556.html>, 19.05.2025.
- [53]. Guelton S., Ledru S., Stone J. Bringing Stack Clash Protection to Clang / X86 – the Open Source Way. 2021. Доступно по ссылке: <https://blog.llvm.org/posts/2021-01-05-stack-clash-protection/>, 19.05.2025.
- [54]. Lutomirski A. Virtually mapped stacks with guard pages (x86, core). 2016. Доступно по ссылке: <https://lkml.org/lkml/2016/6/15/1064>, 19.05.2025.
- [55]. Cook K. VLA removal for v4.20-rc1. 2018, Доступно по ссылке: <https://lkml.org/lkml/2018/10/28/189>, 19.05.2025.
- [56]. Tomlin A. sched: Always check the integrity of the canary. Доступно по ссылке: <https://lore.kernel.org/all/1410527779-8133-1-git-send-email-atomlin@redhat.com/>, 19.05.2025.
- [57]. Pax Team, PaX - kernel self-protection. H2HC, 2012. Доступно по ссылке: <https://pax.grsecurity.net/docs/PaXTeam-H2HC12-PaX-kernel-self-protection.pdf>, 19.05.2025.

- [58]. Popov A. STACKLEAK: A Long Way to the Linux Kernel Mainline. Linux Security Summit 2018, Vancouver, Canada, 2018.
- [59]. Cho H., Park J., Kang J., Bao T., Wang R., Shoshitaishvili Y., Doupé A., Ahn G.-J. Exploiting Uses of Uninitialized Stack Variables in Linux Kernels to Leak Kernel Pointers. WOOT 2020. Доступно по ссылке: <https://www.usenix.org/conference/woot20/presentation/cho>, 19.05.2025.
- [60]. MISRA, MISRA C:2012 – Guidelines for the use of the C language in critical systems. MIRA Ltd, Nuneaton, Warwickshire CV10 0TU, UK (Mar 2013).
- [61]. Pax Team, PaX - GCC plugins galore. H2HC, 2013. Доступно по ссылке: <https://pax.grsecurity.net/docs/PaXTeam-H2HC13-PaX-gcc-plugins.pdf>, 19.05.2025.
- [62]. Cook K. gcc-plugins: Add structleak for more stack initialization. 2017, Доступно по ссылке: <https://lore.kernel.org/kernel-hardening/20170113220256.GA57663@beast/T/#u>, 19.05.2025.
- [63]. Zhao Q. Security Improvements in GCC. Linux Plumbers Conference, 2021. Доступно по ссылке: <https://lpc.events/event/11/contributions/1001/>, 19.05.2025.
- [64]. Zhao Q. add -ftrivial-auto-var-init and variable attribute uninitialized to gcc. GCC Patch Mailing List, February 2021, Доступно по ссылке: <https://gcc.gnu.org/pipermail/gcc-patches/2021-February/565514.html>, 19.05.2025.
- [65]. Cook K. security: Implement Clang's stack initialization. 2019. Доступно по ссылке: <https://lore.kernel.org/all/20190411180117.27704-4-keescook@chromium.org/>, 19.05.2025.
- [66]. Pax Team, RANDKSTACK. Доступно по ссылке: <https://pax.grsecurity.net/docs/randkstack.txt>, 19.05.2025.
- [67]. Reshetova E. Randomize kernel stack offset upon syscall. 2019, Доступно по ссылке: <https://lwn.net/Articles/785484/>, 19.05.2025.
- [68]. Shacham, H. The Geometry of Innocent Flesh on the Bone: Return-into-libc without Function Calls. ACM CCS, 2007, DOI: 10.1145/1315245.1315313.
- [69]. Garnier T. mm: SLAB freelist randomization. 2016, Доступно по ссылке: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=c7ce4f60ac199fb3521c5fcd64da21cee801ec2b>, 19.05.2025.
- [70]. Cook K. mm: Add SLUB free list pointer obfuscation. 2017, Доступно по ссылке: <https://lore.kernel.org/all/20170802180609.GA66807@beast/T/#m5fc98c10905ea25cf6123e723bca8162226d53c4>, 19.05.2025.
- [71]. Cook K. slab: Introduce dedicated bucket allocator. 2024, Доступно по ссылке: <https://lwn.net/Articles/978976/>, 19.05.2025.
- [72]. Potapenko A. add init_on_alloc/init_on_free boot options. 2019, Доступно по ссылке: <https://lore.kernel.org/all/20190628093131.199499-2-glider@google.com/T/#u>, 19.05.2025.
- [73]. Ruiqi G. Randomized slab caches for kmalloc(). 2023, Доступно по ссылке: <https://lore.kernel.org/all/20230714064422.3305234-1-gongruiqi@huaweicloud.com/>, 19.05.2025.
- [74]. Abadi M., Budiu M., Erlingsson Ú., Ligatti J. Control-flow integrity. Proceedings of the 12th ACM conference on Computer and communications security. Alexandria VA USA: ACM, Nov. 2005, pp. 340–353. DOI: 10.1145/1102120.1102165.
- [75]. Abadi M., Budiu M., Erlingsson Ú., Ligatti J. Control-flow integrity principles, implementations, and Applications. ACM Trans. Inf. Syst. Secur., vol. 13, no. 1, pp. 1–40, Oct. 2009, DOI: 10.1145/1609956.1609960.
- [76]. Clercq R. D., Verbauwhede I. A survey of Hardware-based Control Flow Integrity (CFI). 2017.
- [77]. Intel Corporation Control-flow Enforcement Technology Specification. 2021.
- [78]. Shanbhogue V., Gupta D., Sahita R. Security Analysis of Processor Instruction Set Architecture for Enforcing Control-Flow Integrity. Proceedings of the 8th International Workshop on Hardware and Architectural Support for Security and Privacy. Phoenix AZ USA: ACM, Jun. 2019, pp. 1–11. DOI: 10.1145/3337167.3337175.
- [79]. Tice C., Roeder T., Collingbourne P., Checkoway S., Erlingsson Ú., Lozano L., Pike G. Enforcing Forward-Edge Control-Flow Integrity in GCC & LLVM. USENIX Security 14, 2014, pp. 941–955. Доступно по ссылке: <https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/tice>, 19.05.2025.
- [80]. Bento F. M. Control-Flow Integrity for the Linux kernel: A Security Evaluation. 2019.
- [81]. Unified Extensible Firmware Interface Specification. Version 2.7, May 2017, Доступно по ссылке: https://uefi.org/sites/default/files/resources/UEFI_Spec_2_7.pdf, 19.05.2025.

- [82]. Hagl J., Mann O., Pirker M. Securing the Linux Boot Process: From Start to Finish. Proceedings of the 7th International Conference on Information Systems Security and Privacy, 2021, pp. 604–610. DOI: 10.5220/0010313906040610.
- [83]. Safford D. An Overview of The Linux Integrity Subsystem. Whitepaper, Доступно по ссылке: https://cyfuture.dl.sourceforge.net/project/linux-ima/linux-ima/Integrity_overview.pdf, 19.05.2025.
- [84]. Zohar M. EVM. 2011, Доступно по ссылке: <https://lore.kernel.org/all/1309377038-4550-1-git-send-email-zohar@linux.vnet.ibm.com/>, 19.05.2025.
- [85]. Garrett M. security: Add a static lockdown policy LSM. Доступно по ссылке: <https://lore.kernel.org/linux-security-module/20190820001805.241928-4-matthewgarrett@google.com/>, 19.05.2025.
- [86]. Edge J. A seccomp overview, 2015. Доступно по ссылке: <https://lwn.net/Articles/656307/>, 19.05.2025.
- [87]. Arcangeli A. seccomp for 2.6.11-rc1-bk8. 2005, Доступно по ссылке: <https://lwn.net/Articles/120192/>, 19.05.2025.
- [88]. Salaün M. Landlock: From a security mechanism idea to a widely available implementation. 2024, Доступно по ссылке: https://landlock.io/talks/2024-06-06_landlock-article.pdf, 19.05.2025.
- [89]. Salaün M. Landlock LSM. 2021, Доступно по ссылке: <https://lore.kernel.org/linux-security-module/20210422154123.13086-1-mic@digikod.net/>, 19.05.2025.

Информация об авторах / Information about authors

Денис Валентинович ЕФРЕМОВ – старший научный сотрудник. Сфера научных интересов: формальная верификация, статический и динамический анализ.

Denis Valentinovich EFREMOV – senior researcher. Research interests: formal verification, static and dynamic analysis.

Александр Константинович ПЕТРЕНКО – доктор физико-математических наук, профессор, заведующий отделом Технологий программирования ИСП РАН, профессор кафедр Системного программирования ВМК МГУ и ФКН НИУ ВШЭ. Научные интересы: формальные методы программной инженерии, языки спецификаций и моделирования, верификация.

Alexander Konstantinovich PETRENKO – Dr. Sci. (Phys.-Math.), Prof., Head of Software Engineering Department of the Ivannikov Institute for System Programming, Russian Academy of Sciences, Professor of MSU and the Faculty of Computer Science, NUY HSE. Research their use in software development and verification.

Борис Аронович ПОЗИН – доктор технических наук, профессор, главный научный сотрудник Института системного программирования им. В.П. Иванникова РАН, профессор базовой кафедры ЗАО ЕС-лизинг в МИЭМ НИУ ВШЭ, технический директор ЗАО ЕС-лизинг. Сфера научных интересов: программная инженерия, системы обеспечения жизненного цикла доверенного программного обеспечения, автоматизированное тестирование программ.

Boris Aronovich POZIN – Dr. Sci. (Tech.), Prof., Chief Researcher at the Ivannikov Institute for System Programming of the Russian Academy of Sciences, Professor of the Basic Department of CJSC EC-Leasing at the Higher School of Economics, Technical Director of CJSC EC-Leasing. Research interests: software engineering, life cycle ensuring systems for trusted software, automated software testing.

Виталий Адольфович СЕМЕНОВ – доктор физико-математических наук, профессор, заведующий отделом системной интеграции и прикладных программных комплексов Института системного программирования им. В.П. Иванникова РАН с 2015 года. Сфера научных интересов: модельно-ориентированные методологии и инструменты программной инженерии для создания цифровых платформ и мультидисциплинарных программных комплексов, визуализация и компьютерная графика, технологии информационного

моделирования в архитектуре и строительстве, проектное управление и календарно-сетевое планирование.

Vitaly Adolfovich SEMENOV – Dr. Sci. (Phys.-Math.), Prof., Head of the Department of System Integration and Multi-disciplinary Applied Systems of the Ivannikov Institute for System Programming of the RAS since 2015. Research interests: model-driven methodologies and CASE toolkits for creating digital platforms and advanced applied systems, visualization and computer graphics, building information modeling, project management and scheduling.