**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное бюджетное образовательное учреждение высшего образования**
**«Донецкий национальный технический университет»**

# ПРОГРАММНАЯ ИНЖЕНЕРИЯ: МЕТОДЫ И ТЕХНОЛОГИИ РАЗРАБОТКИ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ (ПИИВС–2024)

V Международная научно-практическая
конференция
(г. Донецк, 27–28 ноября 2024 г.)

Сборник материалов и докладов
Том 1

Донецк
2024

УДК 004.41(063)
ББК 32.972
П78

Рекомендовано к изданию
Советом факультета интеллектуальных систем
и программирования
ФГБОУ ВО «ДонНТУ»
(протокол № 9 от 13.12.2024 г.)

**Ответственный редактор –** Бердюкова Светлана Сергеевна

**Редакционная коллегия:**
Зори С. А. (председатель); Мальчева Р. В.

**П78**

**Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС–2024) :** V Междунар. науч.-практ. конф., сб. материалов и докладов, Т. 1, г. Донецк, 27–28 нояб. 2024 г. / Ред. кол.: Зори С. А. (пред.); Мальчева Р. В.; отв. ред.
С. С. Бердюкова. – Донецк, ФГБОУ ВО «ДонНТУ», 2024. – просмотрщик PDF-файлов. – Загл. с титул. экрана.

В сборнике размещены доклады сотрудников академических институтов, высших учебных заведений, а также специалистов научных организаций Донецкой Народной Республики, Луганской Народной Республики, Российской Федерации и Республики Беларусь. Представлены новые идеи, технологии и результаты программной инженерии по спецификации, проектированию, разработке, сертификации, сопровождению и тестированию программного обеспечения компьютерных систем различного назначения.

Издание предназначено для преподавателей, научных сотрудников, обучающихся образовательных учреждений высшего образования, а также всех интересующихся данной тематикой.

Тексты докладов печатаются в авторской редакции.

УДК 004.41(063)
ББК 32.972

УДК 004.41(063)
ББК 32.972
П78

**Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2024):** сборник научных трудов V Международной научно-практической конференции, Том 1. 27-28 ноября 2024 г. – Донецк, ФГБОУ ВО «Донецкий национальный технический университет», 2024. – 327 с.

Цель научно-практической конференции: создание условий для обмена новыми идеями, технологиями и результатами между профессионалами программной инженерии, принимающими непосредственное участие в деятельности по анализу, спецификации, проектированию, разработке, сертификации, сопровождению и тестированию программного обеспечения компьютерных систем различного назначения, а также для расширения сотрудничества специалистов в области индустриального программирования с коммерческими структурами.

Основные направления работы конференции:

– современные языки и технологии программирования;
– информационные системы, базы данных, безопасность и защита данных;
– методы, технологии и системы искусственного интеллекта;
– компьютерное моделирование, системы виртуальной реальности, компьютерной графики и обработки изображений;
– методы и средства автоматизированного проектирования ПО и систем;
– интеллектуальные системы, анализ данных и распознавание образов;
– системы виртуальной реальности, компьютерной графики и обработки изображений;
– проектирование программных и компьютерных систем, средства автоматизированного проектирования по и систем.

Конференция организована Донецким национальным техническим университетом Министерства науки и высшего образования РФ. В организации конференции приняли участие: ФГБОУ ВО «Донецкий национальный технический университет», НОИ «КНТ», факультет ИСП, кафедра программной инженерии им. Л.П. Фельдмана, ФГАОУ ВО «НИУ Московский институт электронной техники» (МИЭТ, г. Москва, РФ), ФГБОУ ВО «Ульяновский государственный технический университет» (г. Ульяновск, РФ), Высшая школа кибертехнологий, математики и статистики РЭУ им. Г.В. Плеханова (г. Москва, РФ), ФБГУН Институт проблем машиноведения Российской Академии наук (ИПМаш РАН, г. Санкт-Петербург, РФ), ФГБУН Институт информатики и математического моделирования им. В.А. Путилова, Кольский научный центр Российской академии наук (ИИММ КНЦ РАН, г. Апатиты, РФ), Учреждение образования «Полесский государственный университет» (г. Пинск, Республика Беларусь), Ферганский медицинский университет общественного здоровья (г. Фергана, Узбекистан).

В первом томе сборника научных трудов представлено 43 доклада сотрудников академических институтов и высших учебных заведений, а также специалистов других научных организаций из Донецкой Народной Республики, Луганской Народной Республики, Российской Федерации, Республики Беларусь, Республики Татарстан.

# ОГЛАВЛЕНИЕ

**ПЛЕНАРНЫЕ ДОКЛАДЫ** ..... 9

**Б.А.Кулик** (Санкт-Петербург, Институт Проблем Машиноведения РАН)
ЗАКОН ПАРАДОКСА И МАТЕМАТИЧЕСКИЕ МЕТОДЫ УСТРАНЕНИЯ ПРОТИВОРЕЧИЙ В ЗНАНИЯХ И РАССУЖДЕНИЯХ ..... 9

**Дж.Ш. Сулейманов** (Казань, Институт прикладной семиотики АН РТ),
**А.Я. Фридман** (Институт информатики и математического моделирования ФИЦ КНЦ РАН),
**Р. Гатиатуллин** (Казань, Институт прикладной семиотики АН РТ)
СЕМИОТИЧЕСКАЯ МОДЕЛЬ АГГЛЮТИНАТИВНОГО ЯЗЫКА КАК ЯДРО КОГНИТИВНОЙ СИСТЕМЫ ОБЪЯСНИТЕЛЬНОГО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ..... 19

**С.А. Амелькин Л.Г. Гагарина (**Москва, НИУ Московский институт электронной техники**)**
СИСТЕМЫ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ КАК СЛОЖНЫЕ ФРАКТАЛЬНЫЕ СИСТЕМЫ. 25

**С.О. Федоров, В.А. Великий, А.Г. Пимонов** (Кемерово, Кузбасский государственный технический университет им. Т.Ф. Горбачева)
ТЕХНОЛОГИИ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИЗВЛЕЧЕНИЯ И АНАЛИЗА ДАННЫХ ИЗ ОТКРЫТЫХ ИСТОЧНИКОВ ДЛЯ СЕЙСМИЧЕСКОГО МОНИТОРИНГА ТЕРРИТОРИИ .. 31

**В.В. Швыров, Д.А. Капустин, Р.Н. Сентяй** (Луганск, Луганский государственный университет)
РАЗРАБОТКА АБСТРАКТНОЙ МОДЕЛИ НЕЙРОСЕТЕВОГО ДЕТЕКТОРА УЯЗВИМОСТЕЙ В ПРОГРАММНОМ КОДЕ ..... 39

**В.Н. Штепа** (Республика Беларусь, УО «Полесский государственный университет»)
РАЦИОНАЛЬНЫЕ ПОДХОДЫ К ВНЕДРЕНИЮ ЦИФРОВЫХ ДВОЙНИКОВ В ВОДОПРОВОДНО-КАНАЛИЗАЦИОННЫЕ ХОЗЯЙСТВА ..... 45

**О.И. Федяев** (Донецк, ДонНТУ)
МОДЕЛИРОВАНИЕ СИСТЕМЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ НА ОСНОВЕ ЕЁ ИНТЕЛЛЕКТУАЛИЗАЦИИ МЕТОДАМИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ..... 46

**А.А. Суханов** (Донецк, ДонНТУ), **Д.Э. Баев** (Malaga, Spain), **О.И. Федяев** (Донецк, ДонНТУ)
ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА КАЧЕСТВА НЕЙРОСЕТЕВОГО РАСПОЗНАВАНИЯ СТУДЕНТОВ ПО ИЗОБРАЖЕНИЯМ ЛИЦ ИЗ БАЗЫ ДАННЫХ ..... 54

**СЕКЦИЯ 1. «СОВРЕМЕННЫЕ ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»** ..... 62

**К.К. Руднев, Т.Г. Дмитрюк** (Донецк, ДонНТУ)
ПИШЕМ ЭТАЛОННЫЙ КОД: ПОДХОД И ОЦЕНКА ..... 62

**С.Н. Евтушенко, Т.Г. Дмитрюк** (Донецк, ДонНТУ)
РАЗРАБОТКА ПРОКСИ-API С ПОДДЕРЖКОЙ IN-MEMORY КЕШИРОВАНИЯ НА NODE.JS . 70

**Д.А. Филипишин** (Донецк, ДонНТУ)
ИССЛЕДОВАНИЕ УДАЛЁННОГО ИСПОЛЬЗОВАНИЯ ОНТОЛОГИЧЕСКОГО ИНЖИНИРИНГА ..... 82

**В.В. Безуглый, А.В. Боднар** (Донецк, ДонНТУ)
ИССЛЕДОВАНИЕ МЕХАНИЗМОВ УСКОРЕНИЯ ВРЕМЕНИ ОТКЛИКА СОВРЕМЕННЫХ WEB-ПРИЛОЖЕНИЙ НА ОСНОВЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ ..... 89

**Е.Д. Пачкин, Д.С. Шидловская** (Кемерово, Кузбасский государственный технический университет имени Т. Ф. Горбачёва)
СИСТЕМА ОТСЛЕЖИВАНИЯ УСПЕВАЕМОСТИ СТУДЕНТОВ С УВЕДОМЛЕНИЕМ ОБ ИЗМЕНЕНИЯХ ..... 95

**СЕКЦИЯ 2. «ИНФОРМАЦИОННЫЕ СИСТЕМЫ, БАЗЫ ДАННЫХ, БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ»** ..... 100

**А.В. Чернышова, А.В. Коржов** (Донецк, ДонНТУ)
АНАЛИЗ ЭФФЕКТИВНОСТИ ПРОТОКОЛОВ ПЕРЕДАЧИ ДАННЫХ, ИСПОЛЬЗУЮЩИХ КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ ..... 100

**Л.А. Лазебная, Д.М. Зеленский** (Донецк, ДонНТУ)
МЕТОДЫ И ИНСТРУМЕНТЫ ДЛЯ ПРОГНОЗИРОВАНИЯ КУРСА ВАЛЮТ ..... 111

**СЕКЦИЯ 3. «МЕТОДЫ, ТЕХНОЛОГИИ И СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»** ..... 117

**А.Ю. Кучмин , И.Л. Тарасова** (Санкт-Петербург, Институт Проблем Машиноведения РАН)
ФОРМИРОВАНИЕ ОБРАЗОВ В ОКРУЖЕНИИ ВЫБОРА МОБИЛЬНОГО РОБОТА И ИХ КЛАССИФИКАЦИЯ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ И АЛГОРИТМОВ ЛОГИКО-ЛИНГВИСТИЧЕСКОЙ КЛАССИФИКАЦИИ ..... 117

**О.В. Булыгина, М.Ю. Воротилова** (Смоленск, филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске)
ПРИМЕНЕНИЕ БИОИНСПИРИРОВАННЫХ АЛГОРИТМОВ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ МАССОВОГО РЕКРУТИНГА ..... 127

**О.В. Булыгина, В.А. Дружинина** (Смоленск, филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске)
ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ БИОИНСПИРИРОВАННЫХ АЛГОРИТМОВ ПРИ ФОРМИРОВАНИИ ПОРТФЕЛЯ ПРОЕКТОВ ..... 133

**О.В. Булыгина, К.В. Хлусович** (Смоленск, филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске)
АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ ОПТИМИЗАЦИИ ПАРАМЕТРОВ ПРОЦЕССА ОБЕСПЕЧЕНИЯ КАЧЕСТВА ТОПЛИВНЫХ БРИКЕТОВ ..... 139

**Я.С. Пикалёв, Р.С. Хакимов** (Донецк, ФБГНУ «Институт проблем искусственного интеллекта»)
К ВОПРОСУ ОБ ИЗВЛЕЧЕНИИ ЗНАНИЙ ИЗ МУЛЬТИМОДАЛЬНЫХ МОДЕЛЕЙ ДЛЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ОБЪЕКТОВ ..... 146

**Б.В.Павленко** (Донецк, ФБГНУ «Институт проблем искусственного интеллекта»)
СТРУКТУРА МЕТАДАННЫХ СЕГМЕНТОВ КАРТ НАВИГАЦИИ ДРОНА ..... 154

**И.В. Чернядьев** (Донецк, ФБГНУ «Институт проблем искусственного интеллекта»)
МАШИННОЕ ОБУЧЕНИЕ В ЗАДАЧЕ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ПАРАМЕТРИЧЕСКОГО АНАЛИЗА КОГНИТИВНЫХ ДИНАМИЧЕСКИХ СИСТЕМ.....159

**Ф.А. Хуссейн** (Таганрог, Южный Федеральный Университет)
МУРАВЬИНЫЙ АЛГОРИТМ КЛИКОГО ПОКРЫТИЯ.....165

**Н.В. Мелещенко, О.И. Федяев** (Донецк, ДонНТУ)
ОПРЕДЕЛЕНИЕ СЕМАНТИЧЕСКОЙ ЭКВИВАЛЕНТНОСТИ ТЕКСТОВ ТРЕБОВАНИЙ ПРЕДПРИЯТИЙ И РАБОЧИХ ПРОГРАММ ДИСЦИПЛИН.....170

**В.В. Кочетуров, О.И. Федяев** (Донецк, ДонНТУ)
СИАМСКИЕ СВЁРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ ДЛЯ РАСПОЗНАВАНИЯ ЛИЦ.....178

**П.С. Похлёбин, О.И. Федяев** (Донецк, ДонНТУ)
ИНТЕГРАЦИЯ ГОЛОСОВЫХ ТЕХНОЛОГИЙ В МЕССЕНДЖЕРЫ: СОЗДАНИЕ ГОЛОСОВОГО АССИСТЕНТА.....185

**А.Р. Мураще нко, О.И. Федяев** (Донецк, ДонНТУ)
ОЦЕНКА СКОРОСТИ ДВИЖЕНИЯ ОБЪЕКТА С ПОМОЩЬЮ НЕЙРОСЕТЕВОЙ МОДЕЛИ YOLO.....193

**Н.И. Петрушан, А.А. Филиппов** (Ульяновск, Ульяновский Государственный Технический Университет)
АНАЛИЗ КАЧЕСТВА РУССКОЯЗЫЧНОГО РЕПОЗИТОРИЯ КЛИНИЧЕСКИХ ДАННЫХ.....200

**Я.А. Рудь, С.А. Зори, И.А. Коломойцева** (Донецк, ДонНТУ)
ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ В ОРГАНИЗАЦИИ РАЗВЛЕКАТЕЛЬНЫХ МЕРОПРИЯТИЙ.....206

**О.И. Федяев, Д.В. Гавриленков** (Донецк, ДонНТУ)
ПАРАМЕТРИЧЕСКОЕ ОПИСАНИЕ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ КАК ОБЪЕКТА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ.....213

**А.О. Истягин, О.В. Рычка** (Донецк, ДонНТУ)
ОБЩИЙ ОБЗОР НЕЙРОСЕТЕВОГО ПОДХОДА ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ ТЕКСТОВОЙ ИНФОРМАЦИИ.....221

**М.К. Слипенко, О.В. Рычка** (Донецк, ДонНТУ)
РЕАЛИЗАЦИИ МЕТОДА RETRIEVAL-AUGMENTED GENERATION ДЛЯ УЛУЧШЕНИЯ ОТВЕТА БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ LANGCHAIN И PGVECTOR.....227

**С.Н. Евтушенко, С.В. Щедрин, И.А. Коломойцева** (Донецк, ДонНТУ)
ТЕОРЕТИЧЕСКИЙ АНАЛИЗ ТЕХНОЛОГИИ TEXT TO SPEECH И ТЕКУЩЕГО СОСТОЯНИЯ ЕЁ РАЗВИТИЯ.....238

**И.А. Коломойцева** (Донецк, ДонНТУ)
ОСОБЕННОСТИ СТРУКТУРЫ ТЕЗАУРУСА ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА ПОИСКА ДОКУМЕНТОВ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ.....245

**СЕКЦИЯ 4. «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ, СИСТЕМЫ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ, КОМПЬЮТЕРНОЙ ГРАФИКИ И ОБРАБОТКИ ИЗОБРАЖЕНИЙ»**.....251

**А.И. Сорокин, А.Ю. Карповский** (Донецк, ГУ "Автоматгормаш им. В.А. Антипова")
ПРОБЛЕМАТИКА ОТОБРАЖЕНИЯ ИНТЕРАКТИВНЫХ ТРЕХМЕРНЫХ СЦЕН В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ .....251

**Я.К. Савельев, А.А. Филиппов** (Ульяновск, Ульяновский Государственный Технический Университет)
ПОДХОД К ОБНАРУЖЕНИЮ И ИЗВЛЕЧЕНИЮ ПРОЦЕССОВ ИЗ ЖУРНАЛОВ СОБЫТИЙ МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ОСНОВЕ РЕТРОСПЕКТИВНОЙ И ЭКСПЕРТНОЙ ИНФОРМАЦИИ.....261

**Н.А. Бездетный, С.А. Зори** (Донецк, ДонНТУ)
ВИЗУАЛИЗАЦИЯ ИНФОРМАЦИИ В СИСТЕМАХ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ: АНАЛИЗ И ВЫБОР АЛГОРИТМОВ ДЛЯ ДИНАМИЧЕСКИХ ГРАФОВ .....267

**А.В. Григорьев, А.В. Синяев** (Донецк, ДонНТУ)
ОБЗОР ФИЗИЧЕСКИХ ДВИЖКОВ ДЛЯ РАЗРАБОТКИ КОМПЬЮТЕРНЫХ ИГР: ТЕНДЕНЦИИ, ХАРАКТЕРИСТИКИ И ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ. .....272

**СЕКЦИЯ 5. «МЕТОДЫ И СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ПО И СИСТЕМ»**.....280

**К.А. Коврик, О.А. Криводубский** (Донецк, ДонНТУ)
ОБ ОСОБЕННОСТЯХ СОВРЕМЕННОГО ПРОИЗВОДСТВА ТРУБ И ИСПОЛЬЗОВАНИИ АСУ ТП В ТРУБНОМ ПРОИЗВОДСТВЕ .....280

**А.В. Григорьев, Е.С. Бондаренко** (Донецк, ДонНТУ)
АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ И РЕШЕНИЙ ДЛЯ ИНТЕГРАЦИИ ЗНАНИЙ В САПР.....287

**К.В. Ржевский, А. В. Григорьев** (Донецк, ДонНТУ)
АНАЛИЗ ЗАДАЧИ РАЗРАБОТКИ УНИВЕРСАЛЬНОГО ЯЗЫКА ПРОЕКТИРОВАНИЯ КАК СРЕДСТВА АВТОМАТИЗАЦИИ MICROSOFT OFFICE VISIO .....294

**Н.Т. Понамарёв, А. В. Григорьев** (Донецк, ДонНТУ)
ИНТЕГРАЦИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ TELEGRAM-БОТОВ.....305

**А.И. Боровиков, О.А. Криводубский** (Донецк, ДонНТУ)
ВИДЫ ЛОГИСТИЧЕСКИХ ПОТОКОВ ПРЕДПРИЯТИЯ .....311

**СЕКЦИЯ 6. «ИНЖЕНЕРИЯ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ»** ..317

**Д.В. Николаенко, Я.И. Выростков** (Донецк, ДонНТУ)**, Л.П. Володько** (Республика Беларусь, УО «Полесский государственный университет»
ОРГАНИЗАЦИЯ АЛУ С УЧЕТОМ АРХИТЕКТУРНЫХ ОСОБЕННОСТЕЙ И ОПТИМИЗАЦИЙ .....317

**А.Е. Воробьев** (Грозный, Грозненский государственный нефтяной технический университет)**,** **А.Н. Корчевский** (Донецк, ДонНТУ)**, К.А. Воробьев** (Москва, Российский университет Дружбы Народов)
ОСНОВНЫЕ ЭЛЕМЕНТЫ И СОСТАВЛЯЮЩИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА .....323

# ПЛЕНАРНЫЕ ДОКЛАДЫ

УДК 004.82

## Закон парадокса и математические методы устранения противоречий в знаниях и рассуждениях

**Б.А. Кулик**
д.ф.-м. н., снс, Институт проблем машиноведения РАН,
ba-kulik@yandex.ru, OrcID: 0000-0001-6193-5588, SPIN-код: 6321-0172

***Кулик Б.А. Закон парадокса и математические методы устранения противоречий в знаниях и рассуждениях.*** *Предлагается в качестве математической основы логического анализа использовать независимый от теории множеств вариант алгебры множеств, содержащийся в книге Р. Куранта и Г. Роббинса «Что такое математика?». В этом варианте законы алгебры множеств, соответствующие законам классической логики, доказываются без аксиом методом перебора вариантов. Сформулированы и обоснованы новые законы алгебры множеств, в том числе закон парадокса. Предложена математическая модель рассуждений, в которой помимо посылок и правил вывода используются ограничения, позволяющие распознавать и анализировать некорректности в рассуждениях. Рассмотрены основанные на законах алгебры множеств методы устранения противоречий.*

***Ключевые слова:*** *алгебра множеств, модель рассуждений, полисиллогистика, граф включений, закон парадокса, закон существования, методы устранения противоречий.*

### *Введение*

В современной науке законы логики, с помощью которых осуществляются обоснования наших знаний и рассуждений, включая доказательства теорем, выводятся с помощью аксиом, причем на аксиомах основаны как классическая логика, так и многочисленные варианты неклассических логик. Тем самым провозглашается равнозначность этих логик, и связанная с этим рассогласованность при выборе правильных методов логического анализа. Такой расплывчатый и, по сути, деструктивный подход к логике был предложен и получил признание в результате многочисленных дискуссий по обоснованиям математики на рубеже XIX и XX столетий. С точки зрения этого подхода в основе логики и математики лежит теория формальных систем. Математическая логика и аксиоматическая теория множеств в настоящее время развиваются и излагаются в учебниках на основе этого подхода.

В то же время в последние десятилетия стали выявляться значительные трудности, связанные с этим подходом. Об этих трудностях много сказано в публикациях, относящихся к философскому направлению под названием *неформальная логика* [1, 2]. Объектами критики неформальной логики являются следующие свойства формальной логики (под этим термином в основном подразумевается математическая логика):

1) стремление максимально освободиться от естественного языка;

2) трудности, связанные с моделированием неопределенностей, сомнений, вопросов и предположений, без которых невозможно развитие любого знания;

3) непригодность в распознавании и анализе ошибок в рассуждениях и обоснованиях;

4) предполагается, что формальная логика, провозгласив свой принципиальный отказ от ориентации на исследование обыденных рассуждений, постепенно теряет свою практическую значимость.

Альтернативный подход к обоснованию логики можно найти в широко известной книге Куранта и Роббинса «Что такое математика?» [3], первое издание которой вышло в 1941 году. В этой книге имеется небольшой раздел «Алгебра множеств» (Algebra of sets), в котором сформулированы 26 законов алгебры множеств, соответствующие законам классической логики, и высказано предположение, что эти законы можно обосновать без аксиом на основе определений основных понятий этой алгебры. То, что этот вариант алгебры множеств не является популярным изложением аксиоматической теории множеств, а предлагает независимый от этой формальной теории подход к обоснованию свойств множеств, можно понять из следующей цитаты в вводной части книги: «В допущении, что математика есть не более чем система следствий, извлекаемых из определений и постулатов, которые должны быть только совместимы между собой, а в остальном являются продуктом свободной фантазии математиков, таится серьезная угроза для самого существования науки. Если бы это было действительно так, математика была бы занятием, недостойным мыслящего человека. Она была бы просто игрой

с определениями, правилами и силлогизмами, не имеющей ни причины, ни цели. Представление, согласно которому человеческий интеллект может творить лишенные какого бы то ни было смысла системы постулатов, есть обман, точнее, полуправда» [3, с. 22].

В настоящее время в математике известны 2 варианта алгебры множеств, В русскоязычной литературе более распространен используемый в теории меры и в теории вероятностей вариант алгебры множеств, в котором определены операции над множествами, но отсутствуют определения отношений включения и равенства [4]. Частным случаем этого варианта алгебры множеств является σ-алгебра. Теоретической основой этого варианта считается аксиоматическая теория множеств. Этот урезанный вариант не позволяет использовать многие необходимые в работе методы логического анализа, в которых часто используется отношение включения множеств. Поэтому в данной публикации за основу принят вариант алгебры множеств, предложенный в [3]. Для этого варианта были разработаны новые методы обоснования законов с помощью перебора вариантов, сформулированы и обоснованы новые законы, в составе которых имеется рассматриваемый ниже закон парадокса.

### *Обоснование законов алгебры множеств*

В книге Куранта и Роббинса перечислено 26 законов алгебры множеств. Ниже приведены некоторые из них, их номера соответствуют номерам в [3, с. 136-138]. Пусть $A$, $B$ и $C$ – произвольные множества, $\boldsymbol{U}$ – универсум рассуждения и $\varnothing$ – пустое множество. Тогда справедливы следующие законы.

2) Если $A \subseteq B$ и $B \subseteq A$, то $A = B$.

3) Если $A \subseteq B$ и $B \subseteq C$, то $A \subseteq C$.

12) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

13) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.

19) $A \cup \overline{A} = \boldsymbol{U}$.

20) $A \cap \overline{A} = \varnothing$.

23) $\overline{\overline{A}} = A$.

24) $A \subseteq B$ эквивалентно $\overline{B} \subseteq \overline{A}$.

25) $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

26) $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

В литературе закон 2 называют *законом эквивалентности*, закон 3 – *законом транзитивности* включения множеств, законы 12 и 13 – *законами дистрибутивности*, закон 23 – *законом двойного дополнения*, закон 24 – *законом контрапозиции*, а законы 25 и 26 – *законами де Моргана*. В логике закону 19 соответствует *закон исключенного третьего*, а закону 20 – *закон непротиворечия*.

В [3] предлагается проверить справедливость этих законов с помощью перебора вариантов соотношений между множествами. Здесь будет показано более подробно, как осуществляется этот перебор. На рисунке 1 изображена диаграмма Эйлера, с помощью которой можно выразить все возможные соотношения между двумя множествами $A$ и $B$ с учетом их общего универсума $\boldsymbol{U}$.

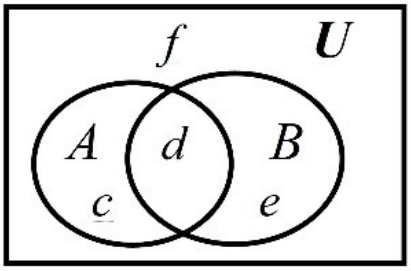



Рисунок 1 – Диаграмма Эйлера для двух множеств

На этом рисунке обратим внимание на области $c$, $d$, $e$ и $f$, не имеющие внутренних границ. Ясно, что эти области не пересекаются друг с другом, а их объединение равно $\boldsymbol{U}$. Тем самым они образуют *разбиение* универсума. Это позволяет использовать эти области в качестве элементов множеств $\boldsymbol{U}$, $A$ и $B$. Тогда для доказательства законов алгебры множеств можно взять за основу следующие исходные данные:

$\boldsymbol{U} = \{c, d, e, f\}$; $A = \{c, d\}$; $B = \{d, e\}$.

Используя эти исходные данные, докажем один из законов де Моргана: $\overline{A \cup B} = \overline{A} \cap \overline{B}$. Для этого в соответствии с определениями операций алгебры множеств последовательно выполним следующие вычисления.

1) $A \cup B = \{c, d, e\}$;

2) $\overline{A \cup B} = \{f\}$;

3) $\overline{A} = \{e, f\}$;

4) $\overline{B} = \{c, f\}$;

5) $\overline{A} \cap \overline{B} = \{f\}$;

6) Сравним 2 и 5. В результате получим: $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

Вычисления подтверждают справедливость закона де Моргана. Однако этого недостаточно, так как необходимо проверить этот закон для всех возможных вариантов соотношений между множествами $A$ и $B$. Эти варианты можно получить, если последовательно исключать элементы $c$, $d$, $e$ и $f$ и их всевозможные сочетания из универсума $\boldsymbol{U}$. Нетрудно убедиться, что таких вариантов 16. Для ясности перечислим их.

1) $\boldsymbol{U} = \{c, d, e, f\}$; $A = \{c, d\}$; $B = \{d, e\}$.

2) $\boldsymbol{U} = \{c, d, e\}$; $A = \{c, d\}$; $B = \{d, e\}$.

3) $\boldsymbol{U} = \{c, d, f\}$; $A = \{c, d\}$; $B = \{d\}$.

4) $\boldsymbol{U} = \{c, e, f\}$; $A = \{c\}$; $B = \{e\}$.

5) $\boldsymbol{U} = \{d, e, f\}$; $A = \{d\}$; $B = \{d, e\}$.

6) $\boldsymbol{U} = \{c, d\}$; $A = \{c, d\}$; $B = \{d\}$.

7) $\boldsymbol{U} = \{c, e\}$; $A = \{c\}$; $B = \{e\}$.

8) $\boldsymbol{U} = \{c, f\}$; $A = \{c\}$; $B = \varnothing$.

9) $\boldsymbol{U} = \{d, e\}$; $A = \{d\}$; $B = \{d, e\}$.

10) $\boldsymbol{U} = \{d, f\}$; $A = \{d\}$; $B = \{d\}$.

11) $\boldsymbol{U} = \{e, f\}$; $A = \varnothing$; $B = \{e\}$.

12) $\boldsymbol{U} = \{c\}$; $A = \{c\}$; $B = \varnothing$.

13) $\boldsymbol{U} = \{d\}$; $A = \{d\}$; $B = \{d\}$.

14) $\boldsymbol{U} = \{e\}$; $A = \varnothing$; $B = \{e\}$.

15) $\boldsymbol{U} = \{f\}$; $A = \varnothing$; $B = \varnothing$.

16) $\boldsymbol{U} = \varnothing$; $A = \varnothing$; $B = \varnothing$.

Например, докажем справедливость закона де Моргана для варианта 6:

$\boldsymbol{U} = \{c, d\}$; $A = \{c, d\}$; $B = \{d\}$.

1) $A \cup B = \{c, d\}$;

2) $\overline{A \cup B} = \varnothing$;

3) $\overline{A} = \varnothing$;

4) $\overline{B} = \{c\}$;

5) $\overline{A} \cap \overline{B} = \varnothing$;

6) Сравним 2 и 5. В результате получим: $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

Рассмотрим доказательства закона контрапозиции (номер 24). В этом законе соотношение $A \subseteq B$ является обязательным условием, поэтому необходимо учитывать те варианты, в которых соблюдается это соотношение. К ним относятся 8 вариантов: 5, 9, 10, 11, 13, 14, 15, 16.

Докажем этот закон для варианта 5: $\boldsymbol{U} = \{d, e, f\}$; $A = \{d\}$; $B = \{d, e\}$.

1) Убедимся, что $A \subseteq B$: $\{d\} \subseteq \{d, e\}$;

2) $\overline{A} = \{e, f\}$;

3) $\overline{B} = \{f\}$;

4) проверка показывает, что $\overline{B} \subseteq \overline{A}$.

Нетрудно убедиться, что закон контрапозиции подтверждается и для всех других перечисленных вариантов. Кроме того, если рассмотреть остальные варианты соотношений между множествами, в которых не соблюдается $A \subseteq B$, то можно убедиться, что для множеств $\overline{B}$ и $\overline{A}$ соотношение $\overline{B} \subseteq \overline{A}$ также не соблюдается во всех этих вариантах. Тем самым подтверждается равносильность соотношений $A \subseteq B$ и $\overline{B} \subseteq \overline{A}$.

Разумеется, предложенный метод доказательства, в котором используется «тупой» перебор вариантов, многим может показаться весьма трудоемким. Например, чтобы доказать законы дистрибутивности, в которых участвуют 3 множества, потребуется рассмотреть 256 вариантов в каждом случае. Можно сократить число вариантов, если для доказательства этого закона воспользоваться другими, менее трудоемкими при обосновании,

законами алгебры множеств. Идею такого доказательства законов дистрибутивности с использованием закона эквивалентности можно найти в книге [5, с. 29].

Однако, несмотря на большой объем вычислений, несомненное преимущество этого метода в том, что он ясно показывает возможность строгого доказательства законов алгебры множеств без аксиом. Поскольку законы алгебры множеств соответствуют законам классической логики, то это означает, что законы классической логики можно обосновать без использования теории формальных систем.

### *Полисиллогистика: модель рассуждений на основе алгебры множеств*

Одной из причин того, что в современной логике возникают трудности при распознавании ошибок и некорректностей в рассуждениях, является недостаточно полная постановка задач логического анализа. Традиционно считается, что правильно построенное рассуждение содержит написанные по определенным правилам формулы (*посылки*) и *правила вывода*, с помощью которых осуществляется получение или проверка правильности следствий. В то же время многие логические некорректности (например, двусмысленность) не отражены в этой модели, в силу чего их распознавание и анализ выходят за рамки строгой математики. С целью устранения этих трудностей предлагается в модель логического анализа помимо посылок и правил вывода добавлять подходящие по смыслу ограничения, нарушение которых свидетельствует о некорректности рассуждения. Примерами таких ограничений являются неравенства, с помощью которых устанавливается невозможность равенства определенных терминов пустому множеству или невозможность равенства между собой двух разных терминов. С помощью таких ограничений можно обнаруживать в рассуждениях парадоксы и двусмысленности. С учетом этого предлагается следующая модель рассуждений, в основе которой лежат законы алгебры множеств.

Пусть в универсуме $\boldsymbol{U}$ задана ***система множеств*** $S_i$ ($i = 1, 2, \dots, n$) и их дополнений $\overline{S_i}$ без перечисления их элементов. ***Утверждениями*** (суждениями) в данной системе являются соотношения между некоторыми парами множеств, причем предусмотрены соотношения двух типов:

***Общие суждения*** заданы как соотношения включения (например, $\overline{S_2} \subseteq S_3$).

***Частные суждения***, выражают то, что некоторые пары множеств содержат общие элементы, т. е. имеют непустое пересечение (например, $S_2 \cap S_6 \neq \varnothing$).

Также в этой системе анализа рассуждений помимо посылок можно использовать два вида ограничений:

- ***ограничения подмены термина*** (или двусмысленности) выражаются как недопустимость равенства некоторых пар множеств (например, $S_2 \neq \overline{S_4}$);
- ***ограничения пустоты*** выражаются как недопустимость равенства пустому множеству некоторых из заданных множеств (например, $S_4 \neq \varnothing$).

Кроме того, в этой модели в качестве одного из условий можно предусмотреть

- ***предполагаемое следствие***, которое выражено в виде общего суждения.

Утверждения в данной системе рассуждений соответствуют Аристотелевым суждениям силлогистики:

***A***: Все $P$ есть $Q$, пример: «Все крокодилы рептилии». В алгебре множеств: $P \subseteq Q$.

***I***: Некоторые $P$ есть $Q$, пример: «Некоторые студенты спортсмены». В алгебре множеств: $P \cap Q \neq \varnothing$.

***E***: Все $P$ не есть $Q$, пример: «Все жирафы не земноводные». В алгебре множеств: $P \subseteq \overline{Q}$.

***O***: Некоторые $P$ не есть $Q$, пример: «Некоторые мои коллеги не любят критику». В алгебре множеств: $P \cap \overline{Q} \neq \varnothing$.

Здесь ***A***, ***I***, ***E***, ***O*** – общепринятые обозначения типов суждений, при этом типы ***A*** и ***E*** называются *общими* суждениями, так как в них идет речь обо *всех* объектах, обозначенных символом $P$, а типы ***I*** и ***O*** – *частными*, поскольку в них говорится о существовании *некоторых* $P$.

В модели может содержаться любое количество исходных утверждений (посылок), поэтому она названа ***полисиллогистикой***. Однако данная модель существенно отличается от традиционной полисиллогистики тем, что, во-первых, содержит ограничения и, во-вторых, она характеризуется свойствами, которые отличаются от свойств традиционных силлогистики и полисиллогистики. К этим свойствами относятся следующие.

1) Утверждения модели отличаются от Аристотелевых суждений тем, что в них не действуют запреты, которые приняты в силлогистике. Во-первых, в утверждениях разрешается использовать термины с отрицанием на первом месте (например, $\overline{P} \subseteq Q$), и допускаются утверждения, в которых оба термина отрицательные (например, $\overline{P} \cap \overline{Q} \neq \varnothing$). Во-вторых, в общих утверждениях типа $P \subseteq Q$ допускается возможность равенства первого термина ($P$) пустому множеству.

2) В качестве множеств $S_i$ или $\overline{S_i}$ в рассматриваемой модели можно использовать формулы исчисления высказываний или предикатов, выраженных с помощью структур *алгебры кортежей*, которая изоморфна алгебре множеств [7].

Эти свойства предложенного варианта полисиллогистики позволяет избежать ошибок [6] и, кроме того, существенно расширяют аналитические возможности системы. В ней возможно решение следующих шести задач:

***Задача 1***: Найти следствия в виде соотношений включения между не заданными в посылках парами множеств (тем самым выводятся следствия в виде новых общих суждений);

***Задача 2***: Проверить, нарушаются ли в данной системе ограничения подмены термина;

***Задача 3***: Если задано предполагаемое следствие, то проверить, выводится ли это следствие из заданных посылок. В случае отрицательного результата осуществляется поиск возможных посылок, добавление которых в модель рассуждений позволяет трансформировать предполагаемое не выводимое следствие в обычное следствие. Найденные варианты таких посылок являются *абдуктивными заключениями*;

***Задача 4***: Проверить, нарушаются ли в данной системе ограничения пустоты (проверяется отсутствие или наличие парадоксов в рассуждении);

***Задача 5***: Найти новые пары множеств, для которых доказывается непустое пересечение (тем самым выводятся следствия в виде новых частных суждений);

***Задача 6***: Установить, для каких множеств, помимо тех, что заданы в ограничениях пустоты, доказывается их безусловное неравенство пустому множеству.

В силлогистике с большими трудностями и не без ошибок решаются только Задачи 1 и 5 [6]. Решения Задач 1, 2, 3, 4, 5 с использованием *E*-структур можно найти в книге [7]. Недавно выяснилось, что для решения Задач 4, 5 и 6 необходимо использовать ранее неизвестные законы алгебры множеств. Эти новые законы рассмотрены в следующем разделе.

Решение всех перечисленных выше Задач удобно начать с построения графа включений. Тогда их анализ существенно упрощается.

***Литералом*** назовем обозначение множества или его дополнения, при этом ясно, что, например, $S_2$ и $\overline{S_2}$ – это разные литералы.

***Графом включений*** назовем граф, на котором в качестве вершин используются литералы, а заданные или полученные в процессе анализа отношения включения между литералами изображаются в виде дуг (т.е. линий со стрелками). Если, например, известно, что $S_4 \subseteq \overline{S_2}$, то на графе включений дуга направлена от литерала $S_4$ к литералу $\overline{S_2}$ ($S_4 \to \overline{S_2}$).

Анализ графа включений предусматривает распознавание в нем путей и циклов.

***Путь*** из литерала $S_k$ в литерал $S_m$ в соответствии с законом транзитивности означает, что между соответствующими множествами имеется отношение включения ($S_k \subseteq S_m$).

***Цикл*** в графе включений в соответствии с законами транзитивности и эквивалентности свидетельствует о том, что все термины, содержащиеся в цикле, представляют эквивалентные друг другу множества.

Для упрощения мы не будем на рисунках изображать точки, как это принято в теории графов, – просто будем рисовать дуги от литерала к литералу.

Порядок действий при решении всех 6-ти перечисленных выше задач начинается одинаково:

1) рисуем граф включений для всех посылок (как изображать на графе частные суждения, будет показано далее);

2) применяем закон контрапозиции ко всем посылкам и добавляем эти новые соотношения в граф включений.

**Пример 1**. Данный, немного измененный, пример взят из книги [8]. В этом примере термин «ростовщик» заменен на термин «мошенник» и добавлена еще одна посылка. Пример интересен тем, что в нем демонстрируется ситуация, когда абсурдное следствие обусловлено не ошибками в логическом выводе, а незаметно присутствует в посылках. Даны посылки:

1) те, кто нарушает свои обещания, не заслуживают доверия;

2) все любители выпить очень общительны;

3) человек, выполняющий свои обещания, честен;

4) ни один трезвенник не мошенник;

5) тому, кто очень общителен, всегда можно верить;

6) все честные люди не мошенники.

Необходимо вывести новые общие суждения и проверить существование мошенников, т.е. проанализировать соблюдается ли ограничение пустоты для термина «мошенники».

Сформулируем посылки полисиллогизма в виде соотношений между следующими множествами: $\boldsymbol{U}$ – некоторое сообщество людей; $S_1$ – нарушающие обещания; $S_2$ – заслуживающие доверия; $S_3$ – любители выпить; $S_4$ – очень общительные; $S_5$ – честные; $S_6$ – мошенники.

Тогда суждения Примера 1 можно выразить в виде следующих соотношений.

1) $S_1 \subseteq \overline{S_2}$ ; 2) $S_3 \subseteq S_4$; 3) $\overline{S_1} \subseteq S_5$; 4) $\overline{S_3} \subseteq \overline{S_6}$ ; 5) $S_4 \subseteq S_2$; 6) $S_5 \subseteq \overline{S_6}$ .

*Ограничение пустоты*: $S_6 \neq \varnothing$.

Теперь построим граф включений (рисунок 2)

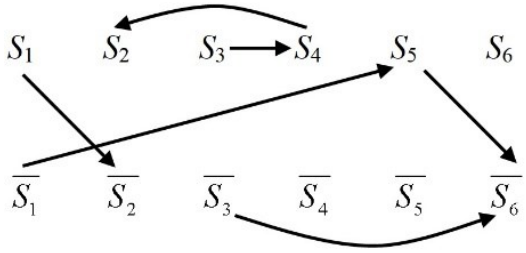



Рисунок 2 – Граф включений для посылок Примера 1

Для упрощения анализа граф включений лучше изображать таким способом: в верхней строке пусть будут расположены все позитивные литералы, в нижней – все негативные. Так мы не пропустим ни одного литерала и к тому же, как будет показано ниже, на такой схеме, в которой противоположные литералы расположены по вертикали, можно легко выводить контрапозиции исходных суждений.

Применим закон контрапозиции (и там, где нужно, закон двойного дополнения) к исходным посылкам. Тогда получим в качестве следствий следующие утверждения:

7) $S_2 \subseteq \overline{S_1}$ ; 8) $\overline{S_4} \subseteq \overline{S_3}$ ; 9) $\overline{S_5} \subseteq S_1$; 10) $S_6 \subseteq S_3$; 11) $\overline{S_2} \subseteq \overline{S_4}$ ; 12) $S_6 \subseteq \overline{S_5}$ .

Построим граф включений для посылок и их контрапозиций (изображены штриховыми линиями).

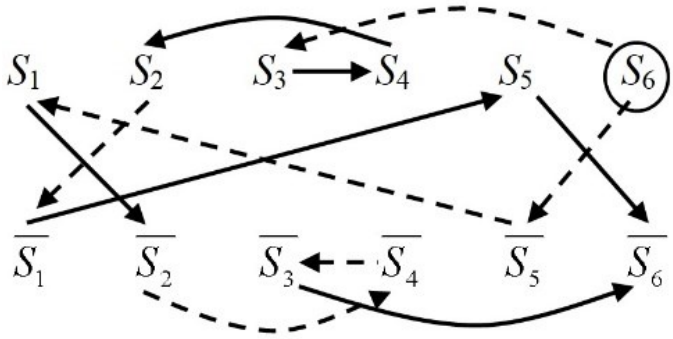



Рисунок 3 – Посылки и следствия Примера 1

Из рисунка 3 видно, что контрапозиции исходных суждений для данного варианта графа включений можно легко построить, используя одно простое правило: *если исходное суждение соединяет пару литералов, то контрапозиция этого суждения соединяет противоположные литералы, при этом направление дуги (вправо или влево) меняется на обратное*.

Теперь легко находится решение задачи. Выберем на графе литералы, в которые не входит ни одна дуга (*начальные литералы*, в данном примере только один литерал $S_6$) и проследим пути из него. В результате получим два *пути* на графе (второй путь является контрапозицией первого):

$S_6 \to S_3 \to S_4 \to S_2 \to \overline{S_1} \to S_5 \to \overline{S_6}$ ; $\quad S_6 \to \overline{S_5} \to S_1 \to \overline{S_2} \to \overline{S_4} \to \overline{S_3} \to \overline{S_6}$ .

Закон транзитивности позволяет нам утверждать, что если имеется путь от одного литерала к другому, то множество, соответствующее начальному литералу, включено в множество, соответствующее конечному литералу. Отсюда ясно, что в обоих случаях получим $S_6 \subseteq \overline{S_6}$ . Эта ситуация названа ***коллизией парадокса*** [7].

При этом, если заменить обозначения литералов соответствующими терминами, то появится ряд взаимно исключающих утверждений: мошенники одновременно трезвенники и любители выпить, общительные и необщительные, честные и нечестные и т. д. Посмотрим, соблюдается ли при этом заданное ограничение $S_6 \neq \varnothing$. В *E*-структурах [7] выражение $S_6 \subseteq \overline{S_6}$ равносильно тому, что $S_6 = \varnothing$. Это говорит о том, что ограничение пустоты в данном примере не выполняется.

Однако вывод о том, что из коллизии парадокса следует равенство соответствующего литерала пустому множеству, в *E*-структурах был основан на аксиомах. Это соотношение не встречается и в известных законах алгебры множеств [3]. Поэтому целесообразно сформулировать и обосновать этот закон. Заодно в следующем разделе будут сформулированы и обоснованы еще два новых закона алгебры множеств, которые необходимы при решении Задач 5 и 6.

Стоит отметить, что если в Примере 1 исключить 6-е суждение «все честные люди не мошенники», то парадокс не появится, но при этом мы получим абсурдное следствие «все мошенники честные», что явно не соответствует значению слова «мошенник».

### *Новые законы алгебры множеств*

Необходимость в новых законах обусловлена хотя бы тем, что без них невозможно решение некоторых задач полисиллогистики. Если учесть, что структуры алгебры кортежей представляют формулы исчисления высказываний и предикатов, а отношение включения между этими структурами изоморфно отношению логического следования [7], то данные законы применимы и в задачах логического вывода в математической логике. Первый закон было предложено назвать законом парадокса.

**(*i*) *Закон парадокса***: если доказано, что $X \subseteq \overline{X}$, то $X = \varnothing$.

*Доказательство*. Из определения операции дополнения следует, что в множестве $\overline{X}$ содержатся те и только те элементы универсума $\boldsymbol{U}$, которые не являются элементами $X$. Предположим, что имеется элемент $b$ такой, что $b \in X$. В то же время из условия $X \subseteq \overline{X}$ следует, что элемент $b$ содержится в $\overline{X}$, т.е. по определению $b \notin X$. Полученное противоречие можно разрешить единственным способом: $X = \varnothing$.

Второй закон используется в тех случаях, когда в рассуждении присутствуют или выводятся частные суждения (Задача 5). Формулировка частных суждений в виде неравенства (например, $S_2 \cap S_6 \neq \varnothing$) не позволяет использовать законы контрапозиции и транзитивности. Поэтому было предложено дать следующую формулировку для частных суждений: соотношение $X \cap Y \neq \varnothing$ равносильно двум соотношениям $\alpha_k \subseteq X$ и $\alpha_k \subseteq Y$ при условии, что $\alpha_k \neq \varnothing$. Для обоснования этой замены потребовалась формулировка еще одного нового закона алгебры множеств

**(*ii*) *Условие непустого пересечения***: $\alpha_k \subseteq X$ и $\alpha_k \subseteq Y$ при условии $\alpha_k \neq \varnothing$ равносильно выражению $X \cap Y \neq \varnothing$.

*Доказательство*. Если $\alpha_k \neq \varnothing$, то существует элемент $b$ такой, что $b \in \alpha_k$. Из $\alpha_k \subseteq X$ и $\alpha_k \subseteq Y$ следует, что $b \in X$ и $b \in Y$, в силу чего справедливо $X \cap Y \neq \varnothing$.

Третий закон алгебры множеств потребовался при решении Задачи 6 (распознавание безусловно непустых множеств среди тех, которые не заданы в ограничениях пустоты).

**(*iii*) *Закон существования***: если $X \neq \varnothing$ и $X \subseteq Y$, то $Y \neq \varnothing$.

*Доказательство*. Поскольку $X \neq \varnothing$, то существует элемент $b$ такой, что $b \in X$. Поскольку $X \subseteq Y$, то $b \in Y$. Следовательно, $Y \neq \varnothing$.

Этот закон интересен тем, что он по форме соответствует давно известному в логике правилу вывода *modus ponens* (MP): если выводимы формулы $A$ и $A \to B$, то выводима формула $B$ ($\to$ – одно из обозначений логической связки импликации). По сути, закон **(*iii*)** – это MP применительно к множествам.

### *Математические методы устранения противоречий*

Противоречие – это не только логическая ошибка. Часто оно становится отправной точкой появления нового знания. В Теории решения изобретательских задач (ТРИЗ) одним из часто используемых методов решения является формулировка проблемы в форме противоречия с последующим применением одного из приемов устранения технических противоречий (в ТРИЗ разработано 40 приемов) [9].

Для логических противоречий типа $X \subseteq \overline{X}$ (коллизия парадокса) можно предложить три математических метода их устранения:

***Метод 1*** (применение закона парадокса): если посылки, на основе которых получена коллизия парадокса, не вызывают сомнения и отсутствует ограничение $X \neq \varnothing$, то использовать закон парадокса, т.е. принять $X = \varnothing$;

***Метод 2*** (корректировка одной из посылок): изменить сомнительную посылку в рассуждении;

***Метод 3*** (ввод нового параметра): если это допустимо для данной ситуации, то добавить новый параметр для $X$, изменение которого влечет изменение свойств $X$ и вместе с этим устранение парадокса.

Пример применения Метода 1 приведен в [7, с. 42]. Суть и алгоритмы реализации Метода 3 изложены в публикациях [10, 11]. Примерами, в которых противоречие устраняется с помощью ввода нового параметра, являются пресуппозиции и базы знаний с аномалиями противоречия.

Рассмотрим более подробно Метод 2. Можно откорректировать сомнительную посылку многими способами. В частности, в [7, с. 40-41] имеется пример, в котором парадокс устраняется после замены общего суждения в посылках на обратное ему общее суждение (например, ошибочное «все простое гениально» можно заменить на более правдоподобное «все гениальное просто»). Здесь предлагается способ, с помощью которого парадокс устраняется после преобразования посылки, выраженной в виде общего суждения, в частное суждение. Рассмотрим пример.

**Пример 2**. В парадоксальный Пример 1 внесем следующее изменение: преобразуем вторую посылку «все любители выпить очень общительны» в более правдоподобное частное суждение «некоторые любители выпить очень общительны». Тогда вместо посылки $S_3 \subseteq S_4$ надо в соответствии с условием непустого пересечения ***(ii)*** ввести две посылки: $\alpha_k \subseteq S_3$ и $\alpha_k \subseteq S_4$. С учетом этого построим граф включений (рисунок 4).

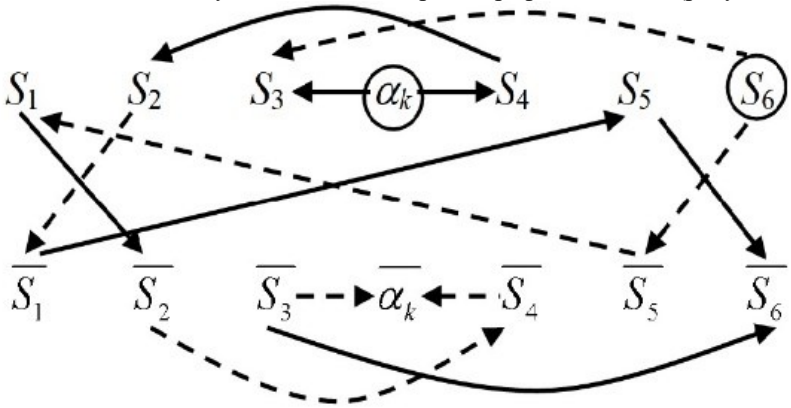



Рисунок 4 – Граф включений Примера 2

Анализ этого графа включений позволяет получить следующие выводы.

1) На графе включений имеются три начальных литерала: $S_6$, $\overline{S_3}$ и $\alpha_k$.

2) Из литерала $S_6$ (мошенники) исходят две ветви: $S_6 \to \overline{S_5} \to S_1 \to \overline{S_2} \to \overline{S_4} \to \overline{\alpha_k}$ и $S_6 \to S_3$. Это означает, что мошенники обладают следующими качествами: не честные, нарушающие обещания, не заслуживающие доверия, необщительные, любители выпить. Среди перечисленных свойств в отличие от Примера 1 нет контрадикторных, что означает отсутствие парадокса.

3) Из непустого литерала $\alpha_k$ исходят две ветви: $\alpha_k \to S_4 \to S_2 \to \overline{S_1} \to S_5 \to \overline{S_6}$ и $\alpha_k \to S_3$. По *условию непустого пересечения* ***(ii)*** получается, что непустое пересечение имеют пары литералов, находящихся на одной из этих ветвей (например, $S_4 \cap \overline{S_1} \neq \varnothing$, так как $\alpha_k \subseteq S_4$ и $\alpha_k \subseteq \overline{S_1}$ ), и пары литералов, содержащихся в разных ветвях, исходящих из $\alpha_k$ (например, $\overline{S_6} \cap S_5 \neq \varnothing$, так как $\alpha_k \subseteq \overline{S_6}$ и $\alpha_k \subseteq S_5$). Нетрудно убедиться, что среди этих пар нет пары ($S_5$, $S_6$), из чего следует, что частное суждение «некоторые мошенники честные» ($S_5 \cap S_6 \neq \varnothing$) не является заключением в данном примере.

Таким образом, замена хотя бы одного общего суждения в цепи включений, ведущей к парадоксу, частным суждением позволяет устранить полностью логическую катастрофу.

Необходимо обратить внимание также на обратное утверждение: возможны случаи, когда преобразование частного суждения в общее суждение преобразует корректное рассуждение в парадокс. При этом допускается логическая ошибка, которая называется *ложное обобщение*. Стоит отметить, что неявное принуждение нас к тому, чтобы мы допустили именно эту логическую ошибку (назовем этот прием *принуждением к ложному обобщению*), оказывается чуть ли не самым популярным способом манипуляции сознанием. При этом часто для манипулятора даже нет необходимости в том, чтобы провозглашать нам эти самые общие суждения – мы сами их послушно принимаем, когда манипулятор преподносит нам факты или сведения, которые ему выгодны, и замалчивает (или оценивает как ложь или происки недругов) сведения, которые не согласуются с его намерениями.

Предложенный в Примере 2 метод устранения противоречий можно обосновать и для общего случая, который выражен в следующей теореме.

***Теорема*** (устранение парадокса с помощью преобразования общего суждения в частное). Если противоречие в рассуждении вызвано цепью включений литералов $L_1 \subseteq L_2 \subseteq \ldots \subseteq L_n$, при условии $L_n = \overline{L_1}$ , то устранить парадокс можно с помощью выбора в этой цепи пары соседних литералов ($L_i$, $L_{i+1}$) и заменой соотношения $L_i \subseteq L_{i+1}$ двумя соотношениями $\alpha_k \subseteq L_i$ и $\alpha_k \subseteq L_{i+1}$ при условии $\alpha_k \neq \varnothing$.

*Доказательство*. При замене, предусмотренной в Теореме, в графе включений разрывается путь, ведущий от $L_1$ к $\overline{L_1}$, так как в исходной цепи дуга $L_i \to L_{i+1}$, преобразуется в подграф $L_i \leftarrow \alpha_k \to L_{i+1}$. В цепи, которая образуется из контрапозиций дуг, входящих в исходную цепь, и также, как и исходная, представляет путь из $L_1$ к $\overline{L_1}$, тоже происходит разрыв, так как дуга $\overline{L_{i+1}} \to \overline{L_i}$ заменяется на подграф $\overline{L_{i+1}} \to \overline{\alpha_k} \leftarrow \overline{L_i}$, который препятствует достижению литерала $\overline{L_1}$ из литерала $L_1$. Тем самым доказывается, что при замене общего суждения $L_i \subseteq L_{i+1}$ частным суждением $L_i \cap L_{i+1} \neq \varnothing$ парадокс устраняется. *Конец доказательства*.

Рассмотрим, как в Примере 2 можно решить Задачу 6 (распознавание безусловно непустых множеств в системе). Для этого используется *закон существования* (***iii***): если $A \neq \varnothing$ и $A \subseteq B$, то $B \neq \varnothing$. Из условий задачи ясно, что изначально заданными непустыми множествами в системе являются $S_6$ и $\alpha_k$. На графе включений (рисунок 4) можно проследить все ветви, исходящие из этих литералов. По закону существования любой литерал, содержащийся в этих ветвях, соответствует безусловно непустому множеству. Из анализа рисунка 4 ясно, что безусловно непустыми множествами являются все за исключением множества, обозначенного литералом $\overline{S_3}$ (трезвенники), так как только этот литерал недостижим ни из $S_6$, ни из $\alpha_k$.

### *Заключение*

В работе показано, что использования варианта алгебры множеств, предложенного в книге Куранта и Роббинса «Что такое математика?», позволяет существенно расширить возможности логического анализа знаний и рассуждений.

Предложено в систему логического анализа помимо посылок и правил вывода вводить ограничения, с помощью которых распознаются и устраняются логические ошибки и некорректности в рассуждениях.

Сформулированы и обоснованы новые законы алгебры множеств, без которых невозможно решение некоторых задач логического анализа.

Рассмотрены математические методы устранения противоречий в знаниях и рассуждениях.


### *Благодарности*

Работа поддержана Минобрнауки Российской Федерации (проект госзадания 124041500008-1)


### *Литература*


1. Groarke, Leo, Informal Logic, The Stanford Encyclopedia of Philosophy (Spring 2024 Edition), Edward N. Zalta & Uri Nodelman (eds.). – Режим доступа: https://plato.stanford.edu/archives/spr2024/entries/logic-informal

2. Грифцова И. Н. Логика как теоретическая и практическая дисциплина. К вопросу о соотношении формальной и неформальной логики. – М.: Эдиториал УРСС, 1998. – 152 с. – Режим доступа: https://studfile.net/preview/3875463/

3. Курант Р., Роббинс Г. Что такое математика? 3-е изд., испр. и доп. – М.: МЦНМО, 2001. – 568 с. – Режим доступа: https://azbyka.ru/deti/wp-content/uploads/2021/09/chto-takoe-matematika.-kurrant-robbins.pdf

4. Сазонов В. В. Алгебра множеств // Математическая энциклопедия. – М.: Советская энциклопедия. Т.1. 1977. – С. 129-130.

5. Столл Р.Р. Множества. Логика. Аксиоматические теории. М.: Просвещение. 1968. – 231 с.

6. Кулик Б.А. Почему в учебниках логики содержатся логические ошибки? // Образовательные ресурсы и технологии. 2023. № 1(42). – С. 7–14.

7. Кулик Б.А. Логика и математика: просто о сложных методах логического анализа (под общ. ред. А.Я. Фридмана). СПб.: Политехника. 2020. – 144 с. – Режим доступа: http://logic-cor.narod.ru/index/knigi/0-9

8. Кэрролл Л. Символическая логика / Пер. с англ. Ю.А. Данилова // Кэрролл Л. История с узелками. М.: Мир, 1985. – С. 189-362.

9. Альтшуллер Г.С. Творчество как точная наука: Теория решения изобретательских задач. - М.: Сов. радио, 1979. - 175 с.

10. Кулик Б.А. Исследование противоречий в естественных рассуждениях на примерах метафор и пресуппозиций // Труды Семнадцатой Национальной конференции по искусственному интеллекту с международным участием. КИИ-2019 (21–25 октября 2019 г., г. Ульяновск, Россия). Ульяновск: УлГТУ, 2019. Т. 2. – С. 192-200.

11. Кулик Б.А., Фридман А.Я. Анализ парадоксов в интеллектуальных моделях систем // Труды Кольского научного центра РАН. Информационные технологии. Вып. 12. 2021. Т. 12, No 5 – С. 171–176.

***Кулик Б.А. Закон парадокса и математические методы устранения противоречий в знаниях и рассуждениях.*** *Предлагается в качестве математической основы логического анализа использовать независимый от теории множеств вариант алгебры множеств, содержащийся в книге Р. Куранта и Г. Роббинса «Что такое математика?». В этом варианте законы алгебры множеств, соответствующие законам классической логики, доказываются без аксиом методом перебора вариантов. Сформулированы и обоснованы новые законы алгебры множеств, в том числе закон парадокса. Предложена математическая модель рассуждений, в которой помимо посылок и правил вывода используются ограничения, позволяющие распознавать и анализировать некорректности в рассуждениях. Рассмотрены основанные на законах алгебры множеств методы устранения противоречий.*

***Ключевые слова:*** *алгебра множеств, модель рассуждений, полисиллогистика, граф включений, закон парадокса, закон существования, методы устранения противоречий.*

***Kulik B.A. The law of paradox and mathematical methods of eliminating contradictions in knowledge and reasoning.*** *It is proposed to use as a mathematical basis for logical analysis a variant of set algebra independent of set theory, contained in the book by R. Courant and H. Robbins "What is Mathematics?". In this variant, the laws of the algebra of sets corresponding to the laws of classical logic are proved without axioms by iterating over the options. New laws of the algebra of sets, including the law of paradox, are formulated and substantiated. A mathematical model of reasoning is proposed, in which, in addition to assumptions and rules of inference, restrictions are used to recognize and analyze inaccuracies in reasoning. The methods of eliminating contradictions based on the laws of set algebra are considered.*

***Keywords:*** *algebra of sets, model of reasoning, polysyllogistics, graph of inclusions, law of paradox, law of existence, methods of eliminating contradictions.*

УДК 004.9

# Семиотическая модель агглютинативного языка как ядро когнитивной системы объяснительного искусственного интеллекта

**Дж.Ш. Сулейманов**[1]
академик АН РТ, д.т.н., профессор, dvdt.slt@gmail.com, OrcID: 0000-0003-1404-0372, SPIN-код: 1476-6020
**А.Я. Фридман**[2]
д.т.н., профессор, fridman@iimm.ru, OrcID: 0000-0003-2408-6892, SPIN-код: 1512-1669
**А.Р. Гатиатуллин**[1]
к.т.н., ayrat.gatiatullin@gmail.com, OrcID: 0000-0003-3063-8147, SPIN-код: 4633-2846

[1] Казань, Институт прикладной семиотики АН РТ
[2] Институт информатики и математического моделирования ФИЦ КНЦ РАН

***Сулейманов Дж.Ш., Фридман А.Я., Гатиатуллин А.Р. Семиотическая модель агглютинативного языка как ядро когнитивной системы объяснительного искусственного интеллекта.*** *Представлена концепция построения когнитивной системы объяснительного ИИ на базе естественных языков агглютинативного типа (АЕЯ), поскольку имманентные им свойства создают основу для контекстно-зависимого синтеза коммуникационных и когнитивных процессов с использованием почти автоматных морфологических средств, не присущих другим группам языков. Система реализует семиотический подход к сопровождению децентрализованной совокупности лексико-грамматических моделей, ее пилотная версия ориентирована на семантику и грамматику татарского языка, поскольку апробация основных идей и результатов будет осуществляться в рамках электронного корпуса татарского языка «Туган тел» и лингвистического портала «Тюркская морфема». В качестве ядра программной системы предлагается семиотическая генеративно-распознающая модель татарского языка (СГРМ ТЯ), которая включает координируемый на метасистемном уровне коллектив компонентов вербализации и распознавания семантики,*

***Ключевые слова:*** *семиотическая генеративно-распознающая модель татарского языка, семантика естественного языка, вербализация и распознавание смысла, семиотическая модель лексико-грамматических средств языка, когнитивная система ИИ, аппарат «гиперадресации» для ускоренного доступа к базам данных и знаний, ситуационный анализ «фокуса внимания» системы моделирования.*

## *Введение*

В исследовании естественных языков можно выделить три аспекта: когнитивный, коммуникативный и технологический [1]. Когнитивный аспект – это характеристика ЕЯ с точки зрения возможностей описания модели мира, представления знаний, кодирования процессов мышления. Коммуникативный аспект отражает потенциал языка для кодирования, приема и передачи, семиотической обработки информации, организации диалога. Технологический аспект определяет формальный и концептуальный потенциал ЕЯ для реализации средств эффективной обработки, адекватного описания и компактного хранения информации на данном языке, а также для разработки интеллектуального программного инструментария, включая операционные системы.

Современные средства накопления и обработки знаний на естественном языке малоэффективны и практически не справляются с такими задачами, как поиск и отбор информации в распределенных базах данных, извлечение знаний, семантический анализ текстовой информации, на наш взгляд, прежде всего потому, что они изначально являются неинтеллектуальными, разработаны на основе примитивных искусственных языков программирования, которые представляют собой подмножества флективно-аналитических языков или искусственных конструкций, созданных на их основе. Еще одна причина сложностей в системах обработки ЕЯ связана с организацией их моделей, строящихся на основе формальных систем, в частности, порождающих грамматик (например, [2]), что создает две принципиальные проблемы: монотонность результатов логического вывода и пассивность инструментов логико-семантического анализа

информации. Такая организация моделей ЕЯ названа в работе [3] *глобальным подходом* к организации исследований ЕЯ.

Для снижения остроты указанных проблем в настоящей работе предлагается обратный подход к задаче моделирования естественных языков: строить для них системы моделирования на базе технологического инструментария вербализации и распознавания смысла, состоящего из семиотических моделей лексико-грамматических средств ЕЯ, в рамках идеологии децентрализованного подхода Г.С. Цейтина [3].

*Предпосылки проекта*: интенсивное распространение систем ИИ в различных областях человеческой деятельности требует решения вопроса об их построении в рамках парадигмы, дружественной к человеку и «понятной» ему.

*Цель проекта*: разработка когнитивной системы ИИ как основы «общего ментального пространства» человека и ИИ, способной интерпретировать, объяснять свои решения, советы и выводы, используя комплексный междисциплинарный подход.

*Метод решения*: развитие семиотической генеративно-распознающей модели татарского языка (СГРМ ТЯ), которая включает децентрализованный, но координируемый на метасистемном уровне коллектив компонентов вербализации и распознавания семантики, на основе применения методов и подходов из таких областей, как семиотика, математическая лингвистика, языкознание и информатика.

*Новизна подхода* определяется следующими основными аспектами:

- использованием *проактивных интеллектуальных средств обработки языковой информации*, отвечающих за семантически ориентированное исследование и преобразование данных, аналогично агентам в многоагентных системах, но функционирующих согласованно по условиям, формируемым координирующими модулями;
- координирующие модули содержат *семиотические модели Поспелова-Полякова*, отвечающие за целостность и полноту использования всей релевантной информации на каждом этапе работы всей системы;
- как во всех современных логических средствах работы с ЕЯ, основной структурой формализации и представления знаний служит *система онтологий, специфика ее построения состоит в поддержке ситуационного подхода*;
- *активностью знаний*, то есть первичностью анализа данных и вторичностью принятия решения на основе этого анализа;
- всеобъемлющим применением процедур объяснения принимаемых решений в целях понимания и адекватного восприятия машиной человека как «старшего», а также *создания «общего здравого смысла», «пространства общей ментальности», «общей картины мира» человека и ИИ*;
- ориентацией на агглютинативные ЕЯ (АЕЯ), в частности, ТЯ, которому свойственна регулярная, почти *автоматная морфология, обеспечивающая кодирование семантически сложной информации и уникальные возможности контекстного управления всеми процессами обработки данных*.

*Возможности реализации проекта* определяются, естественно, финансовыми и календарными ограничениями, но также связаны с имеющимся у коллектива участников проекта заделом:

- показано, что децентрализованное построение систем моделирования естественных языков (ЕЯ) создает обширные возможности применения *прагматически-ориентированного подхода к интеллектуализации* подобных систем (возможность оперировать недоопределенной информацией, семантически управляемой контекстом, и исполнять нечеткие команды), в частности, за счет построения и использования сложных семиотических моделей, изначально ориентированных на решение семантически нетривиальных задач [4];
- для логико-семантической обработки языковой информации разработана алгебра кортежей и *QC*-структуры, реализующие *общую теорию многоместных отношений* [5, 6];
- предложены средства *координации взаимодействий* инструментов анализа информации и контроля корректности хода обработки данных в рамках ситуационного подхода, при этом система моделирования в целом позиционируется как сетецентрическая структура [7];
- апробация идей и результатов выполнения проекта будет проводиться Институтом прикладной семиотики АН РТ с помощью *электронного корпуса ТЯ «Туган тел»*, в котором исследователям доступны 200 млн. словоформ с морфологической разметкой [8] и лингвистического портала «Тюркская морфема» [9].

В результате выполнения первого этапа проекта будет разработана система моделирования, автоматизирующая все этапы построения и сопровождения СГРМ ТЯ и включающая, кроме самой модели, различные вспомогательные подсистемы ее визуализации, заполнения, контроля корректности, постановки задач анализа и синтеза, а также соответствующие базы данных и знаний.

### *Неформальное описание основных идей СГРМ ТЯ*

Рабочий образ, который помогает в разработке нашей модели ТЯ – океан Солярис из одноименного романа Станислава Лема. В этой метафоре фантомы (призраки), порождаемые Солярисом – коммуникационные акты (КА). Для успеха КА фантом должен быть непротиворечивым (внешне и внутренне согласованным) и полным (содержать все существенные аспекты), иначе платье Хари будет без застежки на спине (могло отсутствовать и что-то более существенное), а также (по возможности) не быть избыточным (чтобы не портить красоту образа).

Формально модель можно соотносить с большими генеративными моделями языка (в части генерации языковых явлений, например, словоформ) [10], поэтому далее используется аббревиатура СГРМ ТЯ (Semiotic Generative-Recognizing Model of the Tatar Language – SGRM TL).

Существующие семантические модели ЕЯ практически формировались, преимущественно, на основе флективных индоевропейских языков и структурируются на основе корневых морфем [11], над их совокупностями формируются различные отношения и операции, которые определяют алгоритмы обработки данных на разных уровнях работы модели. Это допустимо в базовых моделях ЕЯ (Соляриса), поскольку в них не требуется оперативно модифицировать параметры модели, но не позволяет моделировать КА, в ходе которых контексты нужно менять в реальном времени, для чего не хватает мощности даже современных вычислительных систем. Поэтому структура СГРМ ТЯ базируется на функциональной классификации аффиксальных морфем ТЯ, позволяющих реализовать следующие полезные в рассматриваемой задаче и эффективные с точки зрения алгоритмизации возможности работы с информацией:

- контекстно-управляемое кодирование семантически сложной информации;
- фиксированная позиция аффиксов;
- компактность представления и обработки именных и глагольных групп;
- морфологический эллипсис и рекурсия;
- проактивность аффиксальных морфем;
- морфологическое выражение недоопределенности явлений (объектов, отношений, ситуаций и др.), нечетких действий и команд, легко восстанавливаемых в контексте [12].

Имеются и другие особенности, например, рекурсия, определяющие создание *цепочек морфем неограниченной длины* за счет присоединения новых аффиксов, становясь последовательностью аффиксальных гиперссылок, причем *каждая новообразованная словоформа обязательно обладает определенным смыслом* благодаря гиперссылке. Такая гиперссылка может даже указывать на концепт, который еще не «обнаружен» или не может быть пока охвачен нашим разумом.

Аффиксальные гиперссылки решают задачи *семантической обработки АЕЯ*, «прослеживания» формирования контекста и ретроспективного просматривания переходов между контекстами, соответственно они *обладают объяснительной силой*. Вместе с тем, задача минимизации времени обработки в условиях больших объемов информации, то есть повышения быстродействия, также остается актуальной, хотя и не столь критичной ввиду улучшения характеристик аппаратного обеспечения моделирования.

Теперь проиллюстрируем примером отмеченные выше особенности ТЯ как потенциального универсального языка интеллектуальных систем.

### *Пример автоматной морфологии ТЯ*

Словоформа *Tatarchalashtyrgalashtyruchylarnykyndagydajmyni* (1)
является полностью корректным татарским словом, означающим следующее:

«Неужели это похоже на то, что на том, что принадлежит тем, кто занимается татарской локализацией время от времени изредка и еще реже?».

Одновременно, это демонстрация потенциала *татарского языка как средства сжатия информации*. Одна словоформа передает смысл, для объяснения которого в языках индо-европеской группы потребуется 10 и более словоформ.

В (1) 13 аффиксальных морфем и одна корневая морфема «*Tatar* (татарин)»:

*Tatar/cha/la/shtyr/gala/shtyr/u/chy/lar/nyky/ndagy/daj/my/ni.*

Корневую морфему «*Tatar*» можно заменить любым концептом из того же грамматического класса без изменения смысла всей словоформы:

(*x*0)*chalashtyrgalashtyruchylarnykyndagydajmyni*,

где, например, *x*0 = {*tatar*, *rus*, *ispan*, *japon*}.

Когнитивное свойство *глубинного эллипсиса* при перечислении разрешает вывести всю одинаковую для всех концептов последовательность аффиксов вправо с сохранением ее в последнем слове. Так, в предложении:

"*Ul minnen alarnyng sanaklary Tatarchalashtyrgalashtyruchylarnykyndagydajmyni, ruschalashtyrgalashtyruchylarnykyndagydajmyni, ispanchalashtyrgalashtyruchylarnykyndagydajmyni, japonchalashtyrgalashtyruchlarnykyndagydajmyni dip sorady*"
применение эллипсиса дает почти четырехкратное сжатие информации:

"*Ul minnen alarnyng sanaklary tatar, rus, ispan, japonchalashtyrgalashtyruchylarnykyndagydajmyni dip sorady*".

При пошаговом (агглютинативном) образовании сложных словоформ типа (1) *каждая новообразованная словоформа обязательно обладает смыслом*.

1. *Tatar* (существительное) – татарин.
2. *Tatarcha* (наречие) – по-татарски: переключение существительного в наречие.
3. *Tatarchala* (глагол, повелительное наклонение) – делай татарским: переключение наречия в глагол.
4. *Tatarchalashtyr* (глагол, повелительное наклонение) – делай татарским время от времени (локализуй ПК на татарский язык время от времени): придание глаголу признака частотности.
5. *Tatarchalashtyrgala* (глагол, повелительное наклонение) – делай татарскую локализацию время от времени, изредка: придание глаголу признака частотности, повторяемости.
6. *Tatarchalashtyrgalashtyr* (глагол) – занимайся татарской локализацией время от времени, изредка и еще реже: придание признака повторяемости.
7. *Tatarchalashtyrgalashtyru* (существительное) – занятие татарской локализацией время от времени, изредка и еще реже: придание признака процесса.
8. *Tatarchalashtyrgalashtyruchy* (существительное) – тот, кто занимается татарской локализацией время от времени, изредка и еще реже.
9. *Tatarchalashtyrgalashtyruchylar* (существительное) – те, кто занимается татарской локализацией время от времени, изредка и еще реже: оператор множественности.
10. *Tatarchalashtyrgalashtyruchylarnyky* (существительное) – то, что принадлежит тем, кто занимается татарской локализацией время от времени, изредка и еще реже: ссылка на объект, принадлежащий тому, кто выражен словоформой слева от этого последнего аффикса.
11. *Tatarchalashtyrgalashtyruchylarnykyndagy* (существительное) – то, что на том, что принадлежит тем, кто занимается татарской локализацией время от времени, изредка и еще реже: ссылка на объект, который находится на том, что выражено словоформой слева от этого аффикса.
12. *Tatarchalashtyrgalashtyruchylarnykyndagydaj* – как то, что на том, что принадлежит тем, кто занимается татарской локализацией время от времени, изредка и еще реже: оператор модальности, отражающего сравнение, со смыслом схожести.
13. *Tatarchalashtyrgalashtyruchylarnykyndagydajmy* – похоже ли это на то, что на том, что принадлежит тем, кто занимается татарской локализацией время от времени, изредка и еще реже? Аффикс модальности, ставящий под вопрос смысл, передаваемый словоформой слева от этого аффикса.
14. *Tatarchalashtyrgalashtyruchylarnykyndagydajmyni* – неужели это похоже на то, что на том, что принадлежит тем, кто занимается татарской локализацией время от времени, изредка и еще реже? Аффикс модальности, придающий вопросу оттенок удивления, восхищения, сомнения.

*Это свойство ТЯ существенно повышает возможности тонкого семантического анализа текстов в парадигме концептуальных пространств* [13]. При этом добавление каждого аффикса соответствует добавлению одной размерности качества (quality dimension [13]) к концептуальному пространству текущего КА, меняя таким образом контекст и перечень концептов, существенных в данном контексте.

В рассмотренной цепочке аффиксов большинство недоопределенных, что позволяет подставлять в них релевантные параметры и таким образом для одной и той же цепочки *генерировать множество контекстов*.

Поаффиксальный разбор словоформы (1):

*Tatar* (*x*0)+*cha* (sw1: adverb) + *la* (sw2: veb1) + *shtyr* (frq: reduction) + *u* (process: localization) + chy (sw2; *x*1: noun: object reference) +*lar* (op1: plural) + *nyky*(*x*2: concept reference) + *dagy*(*x*3: concept reference) + *daj* (op2: mod: assm) + *my* (op3: mod: question) + *ni* (op4: mod: wonder).

Обозначения: *xi* – параметр; swi – switch; frqi – frequency; opi – operator; modi – modality; assm – assimilation.

Пусть: *x*0 = *Tatar*; *x*1 = любой субъект, например, программист; *x*2 = любой объект, принадлежащий программистам, например, стол; *x*3 = любой предмет, который на этом столе, например, компьютер.

Тогда недоопределенная словоформа (1) сгенерирует конкретную фразу:

«Неужели это похоже на компьютер, что на столе, который принадлежит программистам, кто занимается татарской локализацией время от времени, изредка и еще реже?»

Представленное свойство ТЯ, кроме богатства семантической нюансировки текстов, обеспечивает две важных в рассматриваемой задаче возможности:

– типизацию обобщенных семантических схем ситуаций;

– использование параметров цепочек для гиперадресации – ускоренного сужения пространства поиска при генерации КА.

Для реализации этих возможностей в СГРМ ТЯ предназначен специализированный *метауровень семантического поиска* значений параметров аффиксальных цепочек. Алгоритмизация функционирования этого уровня входит в планы будущих исследований по данной теме, которые кратко представлены ниже.

### *Выводы и планы будущих исследований*

По сравнению с аналогами, разрабатываемая система имеет следующие отличительные *особенности и преимущества*:

- модель языка структурируется *с опорой на семантику концептов*, а не грамматические (синтаксические и морфологические) признаки лингвистических единиц, которыми эти концепты выражаются в тексте;
- управление моделью осуществляется *в рамках децентрализованного подхода* с помощью набора проактивных инструментальных средств, реализующих семиотические модели Д.А. Поспелова-Полякова [14, 15];
- быстродействие системы моделирования повышается за счет использования морфологических свойств аффиксальных морфем агглютинативного языка для *гиперадресации* семантических методов поиска по принципу «*кротовых нор*» (червоточин – wormholes, например, [16]) в астрофизике. *Авторы надеются, что наша идея будет скорее реализована на практике.*

При децентрализованной структуре системы моделирования ЕЯ (ее основной состав представлен во Введении) некоторые из ее подсистем вследствие широкого использования приобретают общеязыковое значение, и тогда можно, как в случае синтаксиса, говорить о системе, относящейся ко всему языку, за отдельными исключениями. В случае же семантики, где удается хорошо формализовывать частные подсистемы (например, обозначения отношений родства или моментов времени), такой доминирующей в масштабе всего языка системы не обнаружено. Наиболее развитой из формализованных систем для моделирования семантики является заимствованная из математики теоретико-множественная семантика, использующая аппарат математической логики. Отсутствие в этой системе ориентации на конкретный тип объектов порождает надежды на то, что ее развитие даст возможность описать семантику всего языка [3]. Такая структура, видимо, особенно перспективна для агглютинативных ЕЯ; учитывая профессиональные интересы и имеющийся у авторов значительный задел (например, [17-19]), ее предполагается реализовать на примере татарского языка. С большой вероятностью, представленный здесь подход может позволить продвинуться в моделировании языка для систем искусственного интеллекта.

При анализе публикаций нами выявлены некоторые существенные параллели с моделью мозга К.В. Анохина [20], косвенно подтверждающие целесообразность проводимых исследований:

когнитом (познавательные способности мозга) ↔ Солярис (СГРМ ТЯ);
кротовая нора (ассоциации) ↔ кротовая нора (цепочки аффиксов);
гиперсеть (нейронная) ↔ гиперсеть (сетецентрическая структура модели);
нейрон ↔ семиотическая универсалия (СУ);
специализированный нейрон ↔ локализованная в контексте СУ (с частично или полностью определенными валентностями);
снопы (пучки элементов гиперсети) ↔ цепочки семиотических универсалий;
доступные опции (affordances) ↔ прагматическая ориентация инструментов;
«второй этаж» мозга ↔ метауровень выбора параметров цепочек.

Вероятнее всего, в ходе дальнейшей работы список аналогий расширится.

Кроме непосредственного решения сформулированной здесь задачи, представленный подход перспективен для исследований по управляемости искусственного интеллекта человеком и по разработке ИИ, интерпретируемого и интерпретирующего свои решения, а также созданию «общего здравого смысла», «общей ментальности», «общей картины мира» человека и ИИ. На основе СГРМ ТЯ также могут быть реализованы многие актуальные задачи обработки ЕЯ, включая такие, как общение человека с интеллектуальными системами и взаимодействие систем искусственного интеллекта между собой.

### *Литература*

1. Сулейманов, Д.Ш. К вопросу исследования технологического аспекта естественных языков // Обработка текста и когнитивные технологии: Труды XI Междунар. науч. конф. (Константа, 7–14 сентября 2009 г.). Казань: Изд-во Казан. гос. ун-та, 2010, с. 232-245.

2. Chomsky, N. Syntactic Structures. The Hague: Mouton, 1957.

3. Цейтин Г.С. О соотношении естественного языка и формальной модели. Архив АН СССР. Работа в Научном совете по комплексной проблеме "Кибернетика", 1980 г.

4. Сулейманов, Д. Ш., Гатиатуллин, А. Р. Структурно-функциональная компьютерная модель татарских морфем. – Казань: Фэн, 2003.

5. Kulik, B. A., & Fridman, A. Ya. Complicated Methods of Logical Analysis Based on Simple Mathematics. Cambridge Scholar Publishingб 2022. ISBN: 1-5275-8014-8.

6. Kulik, B., Fridman, A. N-ary Relations for Logical Analysis of Data and Knowledge. IGI Global, 2017.

7. Fridman, A. Situational Modeling: Definitions, Awareness, Simulation. USA: Nova Science Publishers, Inc., 2023. https://doi.org/10.52305/XIKU5849

8. «Туган тел». Режим доступа: https://tugantel.tatar/

9. Gatiatullin, A., Suleymanov, D., Prokopyev, N., Khakimov, B. About turkic morpheme portal // CEUR Workshop Proceedings, Kazan, 12–13 ноября 2020 года. – Kazan, 2020. – P. 226-243.

10. Бахтизин, А. Р., Брагин, А. В., Макаров, В. Л. Большие языковые модели четвёртого поколения как новый инструмент в научной работе // Искусственные общества. – 2023. – Т. 18. – Выпуск 1. URL: https://artsoc.jes.su/s207751800025046-9-1/. DOI: 10.18254/S207751800025046-9

11. Swart, de, H. Introduction to Natural Language Semantics. U.S.A., Stanford, CA: CSLI Publications, 1998. ISBN: 1575861380 (9781575861388).

12. Suleymanov, D. Sh. Natural Cognitive Mechanisms in the Tatar language // In the Collection of the Vienna Proceedings of the Twentieth European Meeting in Cybernetics and Systems Research. Edited by Robert Trappel. Vienna, Austria, 6-9 April, 2010. – 210-213 pp.

13. Gärdenfors, P. The Geometry of Meaning: Semantics Based on Conceptual Spaces. MIT Press, 2014.

14. Поспелов, Д. А. Ситуационное управление: теория и практика. – М.: Наука, 1986.

15. Поляков, В. Н. Проблемы представления, приобретения и использования знаний в свете обработки естественного языка // Когнитивно-семиотические аспекты моделирования в гуманитарной сфере / Научные редакторы – доктор технических наук В. Л. Стефанюк, доктор философских наук Э. А. Тайсина. – Казань: Изд-во АН РТ, 2017. – С. 145–163.

16. Konoplya, R. A.; Zhidenko, A. (4 March 2022). "Traversable Wormholes in General Relativity". Physical Review Letters. 128 (9): 091104. arXiv:2106.05034

17. Сулейманов, Д. Ш., Хакимов, Б. Э., Гильмуллин, Р. А. Концептуальные и лингвистические аспекты разработки корпуса татарского языка // Фэнни Татарстан. 2017. – № 2. – С. 7-16.

18. Сулейманов, Д. Ш., Гатиатуллин, А. Р. Наполнение семантических слотов реляционно-ситуационного фрейма на примере татарских синтаксем // Открытые семантические технологии проектирования интеллектуальных систем. 2014. – № 4. – С. 173-178.

19. Suleimanov, D. Sh., Yakubova, D. D. Lexical and Grammatical Potential of Turkic Languages for the Development of New Information Processing Technologies. В сборнике: Материалы XV международной конференции по компьютерной и когнитивной лингвистике TEL-2018 в 2-х томах. Сер. "Интеллект. Язык. Компьютер". 2018. – С. 361-372.

20. Анохин, К. В. Три пути к сознающему мозгу. 2024. Режим доступа. http://brainseminar.ru.

***Suleymanov Dzhavdet, Fridman Alexander, Gatiatullin Ayrat. Semiotic model of agglutinative language as the core of the cognitive system for explanatory artificial intelligence.*** *The concept of constructing a cognitive system of explanatory AI based on natural languages of the agglutinative type is presented, since the immanent properties of such languages create a fair basis for context-dependent synthesis of communication and cognitive processes with using almost automatic morphological means that are not inherent in other language groups. The system implements a semiotic approach to maintaining a decentralized set of lexical and grammatical models, its pilot version is focused on the semantics and grammar of the Tatar language (TL), since the testing of the main ideas and results will be carried out within the framework of "Tugan Tel", the electronic corpus of the TL. The core of the software system is a semiotic generative-recognizing model of the Tatar language, which includes a group of components for semantic verbalization and recognition coordinated at the metasystem level.*

***Key words:*** *semiotic generative-recognizing model of the Tatar language, semantics of natural languages, verbalization and recognition of meaning, semiotic model of lexical and grammatical means of language, cognitive AI system, "hyperaddressing" apparatus for accelerated access to databases and knowledge bases, situational analysis of the "focus of attention" of the modeling system.*

УДК 519.876

# Системы обработки больших данных как сложные фрактальные системы

**С.А. Амелькин[*1], Л.Г. Гагарина[*2]**

[*1] к.т.н, с.н.с., НИУ Московский институт электронной техники,
amelkin@ist.education, OrcID: 0000-0002-4004-7159, SPIN-код: 3631-1548

[*2] д.т.н., профессор, НИУ Московский институт электронной техники,
gagar@bk.ru, OrcID: 0000-0001-7591-9175, SPIN-код: 8582-4426

***Амелькин С.А., Гагарина Л.Г. Системы обработки больших данных как сложные фрактальные системы.*** *Возможности высокопроизводительных вычислительных сетей при моделировании социально-экономического развития региона находятся на грани метаперехода к предиктивной аналитике на основе исследования генеральной выборке данных. Особенностью структуры как предметной области, данных, так и самой вычислительной сети является инвариантность к масштабу (фрактальность), что необходимо учитывать в ходе формирования прогнозов. Для описания модели социально-экономического развития, рассмотрены два подхода: метод независимого анализа больших данных и агрегированная линейная модель. Показаны предиктивные возможности моделей по предотвращению кризисных ситуаций.*

***Ключевые слова:*** *большие данные, предиктивная аналитика, фрактальные системы, метрическое пространство.*

## *Введение*

Развитие энергоэффективных вычислительных комплексов в обозримом будущем реализует метапереход к новому типу обработки данных в экономической статистике, а именно переход от анализа выборки к анализу генеральной совокупности. Внедрение иммерсионных вычислительных систем [1] позволяет не только формировать сеть высокопроизводительных вычислительных комплексов, но и обеспечивает оптимальные показатели вычислительного комплекса, как экономической системы как за счет снижения капитальных (снижение стоимости и размеров зданий вычислительных центров, возможности установки высокопроизводительных вычислительных комплексов непосредственно в офисах и цехах) и оперативных (снижение затрат на электроэнергию, повышение надежности работы вычислительных устройств) издержек, так и за счет возможностей вторичного использования полученного тепла. Простота и надежность иммерсионных систем настолько высока, что их можно устанавливать в областных и районных центрах, на заводах и в учреждениях, таким образом создавая информационную сеть, охватывающую большие пространства, нивелировав существующее сейчас информационное неравенство в регионах. Формирование иерархической сети высокопроизводительных вычислительных комплексов в интересах создания полного информационного поля, приводит к необходимости фрактальной организации сети. Таким образом, на уровне аппаратного обеспечения создается сложная фрактальная система, инвариантная к преобразованиям масштаба.

На уровне программного обеспечения обработка генеральной совокупности данных социально-экономического состояния также требует фрактальной организации больших массивов данных. Проблема описания метрики усложняется тем, что в проблемах социально-экономического развития рассматриваются не стационарные показатели, а функции времени. Следовательно, задача поиска метрики заключается в определении расстояния между кластерами функциональных зависимостей.

Таким образом, и аппаратное, и программное обеспечение высокопроизводительной вычислительной сети имеют фрактальный характер, также, как и предмет изучения: социально-экономическое развитие. Развитие методов анализа сложных фрактальных систем позволяет найти оптимальный алгоритм обработки больших данных, определить условия устойчивого развития региона.

Исследования в области прогноза социально-экономического развития региона – особый вариант статистического анализа, так как при разделении системы на подсистемы свойства подсистем аналогичны свойствам системы в целом. Такое свойство и обеспечивает инвариантность системы к изменению масштаба

[1] Работа выполнена при поддержке Российского научного фонда (грант №23-21-00173 «Предельные возможности и оптимальные конструктивные решения энергоэффективных систем охлаждения высокопроизводительных вычислительных комплексов»).

(фрактальность). При большом объеме выборки может возникнуть ситуация, когда состояние подсистем с течением времени существенно различается при казалось бы примерно одинаковом состоянии в начальный момент времени. Развитие средств диагностики социально-экономического развития, большое количество наблюдаемых характеристик приводит к тому, что мониторинг состояния региона может проводиться именно методами анализа больших данных. Предлагаем два варианта построения математической модели – модель больших данных и агрегированная линейная модель.

Анализ больших данных может проводиться либо путем учета всех данных, независимо друг от друга, без предварительной их обработки, либо путем агрегирования данных, использования индексных показателей здоровья и поиска соотношений, характеризующих развитие во времени этих индексных показателей. При этом удобно использовать безразмерные показатели, так как интерпретация размерных величин может быть непростой. Рассмотрим каждый из этих подходов к построению математической модели социально-экономического состояния в отдельности.

### *Метод независимого анализа больших данных*

Пусть каждая социально-экономическая подсистема на произвольном уровне детализации характеризуется набором показателей $X(t) = (X_1(t), X_2(t), \dots, X_m(t))$, где $m$ достаточно велико (может достигать нескольких тысяч). Каждый показатель представляет собой некоторое число, тип числовой характеристики определяется инвариантностью к преобразованиям.

Числовые данные – результаты измерения параметров, могут быть дискретными или непрерывными. Классификационные данные – числовые данные, инвариантные к любым преобразованиям, представляющие результат отношения подсистемы к одной из групп по признакам, не имеющим числового значения. К таким данным, например, относится вид ресурсов, которыми обмениваются подсистемы, который может быть задан последовательностью цифр (с учетом фрактальности классификации – рациональным числом от 0 до 1). Естественно, выбор конкретных числовых значений в этом случае произволен. К классификационным данным относится локализация трудовых ресурсов, при этом, вследствие миграции значение может меняться во времени. К классификационным данным относим также индикаторные величины, принимающие значение 1 при наличии некоторого показателя и 0 – при отсутствии этого показателя. Стратификационные данные, инвариантные к монотонным преобразованиям, представляют собой или экспертные оценки, или объективные показатели состояния социально-экономического состояния, которые могут быть получены только в виде семантически значимых дискретных значений. При этом, в отличие от классификаций, возможные значения оценок образуют линейный порядок: при возрастании числового значения оценки изменяется ее семантика. Размерные данные инвариантны к изменению единиц измерения (к линейным преобразованиям) – это основной массив данных, который для описания социально-экономического состояния не только сам зависит от времени, но и от времени может зависеть единица измерения, например, вследствие инфляционных процессов.

Анализ текущего состояния и реализация прогнозных алгоритмов при анализе больших данных может проводиться без предварительной оценки, обработки (нормализация, агрегирование, соотношение числовых данных с предварительно выбранными интервалами). Для проведения анализа построим фазовый портрет подсистемы, для чего вычислим производную (для непрерывных числовых) или относительную разность (для остальных данных). Эти значения показывают скорость изменения параметров $X(t)$. Обозначим вектор полученных скоростей как $X'(t)$, элементы этого вектора имеют размерность $[X_i(t)] / [t]$, где $[X_i(t)]$ – размерность координаты $X_i(t)$, $[t]$ размерность времени. Таким образом, для размерных данных изменение во времени соотносится с семантикой размерности. Для стратификаций изменение во времени необходимо соотносить с семантикой шкалы оценки, для классификаций – с семантикой классификационного графа.

Фазовое пространство размерностью $p = 2m$ представляет собой пространство, в котором для каждого параметра мы отмечаем как его значение, так и скорость изменения во времени. То есть каждая точка в этом пространстве имеет координаты $(X(t), X'(t))$.

Каждая подсистема характеризуется своей траекторией в фазовом пространстве, полученной по ряду наблюдений. Это позволяет выявить кластеры траекторий и найти магистрали (средние значения) траекторий. Чтобы сформировать прогноз, требуется для данной траектории пациента найти расстояния до магистралей и по принципу максимального правдоподобия определить, какому кластеру принадлежит данная траектория.

Обе задачи: кластеризации и определения кластера, которому принадлежит данная траектория, требуют определения метрики на $p$-мерном пространстве, определяющей понятие расстояния.

Под метрикой будем понимать функцию или формулу, определяющую расстояние между любыми траекториями и кластерами (классами траекторий) в метрическом пространстве $R^p$. Метрическое пространство есть множество траекторий (точек $p \times N$-мерного пространства, где $N$ – количество наблюдений) с функцией расстояния $d_S$. Рассмотрим любые три $p \times N$-мерные точки $x, y$ и $z$.

*Определение 1.* Функция, являющаяся метрикой, должна удовлетворять следующим условиям:

1) $d_S(x,y) = d_S(y,x)$ (свойство симметричности),
2) $d_S(x,y) > 0 \Leftrightarrow x \neq y$ (свойство неотрицательности),
3) $d_S(x,y) = 0 \Leftrightarrow x = y$ (свойство совместимости),
4) $d_S(x,y) \leq d_S(x,z) + d_S(z,y)$ (неравенство треугольника).

Эти свойства позволяют формировать метрическое пространство вне зависимости от порядка заполнения и уточнения базы данных. Равенство точек *x* и *y* не обязательно означает, что эти траектории принадлежат одной и той же подсистеме, но говорит о равновесии подсистем, если они в рамках фрактального графа взаимодействий [2] могут обмениваться ресурсами.

Примерами метрики в пространстве состояний фазового портрета являются метрики Евклида и Махаланобиса:

*Определение 2.* Евклидовым расстоянием (Euclidian Distance) между двумя точками $x = (x_1, \ldots, x_p)^T$ и $y = (y_1, \ldots, y_p)^T$ в *p*×*N*-мерном пространстве называется функция вида:

$$d_E(x,y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_p - y_p)^2} = \sqrt{(x-y)^T(x-y)}.$$

При этом $d_E(x,0) = \|x\|_2 = \sqrt{x_1^2 + \ldots + x_p^2} = \sqrt{x^T x}$ является евклидовой нормой $x$. Из этого необходимо следует, что все точки с одним и тем же расстоянием, имеющим норму $d_E(x,0) = \|x\|_2 = c$, удовлетворяют равенству $x_1^2 + \ldots + x_p^2 = c^2$, которое определяет уравнение сфероида.

В задачах кластеризации предпочитают изменение каждого из признаков определять отклонением от центра, причем признаки с высокой изменчивостью должны получить меньший вес, чем признаки с низкой изменчивостью. При этом, поскольку траектории представляют собой совокупность точек (*X*(*t*), *X'*(*t*)), полученных для различных значений *t*, то веса для соответствующих координат каждой точки должны быть равными. Это может быть достигнуто путем масштабирования. Нормализованное расстояние применяют также для признаков, измеренных в различных единицах или существенно различающихся по величине. С учетом того, что показатели $X_i(t)$ предварительно не обрабатываются, нормализация представляется необходимой. Следует учесть, что нормализация никаким образом не меняет свойства показателей, которые являются классификациями или стратификациями. Мы не формируем индексы, что было бы некорректно [3], а расстояния между классами характеризуют особенности различных классов подсистем.

Нормализованное расстояние Евклида между точками $x$ и $y$ вычисляется следующим образом:

$$d(x,y) = d_E(u,v) = \sqrt{\left(\frac{x_1 - y_1}{s_1}\right)^2 + \ldots + \left(\frac{x_{p\times N} - y_{p\times N}}{s_{p\times N}}\right)^2} = \sqrt{(x-y)^T D^{-1}(x-y)},$$

где: $D = \mathrm{diag}(s_{1.}^2, \ldots, s_{p\times N}^2)$, $u = \left(\frac{x_1}{s_1}, \ldots, \frac{x_{p\times N}}{s_{p\times N}}\right)$, $v = \left(\frac{y_1}{s_1}, \ldots, \frac{y_{p\times N}}{s_{p\times N}}\right)$, $s_{i.}$ – масштабирующие коэффициенты. Теперь

$$\|x\| = d(x,0) = d_E(u,0) = \|u\|_2 = \sqrt{\left(\frac{x_1}{s_1}\right)^2 + \ldots + \left(\frac{x_{p\times N}}{s_{p\times N}}\right)^2} = \sqrt{x^T D^{-1} x}$$

и все точки с одинаковым расстоянием с нормой $\|x\| = c$ удовлетворяют уравнению $\left(\frac{x_1}{s_1}\right)^2 + \ldots + \left(\frac{x_{p\times N}}{s_{p\times N}}\right)^2 = c^2$, которое является уравнением эллипсоида.

*Определение 3.* Статистическим расстоянием или расстоянием Махаланобиса (Mahalanobis Distance) между двумя точками $x = (x_1, \ldots, x_{p\times N})^T$ и $y = (y_1, \ldots, y_p)^T$ в *p*×*N*-мерном пространстве называют функцию вида:

$$d_M(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)}$$

и $d_M(x,0) = \|x\|_S = \sqrt{x^T S^{-1} x}$ является нормой $x$. Здесь $S$ – матрица ковариации. Расстояние Махаланобиса можно определить как меру несходства между двумя случайными векторами $x = (x_1, \ldots, x_{pN})^T$ и $y = (y_1, \ldots, y_{pN})^T$ из одного распределения вероятностей с матрицей ковариации $S$. Расстояние Махаланобиса в предположении нормального распределения элементарных объектов в кластере с независимой от исследуемых факторов матрицей ковариаций можно рассматривать, как функцию правдоподобия. Пусть $z_i = (z_{i1}, \ldots, z_{i,(pN)})^T$ и $z_j = (z_{j1}, \ldots, z_{j,(pN)})^T$ – два вектора-строки размерности *p*×*N*. Тогда матрица ковариации размерности (*p*×*N*)×(*p*×*N*) определяется как:

$$S = \frac{1}{pN-1} A^T A,$$

где:

$$A = \begin{pmatrix} (z_{11},\dots,z_{1,(pN)})^T - (z_{01},\dots,z_{0,(pN)})^T \\ (z_{21},\dots,z_{2,(pN)})^T - (z_{01},\dots,z_{0,(pN)})^T \\ \dots \\ (z_{pN,1},\dots,z_{pN,(pN)})^T - (z_{01},\dots,z_{0,(pN)})^T \end{pmatrix}.$$

Здесь $(z_{01},\dots,z_{0,pN})^T-$ точка, относительно которой измеряется расстояние. В дальнейшем под центром будем понимать точку, определяемую средними значениями параметров, т.е. $(\bar{z}_1,\dots,\bar{z}_{pN})^T$. Элемент матрицы ковариации $S$ вычисляют следующим образом: $\sigma_{ij} = \frac{1}{pN-1}\sum_{k=1}^{pN}(z_{ki}-\bar{z}_i)(z_{kj}-\bar{z}_j)$; $i,j = 1,\dots,pN$.

Все точки с одним и тем же расстоянием имеющим норму $\|x\|_S = c$ удовлетворяют равенству $x^T S^{-1} x = c^2$. Расстояние от точки $x$ до центра кластера $\bar{x}$ определяется следующим образом

$$d_M(x,\bar{x}) = \sqrt{(x-\bar{x})^T S^{-1}(x-\bar{x})}.$$

Поскольку метрика Махаланобиса не может быть применена в случае обращения в нуль хотя бы одного элемента главной диагонали матрицы $S$, то применяют ее модификации.

*Определение 4.* Расстоянием Евклида – Махаланобиса (Euclidian-Mahalanobis Distance) [4] между двумя точками $x = (x_1,\dots,x_{pN})^T$ и $y = (y_1,\dots,y_{pN})^T$ в пространстве $R^{p\times N}$ называется функция вида:

$$d_{E-M}(x,y) = \sqrt{(x-y)^T(C+E)^{-1}(x-y)},$$

где $E$ - единичная матрица. Эта метрика устраняет недостаток метрики Махаланобиса. Расстояние Евклида – Махаланобиса может рассматриваться, как расстояние Махаланобиса в системе с ненулевой точностью измерений. В этом случае каждая отдельная траектория представляет собой подсистему, характеризующуюся диагональной матрицей ковариации с единичными дисперсиями.

*Определение 5.* Полиномиальным расстоянием Махаланобиса (Polynomial Mahalanobis Distance) между двумя точками $x = (x_1,\dots,x_{pN})^T$ и $y = (y_1,\dots,y_{pN})^T$ в пространстве $R^{p\times N}$ называется функция вида:

$$d_{M_{\delta^2}}(x,y) = \sqrt{(N-1)(x-y)^T U^T W_{\delta^2}^{-1} U(x-y)},$$

где $A^T A = U^T W_{\delta^2}^{-1} U$ - есть результат сингулярного разложения симметрической матрицы $A^T A$, причем: $W_{\delta^2} = \mathrm{diag}(\omega_1+\delta^2,\dots,\omega_p+\delta^2)$, где $\delta^2$ — малое положительное число

Различаются два случая измерения расстояний. В первом случае требуется решить задачу определения расстояний между траекториями одного и того же кластера (внутрикластерное расстояние). Второй случай охватывает чрезвычайно важные задачи определения расстояний между случайной траекторией и кластером. Эти два вида взаимосвязаны из-за фрактальности рассматриваемой системы. Действительно, каждый кластер представляет собой множество взаимодействующих подсистем, таким образом, расстояние между подсистемами аналогично расстоянию между траекториями в кластере.

В первом случае расстояния между траекториями определяют матрицу ковариаций кластера. Ожидается, что для числовых величин распределение является нормальным, тогда как для классификаций и стратификаций – близким к квазиравномерному. Порядок нумерования классов можно выбрать таким, чтобы минимизировать максимальное внутрикластерное расстояние. В этом случае мы обеспечим минимальные вероятности ошибок, связанные с некорректностью описания переменных.

Во втором случае мы решаем задачу принадлежности траектории подсистемы к тому или иному кластеру.

Кризисы социально-экономического развития в этой модели рассматриваются, как критические области в пространстве состояний. Для определения критических областей должна использоваться обучающая статистика. Алгоритм расчета может быть реализован как для однопроцессорных, так и для многопроцессорных устройств, легко поддается распараллеливанию или фрактализации [1]. Однако, требуется отметить, что построение сложных моделей взаимодействия с большим количеством параметров разного типа требует значительных вычислительных затрат. Возможно использовать системы имитационного моделирования для осуществления процесса составления прогноза.

### *Агрегированная линейная модель*

Процессы социально-экономического развития могут быть описаны в терминах теории управления. Регион рассматривается в этом случае, как линейная система управления, параметры которой испытывают влияние внешних факторов, инициирующих процессы обмена ресурсами. Кризис в этом случае рассматривается как критическая область, а управление системой должно обеспечить нахождение системы в пределах допустимых значений (принцип устойчивого развития [5]).

Линейная модель может быть записана в виде системы дифференциальных уравнений

$$\frac{dX}{dt} = f(X,u,t),$$

где $X$ – множество числовых непрерывных параметров (возможно провести агрегирование параметров в виде

непрерывного индекса), $u$ – возможные управления, связанные с государственным управлением социально-экономическим развитием, также должно быть представлено в виде вектора числовых непрерывных переменных, $t$ – время.

Использование линейных моделей позволяет применить байесовский подход, основанный на сравнении априорных и апостериорных вероятностей выхода в критическую область. Траектории движения системы и состояние системы в данный момент времени позволяют вычислить вероятность истинного и ложного кризиса по формуле:

$$P(R_i|H) = \frac{P(R_i)P(H|R_i)}{P(H)}.$$

Здесь $P(R_i)$ – априорные вероятности кризиса (при $i = 0$ истинного, при $i = 1$ – ложного), $P(H)$ – вероятность выбора подсистемой критического поля траекторий движения, $P(R_i|H)$ – апостериорные вероятности, которые требуется вычислить, $P(H|R_i)$ – вероятности выбора этих траекторий при известном результате прогноза.

Без составления линейной модели необходимо получения выборки большого объема для каждого из возможных состояний подсистемы, что затруднительно. Линейная модель позволяет детерминировать часть результатов, что, в свою очередь, обеспечивает расчетные возможности определения вероятностей $P(H)$ и $P(H|R_i)$. Вероятностный характер траектории развития подсистемы может быть получен за счет снятия ограничений на детерминированность функции $u$.

Прогнозы в байесовской модели должны быть определены по времени прогнозного горизонта. Выделим горизонты краткосрочного и долгосрочного (стратегического) планирования При краткосрочном прогнозе требуется, чтобы с заданной вероятностью (уровнем значимости) траектория движения агрегированных показателей не пересекала критическую область (область возникновения истинного или ложного кризиса). При стратегическом прогнозе требуется одновременное выполнение условий:

– На краткосрочном промежутке пересечение траектории и критической областью отсутствует;
– На долгосрочном промежутке траектория движения агрегированных показателей не пересекает критическую область.

Алгоритм расчета может быть реализован как для однопроцессорных, так и для многопроцессорных устройств, легко поддается распараллеливанию и фрактализации.

Критические области для истинного и ложного кризисов устанавливаются независимо по результатам социально-экономической модели (обучающая выборка). Таким образом, использование линейной математической модели региона позволит точнее прогнозировать возникновение кризисов. Ожидаемая точность прогноза зависит от выбора функции $f(X, u, t)$ и воспроизводимости результатов.

Необходимо отметить, что порядок агрегирования, методы нормализации, приведение параметров к безразмерному виду может существенно влиять на результаты прогноза, поэтому процесс верификации необходим для подтверждения работоспособности алгоритма. Критерием верификации является устойчивость системы: при использовании результатов исследований для коррекции априорных вероятностей априорные вероятности сходятся к значениям апостериорных вероятностей.

Байесовский подход с выделением критической области для каждого возможного исхода (отметим, что возможные исходы представляют собой классификацию, критические области могут также представлять собой фрактал) позволяет существенно снизить требования к вычислительной мощности ЭВМ, используемой для расчета, и к объемам требуемой памяти. С другой стороны, часть информации в процессе агрегирования теряется, что снижает уровень значимости полученных результатов.

### *Выводы*

Особенностью прогнозных моделей является необходимость верификации. Верификация осуществляется с помощью обучающей статистики, то есть набора данных, для которых известен результат прогноза. Обучающая выборка позволяет оценить априорные вероятности в байесовских моделях. При отсутствии обучающей выборки целесообразно в качестве априорных вероятностей выбрать равномерное распределение, однако такой вариант приводит к низкой точности прогноза на первых этапах работы алгоритма. Далее результаты реальной диагностики могут использоваться в качестве обучающей выборки. Если алгоритм работоспособен, то есть если наблюдается повторяемость результата, байесовские модели приходят к устойчивому состоянию, когда априорные вероятности равны апостериорным. Сходимость априорных и апостериорных вероятностей может рассматриваться, как критерий для верификации результатов статистического анализа данных.

Обучающая статистика, полученная до внедрения алгоритма расчета, также может быть использована. Такая обучающая статистика для расчета априорных вероятностей не учитывает многих показателей, поэтому может рассматриваться только как первое приближение. Контроль за сходимостью априорных и апостериорных вероятностей позволяет оценить качество агрегирования переменных и в процессе работы обеспечить правильную настройку алгоритма: изменение порядка агрегирования переменных, выявление факторов,

существенно влияющих на качество прогноза, уточнение констант линейной модели региона.

В кластерном подходе обучающая статистика используется для определения критических областей, в которых вероятность истинного и ложного кризисов превышает допустимый уровень. В ходе работы алгоритма прогнозирования происходит непрерывное уточнение критических областей. Сбор такой статистики позволяет увеличить уровень значимости при принятии гипотез о возможных кризисах в ходе социально-экономического развития.

Отсутствие обучающей статистики называется холодным стартом алгоритма. В связи с большим количеством допущений уровень значимости работы алгоритма при холодном старте следует уменьшить, а систему поддержки решений до выхода на приемлемый уровень значимости использовать только как рекомендательную систему.

Верификация модели на долгосрочном периоде требует длительного использования прогнозной модели. При изменении модели прогноза могут потребоваться дополнительные данные, поэтому целесообразно формировать сложную фрактальную сеть вычислительных комплексов, обеспечивающих управление по данным генеральной совокупности данных, при этом параллельно использовать как кластерную, так и агрегированную модели.

### *Литература*


1. Амелькин, С.А. Разработка системы автоматического управления погружным жидкостным охлаждением высокопроизводительных вычислительных комплексов / С.А.Амелькин, С.В.Карпеш, А.Д.Клементьев, А.А.Петров // Программные системы: теория и приложения. – 2016. – №1(28). – С. 209-219.

2. Амелькин, С.А. Фрактальная модель макросистем / С.А.Амелькин // Программные системы: теория и приложения. – 2024. – №1 (60) . – С. 41-62.

3. Овчинникова, О.П. Проблемы оценки социально-экономического развития муниципальных систем и результативности управленческих воздействий / О.П.Овчинникова, О.А.Судоргин, В.В.Лукашов // Финансы и кредит. – 2013. – 43 (571). – С. 2-7.

4. Амелькин, С.А. Обобщенное расстояние Евклида–Махаланобиса и его свойства / С.А.Амелькин, А.В.Захаров, В.М.Хачумов // Информационные технологии и вычислительные системы. – 2006. – № 4. – С. 40-44.

5. Кузьменкова, В.Д. Устойчивое развитие регионов России / В.Д.Кузьменкова // Вестник Воронежского государственного университета инженерных технологий. – 2016. – 2 (68). – С. 257-261.



***Амелькин С.А., Гагарина Л.Г. Системы обработки больших данных как сложные фрактальные системы.*** *Возможности высокопроизводительных вычислительных сетей при моделировании социально-экономического развития региона находятся на грани метаперехода к предиктивной аналитике на основе исследования генеральной выборке данных. Особенностью структуры как предметной области, данных, так и самой вычислительной сети является инвариантность к масштабу (фрактальность), что необходимо учитывать в ходе формирования прогнозов. Для описания модели социально-экономического развития, рассмотрены два подхода: метод независимого анализа больших данных и агрегированная линейная модель. Показаны предиктивные возможности моделей по предотвращению кризисных ситуаций.*



***Ключевые слова:*** *большие данные, предиктивная аналитика, фрактальные системы, метрическое пространство.*



***Amelkin S.A., Gagarina L.G. Big data processing systems as complex fractal systems.*** *The capabilities of high-performance computing networks in modeling the socio-economic growth of a region are on the verge of a meta-transition to predictive analytics based on a study of a general totality. A feature of the structure of both the subject area, data, and the computer network itself is scale invariance (fractality), which must be taken into account when forming forecasts. To describe the model of socio-economic growth, two approaches are considered: the method of independent analysis of big data and the aggregated linear model. The predictive capabilities of models for preventing crisis situations are shown.*



***Keywords:*** *big data, predictive analytics, fractal systems, metric space.*

УДК 004.9:550.34

# Технологии и программное обеспечение извлечения и анализа данных из открытых источников для сейсмического мониторинга территории

**С.О. Федоров[*1], В.А. Великий[*2], А.Г. Пимонов[*3]**

[*1] аспирант, Кузбасский государственный технический университет им. Т.Ф. Горбачева, sergej-fyodorov-1999@mail.ru,

[*2] аспирант, Кузбасский государственный технический университет им. Т.Ф. Горбачева, vova.velikiy.00@mail.ru,

[*3] д.т.н., профессор, Кузбасский государственный технический университет им. Т.Ф. Горбачева, pag_vt@kuzstu.ru,

***Федоров С.О., Великий В.А., Пимонов А.Г. Технологии и программное обеспечение извлечения и анализа данных из открытых источников для сейсмического мониторинга территории.*** *В статье приведены результаты анализа возможностей использования открытых сейсмических данных для проведения исследований в сфере сейсмического мониторинга. Спроектирована и разработана программная платформа, реализующая модули сбора, обработки, и визуализации открытых сейсмических данных для решения ряда задач мониторинга. Перспективой развития продукта является расширение числа источников данных и применяемых технологий с целью улучшения точности результатов анализа и дальнейшего внедрения модулей в реальные системы мониторинга.*

***Ключевые слова***: *сейсмический мониторинг, открытые данные, IRIS DMC, программные комплексы, Python.*

## *Введение*

В настоящее время одним из ключевых феноменов, связанных с развитием информационных технологий, являются т.н. «большие данные» или Big Data. Важнейшая характеристика данного феномена – возможность автоматического сбора, обработки и анализа всевозможных данных огромных размеров [1]. Автоматизированные системы управления, основанные на такой концепции, уже проявили себя во многих областей деятельности, включая образование, медицину, банковскую сферу, промышленность и т. д. Процесс внедрения подобных систем, с одной стороны, требует установки развитой сети устройств для сбора данных (датчиков, видеокамер, измерительных приборов и т. п.) и мощных вычислительных ресурсов (дата-центры, центры обработки данных). С другой стороны, для получения полезной информации необходимо применять различные методы анализа «сырых» данных, в связи с чем для каждой области проводятся исследования по созданию эффективных программных комплексов для сбора, обработки и визуализации данных.

Сейсмический мониторинг традиционно является областью деятельности, основанной на данных. Автоматизация сейсмического мониторинга, тем более процессов в реальном времени, является не только важнейшей, но и одновременно сложнейшей задачей сейсмологической практики. Его важность определяется такими потребностями [2], как:

- оперативная корректировка карт тектонической активности региона, карт балльности и сотрясаемости;
- необходимость принятия экстренных мер и исполнения оперативных мероприятий в зависимости от текущей сейсмической обстановки на основе автоматических уведомлений, обеспечиваемых системой мониторинга;
- автоматическое формирование баз сейсмологических данных, включая их наполнение непрерывными волновыми формами наблюдаемых процессов, бюллетенями сейсмических событий и прочей сопроводительной информацией;
- обмен информацией с другими сейсмологическими центрами и сетями сбора данных;
- автоматическое использование непрерывных данных реального времени от других сейсмических сетей с целью улучшения локации сейсмических событий и оценки их параметров.

На текущий момент эффективность сейсмического мониторинга во многом зависит от плотности покрытия измерительными сейсмическими станциями интересующей территории как в случае локального

уровня конкретных объектов (населённых пунктов, добывающих предприятий, объектов энергетики и т. д.), так и в случае регионального уровня в задаче сейсмического районирования. Тенденция увеличения сейсмостанций приводит к ситуации, когда необходимо в реальном времени обрабатывать множество потоков сигналов датчиков, при этом сами данные могут содержать неточности, вследствие чего задача автоматизации процессов мониторинга особенно актуальна и сложна. Из этого можно сделать вывод о важности проведения исследований научным сообществом с целью разработки новых технологий, методов и средств анализа сейсмических и прочих данных, связанных с процессом мониторинга.

В данной работе рассматривается вопрос о возможности проведения таких исследований независимыми учеными на основе открытых данных, поставляемых различными исследовательскими организациями, в области сейсмологии. Целью работы являлось создание программного продукта для решения нескольких задач сейсмического мониторинга с использованием открытых данных. Для достижения цели поставлены и решены три задачи:

1) определить перечень решаемых в рамках исследовательской работы задач сейсмического мониторинга, перечень открытых источников информации и формат входных данных для их интеллектуального анализа;
2) сформировать архитектуру разрабатываемого программного комплекса на основе выбранных задач сейсмического мониторинга, определить набор средств и технологий разработки;
3) выполнить разработку программного комплекса, провести тестирование отдельных модулей автоматической обработки данных и провести анализ эффективности использования открытых сейсмических данных и технологий их обработки.

### *Задачи сейсмического мониторинга и методы обработки данных*

Для демонстрации возможностей использования открытых баз сейсмологической информации поставлены четыре задачи сейсмического мониторинга:

1) анализ и классификация сейсмических сигналов;
2) определение координат и глубины гипоцентра сейсмического события;
3) расчёт сейсмичности;
4) расчёт сейсмического воздействия.

Под первой задачей понимается получение информации о вступлениях сейсмических волн среди записей сейсмодатчиками колебаний земной поверхности. Сложность задачи определяется, во-первых, регистрацией датчиками шумов различного характера наряду с сейсмическими сигналами, и, во-вторых, большими объёмами непрерывных записей колебаний [3]. В настоящее время активно проводятся исследования по внедрению моделей машинного обучения для решения задачи разделения сейсмических волн от волн иного происхождения. Например, в работе [4] описывается набор моделей, как линейных (линейный дискриминант Фишера, логистическая регрессия), так и нелинейных (искусственные нейронные сети, метод опорных векторов), для идентификации микросейсмических сигналов. В рамках данного исследования использованы две модели глубокого обучения, Seismic-Performer и Spec-CNN, разработанные и обученные российскими учеными для классификации отрезков волновых форм измерений сейсмодатчиков с выделением вступлений продольной (P) и поперечной (S) волн, а также шума (N) [5]. Обе модели с обученными весами выложены в открытый доступ в качестве репозитория GitHub [6]. Для решения поставленной задачи в рамках данного исследования реализованы процессы выгрузки волновых форм из открытых источников, их предобработки и загрузки в выбранные модели нейронных сетей, а также интерпретации результатов.

Вторая задача следует после первой и является частью работ по определению параметров зарегистрированного сейсмического события. Определение местоположения очага землетрясения влияет на дальнейший расчёт магнитуды и области сотрясаемости. Вместе с этим наиболее точная локализация землетрясения возможна в случае, когда оно произошло «внутри» расстановки не менее трёх сейсмостанций [7]. В выбранном для решения задачи локализации методе сфер [8] в качестве исходных данных при известных координатах трёх сейсмодатчиков используются разности времен прихода продольной и поперечной сейсмической волны на каждый сейсмодатчик. Важно отметить, что скорости распространения продольной волны и отношение поперечной волны к продольной принимаются как константы для всей среды. Также в работе используется алгоритм перевода координат согласно ГОСТ Р 51794-2008, так как метод предполагает проведение вычислений в прямоугольной системе координат, а также выбор одной из сейсмостанций в качестве начала координат. Для наглядности влияния описанных факторов на конечный результат расчётов используются несколько итераций локализации с изменяемыми параметрами – скоростями сейсмических волн и точки с нулевыми координатами. Принимается к сведению, что реальный эпицентр землетрясения может располагаться вблизи итоговой области точек.

Третья задача (расчёт сейсмичности) предполагает определение параметров исторических землетрясений на интересующей территории, а также степени их воздействия на различные объекты. В качестве обобщенной

характеристики сейсмического воздействия на здания и сооружения принимается интенсивность землетрясения, измеряемая в баллах и зависящая от глубины очага и магнитуды землетрясения, служащей мерой его энергии. В России и в большинстве других стран интенсивность оценивается по 12-бальной шкале.

В данном исследовании интенсивность землетрясения определяется с помощью уравнения макросейсмического поля (УМП) [9], дающего эмпирическую корреляционную связь между наблюдаемой макросейсмической интенсивностью, магнитудой землетрясения, эпицентральным расстоянием и глубиной очага. Коэффициенты УМП определены для некоторых регионов России, однако данном исследовании используются средние значения, так как расчеты не проводятся в пределах конкретного региона со специфичными оценками коэффициентов.

Четвертая задача относится к оценке сейсмического воздействия на конкретные объекты с учетом их особенностей. Прогнозирование последствий землетрясений в рамках населённого пункта позволяет получить данные: о количестве зданий и сооружений, получивших определенные степени разрушения; о качественном описании разрушений зданий и сооружений и т. д. В рамках данной исследовательской работы используется методика [10], позволяющая при известных характеристиках материала традиционных построек (без антисейсмических мероприятий) определить сейсмостойкость каждого здания. Далее для группы однотипных зданий в зависимости от их сейсмостойкости и реальной интенсивности землетрясения может быть найдена осредненная степень разрушения, которая используется для приближенной оценки потерь населения, находящегося в этих зданиях. В данной работе для каждого из заданных населённых пунктов рассчитывается процентное соотношение степеней разрушения зданий на основе данных о материале конструкций и характеристиках исторических землетрясений. Важно отметить, что в работе для населённых пунктов допускается соответствие реальной интенсивности землетрясения и интенсивности, найденной с помощью УМП.

### *Источники данных и технологии их обработки*

Для решения набора описанных выше задач требуется выгрузка следующих данных из внешних источников:

– волновые формы регистрируемых сейсмодатчиками колебаний для решения первой задачи;

– времена вступлений P- и S- волн конкретного землетрясения на заданные сейсмостанции, а также метаданные сейсмостанций, включающие их координаты и высоту над уровнем моря, для решения второй задачи;

– параметры исторических землетрясений, включающие магнитуду, координаты эпицентра и глубину очага, для решения третьей задачи;

– параметры исторических землетрясений, координаты интересующих населённых пунктов и характеристики находящихся в них зданий, позволяющие определить сейсмостойкость, для решения четвертой задачи.

В качестве источников этих данных выбраны три открытых сервиса. Первый сервис – IRIS Data Management Center (далее IRIS DMC), управляемый объединением научно-исследовательских институтов по сейсмологии (сайт www.iris.edu). Сервис предоставляет данные крупнейших сейсмических сетей со всего мира посредством веб-сервисов, онлайн-средств и приложений. Ключевыми форматами описания данных являются: miniSEED для волновых форм; FDSN StationXML для идентификаторов и метаданных сейсмических сетей, станций, датчиков и каналов данных; QuakeML для описания сейсмических событий.

Вторым сервисом является сайт Дом.МинЖКХ.РУ (далее МинЖКХ), предоставляющий данные о зданиях и сооружениях для дальнейшей оценки потенциального ущерба от сейсмического события (сайт dom.mingkh.ru). В рамках данного исследования для определения класса сейсмостойкости здания используются такие поля, как «Серия, тип постройки», «Тип перекрытий» и «Материал несущих стен». Важно отметить, что на текущий момент проект не предоставляет интерфейса для автоматической выгрузки данных.

Третий сервис – MapBox – предоставляет платформу для создания пользовательских онлайн-карт на основе открытых карт OpenStreetMap и других источников (сайт www.mapbox.com). Базовый функционал предоставляется бесплатно после регистрации. Картографическая информация в данном исследовании используется для нахождения населённых пунктов, а также для формирования запросов на получение объектов землетрясений и сейсмостанций.

Таким образом, описанные публичные сервисы предоставляют почти все необходимые данные. Недостающими являются времена вступлений P- и S- волн для решения второй задачи, в рамках данного исследования эти данные получаются из результатов решения первой задачи.

Для реализации алгоритмов сбора и обработки входных данных составлен стек технологий программного комплекса. В основе разработки лежит трёхуровневая архитектура, использование принципов объектно-ориентированного программирования и паттерна проектирования Model-View-Controller, что обеспечивает изолированность логики программного комплекса, то есть алгоритмов сбора и обработки сейсмических данных,

от выбранных способов хранения и отображения данных. Конечный продукт представляет собой веб-приложение с серверной частью на языке программирования Python, клиентской частью на языках HTML, CSS и JavaScript, а также системой управления базой данных (СУБД) PostgreSQL. Выбор этих технологий обусловлен их открытостью, относительной лёгкостью реализации проектов и, что наиболее важно, наличием библиотек для решения поставленных задач исследования. В частности, используются следующие библиотеки языка Python:

– микрофреймворк Flask позволяет автоматизировать создание каркаса веб-приложения, обеспечивающего маршрутизацию запросов клиентов и отправку результатов;

– библиотека Math содержит инструменты для проведения математических операций и используется в работе для программной реализации алгоритмов и формул;

– библиотека NumPy является популярной библиотекой для обработки числовых массивов, которыми в данной работе являются волновые формы регистрируемых сейсмодатчиками сигналов;

– библиотека pandas предназначена для работы с табличными данными и используется для обработки параметров зданий и сооружений, получаемых в виде файла от пользователя системы;

– библиотека TensorFlow является популярным решением для работы в сфере машинного обучения, в данной работе используется для взаимодействия с моделями нейронных сетей Seismo-Performer и Spec-CNN;

– библиотека ObsPy разработана специально для взаимодействия с API IRIS DMC, формирования запросов на выгрузку данных и получения результатов в виде объектов классов языка Python, отражающих описанные выше форматы данных;

– библиотека GeoPy позволяет выполнять операции с геоданными, например, расчёт расстояния между двумя точками;

– библиотека Matplotlib используется для построения графиков волновых форм с указанием времён вступления сейсмических волн в соответствии с задачей анализа и классификации сейсмических сигналов.

Ключевой библиотекой на стороне клиента является MapBox.JS для формирования интерактивных карт и работы с различными элементами на карте – точечными и полигональными объектами, метками и др.

### *Результаты исследования*

Разработанный авторами программный комплекс для сбора, обработки и анализа данных сейсмического мониторинга представляет собой набор модулей, взаимодействующих друг с другом в рамках веб-сервера (рис. 1).

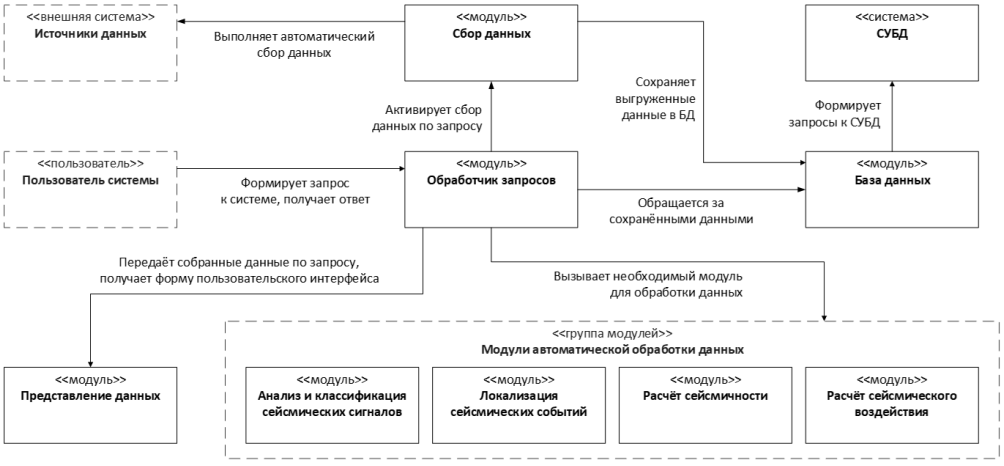



Рисунок 1 – Структура программного комплекса

Модуль сбора данных предназначен для обращения к внешним источникам данных и выгрузки необходимой информации для дальнейшей обработки. Как было описано раннее, такими источниками являются публичный веб-сервис IRIS SAGE DMC сервис ДОМ.МинЖКХ.ру, при этом в первом случае выгрузка данных производится через HTTP-запросы (обёрнутые в функции библиотеки obspy), а во втором случае реализуются функции загрузки и обработки табличных файлов с информацией о материалах зданий, сформированных

пользователем вручную. Данный модуль также отвечает за обработку информации, введённой пользователем через интерфейс.

Модуль базы данных является прослойкой между всеми модулями системы и используемой системой управления базой данных (СУБД). Модуль определяет интерфейс для получения, сохранения, изменения и удаления данных, благодаря чему возможно скрыть нежелательные операции с данными в БД и отклонить ошибочные. СУБД также является неотъемлемым компонентом системы ввиду необходимости сохранения в местном хранилище выгруженных извне данных различных сущностей: метаданные сейсмических станций; данные о землетрясениях; данные о населённых пунктах и зданиях.

Группа модулей автоматической обработки данных реализует набор описанных выше четырех задач сейсмического мониторинга. Модуль анализа и классификации сейсмических сигналов предназначен для предобработки выгруженных из IRIS волновых форм, загрузки их в модель нейронной сети и формирования результатов. Модуль локализации сейсмических событий предназначен для расчёта координат очага землетрясения с использованием метода сфер. Модуль расчёта сейсмичности предназначен для расчёта изосейст в виде набора окружностей, а также для расчёта интенсивности выбранного землетрясения в любой точке вокруг него. Модуль расчёта сейсмического воздействия предназначен для расчёта процентного соотношения степеней разрушения зданий от сейсмического события заданной интенсивности в имеющихся населённых пунктах.

Модуль обработчика запросов фактически представляет собой набор функций для обработки HTTP-запросов. Модуль предназначен для вызова команд остальных модулей с целью обработки и получения результирующих данных, которые требует пользователь системы.

Модуль представления данных составляет набор шаблонов и команд для формирования элементов пользовательского интерфейса в зависимости от входных данных. Функции для составления конкретной формы вызываются модулем обработчика запросов.

Таким образом, описанная структура программного комплекса с одной стороны обеспечивает возможность независимой работы над отдельными модулями, а с другой связывает модули в единую систему. Во многом это достигается за счёт размещения управляющих элементов пользовательского интерфейса на одной веб-странице, включающей интерактивную карту (рис. 2) для управления пространственными объектами (населёнными пунктами, сейсмостанциями, сейсмическими событиями), панели управления данными (просмотра, создания, изменения и удаления записей, а также формирование запроса на выгрузку из внешних источников) и панелей обработки данных для решения четырёх задач сейсмического мониторинга.


Рисунок 2 – Внешний вид пользовательского интерфейса с интерактивной картой и кнопками открытия панелей управления

Панель классификации сейсмических сигналов и локализации землетрясения (рис. 3) позволяет пользователю сформировать запрос на выгрузку и классификацию волновых форм с помощью модели нейронной сети. Для волновых форм строятся графики с отметками вступлений сейсмических волн. Для локализации составляется набор точек с координатами эпицентра в зависимости от скоростей сейсмических волн согласно описанным раннее допущениям. Предполагается что вблизи этих точек лежит реальный эпицентр.

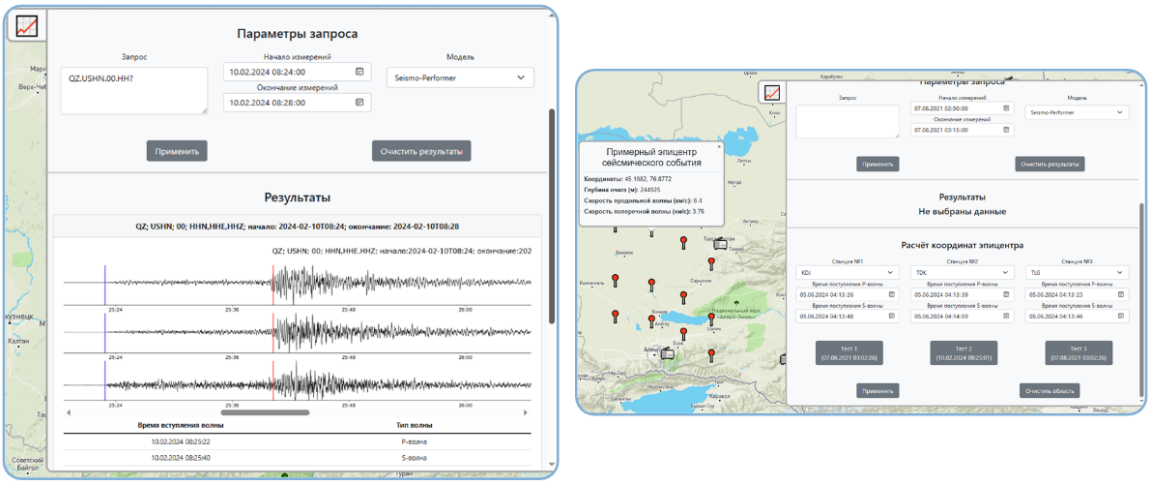



Рисунок 3 – Результаты классификации и локализации

На панели расчёта сейсмичности и сейсмического воздействия (рис. 4) пользователь может отправить запрос на построение областей интенсивности выбранного землетрясения. Кроме того, для каждого населённого пункта рассчитываются интенсивность и процентное соотношение степеней повреждения зданий.

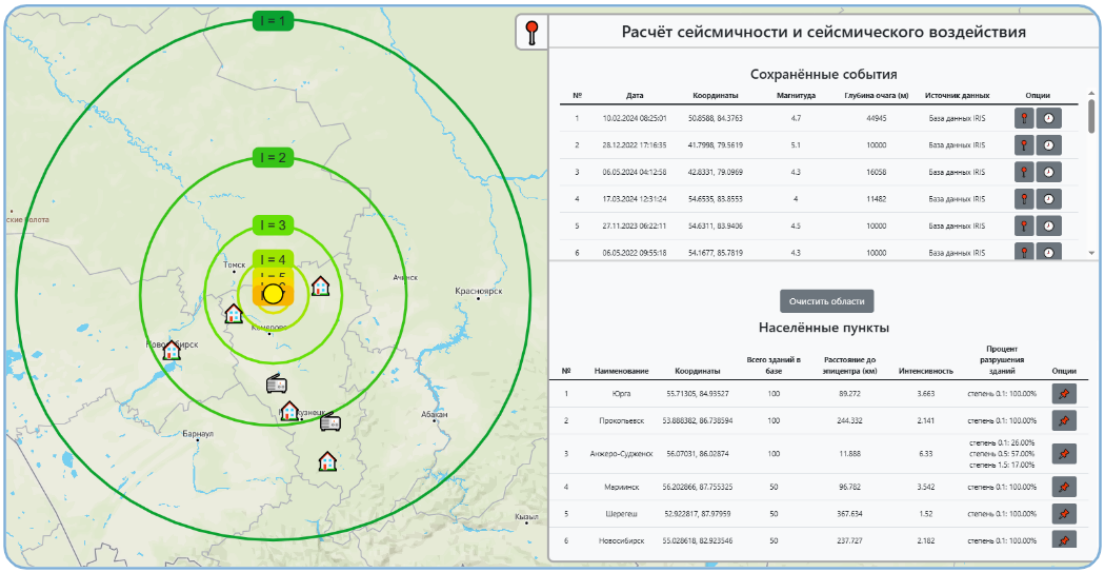



Рисунок 4 – Результаты расчёта сейсмичности и сейсмического воздействия

Для тестирования корректности обработки сейсмических данных в рамках модулей автоматической обработки произведено сравнение результатов работы модулей с существующими сервисами и ориентировочными расчётными показателями. Для модуля анализа и классификации сейсмических сигналов выполнено визуальное сравнение отмеченных времён вступлений и усилению колебаний волновых форм для крупных землетрясений (рис. 3). Дополнительно примерные времена вступления волн на сейсмостанции рассчитаны на основе координат эпицентра и времени возникновения нескольких исторических сейсмических событий. В конечном итоге сделан вывод о том, что данные, выгружаемые из IRIS, обрабатываются и подаются на вход модели в той форме, в которой предполагалось разработчиком, а результаты работы моделей правильно интерпретируются в виде графиков волновых форм.

Для модуля локализации землетрясения выполнено сравнение результатов локализации с координатами эпицентра из IRIS для четырех сейсмических событий. По результатам анализа отмечается, что реальный эпицентр располагается внутри образуемой маркерами «фигуры» (пример на рис. 3) в случае невысокого расстояния от сейсмостанций, в остальных случаях выходя за его пределы на небольшое расстояние.

Для модуля расчёта сейсмичности выполнено сравнение результирующих изосейст с изосейстами, рассчитанными сервисом EQA!ert (сайт eqalert.ru). В трёх из четырёх случаев изосейсты совпадали, тогда как в одном случае предположительно использовались специфичные значения коэффициентов УМП.

Для модуля расчёта сейсмического воздействия выполнено сравнение результатов с расчётными путём моделирования нескольких сейсмических событий вокруг имеющихся в базе данных населённых пунктов, для каждого из которых вручную выгружен набор зданий из сайта МинЖКХ. Результаты расчётов степеней разрушения соответствовали ожидаемым.

### *Заключение*

В результате проведенного исследования разработан программный комплекс сейсмического мониторинга, включающий модули сбора, обработки и представления сейсмических данных на основе открытых источников. Конечный продукт ввиду удобства модификации отдельных компонентов может быть использован для тестирования и отладки алгоритмов извлечения и анализа сейсмических данных с целью их внедрения в существующие системы сейсмического мониторинга. К вопросу об особенностях использования открытых данных для проектирования программных комплексов сейсмического мониторинга сделан вывод о том, что на настоящий момент многообразие данных, предоставляемых публичными сервисами, позволяет проводить исследования в большом числе задач мониторинга: от регистрации сейсмических событий до расчёта сейсмического воздействия. Тем не менее, для реализации высокоточных методов в ряде задач дополнительно требуется использование информации о специфичных характеристиках интересующей территории, что может отсутствовать в открытом доступе.

Дальнейшие исследования открытых сейсмологических данных могут быть направлены в сторону поиска новых источников данных и модификации алгоритмов их обработки и анализа, а также реализации с их помощью задач сейсмического мониторинга, не рассмотренных в данной работе. Разработанный программный комплекс также может быть усовершенствован в следующих направлениях: поддержка большего числа источников данных, в том числе форматов их описания; реализация хранения истории расчётов модулей для более тщательного анализа работы алгоритмов; возможность редактирования глобальных параметров из пользовательского интерфейса, в том числе параметров моделей нейронных сетей (порог, частота, веса и др.), скоростей сейсмических волн, коэффициентов уравнения макросейсмического поля и др.

### *Литература*


1. Пестунов, А.И. Big data как феномен: причины и следствия появления больших данных / А.И. Пестунов, А.С. Гинтофт, О.В. Криветченко // ЭКО. – 2023. – № 9(591). – С. 137-154.

2. Зараменских, Е.П. Интернет вещей. Исследования и область применения : монография / Е.П. Зараменских, И.Е. Артемьев. // Москва : ИНФРА-М, 2023. – 188 с.

3. Машинное обучение в сейсмологии [Электронный ресурс] // Хабр. – 2021. – Режим доступа: https://habr.com/ru/articles/587690/ (дата обращения: 15.10.2024).

4. Machine Learning Approaches for Long-term Rock Burst Prediction [Электронный ресурс] / Y. Pu. // 2019. – Режим доступа: https://research.ebsco.com/linkprocessor/plink?id=360b5c50-2068-3c5e-96ed-a837c244f119 (дата обращения: 16.09.2024).

5. Stepnov A. The Seismo-Performer: A Novel Machine Learning Approach for General and Efficient Seismic Phase Recognition from Local Earthquakes in Real Time / A. Stepnov, V. Chernykh, A.Konovalov // Sensors. – 2021. – № 21. – [Электронный ресурс]. – Режим доступа: https://www.mdpi.com/1424-8220/21/18/6290 (дата обращения: 16.10.2024).

6. Seismo-Performer // GitHub : [сайт]. – URL: https://github.com/jamm1985/seismo-performer (дата обращения: 21.05.2023).

7. Кузнецов, В.В. Физика Земли. – 2011. – 842 с. – Режим доступа: https://www.geokniga.org/sites/geokniga/files/inbox/5226/9.pdf (дата обращения: 13.02.2024).

8. Асланов, Г.К. Об одном методе определения очага землетрясения с одновременным определением скоростей сейсмических волн / Г.К. Асланов, М.Г. Даниялов, Т.Г. Асланов, Х.Д. Магомедов // Труды Института геологии Дагестанского научного центра РАН. – 2010. – № 56. – С. 54-59.

9. Чудинова, О.Н. Прогнозирование последствий чрезвычайных ситуаций природного и техногенного характера / О.Н. Чудинова, Т.В. Чередова, А.А. Бутакова, Ю.С. Воронина. – Улан-Удэ: Изд-во ВСГУТУ, 2022. – 88 с.

10. Сборник методик по прогнозированию возможных аварий, катастроф, стихийных бедствий в РСЧС / Министерство российской федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий. – 1 изд. – Москва : 1994. – 34 с.

***Федоров С.О., Великий В.А., Пимонов А.Г. Технологии и программное обеспечение извлечения и анализа данных из открытых источников для сейсмического мониторинга территории.*** *В статье приведены результаты анализа возможностей использования открытых сейсмических данных для проведения исследований в сфере сейсмического мониторинга. Спроектирована и разработана программная платформа, реализующая модули сбора, обработки, и визуализации открытых сейсмических данных для решения ряда задач мониторинга. Перспективой развития продукта является расширение числа источников данных и применяемых технологий с целью улучшения точности результатов анализа и дальнейшего внедрения модулей в реальные системы мониторинга.*

***Ключевые слова:*** *сейсмический мониторинг, открытые данные, IRIS DMC, программные комплексы, Python.*

***Fedorov Sergey, Vladimir Velikiy, Pimonov Alexander. Technologies and software for open sources data extraction and analysis in territorial seismic monitoring task.*** *The article provides the analysis results of open seismic data usage possibilities in the field of seismic monitoring research. In order to solve several seismic monitoring tasks, a software platform implementing open seismic data collection, processing, and visualization has been developed. The prospects for this research include expanding data sources as well as the technologies used in order to improve analysis results accuracy and future modules implementation into real monitoring systems.*

***Key words:*** *seismic monitoring, open data, IRIS DMC, software complexes, Python.*

УДК 004.056, 004.043

# Разработка абстрактной модели нейросетевого детектора уязвимостей в программном коде

**В.В. Швыров[*1], Д.А. Капустин[*2], Р.Н. Сентяй[*3]**

[*1] к.ф.-м.н, доцент, Луганский государственный университет,
slshj@yandex.ru, OrcID: 0009-0005-0163-2952, SPIN-код: 1562-9454

[*2] д.т.н, доцент, Луганский государственный университет,
kap-kapchik@mail.ru, OrcID: 0009-0008-4254-319X, SPIN-код: 2350-9390

[*3] ст. преп., Луганский государственный университет,
sentyayroman@yandex.ru, SPIN-код: 7056-3049

***Швыров В.В., Капустин Д.А., Сентяй Р.Н. Разработка абстрактной модели нейросетевого детектора уязвимостей в программном коде.*** *В последние годы методы машинного обучения и большие языковые модели все более активно используются в самых различных областях программной инженерии. Одной из перспективных сфер применения больших языковых моделей является статический анализ программного кода, в котором нейросетевые модели используются для обнаружения и классификации различных видов уязвимостей. В данной работе представлена математическая модель абстрактного нейросетевого детектора уязвимостей в программном коде. Модель включает описание уровня представления программного кода, формальное теоретико-множественное определение нейросетевого детектора, а также ряд необходимых определений и обозначений. Кроме того, с использованием предложенной модели разработан прототип класса, реализующего обнаружение уязвимостей в программном коде с использованием нейросетевого подхода.*

***Ключевые слова:*** *уязвимость, программный код, статический анализ, анализ безопасности, математическая модель, большие языковые модели.*

## *Введение*

Одним из методов анализа безопасности программного кода является метод статического анализа, который подразумевает проверку кода без его фактического выполнения. Данный метод был предложен в работах Кузо [1] и активно развивался различными исследователями [2-5].

Стремительное развитие методов машинного обучения и появление архитектуры Transformer [6] привело в росту использования различных нейросетевых моделей в задачах генерации, анализа, форматирования, рефакторинга программного кода. В систематическом обзоре [7] представлены результаты анализа публикаций, связанных с использованием больших языковых моделей в различных областях программной инженерии.

Одним из перспективных, но и наиболее сложных направлений использования нейросетевых моделей, является анализ безопасности программного кода. Ряд исследований свидетельствуют о том, что современные большие языковые модели, такие как chatGPT зачастую не справляются с задачей обнаружений уязвимостей в программном коде [8, 9]. В то же время, нейросетевые модели, которые были обучены на специально сформированных наборах данных, содержащих примеры кода с уязвимостями, имеют значительно более высокие показатели точности [10, 11].

Стандартной практикой проверки эффективности нейросетевой модели, является проверка модели на известном наборе тестовых случаев, например SATE [12]. Однако, такой подход не всегда точно отражает фактическую точность модели и результаты проверки модели на реальном коде оказываются значительно ниже тестовых.

Основной проблемой является отсутствие строгой математической модели для статических анализаторов, которые используют нейросетевые модели для обнаружения уязвимостей в программном коде, а также недостаток формальных метрик для подобных анализаторов.

Необходимо отметить, что для оценки уязвимостей примером метрики может быть CVSS 3.0, которая отражает самые различные характеристики уязвимостей.

Целью работы является разработка математической модели абстрактного нейросетевого детектора и формирование понятийного аппарата для описания свойств подобных детекторов.

Для достижения поставленной цели необходимо решить ряд задач, в частности:

- описать формальную модель уровня представления программного кода, подходящую для ее использования в контексте статического анализа с использованием нейросетевых детекторов;

- разработать концептуальную модель абстрактного детектора, для определения ключевых сущностей и параметров;

- описать математическую модель абстрактного детектора с использованием теоретико-множественного подхода и концептуальной модели.

### *Модель представления программного кода*

Программный код может быть представлен на различных уровнях абстракции. Можно выделять уровень исходного кода, функциональный уровень, промежуточное представление. Распространенным способом представления программного кода является абстрактное синтаксическое дерево, граф потока управления, граф потока данных, а также графы, учитывающие семантическую структуру программы. Выбор формы представления программы может быть обусловлен архитектурой нейросетевой модели, которая используется для дальнейшей обработки программного кода.

В данной работе будем использовать представление программного кода на уровне строк кода исходной программы. Введем ряд необходимых определений и обозначений.

**Определение 1.** Пусть $P$ – программа на некотором языке программирования, которая представлена в виде последовательности строк исходного кода $p_1, p_2, ..., p_s$ обозначим через $C(P)$ - множество всех уникальных строк программы $P$, $|C(P)| \le s$. Обозначим через $C(P)_k = \{p \in C(P) \mid len(p) \le k\}$. Множество $C(P)_k$ назовем $k$-фильтрацией программы $P$. Назовем множество $\overline{C(P)_k} = \{p \in C(P) \mid len(p) > k\}$ $k$-дополнением программы P.

На данный момент все существующие нейросетевые модели имеют ограничения по длине входных данных. В связи с этим, если входная строка будет иметь длину, которая превышает входную длину нейросетевой модели, то строка обычно обрезается, что, в итоге, может привести в потере точности анализа кода. По сути, использование $k$-фильтрации программы, где $k$ – ограничение на входную длину строки для конкретной нейросетевой модели, формирует разбиение множества всех строк программы на два класса строк. Класс $C(P)_k$ - это строки, которые будут обработаны нейросетевой моделью без обрезки, $\overline{C(P)_k}$ - класс строк, которые превышают допустимую длину и будут обрезаны. Для таких строк имеет смысл использовать модели с длиной входных данных $t>k$.

**Определение 2.** Пусть $P$ – программа на некотором языке программирования, которая представлена в виде последовательности строк исходного кода $p_1, p_2, ..., p_s$, $p_k$ - произвольная строка, где $1 \le k \le s$. Назовем контекстом длины $m$, любую упорядоченную последовательность строк $p_g, ...., p_h$ программы P, такую, что $1 \le g \le k, k \le h \le s, h - g = m$, $m \le s-1$. Обозначим через $Con(P)$ - множество всех возможных контекстов всех строк программы P.

Согласно данному определению, контекст длины 0 для некоторой строки $p_k$ совпадает с этой строкой. Будем обозначать контекст длины $m$ строки $p_k$ через $Con^m_{p_k}$, положим, что $SCon^m_{p_k}$ - множество всех контекстов длины m для строки $p_k$ и определим множество всех возможных контекстов строки $p_k$ как

$$Con_{p_k} = SCon^0_{p_k} \cup SCon^1_{p_k} \cup ... \cup SCon^{s-1}_{p_k},$$

тогда,

$$Con(P) = Con_{p_1} \cup ... \cup Con_{p_s}$$

**Пример**. Пусть $P = p_1, p_2, p_3$, $C(P)_k = \{p_1, p_2, p_3\}$, т.е. все строки программы различные и имеют длину меньше, чем $k$. Тогда

$$SCon^0_{p_1} = \{(p_1)\}, SCon^1_{p_1} = \{(p_1, p_2)\}, SCon^2_{p_1} = \{(p_1, p_2, p_3)\},$$
$$SCon^0_{p_2} = \{(p_2)\}, SCon^1_{p_2} = \{(p_1, p_2), (p_2, p_3)\}, SCon^2_{p_1} = \{(p_1, p_2, p_3)\},$$
$$SCon^0_{p_3} = \{(p_3)\}, SCon^1_{p_3} = \{(p_2, p_3)\}, SCon^2_{p_3} = \{(p_1, p_2, p_3)\},$$
$$Con(P) = \{(p_1), (p_2), (p_3), (p_1, p_2), (p_2, p_3), (p_1, p_2, p_3)\}.$$

**Утверждение 1.** Пусть $P$ – программа на некотором языке программирования, которая представлена в виде последовательности строк исходного кода $p_1, p_2, ..., p_s$, тогда

$$|Con(P)| = \frac{s+1}{2} s.$$

**Доказательство**. Действительно, всего существует ровно s контекстов длины 0. Поскольку контексты длины 1 представляют собой упорядоченную пару строк программы, то существует всего *s-1* возможных контекстов длины 1. Далее, рассуждая аналогично подсчитаем количество всех контекстов вплоть до контекста, который содержит все строки программы, такой способ выбора строк будет всего один, т.е. имеем, *s-(s-1)* контекстов длины *s-1*. Используя формулу для вычисления суммы арифметической прогрессии получаем искомое значение.

**Замечание**. Рассматривая каждый контекст $X \in Con(P)$ как независимую программу *X*, используя определение 1, аналогично, можно определить k-фильтрацию и k-дополнение для *X*.

### *Абстрактная модель нейросетевого детектора*

Используя модель представления программного кода, введенную выше, дадим формальное определение нейросетевого детектора на уровне строк.

**Определение 3**. Пусть $L = \{l_1, l_2, ..., l_m\}$ – множество языков программирования, $V = \{v_1, v_2, ..., v_n\}$ – множество типов уязвимостей, P – программа на некотором языке $l \in L$, которая представлена в виде списка строк программного кода $s_1, s_2, ..., s_t$, $C(P)_k$ - k-фильтрация программы P. Нейросетевым детектором уязвимостей D на уровне строк назовем кортеж $(L, V_z, M, k)$, где M – нейросетевая модель такая, что для любого $v \in V_z \subseteq V, |V_z| = q$ и для любой строки $s \in C(P)_k$ она формирует выходной вектор $M(s) = (w_1, w_2, ..., w_q)$, где $0 \le w_j \le 1, j = 1, ..., q$, $w_j$ - вероятность наличия в строке s уязвимости $v_j$.

**Определение 4.** Нейросетевой детектор уязвимостей уровня строк $D = (L, V, M, k)$ назовем детектором для языка $l$, если $L = \{l\}$. Будем обозначать в таком случае его через $(l, V, M, k)$.

**Определение 5.** Пусть $V = \{v_1, v_2, ..., v_n\}$ – множество типов уязвимостей. Нейросетевой детектор уязвимостей $D = (L, V_z, M, k)$ назовем *V*-полным, если $V_z = V$.

**Определение 6.** Пусть $V = \{v_1, v_2, ..., v_n\}$ – множество типов уязвимостей. Нейросетевой детектор уязвимостей $\overline{D} = (L, V_z, M, k), |V_z| < |V|$ назовем *V*-дополнением, если для любого $v_i \in V$ для любой строки $s \in C(P)_k$ определен вектор $(a_1, a_2, ..., a_n), i = 1, ..., n,$

$$a_i = \begin{cases} 0, v_i \notin V_z, \\ w_j, v_i \in V_z, \end{cases}$$

где $(w_1, w_2, ..., w_q) = M(s)$, j – индекс соответствующей вероятности в M(s).

Замечание. Фактически *V*-дополнение является *V*-полным детектором, в котором для тех типов угроз, вероятности которых модель *M* не может прогнозировать, соответствующие значения заданы нулевыми.

Далее, введем обобщенное понятие нейросетевого детектора, для случая представления кода с использованием множества контекстов программы P.

**Определение 7.** Пусть $L = \{l_1, l_2, ..., l_m\}$ – множество языков программирования, $V = \{v_1, v_2, ..., v_n\}$ – множество типов уязвимостей, P – программа на некотором языке $l \in L$, которая представлена в виде списка строк программного кода $s_1, s_2, ..., s_t$. Нейросетевым детектором уязвимостей $D$ на уровне строк с контекстом назовем кортеж $(L, V_z, M^C, k)$, где $M^C$ – нейросетевая модель такая, что для любого $v \in V_z \subseteq V, |V_z| = q$ и для любого контекста $C_i \in Con_{p_k}$, $|Con_{p_k}| = c$, $1 \le i \le c$ строки $p_k, 1 \le k \le t$ программы *P*, она формирует выходной вектор $M^C(p_k) = (w_1, w_2, ..., w_q)$, где $0 \le w_j \le 1, j = 1, ..., q$, $w_j$ - вероятность наличия в строке $p_k$ уязвимости $v_j$ с учетом ее контекста.

Последовательная проверка с помощью модели $M^C$ каждого контекста из множества $Con_{p_k}$, $|Con_{p_k}| = c$ строки $p_k$ позволяет сформировать матрицу

$$\begin{pmatrix} w_{11} & w_{12} & ... & w_{1q} \\ w_{21} & w_{22} & ... & w_{2q} \\ ... & ... & ... & ... \\ w_{c1} & w_{c2} & ... & w_{cq} \end{pmatrix}, \text{ где } 0 \le w_{ij} \le 1, i = 1, ..., c, j = 1, ..., q,$$

где $w_{ij}$ - вероятность наличия в строке $p_k$ уязвимости $v_j$ при ее проверке нейросетевой моделью $M^C$ с учетом некоторого контекста $C_i$ из множества контекстов $Con_{p_k}$, $\left|Con_{p_k}\right| = c$.

Замечание. Поскольку максимально возможный контекст для строки представляет собой всю программу P, то модели, которые могут использоваться для обнаружения уязвимостей строк с учетом их контекстов должны обладать возможностью обрабатывать входные данные длиной $k \geq len(P)$, где $len(P)$ - длина всей программы P. Кроме того, формирование набора данных для обучения подобных моделей, также значительно усложняется. Так, в обучающей выборке на вход должен подаваться контекст некоторой строки s, и сама строка s.

### *Разработка класса нейросетевого детектора и прототипа приложения*

Рассмотрим пример реализации абстрактной модели на практике в виде некоторого класса детектора программного кода. Теоретико-множественный подход, который используется при описании математической модели является удобным для задания класса, который будет реализовывать функционал нейросетевого детектора. На рис. 1. Приведена UML диаграмма класса нейросетевого детектора на уровне строк.

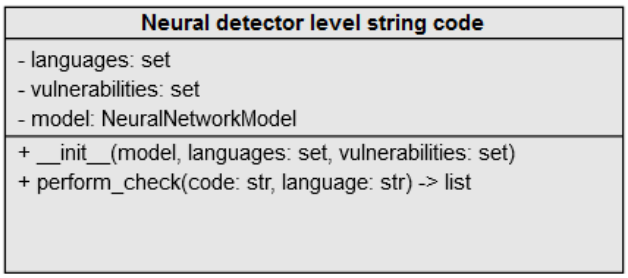



Рисунок 1 – Диаграмма класса детектора

Данный класс включает такие атрибуты:

- languages: множество поддерживаемых языков программирования, на которых детектор может анализировать код;
- vulnerabilities: множество поддерживаемых типов уязвимостей, которые детектор может обнаружить.
- model: поле для хранения нейросетевой модели, которая будет использоваться для анализа уязвимостей.

А также методы:

- __init__(model, languages: set, vulnerabilities: set): конструктор, инициализирующий поля класса, такие как поддерживаемые языки программирования, типы уязвимостей и модель.
- perform_check(code: str, language: str) -> list: основной метод, выполняющий проверку исходного кода на уязвимости. Проверяет, поддерживается ли язык программирования, и, если поддерживается, запускает анализ с помощью модели. Возвращает результаты анализа в виде списка с вероятностями наличия уязвимостей в анализируемой строке.

Необходимо отметить, что строки, превышающие входную длину модели, по умолчанию, обрезаются на этапе обработки их самой моделью, поэтому для исключения потери данных в таких строках необходимо корректно выполнить подготовку строк для проверки и сформировать множество $C(P)_k$ согласно модели представления исходного кода.

В случае, если множество уязвимостей содержит только один конкретный тип уязвимости, то на выходе модели может быть всего одно значение, которое показывает вероятность наличия в строке уязвимости данного типа. На рис. 2. Представлен прототип интерфейса приложения, которое иллюстрирует данный подход.


Рисунок 2 –Интерфейс приложения

Задание множества языков – определяется выбором пользователя, исходя из конкретной задачи. Разработанный нейросетевой детектор позволяет обнаруживать уязвимость типа CWE-259 - Жестко вписанные в программный код пароли или учетные данные. В качестве модели для данного детектора, с использованием техники низко ранговой адаптации LoRA [13] была обучена модель на архитектуре Transformer – Roberta-base, эти результаты представлены в работе авторов [14]. По результатам проверки, формируется файл отчета.

### *Выводы*

Оценка безопасности и обнаружение уязвимостей в программном коде является актуальной задачей программной инженерии. Одним из методов анализа кода является метод статического анализа программного кода. В последние годы активно развиваются новые методы статического анализа, которые связаны с использованием методов машинного обучения и больших языковых моделей. Однако, развитие данного подхода с одной стороны требует формирования новых качественных наборов данных содержащий примеры различных видов уязвимостей, с другой стороны возникает необходимость стандартизации и оценки качества полученных нейросетевых моделей.

В данной работе с использованием теоретико-множественного подхода представлена формальная модель представления программного кода и модель нейросетевого детектора, которая может быть использована для оценки макропараметров нейросетевого статического анализатора, например по такой характеристике как поддерживаемое число типов уязвимостей, поддерживаемые языки, а также стандартные меры оценки точности нейросетевых моделей.

Сильной стороной нейросетевого подхода в статическом анализе является высокая степень эвристики при анализе угроз. Однако, всегда возможны ложно положительные прогнозы. Наиболее перспективно использование подобных анализаторов в сочетании с классическими методами анализа программного кода.

### *Литература*

1. Cousot, P. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints / P. Cousot // Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. — 1977. - pp. 238–252.

2. Allen, F.E. Control flow analysis / F.E. Allen // ACM SIGPLAN Notices. - Volume 5 Issue 7 July, 1970. pp. 1–19.

3. Johnson, S. C. Lint, C Program Checker / S.C. Johnson // COMP. SCI. TECH. REP. — 1978. — pp. 78–90.

4. Аветисян, А. И. Использование статического анализа для поиска уязвимостей и критических ошибок в исходном коде программ / А.И. Аветисян, А.А. Белеванцев, А.Е. Бородин, В. Несов // Труды ИСП РАН. - Том 21. - 2011. - С. 23-38.

5. Smith, J. How Developers Diagnose Potential Security Vulnerabilities with a Static Analysis Tool / J. Smith, B. Johnson, E. Murphy-Hill, B. Chu, H. R. Lipford // IEEE Transactions on Software Engineering. – 2019. - 45(9). Pp. 877–897.

6. Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need / A. Vaswani, N. Shazeer, N. Parmar et al // Advances in Neural Information Processing Systems. - 2017. - pp. 5998–6008.

7. Hou, X. Large Language Models for Software Engineering: A Systematic Literature Review / X. Hou, Y. Zhao,

Y. Liu et al. - 2023. ArXiv, abs/2308.10620. - DOI:10.48550/arXiv.2308.10620.

8. Cheshkov A. Evaluation of ChatGPT Model for Vulnerability Detection / A. Cheshkov, P.A. Zadorozhny, R. Levichev. ArXiv, abs/2304.07232, 2023. - DOI:10.48550/arXiv.2304.07232.

9. Fu M., Tantithamthavorn C.K., Nguyen V., Le T. ChatGPT for Vulnerability Detection, Classification, and Repair: How Far Are We? / M. Fu, C.K. Tantithamthavorn, V. Nguyen, T. Le // 30th Asia-Pacific Software Engineering Conference (APSEC). - 2023. - pp. 632–636. - DOI: 10.1109/APSEC60848.2023.00085.

10. Chan, A. Transformer-based Vulnerability Detection in Code at EditTime: Zero-shot, Few-shot, or Fine-tuning / A. Chan, A. Kharkar, R.Z. Moghaddam, Y. Mohylevskyy et al. - 2023. ArXiv, abs/2306.01754. DOI:10.48550/arXiv.2306.01754.

11. Luo, Z. WizardCoder: Empowering Code Large Language Models with Evol-Instruct / Z. Luo, C. Xu, P. Zhao et al. - 2023. - ArXiv, abs/2306.08568. - DOI: 10.48550/arXiv.2306.08568.

12. Okun, V. Report on the Static Analysis Tool Exposition (SATE) IV / V. Okun, A. Delaitre, P.E. Black // NIST Special Publication. - 2013. - Vol. 500. - DOI: 10.6028/NIST.SP.500-297.

13. Hu E., Shen Y., Wallis P. et al. LoRA: Low-Rank Adaptation of Large Language Models / E. Hu, Y. Shen, P. Wallis et al. ArXiv, abs/2106.09685, 2021.

14. Швыров, В.В. Дообучение больших языковых моделей с использованием техники LoRA для решения задач статического анализа программного кода / В.В. Швыров, Д.А. Капустин, А.В. Кущенко, Р.Н. Сентяй // Вестник Луганского государственного университета имени Владимира Даля. – 2023. - № 12(78). – С. 210-215.

***Швыров В.В., Капустин Д.А., Сентяй Р.Н. Разработка абстрактной модели нейросетевого детектора уязвимостей в программном коде.*** *В последние годы методы машинного обучения и большие языковые модели все более активно используются в самых различных областях программной инженерии. Одной из перспективных сфер применения больших языковых моделей является статический анализ программного кода, в котором нейросетевые модели используются для обнаружения и классификации различных видов уязвимостей. В данной работе представлена математическая модель абстрактного нейросетевого детектора уязвимостей в программном коде. Модель включает описание уровня представления программного кода, формальное теоретико-множественное определение нейросетевого детектора, а также ряд необходимых определений и обозначений. Кроме того, с использованием предложенной модели разработан прототип класса, реализующего обнаружение уязвимостей в программном коде с использованием нейросетевого подхода.*

***Ключевые слова:*** *уязвимость, программный код, статический анализ, анализ безопасности, математическая модель, большие языковые модели.*

***Shvyrov Vjacheslav, Kapustin Denis, Sentay Roman Development of an Abstract Model for a Neural Network-Based Vulnerability Detector in Program Code.*** *In recent years, machine learning methods and large language models have been increasingly applied in various fields of software engineering. One promising area for the application of large language models is static analysis of program code, where neural network models are used to detect and classify different types of vulnerabilities. This work presents a mathematical model for an abstract neural network-based vulnerability detector in program code. The model includes a description of the program code representation layer, a formal set-theoretic definition of the neural network detector, as well as a series of necessary definitions and notations. Additionally, using the proposed model, a prototype class has been developed that implements vulnerability detection in program code through a neural network approach..*

***Keywords:*** *vulnerability, program code, static analysis, security analysis, mathematical model, large language models.*

УДК 628.3:621.3

# Рациональные подходы к внедрению цифровых двойников в водопроводно-канализационные хозяйства

**В.Н. Штепа**[*1]

[*1] д.т.н, доцент, Белорусский государственный технологический университет,
shtepa@belstu.by, OrcID: 0000-0002-2796-3144, SPIN-код: 2834-2138

***Штепа В.Н. Рациональные подходы к внедрению цифровых двойников в водопроводно-канализационные хозяйства.*** *Проанализированы основные недостатки функционирования существующих систем автоматизации водопроводно-канализационных хозяйств. Определено одним из наиболее перспективных направлений повышения эффективности соответствующих технологических процессов использование цифрового моделирования (концепты цифровых двойников). С учётом значительной стоимо automated process control systemscти создания таких продуктовых решений обоснованы и предложены концептуальные схемы их практического внедрения в рамках уже работающих SCADA и АСУТП с дополнением ресурсными базами знаний, лабораторными информационно-моделирующими системами и виртуальными физико-математическими моделями.*

***Ключевые слова:*** *водоотведение, водоснабжение, водопроводно-канализационные хозяйства, управление, цифровое моделирование, цифровой двойник.*

***Shtepa V.N. Rational approaches to the implementation digital twins in water supply and sewerage systems.*** *The main shortcomings existing automation systems for water supply and sewerage facilities are analyzed. One of the most promising results increasing the efficiency technological processes using conceptual analysis (the concept of digital twins) is determined. Taking into account the cost of creating products such solutions, conceptual schemes for their practical application within the framework of the already industrial SCADA and automated process control systems with additions resource knowledge bases, laboratory information-modeling mechanisms and virtual physical and mathematical ones are substantiated and proposed.*

***Keywords:*** *water disposal, water supply, water supply and sewerage, management, digital modeling, digital twin.*

**Полный текст статьи опубликован в научном журнале «Информатика и кибернетика», № 3(37), 2024 г.**

УДК 004.89

# Моделирование системы подготовки специалистов на основе её интеллектуализации методами искусственного интеллекта

**О.И. Федяев**[*1]

[*1] к.т.н, доцент, Донецкий национальный технический университет,
olegfedyayev@mail.ru, OrcID: 0000-0001-6822-7306, SPIN-код: 7777-3791

***Федяев О.И. Моделирование системы подготовки специалистов на основе её интеллектуализации методами искусственного интеллекта.*** *Приведена математическая постановка задачи мультиагентного моделирования процесса распределения студентов на фирмы. Она представлена как задача коллективного выбора с учётом взаимных требований сторон. Для каждого типа агентов формализованы его роли и протоколы взаимодействия с другими агентами. Предложен подход к реализации сотрудничества агентов при поиске компромиссного решения в распределении студентов. Эксперимент по анализу трудоустройства проведен в режиме координированного взаимодействия агентов.*

***Ключевые слова:*** *Мультиагентная система, программный агент, имитационная модель, сотрудничество агентов, трудоустройство студентов.*

## *Введение*

В работе рассматривается применение теории и практики двусторонних рынков к решению задачи распределения выпускников кафедры университета по фирмам. Если учесть, сколько участников взаимодействуют с каждой стороны, то рынок труда, состоящий из выпускников и фирм, - пример рынка «один-на-много» [1]. В таком двустороннем рынке главное уметь формально доказать существование устойчивого распределения участников рынка и благодаря этому определить конкретные связи между работодателями и молодыми специалистами. Проблема современных рынков труда состоит в сложности координации одновременного поиска работы и работников.

Одно из возможных решений этой проблемы – централизация координационного процесса трудоустройства. Для решения задачи трудоустройства студентов в работе предложен способ коллективного выбора с учётом взаимных интересов, основанный на построении централизованной системы распределения выпускников на уровне кафедры университета [1,2]. С этой целью разработана методика и прототип программной системы, позволяющей, исходя из полученных студентами знаний по базовым дисциплинам, прогнозировать успешность их трудоустройства с учётом профессиональных требований фирмы. Эта система также позволит выявлять возможные пробелы в подготовке студентов на выпускающей кафедре и оценивать качество подготовки.

Чтобы учесть особенности анализируемых процессов и построить адекватную модель, в данной работе имитационная система строится методами агентно-ориентированного моделирования, которые сейчас успешно применяются для описания поведения неоднородных динамических систем с распределённым интеллектом [3].

## *Структура система моделирования подготовки и трудоустройства студентов*

Общая структура системы моделирования, частью которой является подсистема трудоустройства выпускников, представлена на рис. 1. В системе представлены три группы людей, между которыми происходит взаимодействие. Каждая из групп передаёт свои цели и способы их достижения своим программным агентам, которые в свою очередь моделируют основные функции реальных людей, используемые в процессе обучения и трудоустройства студентов. В начале рассмотрим общий подход к моделированию.

Самую большую группу агентов образуют агенты студентов, которые обладают полученными в процессе обучения определёнными компетенциями (знаниями и умениями). В модели рассматриваются два способа извлечения у студентов остаточных знаний и умений, которые делегируются искусственным агентам - двойникам студентов. Первый способ – это передача профессиональных навыков от молодого специалиста (источника знаний) к нейросетевому программному агенту на основе коммуникативного метода извлечения знаний из реальных студентов выпускников. Второй способ основан на прогнозировании качества усвоения студентом

знаний по отдельным дисциплинам в зависимости от его личностных характеристик. Каждый студент обладает определённым ментальным портретом (темпераментом, уровнем IQ, типом личности и т.д.). Предполагается, что существует функциональная зависимость полученных конкретным студентом знаний и умений от ментальных портретов студента и преподавателя, а также среды обучения. Для того, чтобы агенты преподавателей могли прогнозировать уровень остаточных знаний и умений, каждый студент должен предоставить им свой ментальный портрет. Эти способы описаны в публикации [4] и в данной статье не рассматриваются.

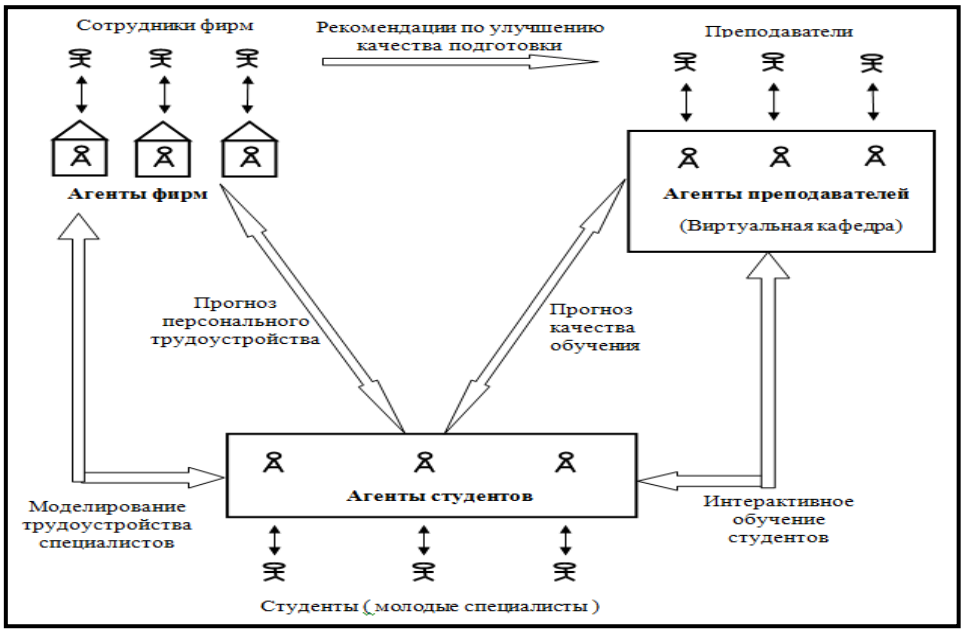



Рисунок 1 – Многоагентная система имитационного моделирования процесса подготовки и распределения студентов на фирмы

Далее, после того, как для агентов-студентов определились остаточные знания, они начинают поиск привлекательных фирм, для этого они «обходят» все фирмы и узнают у них условия работы: размер зарплаты, количество рабочих часов и другие важные социальные факторы. После этого студент формирует для каждой фирмы оценку её привлекательности, на основе которых он определяет очерёдность посещения фирм. В свою очередь фирмы, при выявлении желания студента работать у них, запрашивают у него уровни остаточных знаний и умений по профильным дисциплинам. В заключении каждая фирма оценивает компетенции кандидатов, предлагая им «решить» набор тестовых заданий по профилю фирмы, и успешно прошедшим этап тестирования кандидатам предлагает трудоустроиться.

### *Двухкритериальная модель задачи трудоустройства*

Агентно-ориентированная модель динамического процесса трудоустройства студентов на фирмы (предприятия) представлена двумя группами взаимодействующих искусственных агентов: множеством агентов $X = \{x_1, x_2, \dots, x_N\}$, представляющих студентов, которые хотят трудоустроиться по специальности (N - количество студентов); множеством агентов $F = \{f_1, f_2, \dots, f_M\}$, представляющих фирмы разного профиля, которые предлагают вакантные места для работы (M – количество фирм).

Процесс трудоустройства начинается с того, что каждый студент по своим критериям оценивает для себя привлекательность каждой фирмы

$$t_{n,m} = \varphi_n(c_m) \ , \quad t_{n,m} \in [0,1]$$

где $t_{n,m}$ – оценка привлекательности m-ой фирмы для n-го студента; $\varphi_n$- многомерная функция субъективной оценки студентом $x_n$ привлекательности фирмы $f_m$; $c_m$ – вектор значений социально-экономических характеристик, т. е. социальный пакет, который предлагается студенту на фирме $f_m$. Компоненты вектора $c_m$ определяют размер зарплаты, продолжительность рабочего дня, форму собственности, обеспеченность жильём, возможность удалённой работы и другие показатели, значения которых характеризуют условия работы на m-ой фирме.

В свою очередь каждая фирма оценивает уровень знаний и умений претендентов, предлагая им тестовые задания по профилю деятельности фирмы

$$g_{n,m} = \sum_{j=1}^{J_m} \mu_{m,j}(z_n) \ , \ g_{n,m} \in [0,1]$$

где $z_n$ – вектор значений уровней знаний и умений, которыми обладает студент $x_n$ (в частном случае – по одной дисциплине); $\mu_{m,j}$ - многомерная функция субъективного оценивания m-ой фирмой способность студента $x_n$ решать j-ое задание, предлагаемое фирмой; $g_{n,m}$ - оценка профессиональных компетенций n-го студента, выставленная m-ой фирмой; $J_m$ - количество тестовых заданий у m-ой фирмы ($1 \le j \le J_m$).

Функции $\varphi_n(.)$ и $\mu_{m,j}(.)$ находятся из решения обратных задач методами машинного обучения. Функция $\varphi_n(.)$ отражает мнение конкретного студента или группы студентов и описывает зависимость привлекательности фирмы от предлагаемого фирмой социального пакета. Вторая функция $\mu_{m,j}(.)$ имитирует поведение работников фирмы, занимающихся подбором кадров, при тестировании уровня компетенции студентов в зависимости от их знаний и умений по профилю деятельности фирмы. Поскольку обе функции связывают качественные данные, то для их конструирования применена нейросетевая методология как универсальное средство функциональной аппроксимации. Для обучения нейросетевых моделей функций использовались данные реально проводимых опросов нескольких десятков респондентов. Решение этих задачи не рассматривается в данной статье.

Основная задача, связанная с распределением студентов по фирмам, относится к проблеме коллективного выбора с учётом взаимных требований сторон, которая типична для задач многокритериального принятия решений. В содержательной форме эта задача состоит в распределении студентов на фирмы так, чтобы отклонения от планов приёма отобранных лучших студентов были минимальными, а желания студентов были максимально учтены.

Формально данная задача относится к классу задач о назначениях с дополнительным требованием [5]. В данной работе рассматривается модель задачи, в которой каждый студент-выпускник сумеет выполнять предложенные фирмой работы на уровне $g_{n,m}$ , но имеет также предпочтения по выбору фирмы с учётом её привлекательности $t_{n,m}$. Обозначим через $s_{n,m}$ логическую переменную, которая при значении равном «1» означает, что $n$-й студент принят на $m$-ю фирму ($s_{n,m} \in \{0,1\}$). Тогда задачу о назначениях с предпочтениями можно представить как двухкритериальную. Фирма стремится максимизировать компетенции выпускников, принятых на работу, а выпускники, в свою очередь, удовлетворить свои предпочтения путём выбора наиболее привлекательной фирмы. Первая целевая функция отвечает за максимизацию суммарных компетенций у принятых на работу

$$H_1(X) = \sum_{n=1}^{N} \sum_{m=1}^{M} g_{n,m} * s_{n,m} \to \max \ .$$

а вторая целевая функция связана с учётом предпочтения студентов по выбору фирмы

$$H_2(X) = \sum_{n=1}^{N} \sum_{m=1}^{M} t_{n,m} * s_{n,m} \to \max \ ,$$

Стоит заметить, что предпочтения студентов не всегда совпадают с кадровыми потребностями фирмы. Отсюда возникает конфликт целевых функций $H_1(X)$ и $H_2(X)$. Поиск экстремумов выполняется с учётом следующих ограничений:

$$\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} s_{n,m} \le \boldsymbol{l_m} \ ,$$

$$\sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} s_{n,m} \le \boldsymbol{1} \ ,$$

$\boldsymbol{l_m}$ - количество вакансий на m-ой фирме ($0 \le l_m \le N$). Вид функций $H_1(X)$ и $H_2(X)$ даёт возможность использовать вместо двух следующий один обобщённый критерий

$$H(X) = \sum_{n=1}^{N} \sum_{m=1}^{M} (t_{n,m} + g_{n,m}) * s_{n,m} \to \max \ ,$$

что позволяет задачу о назначениях решать венгерским методом [4].

### *Агентно-ориентированная имитационная модель процесса трудоустройства студентов*

В виде программы многоагентная система моделирования была реализована с помощью инструментальной среды Jason, которая ориентирована на создание интеллектуальных программных агентов с

открытой BDI-архитектурой, позволяющей описывать на языке Java поведение агентов с учётом специфики решаемой задачи [6]. На рис. 2 представлен интерфейс программного обеспечения системы, реализующей взаимодействие с системой посредством удобных окон.

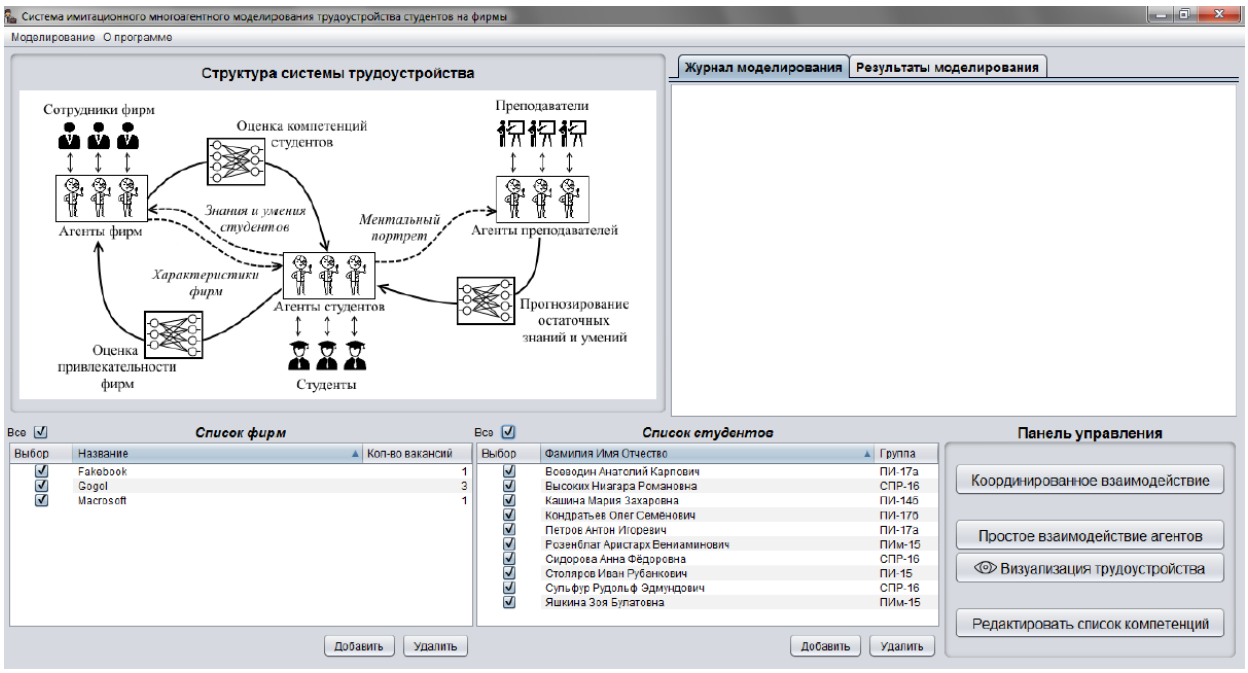



Рисунок 2 - Пользовательский интерфейс системы имитационного моделирования

Окна позволяют настраивать агентов-студентов и агентов-фирм, задавая их характеристики, а также редактировать список доступных компетенций и выбирать различные режимы моделирования процесса трудоустройства. Окно диалога для задания социально-экономических характеристик фирмы представлено на рис. 3.

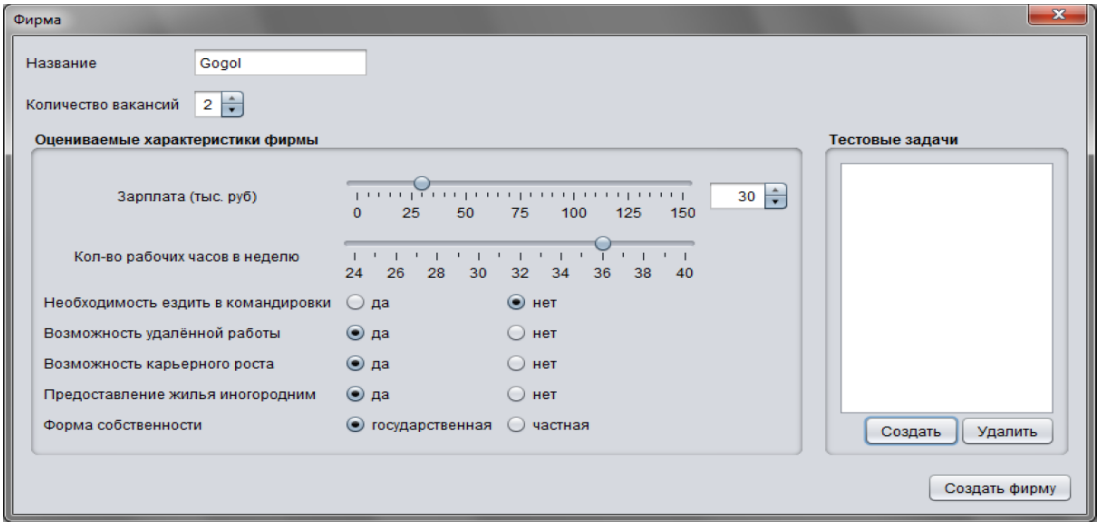



Рисунок 3 - Диалоговое окно для задания условий работы на фирме

Первым этапом моделирования процесса трудоустройства является «собеседование», которое происходит между агентами студентов и работниками фирм. На рис. 4 представлена диаграмма последовательности на языке UML, показывающая динамику взаимодействия агентов при наличии Координатора (арбитра), роль которого может выполнять, например, ответственный за распределение студентов на кафедре или заведующий кафедрой.

При координированном взаимодействии агентов для получения плана распределения агент Координатор, получив на этапе знакомства списки предпочтения агентов, может сразу применять оптимизационный метод и выдать оптимальное решение по трудоустройству. При таком виде трудоустройства агенты взаимодействуют

только на этапе знакомства.

Функция вычисления оценки привлекательности фирмы, которая используется в самом начале взаимодействия агентов, реализована в нейросетевом базисе, так как она является трудно формализуемой. Поэтому для её построения использовалась модель нейронной сети - многослойный персептрон, а проверка адекватности модели нейросети выполнялась на реальных данных (см. рис. 3), полученных от нескольких десятков студентов-респондентов. После настройки нейросетевой функции по стратегии «обучение с учителем» были проведены эксперименты для получения субъективной оценки привлекательности фирмы со стороны конкретного студента (см. рис.5).

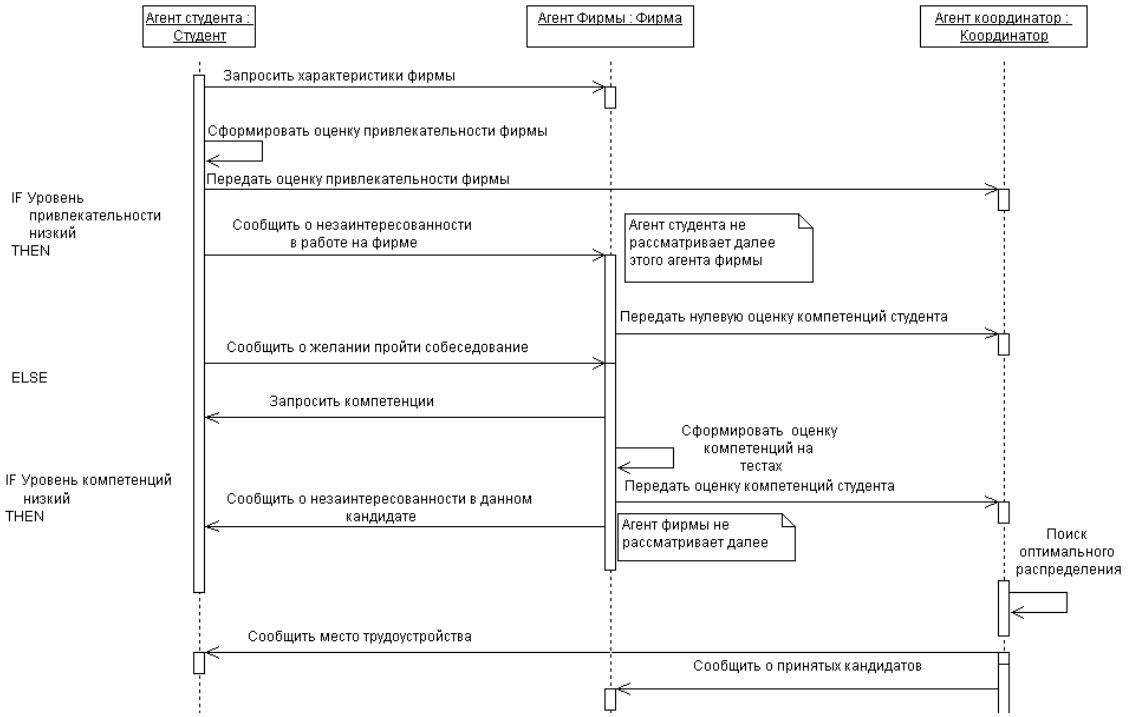



Рисунок 4 - Взаимодействие агентов при координируемом сотрудничестве


Рисунок 5 - Эксперименты с нейросетью, обученной на основе предпочтений конкретного студента

Следующая важная задача связана с получением оценки уровня владения компетенциями (знаниями и умениями) в виде коэффициентов уверенности из диапазона [0,1], которые отражают наличие остаточных знаний по той или иной дисциплине. В свою очередь работодатель формирует список тестовых задач, которые являются профильными для его предприятия, попутно выбирая те знания и умения, которые, по его мнению, действительно важны в решении соответствующей задачи и ставит каждой компетенции свой коэффициент важности по фиксированной четырёхбальной шкале (1 – высокий, 0.8 – выше среднего, 0.5 – средний, 0.3 – малый). Для удобства коэффициенты важности ($P_i$) целесообразно нормализовать в диапазон от 0 до 1, так чтобы $\sum P_i = 1$. Поскольку все входные сигналы изменяются в одном диапазоне от 0 до 1 и оценки важности, установленные экспертом, также нормализованы, то оценку успешности решения одной задачи можно реализовать одним нейроном с линейной функцией активации без обучения (весовыми коэффициентами выступают нормализованные коэффициенты важности Pi ). Также необходимо отметить, что итоговым результатом оценки уровня компетенций студента является обобщающий результат оценок по всем задачам фирмы, поэтому итоговая оценка вычисляется по формуле

$$\text{Итоговая оценка} = \frac{1}{k}\sum_{j=1}^{k}\sum_{i=1}^{N+M} x_i * P_i \quad ,$$

где k - количество тестовых задач; N, M – количество соответственно знаний и умений по дисциплине; $x_i$ - степень владения компетенцией студентом; $P_i$- нормализованный коэффициент важности.

Ниже приведены результаты моделирования процесса трудоустройства, в котором участвовали 3 фирмы (у всех по 4 вакантных места) и 10 студентов (у всех принята минимальная привлекательность и минимальный уровень знаний, равные 0). При запуске процесса моделирования все агенты студентов проходят «собеседование» с агентами фирм в соответствии с алгоритмом на рис. 4 и по ранее рассмотренным методикам агенты с нейросетевой архитектурой формируют значения привлекательности и компетенции, которые передаются агенту Координатору (см. табл. 1).

Таблица 1. Предпочтения агентов

| Студенты | $f_1$ – кол-во вакансий - 4 | | $f_2$ – кол-во вакансий - 4 | | $f_3$ – кол-во вакансий - 4 | |
|---|---|---|---|---|---|---|
| | Привлека-тельность | Компетен-ция | Привлека-тельность | Компетен-ция | Привлека-тельность | Компетен-ция |
| $x_1$ | 0,44 | 0,89 | 0,38 | 0,76 | 0,5 | 0,25 |
| $x_2$ | 0,62 | 0,25 | 0,41 | 0,23 | 0,23 | 0,84 |
| $x_3$ | 0,63 | 0,79 | 0,5 | 0,73 | 0,43 | 0,37 |
| $x_4$ | 0,46 | 0,72 | 0,57 | 0,44 | 0,35 | 0 |
| $x_5$ | 0,79 | 0,04 | 0,77 | 0,85 | 0,76 | 0,86 |
| $x_6$ | 0,4 | 0,27 | 0,74 | 0,76 | 0,3 | 0,26 |
| $x_7$ | 0,51 | 0,24 | 0,08 | 0,91 | 0,43 | 0,84 |
| $x_8$ | 0,06 | 0,48 | 0,51 | 0,1 | 0,92 | 0,77 |
| $x_9$ | 0,82 | 0,22 | 0,99 | 0,42 | 0,5 | 0,24 |
| $x_{10}$ | 0,01 | 0,04 | 0,31 | 1 | 0,47 | 0,8 |

На рис. 6 представлен журнал моделирования, в котором записываются все действия, совершённые агентами в ходе моделирования (строки без сдвига – действия агентов-фирм, строки со сдвигом – действия агентов-студентов).

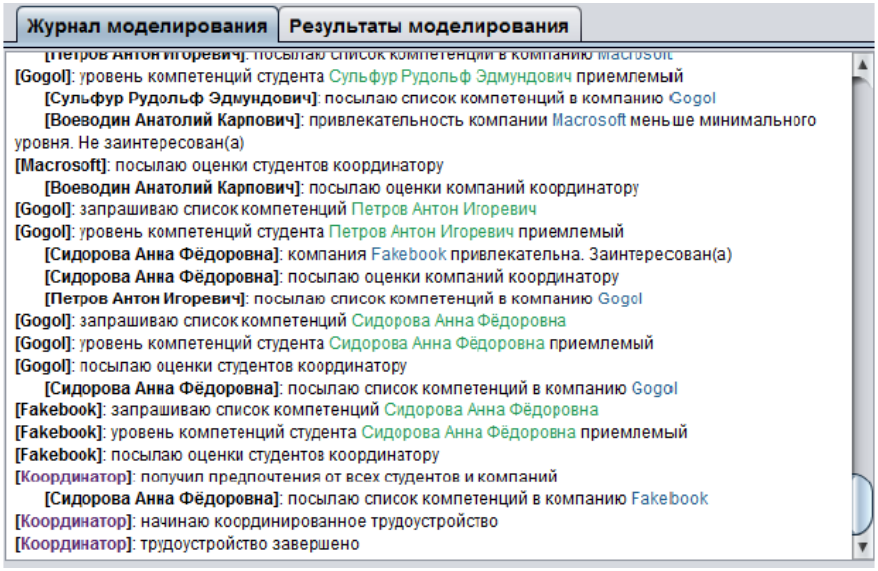



Рисунок 6 - Журнал моделирования

На рис. 7 представлены результаты моделирования трудоустройства с помощью координированного взаимодействия, используя венгерский метод оптимизации [5].

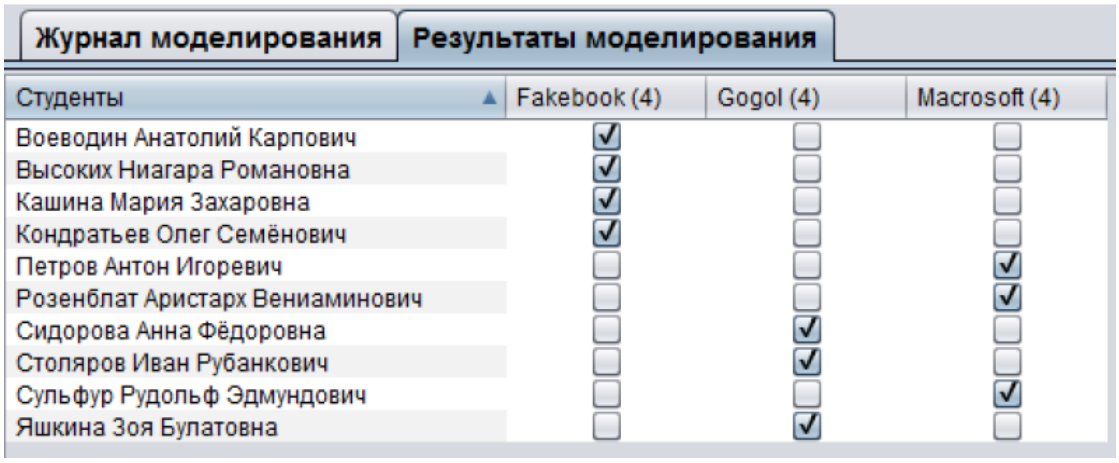



Рисунок 7 – Результаты трудоустройства с помощью координированного взаимодействия

В итоге все студенты нашли себе рабочие места. Однако, необходимо отметить, что не было элемента соревнования, т.е. везде были места и студенты беспрепятственно занимали их.

### *Выводы*

Имитационная модель представлена в виде многоагентной системы, учитывающей распределённость, неоднородность и интеллектуальность объекта исследования. Кроме того, в статье рассмотрена постановка задачи моделирования процесса трудоустройства студентов. Динамика процесса распределения студентов на фирмы реализована совокупностью взаимодействующих программных агентов. Для каждого типа агентов формализованы его роли и протоколы взаимодействия с другими агентами. Предложен подход к реализации сотрудничества агентов при поиске компромиссного решения, в основе которого лежит принцип гомеостатического (равновесного) управления динамикой многоагентной системы. С помощью разработанной имитационной модели проведен эксперимент, посвящённый анализу процесса трудоустройства студентов в режиме координированного взаимодействия агентов.

### *Литература*

1. Железова Е.Б., Измалков С.Б., Сонин К.И., Хованская И.А. Теория и практика двусторонних рынков // Вопросы экономики, №1, 2013. – С. 4-26.

2. Андрейчиков А.В., Андрейчикова О.Н. Интеллектуальные информационные системы: Учебник. – М.: Финансы и статистика, 2006. – 424 с.
3. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.
4. Федяев О.И. Прогнозирование остаточных знаний студентов по отдельным дисциплинам с помощью нейронных сетей // Известия ЮФУ. Технические науки. – 2016. – №7 (180). – С. 122 - 136.
5. Таха Х. Введение в исследование операций. – М.: Издательский дом Вильямс, 2003. – 407 с.
6. Programming multi-agent systems in AgentSpeak using Jason – https://www.gbv.de/dms/ilmenau/toc/522511953.PDF

***Федяев О.И. Моделирование системы подгоовки специалистов на основе её интеллектуализации методами искусственного интеллекта.*** *Приведена математическая постановка задачи мультиагентного моделирования процесса распределения студентов на фирмы. Она представлена как задача коллективного выбора с учётом взаимных требований сторон. Для каждого типа агентов формализованы его роли и протоколы взаимодействия с другими агентами. Предложен подход к реализации сотрудничества агентов при поиске компромиссного решения в распределении студентов. Эксперимент по анализу трудоустройства проведен в режиме координированного взаимодействия агентов.*

***Ключевые слова:*** *мультиагентная система, программный агент, имитационная модель, сотрудничество агентов, трудоустройство студентов.*

***Fedyaev O.I. Modeling of a specialist training system based on its intellectualization by artificial intelligence methods.*** *The mathematical formulation of the problem of multi-agent modeling of the process of distributing students to firms is given. It is presented as a task of collective choice, taking into account the mutual requirements of the parties. For each type of agent, its roles and protocols for interaction with other agents are formalized. An approach to the implementation of agent cooperation in the search for a compromise solution in the distribution of students is proposed. The experiment on the analysis of employment was conducted in the mode of coordinated interaction of agents.*

***Key words:*** *multi-agent system, software agent, simulation model, agent collaboration, student employment.*

УДК 004.93’1

# Экспериментальная оценка качества нейросетевого распознавания студентов по изображениям лиц из базы данных

**А.А. Суханов*1, Д.Э. Баев*2. О.И. Федяев*3**

*1 аспирант, Донецкий национальный технический университет,
studysukhanov@mail.ru

*2 Lead QA Engineer, TradingView, Malaga, Spain
azo.cw@yandex.ru

*3 к.т.н, доцент, Донецкий национальный технический университет,
olegfedyayev@mail.ru, OrcID: 0000-0001-6822-7306, SPIN-код: 7777-3791

***Суханов А.А., Баев Д.Э., Федяев О.И. Экспериментальная оценка качества нейросетевого распознавания студентов по изображениям лиц из базы данных.*** *Рассматривается задача автоматизации видеорегистрации присутствия студентов на учебном занятии. Выполнена постановка задачи компьютерного распознавания и разработана функциональная схема системы. Описана архитектура нейронной сети VGGFace и её использование для распознавания лиц из видеопотока. Проведена экспериментальная оценка качества распознавания лиц в зависимости от различного вида помех, и от объёма видеоданных, определяемого количество студентов.*

***Ключевые слова:*** *компьютерное зрение, свёрточная нейронная сеть, распознавание лиц, видеопоток, регистрация студентов в аудитории, VGGFace16.*

## *Введение*

Рассматривается решение задачи, связанной с автоматизацией контроля присутствия студентов в аудитории, с помощью компьютерного зрения. Актуальность данной задачи обусловлена множеством факторов, во-первых, потерями времени преподавателя, которые возникают при ручной регистрации присутствия большого количества учащихся в аудитории, к примеру, в потоке из 70 человек потери на «перекличку» составляют в среднем 85 мин в семестре и даже больше; во-вторых, «перекличка» проходит шумно и не всегда достоверно [1].

Кроме того, в стандартных условиях преподаватель ведёт вручную журнал посещений на бумажном носителе, в котором отмечает посещаемость студентами лекций и практических занятий. Как правило, на одну группу по одной дисциплине приходится два журнала – для лекций и для практических занятий. В итоге для одного потока, который обычно состоит из трёх или же четырёх групп, у преподавателя находится от шести до восьми журналов посещаемости. Время и ресурсы, затраченные на их заполнение в течение семестра, просто огромные. Поэтому разработанная система визуального контроля присутствия студентов в аудитории поможет упростить и автоматизировать данные аспекты в работе преподавателя.

В этой связи такая система видеоконтроля должна быть ориентирована на решение следующих подзадач:
локализацию лица входящих в аудиторию студентов из видеопотока в режиме реального времени;
распознавание студентов путём сопоставления признаков лица с лицами из базы данных;
занесение в электронный журнал посещаемости пометки о явке или неявки студента на занятие;
сохранение пометки с журнала в базу данных для дальнейшего их использования.

Создание системы с таким функционалом позволит преподавателю получить дополнительное время, которое он может выделить на предоставление студентам большего объёма учебного материала, не затрачивая время на перекличку. Собранная информация в виде электронной базы данных также даст возможность преподавателю проводить статистический анализ учебной дисциплины студентов.

Основные трудности компьютерного распознавания лиц в реальном времени связаны с помехами и быстрой изменчивостью изображений лиц в видеопотоке: положение, размер и ракурс лица в кадре, освещение и т. д. [1]. В настоящее время большие перспективы в преодолении перечисленных проблем связывают с применением глубоких нейронных сетей [2, 3]. Оценке эффективности этого класса нейросетей в решении задачи распознавания лиц и посвящена данная статья.

### *Постановка задачи компьютерного распознавания лиц*

Введём следующие обозначения: L – множество фамилий распознаваемых людей по лицам (например, список студентов в группе); $\overline{X}$ – множество подготовленных фотографий распознаваемых лиц, т.е. эталоны изображений распознаваемых лиц; l – фамилия человека ($l \in L$), фотография которого изображена на снимке $\bar{x}$ ($\bar{x} \in \overline{X}$).

Предположим, что есть функция f, которая способна выявлять признаки лица по его изображению. С её помощью можно сформировать множество $\overline{Y}$ из векторов признаков, выявленных на фотографиях $\overline{X}$ :

$$f: \overline{X} \to \overline{Y}.$$

Тогда элемент этого множества $\bar{y}$ – это вектор признаков одной из фотографий $\bar{x}$ лица человека с фамилией l ($l \in L$). В результате получаем базу данных для распознавания людей по лицам в виде множества V:

$$V = \left\{(\bar{y}, l) \middle| \, \bar{y} = f(\bar{x}),\ \bar{x} \in \overline{X},\ l \in L\right\} .$$

Соотношения между мощностями множеств можно представить в виде следующих неравенств:

$$|\overline{X}| \geq |L| \ , \quad |\overline{X}| = |\overline{Y}| \ .$$

Система компьютерного зрения в режиме распознавания человека по изображению его лица x должна, во-первых, применить к x свой функционал f для формирования вектора признаков y ($f: x \to y$) и, во-вторых, для него найти в базе данных лиц V наиболее похожий вектор $\bar{y}$ и из найденной пары $(\bar{y}, l)$ взять соответствующую ему фамилию l :

$$l = \min_{(\bar{y} \in \overline{Y}) \& ((\bar{y}, l) \in V)} \|\bar{y} - y\|$$

где l – фамилия распознанного человека по фотографии x ($l \in L$); $\|\cdot\|$ - вычисление косинусного сходства векторов.

Таким образом, задача состоит в том, чтобы построить функционал f, на котором основана рассмотренная идея распознавания образов.

### *Функциональная схема системы видеоконтроля студентов*

Видеорегистрации студентов при входе в аудиторию осуществляется в реальном времени. На рис. 1 показана функциональная схема распознавания студентов по изображению лица из видеопотока.


Рисунок 1 – Схема видеорегистрации студентов при входе в аудиторию

При входе в аудиторию видеокамера, подключенная к компьютеру преподавателя, локализует лицо студента, в результате чего получается изображение выделенного лица. Далее данное изображение кодируется свёрточной нейронной сетью VGGFace [4, 5], т.е. выделенному лицу ставит в соответствие вектор признаков, который запоминается в базе данных при настройке системы.

В штатном режиме работы системы, т.е. при распознавании, в блоке сравнения вектор признаков *y* распознаваемого лица *x*, полученный с выхода свёрточной нейронной сети, сравнивается со всеми векторами $\bar{y}$ базы данных *V*. Процедура сравнения основывается на методе вычисления косинусного сходства вектора распознаваемого лица с каждым вектором-эталоном из базы данных по следующей формуле

$$\text{Сходство} = \frac{Y \cdot \bar{Y}}{\|Y\| \cdot \|\bar{Y}\|} = \frac{\sum_{i=1}^{n} y_i \cdot \bar{y}_i}{\sqrt{\sum_{i=1}^{n} y_i^2} \cdot \sqrt{\sum_{i=1}^{n} \bar{y}_i^2}}$$

где $Y$ и $\bar{Y}$ – вектора признаков соответственно распознаваемого лица и лица-эталона из базы данных; $n$ = 2622 ($n$ – количество классов).

Для выработки признаков лица свёрточная нейронная сеть предварительно была обучена её создателями на примерах фотографий 2622-х человек (по 1000 фотографий на человека) [6]. Сеть настроена на классификацию распознаваемого лица, используя в качестве классов лица из обучающего множества. Поэтому результатом работы сети является 2622-мерный вектор, каждый элемент которого представляет собой вероятность сходства лица с одним из обучающего множества. Считается, что два изображения лица относятся к одному человеку, если они в одинаковой мере похожи на каждое лицо из обучающего множества. Для этого вектора признаков этих изображений в пространстве лиц из обучающего множества должны образовывать между собой достаточно острый угол.

Распознаваемое лицо считается соответствующим эталону, если полученный коэффициент сходства выше определённого значения (в работе использовалось значение 0,7). В случае успешного сравнения лиц, происходит запись в электронный журнал о явке студента на занятие.

### *Архитектура свёрточной нейронной сети*

Архитектура VGGFace16 представляет собой глубокую свёрточную нейронную сеть (CNN), специально разработанную для распознавания лиц, была предложена К. Симоняном и А. Зиссерманом из Оксфордского университета. Данная нейронная сеть имеет многослойную структуру, показанную на рисунке 2.

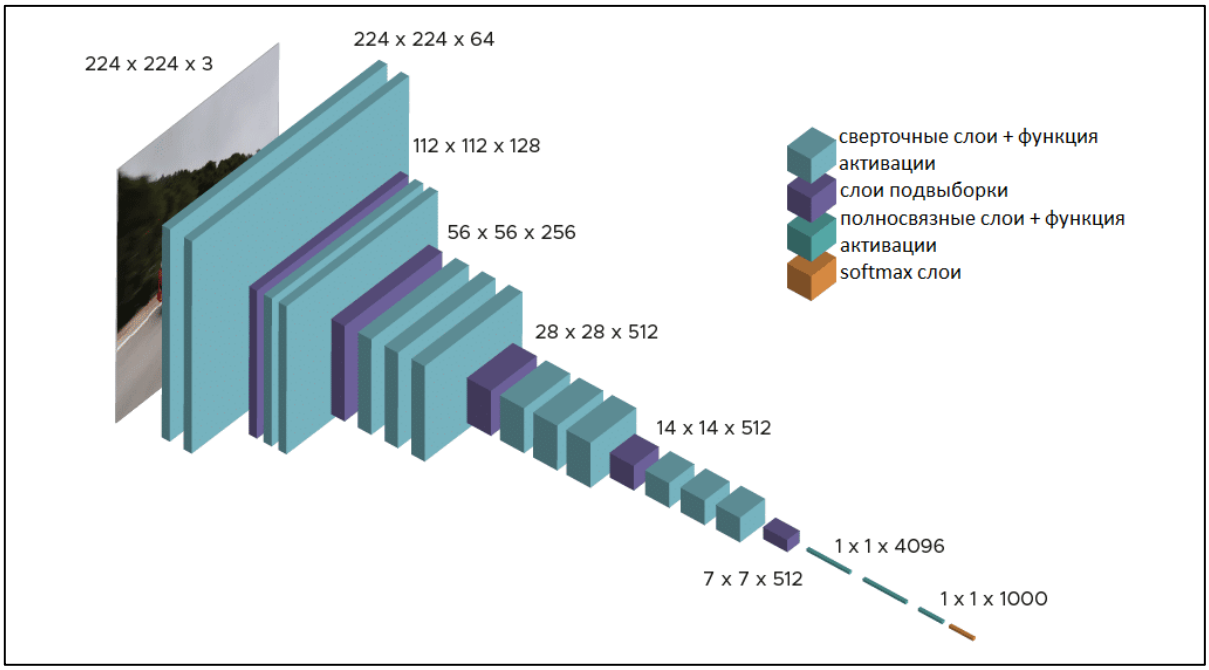



Рисунок 2 – Архитектура модели многослойной свёрточной нейронной сети VGGFace

Входной слой в архитектуре VGGFace16 предназначен для приёма и предварительной обработки входного изображения. Входной слой принимает изображение размером 224x224 пикселей. Это стандартный размер для многих свёрточных нейронных сетей, так как он позволяет сети эффективно обрабатывать и извлекать признаки из изображения. Изображение представляется в формате RGB, что означает наличие трех цветовых каналов (красного, зеленого и синего). Каждый канал отвечает за интенсивность соответствующего цвета в каждом пикселе изображения. Интенсивности пикселей изображения могут быть нормализованы. Например, значения пикселей могут быть масштабированы в диапазон от 0 до 1 или изменены таким образом, чтобы иметь нулевое среднее и единичное стандартное отклонение. Это помогает улучшить производительность сети и ускорить процесс обучения. Входные данные представляются в виде трехмерного тензора (tensor) размером 224x224x3 (высота, ширина, каналы цвета). Этот тензор передается в первый свёрточный слой сети [7, 8].

Свёрточные слои (convolutional layers) являются основными строительными блоками свёрточных нейронных сетей. Они предназначены для извлечения признаков из входных данных через серию фильтров. Каждый свёрточный слой имеет набор фильтров (также называемых ядрами), которые представляют собой маленькие матрицы весов (например, размером 3x3 или 5x5). Фильтры применяются к входным данным для извлечения пространственных признаков (features). Каждый фильтр может распознавать разные аспекты изображения, такие как края, текстуры, формы и т. д. Фильтры скользят (или конволюционируют) по входному изображению, вычисляя скалярное произведение между весами фильтра и соответствующими пикселями в области изображения. Это дает карту признаков (feature map), где каждый элемент представляет собой результат применения фильтра к одной части изображения. После применения фильтра результат проходит через нелинейную функцию активации, например, как ReLU (Rectified Linear Unit) которая имеет следующий вид *f(x) = max (0, x)*. ReLU заменяет все отрицательные значения в карте признаков на ноль, вводя нелинейность в модель, что позволяет ей лучше обучаться сложным зависимостям. Иногда к входному изображению добавляются дополнительные строки и столбцы из нулей (zero padding), чтобы сохранить размерность карты признаков и предотвратить потерю информации по краям. Математически это описывается следующим образом:

$$y_{i,j,k} = f(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{c=0}^{C-1} w_{m,n,c,k} * x_{i+m,j+n,c} + b_k)$$

где $y_{i,j,k}$ – выходное значение на позиции (i, j) и канале k; $x_{i,j,c}$ – входное значение на позиции (i, j) и канале c; $w_{m,n,c,k}$ – значение фильтра на позиции (m, n) и канале c для фильтра k; $b_k$ – смещение фильтра k; M, N – размер фильтра; C – количество входных каналов.

Слой подвыборки (pooling) является важным компонентом свёрточной нейронной сети. Главная цель этого слоя - уменьшение размерности карт признаков, представленных плоскостями свёрточного слоя, с сохранением отличительных особенностей изображения. Это позволяет уменьшить размерность изображения для дальнейшего анализа и снизить вычислительные затраты. С этой целью карта признаков разбивается на квадраты (области объединения), в которых находится максимальное значение признаков. Этот метод часто используется, потому что он хорошо сохраняет важные признаки изображения и описывается следующей формулой:

$$y_{i,j,k} = max_{(m,n)\,\in\, pool}(x_{\bar{\imath}+m,\bar{\jmath}+n,k})$$

где $y_{i,j,k}$ – выходное значение на позиции (i, j) в плоскостях слоя подвыборки канала k; $x_{\bar{\imath}+m,\bar{\jmath}+n,k}$ - входное значение для функции *max*, а фактически это выходные сигналы нейронов на позиции ($\bar{\imath}$, $\bar{\jmath}$,) в соответствующих плоскостях предыдущего свёрточного слоя канала k*;* pool – область объединения в свёрточной плоскости для позиции (i, j) в соответствующей плоскости слоя подвыборки.

Полносвязные слои (fully connected layers) являются также одним из важных компонентом нейронных сетей и играют ключевую роль в задачах классификации. Они работают аналогично традиционным многослойным перцептронам. В полносвязном слое каждый нейрон соединен с каждым нейроном предыдущего слоя, отсюда и название "полносвязный". Эти связи представлены весами, которые настраиваются во время обучения модели. Математически это описывается так:

$$y_i = f(\sum_{j=0}^{N-1} w_{i,j} * x_j + b_i)$$

где $y_i$ – выходное значение нейрона i; $x_j$ – входное значение нейрона j; $w_{i,j}$ – вес соединения между нейронами i и j; $b_i$ – смещение нейрона i; N – количество входных нейронов.

Входные данные полносвязного слоя представляют собой одномерный вектор, полученный после прохождения через свёрточные и pooling слои. Этот вектор умножается на матрицу весов, что позволяет преобразовать входные данные в новый набор признаков. Результат матричного умножения проходит через нелинейную функцию активации, такую как ReLU (Rectified Linear Unit), сигмоида или tanh. Это вводит нелинейность в модель, что позволяет ей моделировать сложные зависимости. В последнем полносвязном слое часто используется функция активации softmax, которая преобразует выходные значения в вероятности принадлежности к разным классам. Полносвязные слои передают свои выходные данные дальше по сети или предоставляют окончательные результаты классификации.

Softmax слой — это компонент нейронных сетей, который используется для преобразования выходных данных в вероятности, соответствующие различным классам. Softmax слой преобразует необработанные

выходные значения (логиты) последнего полносвязного слоя в вероятности, которые суммируются до 1. Это позволяет интерпретировать выходные данные как вероятности принадлежности к различным классам.

Результатом работы softmax слоя является вектор вероятностей, где каждая вероятность соответствует одному из классов. Значения вектора находятся в диапазоне от 0 до 1 и в сумме дают 1. Softmax слой часто используется в задачах многоклассовой классификации, где выходными данными сети являются вероятности принадлежности к нескольким классам. Например, если нейронная сеть обучается распознавать лица, softmax слой будет выводить вероятность того, что изображение принадлежит каждому из известных классов лиц [9].

В системе видеоконтроля студентов в качестве модели свёрточной нейронной сети использовалась нейросеть VGGFace (рис. 2) [4, 5]. Сеть VGGFace принимает на входе RGB изображение лица размером 224x224 (фрагмент изображения, вырезанный по координатам, полученным методом Виолы-Джонса, расширяется или сжимается до этого размера). Далее изображения проходят через стек свёрточных слоёв, в которых используются фильтры с очень маленьким рецептивным полем размера 3x3.

Сеть предварительно обучена на множестве из 2,6 миллионов фотографий (2622 человека, 1000 фотографий каждого). Координаты каждого измерения вектора представляют собой вероятность того, что исходное лицо принадлежит одному из людей из обучающего множества.

Были проведены исследования качества распознавания и производительности системы. Для условий дневного света в помещении университета определены такие граничные значения способности распознавания, как углы поворота головы, уровень освещённости и расстояние до камеры. Распознавание с помощью указанной видеокамеры было устойчивым на расстоянии до 6 м от камеры. Эксперименты показали, что нейросеть распознавала изображения лиц даже размером 22x22 пикселя.

### *Экспериментальная оценка качества распознавания лиц*

Она выполнена на одном компьютере с использованием одной камеры. Технические характеристики веб-камеры: разрешение в pix: 1280*720; тип матрицы: CMOS; фокусное расстояние: автоматическое; частота кадров: до 30 кадров/секунду при разрешении в 1280*720.

Технические характеристики компьютерной системы: процессор: Intel Core i5-9400F CPU 2.90GHz; оперативная память: 16 ГБ; тип системы: 64-разрядная операционная система, процессор x64; видеокарта: NVIDIA GeForce RTX 4060 TI; система: Windows.

Эксперименты направлены на оценку влияния внешних условий на качество распознавания, а именно: 1) влияние расстояния человека от камеры; 2) влияние различного вида помех; 3) влияние угла поворота головы; 4) влияние уровня освещения. Например, при исследовании степени влияния различного вида помех, в качестве помех использовались элементы одежды и аксессуары (рис. 3). Результаты экспериментов представлены в табл. 1.

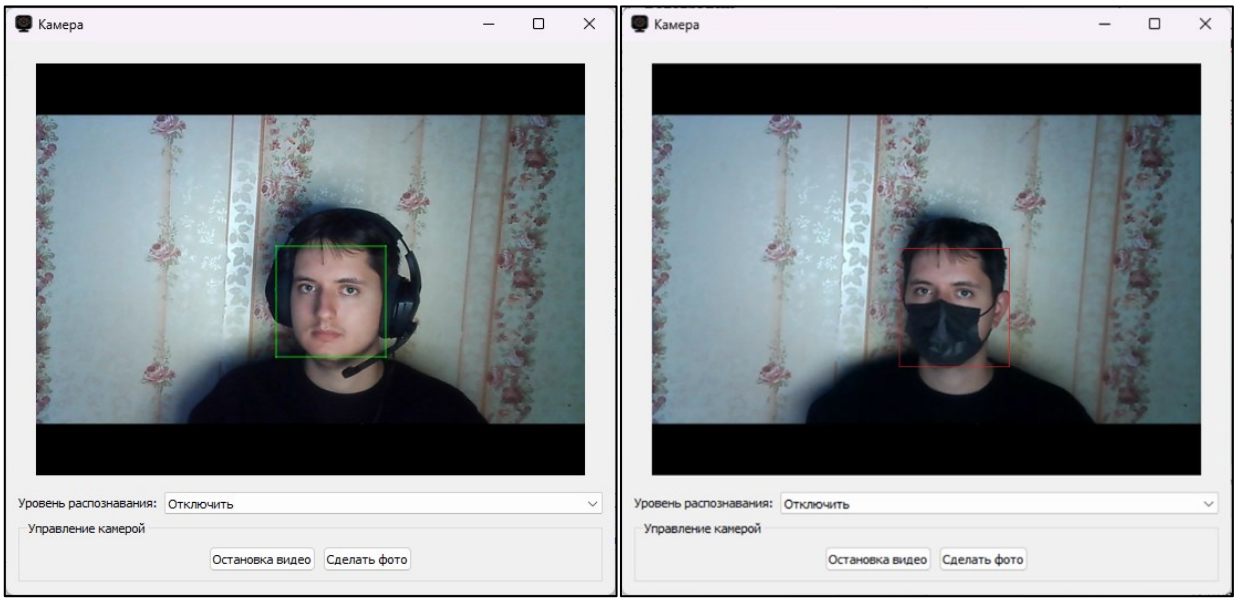



Рисунок 3 - Расположение головы человека перед видеокамерой с помехами

Таблица 1 – Результаты оценки качества распознавания при помехах

| Вид помехи (одежда, аксессуар) | Успешность локализация лица на изображении (да, нет) | Успешность распознавания (да, нет) |
|---|---|---|
| Кепка | да | да |
| Маска | да | нет |
| Очки | да | да |
| Наушники | да | да |
| Кепка+очки | да | да |
| Маска+кепка+очки | да | нет |

По результатам оценки влияния внешних условий на качество распознавания с видами помех можно прийти к выводу что помехи не влияют на успешность локализации лица. На успешность распознавания сильно влияет наличие маски.

Также проведена оценка влияния реального объёма видеоданных, определяемого количеством студентов, на качество распознавания. В связи с тем, что в настоящее время обучение студентов происходит дистанционно, эксперименты проводились с использованием бумажных носителей. Имитация входа студентов в аудиторию заключалась в следующем. К видеокамере преподносили на бумажном носителе формата А4 фотографии студентов размером 5,29 на 4,23 см., сначала одной группы, потом двух выбранных групп, а после всего потока. На этапе настройки системы с данными фотографиями были связаны анкеты (фамилии) студентов, которые распределены по группам. Всего было создано три группы с общим количеством студентов 69. В качестве результата фиксировался процент распознавания.

Учитывая небольшие размеры фотографий, бумажный носитель располагался перед видеокамерой на расстоянии 80 см. Для улучшения результатов локализации лиц и их распознавания это расстояние могло меняться в небольших пределах (рис 3).

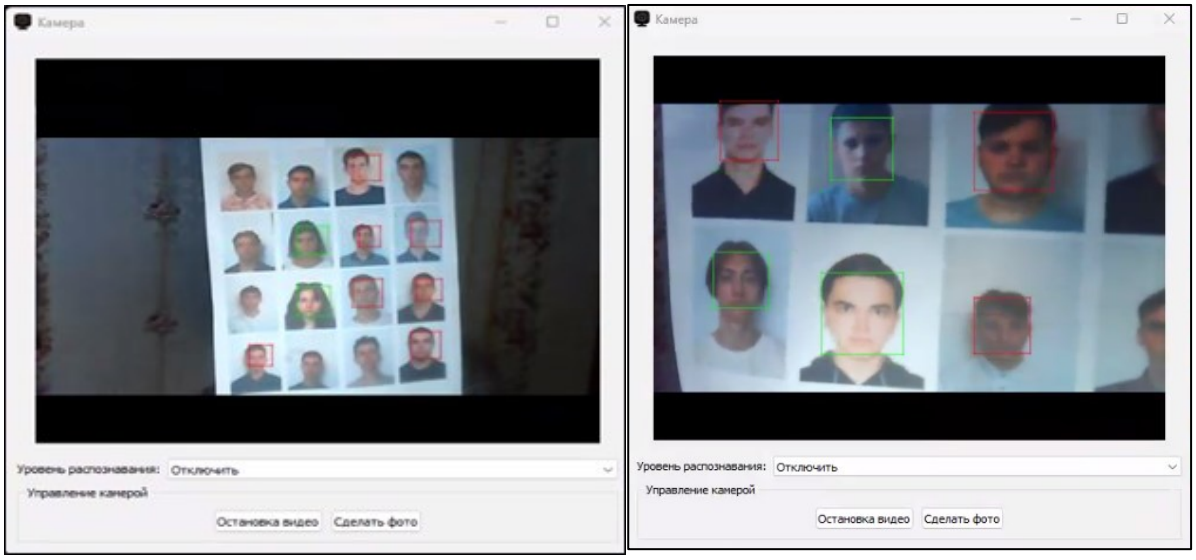



Рисунок 3 – Расположение фотографий лиц перед камерой

Было отмечено, что для фотографий такого размера качество распознавания улучшалось с уменьшением расстояния до веб-камеры. Наилучшее расстояние составило 60 см. В первом эксперименте проводилась видеорегистрация студентов одной группы, состоящей из 22 человек. Во второй колонке табл. 2 приведены результаты того, как система отреагировала на показанные фотографии студентов этой группы.

Таблица 2 - Влияние объема видеоданных на качество распознавания

| Результат распознавания | Количество студентов в группе | | |
|---|---|---|---|
| | Группа из 22 | Группа из 47 | Группа из 69 |
| Количество распознанных | 18 | 41 | 58 |
| Количество не распознанных | 4 | 6 | 11 |
| Процент распознавания | 81% | 87% | 84% |

Пример видеорегистрации студентов в режиме реального времени показа фотографий продемонстрирован на рис. 4 (главное диалоговое окно преподавателя).

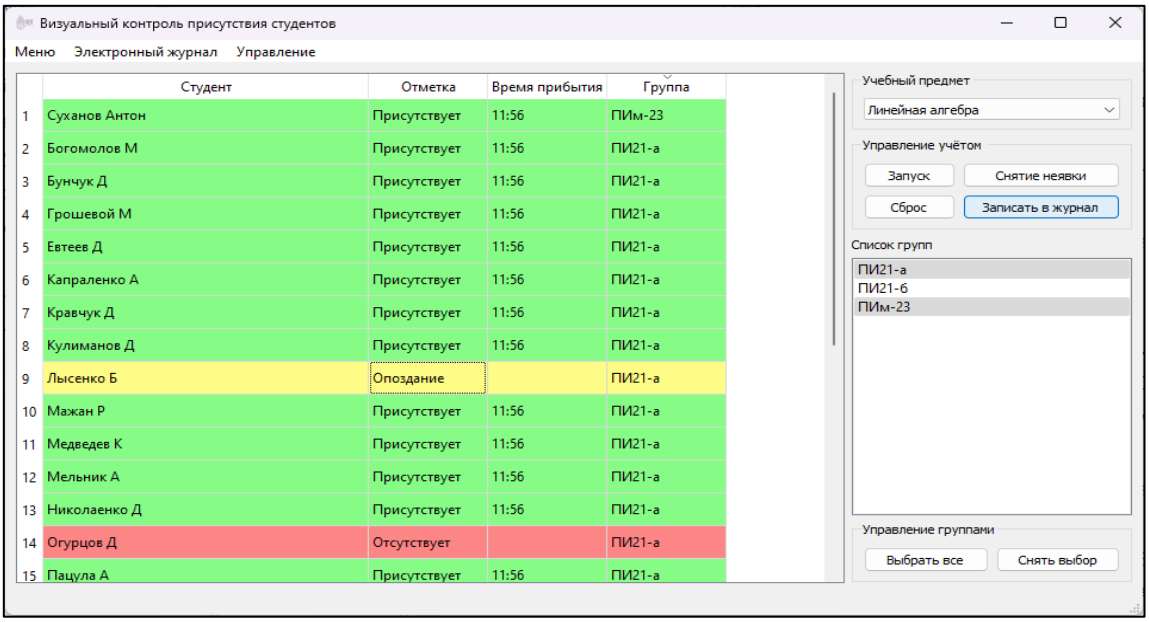



Рисунок 4 – Диалоговое окно преподавателя при видеорегистрации студентов

По фотографиям студентов этой группы система успешно распознало 18 и не распознало 4 студентов, т к. их фотографии были затемнены или же имели маленькое разрешение и плохое качество.

В последующих двух экспериментах количество распознаваемых студентов увеличивалось. Одна группа включала 47 студентов, а другая – 69 студентов. Эксперименты проводились в аналогичных условиях. Результаты видеорегистрации представлены в третьей и четвёртой колонках табл. 2. Причина неудачного распознавания ряда студентов в этих группах состояла в том, что их фотографии были также затемнены и имели плохое качество картинки. Полученные результаты показывают, что, несмотря на некоторую искусственность условий проведения последних экспериментов, качество распознавания с помощью применяемой нейросети не снижается с ростом количества студентов до размера типового потока.

### *Выводы*

Произведена формальная постановка задачи распознавания лиц на основе предобученной нейронной сети. Представлена архитектура и математическое описание нейронной сети VGGFace. Использование такой нейросети облегчает настройку системы на видеоконтроль людей благодаря упрощению формирования датасет. Также предложена функциональная схема видеорегистрации студентов при входе в аудиторию. Изображения лиц поступают с видеокамеры в режиме реального времени. Экспериментальные исследования были ориентированы на оценку влияния различных видов помех на качество распознавания с учётом реального объёма видеоданных по студентам потока. Часть экспериментов проведена по фотографиям реальных студентов в количестве 69 человек.

### *Литература*

1. Федяев О.И., Коломойцева И.А. Автоматическая регистрация присутствия студентов на учебном занятии с помощью компьютерного зрения // XXI Национальная конференция по искусственному интеллекту с международным участием КИИ-2023 (Смоленск, 16-20 октября 2023 г.). Труды конференции. В 2-х томах. Т.1. – Смоленск: Принт-Экспресс, 2023. – С. 294-303.

2. Суханов А.А., Ткачев Н.М., Федяев О.И. Визуальный контроль присутствия студентов в аудитории на основе глубокой нейронной сети // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2022): Сборник материалов IV Международной научно-практической конференции, Донецк, 29–30 ноября 2022 года. Том 1. – Донецк: Донецкий национальный технический университет, 2022. – С. 111-115.

3. Sukhanov A.A., Fedyaev O.I., Kaverina O.G. Young scientists' researches and achievements in science. // Научные исследования и достижения молодых ученых: материалы научно-практической конференции для молодых ученых Донецк, 20 апреля 2023 года / под общей редакцией Е.Н. Кушниренко. – Донецк: ДонНТУ, 2023. – С. 310 -318.

4. Антонио Джулли, Суджит Пал. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow / пер. с англ. Слинкин А.А. – М.: ДМК Пресс, 2018. – 294 с.

5. VGG Face Descriptor // robots.ox.ac.uk. [Электронный ресурс]. – Режим доступа: http://www.robots.ox.ac.uk/~vgg/software/vgg_face/ - Загл. с экрана.

6. K Wickrama Arachchilage, S. P. Deep-learned faces: a survey / S. P. K Wickrama Arachchilage, E. Izquierdo // Eurasip Journal on Image and Video Processing. – 2020. – Vol. 2020, No. 1. – P. 1-33.

7. Cornell University. Computer vision and Pattern Recognition. Very Deep Convolutional Networks for Large-Scale Image Recognition. Karen Simonyan, Andrew Zisserman [Электронный ресурс]. – Режим доступа: https://arxiv.org/abs/1409.1556 - Загл. с экрана.

8. Faris Abdlkader, D. Design and analysis of face recognition system based on VGGFace-16 with various classifiers / D. Faris Abdlkader, M. Faris Ghanim // IAES International Journal of Artificial Intelligence. – 2024. – Vol. 13, No. 2. – P. 1499.

9. Jason {osa-jima}. Convolutional Networks - VGG16 [Электронный ресурс]. – Режим доступа: https://www.jasonosajima.com/convnets_vgg.html - Загл. с экрана.

***Суханов А.А., Баев Д.Э., Федяев О.И. Экспериментальная оценка качества нейросетевого распознавания студентов по изображениям лиц из базы данных.*** *Рассматривается задача автоматизации видеорегистрации присутствия студентов на учебном занятии. Выполнена постановка задачи компьютерного распознавания и разработана функциональная схема системы. Описана архитектура нейронной сети VGGFace и её использование для распознавания лиц из видеопотока. Проведена экспериментальная оценка качества распознавания лиц в зависимости от различного вида помех, и от объёма видеоданных, определяемого количество студентов.*

***Ключевые слова:*** *компьютерное зрение, свёрточная нейронная сеть, распознавание лиц, видеопоток, регистрация студентов в аудитории, VGGFace16.*

***Sukhanov A.A., Baev D.E., Fedyaev O.I. Experimental evaluation of the quality of neural network recognition of students using face images from the database.*** *The problem of automation of video registration of students' presence at the training session is considered. The problem formulation of computer recognition is performed and the functional scheme of the system is developed. The architecture of VGGFace neural network and its use for face recognition from video stream is described. Experimental evaluation of the quality of face recognition depending on different types of interference and on the volume of video data determined by the number of students is carried out.*

***Keywords:*** *computer vision, convolutional neural network, face recognition, video stream, student registration in the classroom, VGGFace16.*

# СЕКЦИЯ 1. «СОВРЕМЕННЫЕ ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»

УДК 004.054

## Пишем эталонный код: подход и оценка

**К.К. Руднев[*1], Т.Г. Дмитрюк[*2]**

[*1] студент, Донецкий национальный технический университет,
thebooom@yandex.ru

[*2] ассистент, Донецкий национальный технический университет,
tnauka@lenta.ru, OrcID: 0000-0003-4803-5555, SPIN-код: 9249-2928

***Руднев К.К., Дмитрюк Т.Г. Пишем эталонный код: подход и оценка.*** *Данная работа посвящена анализу принципов и метрик, которые помогают разработчикам создавать качественный, поддерживаемый и масштабируемый код. Рассмотрена важность качественной и количественной оценок «чистоты кода» в работе программиста. Предложен и разработан интегральный показатель чистоты написания кода, учитывающий требования и ограничения конкретного проекта программного обеспечения.*

***Ключевые слова:*** *чистый код, метрики оценки кода, принципы программирования, программная инженерия.*

### *Введение*

Существуют сотни языков программирования, тысячи принципов и паттернов разработки программного обеспечения — и всё это применяется ситуативно. Каждый раз, когда разработчик видит перед собой новую задачу, первым делом, он анализирует все свои знания, умения и навыки и старается совершенствовать решение поставленных задач. Порой это получается не с первого раза, не со второго и даже не с третьего. Это приходит с опытом. Часто стоит посоветоваться и обсудить свое виденье решения с командой, услышать чужие мнения, критику — и еще раз всё обдумав, приступать к разработке проекта.

С большой долей вероятности, во время того, как нужно внедрить что-то новое в продукт, выбор языка программирования и фреймворка не будет сильно обширен, а будет строго ограничен тем стеком, который использует команда. Это упрощает задачу выбора решения разработчика, потому что, как правило, определены основной язык продукта и условный фреймворк, на котором будет выполнено решение. Это не относится к выбору базы данных или решения каких-то узконаправленных задач, где пользователь вынужден использовать тот или иной инструмент.

### *Постановка проблемы*

Актуальность данной проблемы заключается в потребности формального определения чистоты написанного разработчиками кода. Проблематика заключается в том, что перед проверкой другим специалистом код никак не проверяется программно, что позволило бы превентивно избежать лишних затрат времени на ревью. Целью исследования является формулирование постулатов и принципов написания «хорошего» кода, а также разработка модели формулы для математической оценки «чистоты» кода.

### *Определение идеального кода*

Что такое идеальный код? Это код, который удобно читать и приятно дополнять. Несомненно, у каждого своё видение идеального и удобного кода. Роберт Мартин, создатель книги «Чистый код: создание, анализ и рефакторинг», считает, что чистый код — это тот, который можно без труда читать как книгу — сверху вниз [1, с. 61]. Умение писать опрятный и правильный код — это совершенствование навыков хорошего специалиста в сфере программирования. Можно считать, что эти навыки прокачиваются параллельно изучению языка, его тонкостей и новых функций. С каждым годом в языках программирования выходят новые методы и возможности, а старые, в свою очередь, посыпаются «синтаксическим сахаром» для упрощения чтения и разработки.

Основные тезисы написания понятного пользователю кода, во-первых, называть все вещи осмысленно, во-вторых, делать функции компактными и сосредоточенными на одной задаче за раз, и наконец, писать легкий для прочтения код. Вышеперечисленные и другие общепринятые догмы можно оспаривать, особенно, если они не соответствуют установленному стилю проекта или в корне противоречат восприятию разработчика.

***Классификация принципов программирования***

Существуют отдельные паттерны и принципы разработки программного обеспечения, следуя которым получится «хороший» код, который в дальнейшем будет качественно масштабироваться и поддерживаться в рабочем состоянии [1]. Эти качественные характеристики позволяют оформить

Например, принцип разработки «DRY» (Don't repeat yourself), что в переводе обозначает «не повторяйся». Это будто бы очевидно, ведь зачем писать то, что уже написано. Написанный код должен уметь быть переиспользованным, чтобы один метод мог быть использован в разных местах при разных условиях. Но вернемся назад и вспомним, что один метод может решать только одну задачу, по возможности, используя минимальное число аргументов.

Огромный свод ограничений и правил задает самый популярный принцип разработки — «SOLID». Рассмотрим отдельно каждый пункт принципа с примерами на языке C#.

1. S – Single Responsibility Principle – принцип единственной ответственности, где для каждого класса выделяется своя единая зона ответственности (см. рис. 1).

```
/// <summary>
/// Класс, без соблюдения принципа
/// </summary>

public class Invoice
{
    public void AddInvoice(InvoiceItem item)
    {
        // Логика добавления инвойса
    }

    public void DeleteInvoice(InvoiceItem item)
    {
        // Логика удаления инвойса
    }

    public void LogError(string message)
    {
        // Логика логирования ошибок
    }

    public void SendEmail(string recipient, string subject, string body)
    {
        // Логика отправки электронного письма
    }
}
```

```
/// <summary>
/// Классы, с разделенной ответственностью
/// </summary>

public class Invoice
{
    public void AddInvoice(InvoiceItem item)
    {
        // Логика добавления инвойса
    }

    public void DeleteInvoice(InvoiceItem item)
    {
        // Логика удаления инвойса
    }
}

public class ErrorLogger
{
    public void LogError(string message)
    {
        // Логика логирования ошибок
    }
}

public class EmailSender
{
    public void SendEmail(string recipient, string subject, string body)
    {
        // Логика отправки электронного письма
    }
}
```

Рисунок 1 – Пример работы принципа единственной ответственности

2. O – Open closed Principle – принцип открытости-закрытости, согласно которому классы должны иметь механизм модификации и расширения, но не изменения (см. рис. 2).

```
// 1. Определение базового класса
public abstract class Invoice
{
    public abstract void AddInvoice(InvoiceItem item);
}

// 2. Реализация класса
public class BasicInvoice : Invoice
{
    public override void AddInvoice(InvoiceItem item)
    {
        // Логика добавления инвойса
    }
}

// 3. Фабрика
public class InvoiceFactory
{
    public static Invoice CreateInvoice(string type, IInvoiceLogger logger,
IInvoiceEmailSender emailSender)
    {
        switch (type)
        {
            case "Basic":
                return new BasicInvoice(logger, emailSender);
            // Другие типы инвойсов здесь
            default:
                throw new ArgumentException("Неизвестный тип инвойса", nameof(type));
        }
    }
}
```

Рисунок 2 – Пример реализации принципа открытости-закрытости

3. L – Liskov substitution Principle – принцип подстановки Барбары Лисков. При разработке должна быть возможность заменить родительский класс классом наследником, чтобы при этом работа программы не изменилась (см. рис. 3).

```
// Рассмотрим подобный пример класса, который можно заменить вместо родительского
public class InvoiceForAccounting : Invoice
{
    public override void AddInvoice(InvoiceItem item)
    {
        // Измененная логика работы метода
        emailSenderService.SendEmail("foo", "foo", "foo");
    }

    public override void DeleteInvoice(InvoiceItem item)
    {
        // Измененная логика работы метода
        emailSenderService.SendEmail("foo", "foo", "foo");
    }

    // Остальные методы можно оставить без изменений
}
```

Рисунок 3 – Пример реализации принципа подстановки Барбары Лисков

4. I – Interface Segregation Principle – принцип разделения интерфейсов. Данный принцип устанавливает правило, согласно которому не нужно заставлять клиента (класс) реализовывать интерфейс, который не имеет к нему отношения (см. рис. 4).

```
public interface IInvoiceManager
{
    void AddInvoice(InvoiceItem item);
    void DeleteInvoice(InvoiceItem item);
}

public class Invoice : IInvoiceManager
{
    public void AddInvoice(InvoiceItem item)
    {
        // Логика добавления инвойса
    }

    public void DeleteInvoice(InvoiceItem item)
    {
        // Логика удаления инвойса
    }
}
```

Рисунок 4 – Пример реализации принципа разделения интерфейсов

5. D – Dependency Inversion Principle – принцип инверсии зависимостей. Модули зависят исключительно от абстракций, а сами абстракции не должны зависеть от деталей. Детали зависят от абстракций (см. рис. 5).

```
// Выносим все в интерфейсы (абстракции) и зависим только от них, а не от реализаций
public interface IEmailService
{
    void SendEmail(string recipient, string subject, string body);
}


public class EmailService : IEmailService
{
    public void SendEmail(string recipient, string subject, string body)
    {
        // Реализация отправки электронного письма
    }
}

public class Invoice
{
    private readonly IEmailService _emailService;

    public Invoice(IEmailService emailService)
    {
        _emailService = emailService;
    }

    public void AddInvoice(InvoiceItem item)
    {
        // Логика добавления инвойса
        _emailService.SendEmail("recipient@example.com", "Инвойс", "Инвойс добавлен");
    }

    public void DeleteInvoice(InvoiceItem item)
    {
        // Логика удаления инвойса
        _emailService.SendEmail("recipient@example.com", "Инвойс", "Инвойс удален");
    }

    public void SendEmail(string recipient, string subject, string body)
    {
        // Логика отправки электронного письма
        _emailService.SendEmail(recipient, subject, body);
    }
}
```

Рисунок 5 – Пример реализации инверсии зависимостей

Придерживаясь правил вышеизложенных принципов, разработчик программного обеспечения создаёт код понятным и читаемым, более адаптивным и менее связным. Главная причина написания хорошего кода — это не его красота или размер, а сложность поддержки. Представим ситуацию, что нужно написать сложный сервис для расчета аналитических данных. Разработка по условию проводится в течение месяца, и в результате написан не код, а целое произведение искусства — все кратко и понятно, скорость работы сервиса поражает, сам он покрыт тестами и сложные куски бизнес-логики задокументированы. По истечении года источник выгрузки для аналитики изменился и специалисту требуется внести правки в уже работающую систему - задача не из сложных. Однако в ходе анализа кода приходит осознание, что базовые принципы были проигнорированы и код получился чересчур связным и практически невозможно настроить новые модели исходных значений без изменения уже существующего кода (нарушение принципа открытости-закрытости), что приводит к усложнению задачи относительно человеческих ресурсов и, скорее всего, ухудшения качества кода.

Помимо вышеупомянутых SOLID и DRY, хотелось бы осветить ещё несколько самых популярных принципов — это «KISS» и «YAGNI».

YAGNI (аббревиатура в переводе «Это вам не понадобится») — обозначает, что стоит писать только тот код, который будет использован. Это уменьшает вес проекта и его захламлённость. Не стоит также писать код, который возможно будет нужен разработчику через неделю.

KISS (аббревиатура в переводе «Будь проще») — обозначает, что нужно быть проще, ведь чем проще код — тем проще с ним работать.

Принципы программирования можно сравнить с законами, нарушение которых влечёт за собой некоторые последствия. Если их обобщить вышеупомянутые принципы, то они все гласят выполнять всё максимально просто и таким образом, чтобы не нужно было переделывать.

Для увеличения роли понимания кода между разными программистами (и соблюдение вышеупомянутых принципов) были придуманы паттерны проектирования, которых стоит придерживаться. Поставленную задач скорее всего уже кто-то решал, и в большинстве случаев для её решения был придуман шаблон. Рассмотрим на примере один из самых простых и самых прикладных шаблонов — «Посредник» («Mediator»). В его основе заложена инкапсуляция способов взаимодействия между модулями, что позволяет избавиться от прямой ссылки объектов друг на друга и, тем самым, уменьшает связность кода. К паттернам, в отличие от принципов, стоит подходить с меньшей серьезностью. После их изучения не стоит всё писать, используя только паттерны.

Таким образом, принципы программирования и паттерны написания кода в своей совокупности формируют качественные характеристики оформления разрабатываемого продукта, позволяющие создать для кода «красивую рамку» и хорошую репутацию для самого программиста.

***Комплексная оценка чистоты кода***

Ознакомившись с тем, что может придать программному коду пристойный вид, теперь стоит определиться с тем, как вообще оценивать его качество написания? К сожалению, нет единого ГОСТа оценки качества кода и каждый специалист, команда или анализаторы кода формируют свои требования и правила. Однако автором статьи предложена попытка собрать единый стандарт чистоты написания кода, который можно оценить количественно, что в последствии позволит сформулировать «эталонный» код. Рассмотрим некоторые количественные характеристики оценки качества кода, которые по мнению автора дадут наиболее полную информацию для разрабатываемого «эталона».

1. **Удобство поддержки кода** (U) — показатель, который подразумевает под собой индекс простоты выполнения дальнейшей поддержки программистом написанного кода (представлен формулой (1)):

$$U = Max(0, (171 - 5{,}2\ln(V) - 0.23D - 1.62\ln(Cl)) * 100/171, \quad (1)$$

где функция *Max(0, ...)* означает, что результат не может быть меньше 0; *ln(V)* – натуральный логарифм объема Хэлстеда; *D* – цикломатическая сложность кода; *ln(Cl)* - натуральный логарифм количества строк кода; *100 / 171* – коэффициент для масштабирования результата в диапазон значений от 0 до 100.

Диапазон значений индекса: от 0 до 100. Чем ниже индекс, тем труднее разработчику поддерживать код [2].

2. **Цикломатическая сложность** (D) — условная метрика читаемости кода [3], которая вводит штрафы от 1 до 100 единиц за каждый прерывающий исполнение кода оператор. Чем меньше штрафов, тем меньше вложенность и читабельнее код. Подсчитывается он согласно (2):

$$D = Min\left(\frac{Dc+Di+Dr+Dl}{Dsum}, 100\right), \quad (2)$$

где $D_c$ – циклические конструкции *(for, while, do-while):* +1 за каждый цикл; $D_i$ – прерывающие действия *(break, continue, return):* +1 за каждое прерывающее действие; $D_r$ – рекурсия*:* +1 за каждый рекурсивный вызов; $D_l$ – логические операторы*:* +1 за каждую последовательность логических операторов (&&, ||), но последовательные операторы одного типа считаются как один; $D_{sum}$ – сумма логических операторов кода в одном файле.

3. **Метрики Хэлстеда** представляют собой набор количественных характеристик, которые используются для оценки сложности и других свойств программного кода [4]. **Объем Холстеда** описывает объём программы в контексте теории сложности алгоритмов и считается по формуле (3):

$$V = \text{Min}(N\log(2)n, 100), \quad (3)$$

где *N* – общее количество операторов и операндов в программе; *n* – количество различных операторов и операндов.

4. **Процент покрытия кода тестами** – показатель доли того, какая часть исходного кода была охвачена тестами (формула (4)).

$$T = \frac{T_l}{Cl}100\%\,, \quad (4)$$

где: *Tl* – общее количество покрытых кодом строк; *Cl* – общее количество строк кода.

Эта характеристика помогает оценить качество комплекта тестов и определить, какие части кода остались непроверенными. Мы считаем её в процентах от 0 до 100, где 100% − это идеальный, полностью покрытый тестами код.

5. **Процент работоспособности кода** – показатель доли, какая часть написанных тестов успешно выполняется (5).

$$W = \frac{T_s}{T_c}100\%\,, \quad (5)$$

где $T_s$ – общее количество успешных тестов; $T_c$ – общее количество тестов.

6. **Процент портативности кода** – показатель доли кода, которая запускается на всех платформах (Windows, MacOS, Linux) (6).

$$Pp = \frac{Pdc+Cc+Oc}{3} 100\% , \quad (6)$$

где *Pdc* – процент платформо-зависимого кода; *Cc* – количество файлов конфигурации умножить на число платформозависимых параметров и разделить на 100%; *Oc* – процент успешных тестов на всех системах.

7. **Процент зависимости кода** – показатель доли кода, который вызывает функции из других программных модулей (7).

$$Pd = \frac{Cd}{Cl} 100\% , \quad (7)$$

где *Cd* – число зависимых строк кода; *Cl* – число всех строк кода.

8. **Процент повторяющегося кода** – показатель доли, который повторяется (8).

$$Pr = \frac{Cr}{Cl} 100\% \quad (8)$$

где *Cr* – число зависимых строк кода; *Cl* – число всех строк кода.

Эта характеристика — негативная, имеющая обратную корреляционную связь, стоит избегать её роста.

***Анализ оценки***

Исходя из логики способа оценивания рассмотренных количественных характеристик, ниже в таблице 1 предложены их предельные граничные значения, нормированные значения и удельные веса оценок.

Таблица 1 – Количественные характеристики показателей чистоты кода

| **Оценка** | **Границы** | **Нормированное значение** | **Вес** |
|---|---|---|---|
| Удобство *(U)* | (0, 100) | 0.8 | 0.15 |
| Цикломатическая сложность *(D)* | (0, 100] | 0.12 | 0.20 |
| Объем Хэлстеда *(V)* | (0, 100] | 0.5 | 0.20 |
| Процент покрытия тестами *(T)* | (0, 100) | 0.9 | 0.15 |
| Процент работоспособности *(W)* | (0, 100) | 1 | 0.10 |
| Процент портативности *(Pp)* | (0, 100) | 0.8 | 0.05 |
| Процент зависимости *(Pd)* | (0, 100) | 0.6 | 0.05 |
| Процент повторений *(Pr)* | (0, 100) | 0.15 | 0.10 |

Нормированные значения, представленные в таблице, соответствуют среднему значению нормального распределения данной величины. Вес оценки определялся из субъективных соображений, как мера важности. Границы были заданы исходя из семантики формулы и определены как максимальные и минимальные значения коэффициентов.

Классификация показателей чистоты кода позволяет разработать модель индекса чистоты кода. Опираясь на таблицу 1, представим статистическую модель в виде линейного полинома (см. рис. 6).

| | Коэффициент регрессии | Коэффициент значимости | Весовые коэффициенты |
|---|---|---|---|
| $c$ | -0.12143 | -0.97354 | 0.00000 |
| $x_1$ | 0.15664 | 1.10102 | 0.11907 |
| $x_2$ | 0.29007 | 3.14935 | 0.34057 |
| $x_3$ | 0.48122 | 2.08592 | 0.22557 |
| $x_4$ | 0.02938 | 0.21296 | 0.02303 |
| $x_5$ | -0.05665 | -0.50849 | -0.05499 |
| $x_6$ | 2.40041 | 1.77662 | 0.19213 |
| $x_7$ | 0.22061 | 0.12567 | 0.01359 |
| $x_8$ | 0.98383 | 1.30415 | 0.14103 |

Рисунок 6 – Коэффициенты линейного полинома

Обработка статистических данных позволила на этапе каскадно-регрессионного анализа рассчитать модель линейного вида интегрального показателя чистоты кода (9):

$$I_{cc} = -0.12 + 0.15U + 0.29D + 0.48V + 0.029T - 0.05W + 2.4Pp + 0.22Pd + 0.98Pr. \quad (9)$$

Уравнение вида (10) составлено из весовых коэффициентов, каждый его параметр указывает долю от общей оценки $\boldsymbol{I_{cc}}$:

$$\boldsymbol{I_{cc} = 0.11U + 0.34D + 0.22V + 0.2T - 0.05W + 0.19Pp + 0.01Pd + 0.14Pr}\ . \quad (10)$$

Полученное уравнение характеризуется коэффициентами значимости $t_j$, показывающими степень влияния каждого фактора на отклик, в соответствии с которыми может быть сформулирован ряд значимости (11):

$$\boldsymbol{D > V > T > W > Pr > U > Pd > Pp}\ . \quad (11)$$

Наибольший вклад в формирование индекса чистоты кода $I_{cc}$ составляют показатели: «Цикломатическая сложность кода», «Объём Хэлстеда» и «Процент покрытия кода» программы.

Оценки линейного полинома представляют следующие характеристики полученной модели.

1) Величина дисперсии остаточной (масштабированной) $S_{1z}^2 = 0.037$ свидетельствует о том, что погрешность модели не превышает и 5%. Модель достаточно точно предсказывает зависимую переменную по заданным значениям независимых переменных.

2) Отношение Фишера $F_1 = 18$ (должно удовлетворять условию $F_1 \geq 1{,}5$) показывает во сколько раз полученная зависимость лучше полинома $V = V_{cp}$ (где $V_{cp}$ — математическое ожидание V).

3) Коэффициент множественной детерминации R = 0.962 характеризует степень близости полученного уравнения к функциональной зависимости.

Оценки уравнения характеризуют модель как адекватную.

***Вывод***

Проведенный анализ принципов программирования позволил сформировать постулаты и принципы написания «хорошего» кода. Выявлены основные проблемы на ревью кода. Разработана модель и формулы для математической оценки «чистоты» кода в качестве основы для будущих исследований по подготовке разработки средств автоматизированного анализа программного кода.

Научная новизна представленной работы состоит в систематизации качественной оценки и формализации впервые разработанной модели количественной оценки чистоты написания кода. Практическая значимость исследования заключается в возможности использования формулы интегрального показателя расчета чистоты кода в практике программистов для получения прогнозных значений оценок написания кодов с целью самоконтроля проделанной работы.

### *Литература*

1. Мартин Р. С. Чистый код: Создание, анализ и рефакторинг / Р. С. Мартин, 2013. – 464 с.
2. Принципы SOLID на примерах [Электронный ресурс] – Режим доступа: https://habr.com/ru/articles/688530. — Загл. с экрана.
3. Метрики кода — диапазон индексов удобства поддержки и их значения [Электронный ресурс] – Режим доступа: https://learn.microsoft.com/ru-ru/visualstudio/code-quality/code-metrics-maintainability-index-range-and-meaning. — Загл. с экрана.
4. Цикломатическая сложность [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Цикломатическая_сложность — Загл. с экрана.
5. Применение метрик Холстеда для количественного оценивания характеристик программ ЭВМ [Электронный ресурс] – Режим доступа: https://cyberleninka.ru/article/n/primenenie-metrik-holsteda-dlya-kolichestvennogo-otsenivaniya-harakteristik-programm-evm — Загл. с экрана.

***Руднев К.К., Дмитрюк Т.Г. Пишем эталонный код: подход и оценка.*** *Данная работа посвящена анализу принципов и метрик, которые помогают разработчикам создавать качественный, поддерживаемый и масштабируемый код. Рассмотрена важность качественной и количественной оценок «чистоты кода» в работе программиста. Предложен и разработан интегральный показатель чистоты написания кода, учитывающий требования и ограничения конкретного проекта программного обеспечения.*

***Ключевые слова:*** *чистый код, метрики оценки кода, принципы программирования, программная инженерия.*

***Rudnev K.K., Dmitriuk T.G. Writing the reference code: approach and evaluation.*** *This work is devoted to the analysis of principles and metrics that help developers create high-quality, supported and scalable code. The importance of qualitative and quantitative assessments of "code purity" in the work of a programmer is considered. An integral indicator of the purity of writing code is proposed and developed, taking into account the requirements and limitations of a specific software project.*

***Keywords:*** *clean code, code evaluation metrics, programming principles, software engineering.*

УДК 004.054

# Разработка прокси-API с поддержкой in-memory кеширования на Node.js

**С.Н. Евтушенко[*1], Т.Г. Дмитрюк[*2]**

[*1] студент, Донецкий национальный технический университет,
s.scorpi-on@ya.ru

[*2] ассистент, Донецкий национальный технический университет,
tnauka@lenta.ru, OrcID: 0000-0003-4803-5555, SPIN-код: 9249-2928

***Евтушенко С.Н., Дмитрюк Т.Г. Разработка прокси-API с поддержкой in-memory кеширования на Node.js.*** *В статье приведён системный анализ предметной области для разработки программного интерфейса, расширяющего возможности публичного API сервиса IMDb. Рассмотрено влияние кеширования на производительность API. Представлено обоснование выбора стека технологий для разработки, исходя из требований предполагаемого технического задания. Разработанное API поддерживает все возможности донорского API, CRUD-операции с кешем и управление его размером. Документация к API реализована при помощи набора инструментов Swagger.*

***Ключевые слова:*** *API, прокси, кеш, системный анализ, Node.js, JavaScript, Express, CRUD, Swagger.*

## *Введение*

В современном веб-пространстве кеширование играет ключевую роль в обеспечении высокой производительности и отзывчивости приложений. С увеличением объема данных и числа пользователей, которые одновременно обращаются к веб-сервисам, необходимость в использовании эффективных механизмов кеширования становится все более актуальной. Кеширование позволяет значительно сократить время отклика, уменьшить нагрузку на серверы и оптимизировать использование сетевых ресурсов.

В рамках данной работы поставлена цель разработать программный интерфейс, который расширяет возможности публичного API сервиса IMDb, поддерживает сохранение ответов сервера в кеш, а также управление этим кешем. В числе требований указаны адаптация под высокую масштабируемость и скорость внедрения изменений.

В соответствие с поставленной целью предполагается решение следующих задач:

- провести системный анализ требований к API и возможностей донорского API;
- выбрать соответствующий стек технологий для разработки с учетом ограничений по времени;
- реализовать API и написать документацию к нему.

## *Системный анализ предметной области*

Существует несколько публичных API для работы с IMDb. В настоящей работе предложено использовать сервис OMDb API, который достаточно быстро работает и выполняет две основные функции:

- поиск фильмов по названию и году выхода в прокат;
- получение информации о конкретном фильме по названию или идентификатору.

Функционал также позволяет передавать параметры страницы, формата и длины ответа сервера. Обязательной является передача авторизационного токена, который необходим для защиты от спама и без которого запросы к API возвращают ошибку. Поддерживаемые GET-запросы осуществляются по единому адресу [1] и подразумевают передачу параметров в URL-адресе. На рисунке 1 изображена выдержка из официальной документации, описывающая параметры запросов к донорскому API.

By ID or Title

| Parameter | Required | Valid Options | Default Value | Description |
|---|---|---|---|---|
| i | Optional* | | <empty> | A valid IMDb ID (e.g. tt1285016) |
| t | Optional* | | <empty> | Movie title to search for. |
| type | No | movie, series, episode | <empty> | Type of result to return. |
| y | No | | <empty> | Year of release. |
| plot | No | short, full | short | Return short or full plot. |
| r | No | json, xml | json | The data type to return. |
| callback | No | | <empty> | JSONP callback name. |
| v | No | | 1 | API version (reserved for future use). |

*Please note while both "i" and "t" are optional at least one argument is required.

By Search

| Parameter | Required | Valid options | Default Value | Description |
|---|---|---|---|---|
| s | Yes | | <empty> | Movie title to search for. |
| type | No | movie, series, episode | <empty> | Type of result to return. |
| y | No | | <empty> | Year of release. |
| r | No | json, xml | json | The data type to return. |
| page New! | No | 1-100 | 1 | Page number to return. |
| callback | No | | <empty> | JSONP callback name. |
| v | No | | 1 | API version (reserved for future use). |

Рисунок 1 — Параметры запросов, с которыми работает донорское API

Согласно требованиям к программному продукту, разрабатываемое API должно дублировать вышеуказанный интерфейс, но с условием кеширования ответов сервера. Вообще кеширование — это процесс хранения копий данных в более быстрой доступности для уменьшения времени отклика при их извлечении [2]. Внедрение кеширования в рассматриваемой задаче необходимо для уменьшения количества запросов к серверам донорского API и снижения нагрузки на него. На рисунке 2 представлен принцип работы кеширования запросов к API.

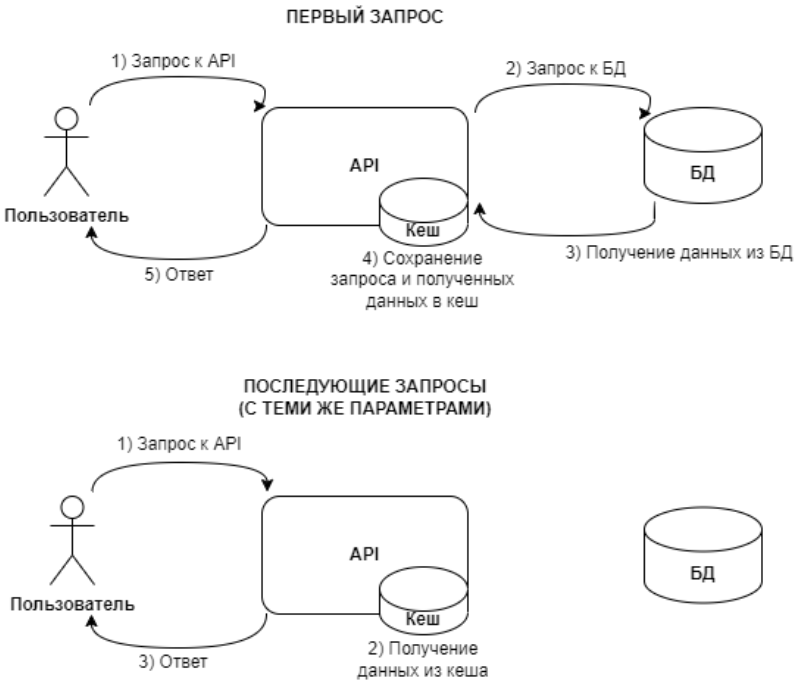



Рисунок 2 — Схема работы кеширования API

Из этого примера видно, что только первый запрос данных приводит к обращению к базе данных (БД). Все последующие запросы с теми же параметрами обращаются к кешу до тех пор, пока данная запись не будет из него удалена. Важно отметить, что параметры в URL могут передаваться в любом порядке, а значит смена этого порядка не должна влиять на кеш.

Кеширование по месту хранения данных подразделяется на два типа:

- кеширование на стороне клиента: данные хранятся в браузере пользователя, что позволяет избежать повторных запросов к серверу при повторном обращении к тем же данным;

- кеширование на стороне сервера: данные хранятся на сервере, что позволяет ему быстро предоставлять их пользователям без необходимости повторных обращений к базам данных или внешним API.

В данной работе реализовано кеширование на стороне прокси-сервера. Это позволяет не только снизить нагрузку на API в количестве запросов, но и снять с него расходы на кеширование. Так как обычный пользователь не связан с донорским API напрямую, решение создать расширенный клон данного API на собственном сервере является оптимальным для решения поставленной задачи.

Одним из распространённых способов хранения кешируемых данных является in-memory кеширование. Этот подход предполагает хранение данных в оперативной памяти сервера. В числе основных преимуществ такого подхода можно выделить:

- высокую скорость доступа (данные, хранящиеся в памяти, могут быть извлечены значительно быстрее, чем данные, хранящиеся на диске или получаемые через сетевые запросы);

- простоту реализации (in-memory кеширование легко интегрируется в существующие приложения и не требует сложной настройки);

- поддержку различных стратегий кеширования (in-memory кеши могут поддерживать различные стратегии управления данными, такие как LRU (Least Recently Used), TTL (Time to Live) и другие, что позволяет эффективно управлять памятью).

Рассмотрим также недостатки in-memory кеширования:

- ограниченный объем оперативной памяти в (при больших объемах данных кеш может не вмещать всю необходимую информацию);

- опасность потери данных (в случае сбоя системы или перезагрузки сервера данные в кеше могут быть потеряны, если не предусмотрены механизмы для их восстановления);

- слишком примитивный принцип работы (более продвинутое управление кешем может требовать модификации системы кеширования и усложнения всего программного продукта).

Эти особенности хорошо коррелируют с потребностью в высокой скорости внедрения изменений, поэтому в рамках данной работы выбор пал именно на in-memory кеширование.

### *Выбор инструментария для разработки*

Разработка API является типовой задачей для веб-разработчика. Соответственно, для её решения существует множество инструментов с различными характеристиками, преимуществами и недостатками. В качестве стандартного выбора для реализации API в соответствии с поставленными требованиями была предложена платформа Node.js. Это платформа с открытым исходным кодом для работы с языком JavaScript, позволяющая запускать программный код вне браузера. Рассмотрим её особенности, которые будут полезны при разработке, и сравним их с аналогами.

Во-первых, Node.js основан на неблокирующей модели ввода-вывода, что позволяет обрабатывать множество запросов одновременно. Это особенно важно для прокси-API, который должен эффективно управлять большим количеством входящих запросов и обеспечивать быструю реакцию на них. В отличие от традиционных серверных платформ, таких как Java или Ruby, где каждый запрос может блокировать поток, Node.js позволяет избежать таких узких мест, что значительно повышает производительность приложения.

Во-вторых, так как Node.js базируется на JavaScript, это делает его особенно привлекательным для разработчиков, работающих как на стороне клиента, так и на стороне сервера. Это позволяет использовать единый стек технологий, что упрощает процесс разработки и уменьшает время на обучение. Тогда как другие языки программирования не обладают такой универсальностью и требуют знания нескольких технологий.

Кроме того, экосистема Node.js располагает огромным количеством библиотек и модулей, которые могут быть использованы для реализации различных функций, включая кеширование. Это позволяет быстро интегрировать необходимые решения и сосредоточиться на бизнес-логике приложения, а не ручной разработке его инфраструктуры. В этом плане с Node.js может сравниться, например, язык Python, также обладающий большим набором, в частности, веб-инструментов.

Помимо Node.js как платформы для разработки, следует выбрать и веб-фреймворк. В этом контексте одним из наиболее подходящих вариантов является широко распространённый Express [3]. Его популярность

среди разработчиков объясняется тем, что Express предлагает минималистичный и гибкий подход к созданию веб-приложений, что позволяет быстро разрабатывать и настраивать API. В отличие от более сложных фреймворков, таких как NestJS или Koa, Express предоставляет разработчикам возможность сосредоточиться на реализации бизнес-логики, не отвлекаясь на сложные конфигурации. Это особенно важно в условиях ограниченного времени, когда необходимо быстро реализовать функционал, не предполагающий сложных взаимодействий.

Также стоит отметить, что Express имеет богатую экосистему модулей и плагинов, что позволяет легко интегрировать дополнительные функции, такие как аутентификация, логирование и обработка ошибок. Это дает возможность быстро расширять функциональность приложения без необходимости писать всё с нуля. Так, хотя механизм in-memory кеширования и возможно реализовать стандартными средствами Node.js, в работе предложен к использованию плагин apicache, который позволяет максимально упростить настройку кеширования ответов API на Express, а также поддерживает автоочистку кеша и его хранение в Redis [4]. На рисунке 3 представлен пример минимальной программы на Express, реализующей кеширование ответов API.

```
const express : e | () => core.Express = require('express');
const apicache : ApiCache | {...} = require('apicache');

const app : any | Express = express();
const port : number = 3000;
let cache = apicache.middleware;
app.use(cache( strDuration: '5 minutes'));

app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : void => {
    res.send( body: 'Hello World!');
});

app.listen(port, hostname: () : void => { new *
    console.log(`Example app listening on port ${port}`);
});
```

Рисунок 3 — Пример минимальной программы на Express с кешированием ответов API на 5 минут

Для обеспечения масштабируемости кода необходимо вести разработку не только быстро, но и качественно. Код в итоговом программном продукте должен быть легко читаем и соответствовать принятым во фреймворке паттернам дизайна (например, модульной архитектуре). Что касается стека, хорошей практикой является внедрение в проект линтера, который отвечает за форматирование кода и позволяет поддерживать кодовую базу в едином стиле по настраиваемым правилам. Для JavaScript стандартным решением является использование ESLint, который, помимо кастомизации, предоставляет возможность использовать уже встроенный рекомендованный набор правил. Пример минимальной конфигурации ESLint с подключением таких правил приведён на рисунке 4.

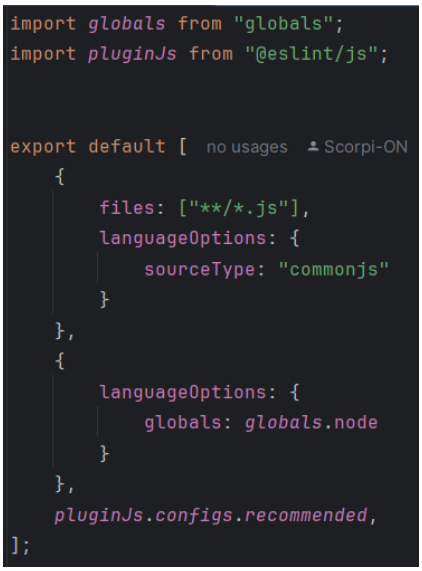



Рисунок 4 — Пример минимального конфига ESLint с подключённым рекомендованным набором правил

Документирование веб-API в наше время выходит за рамки написания текстового описания. Для этой задачи существуют продвинутые инструменты, которые не только наглядно представляют возможности API, но и позволяют тестировать его в удобном графическом интерфейсе. Одним из технических решений является открытый набор инструментов Swagger. Его главное достоинство в том, что он обладает большой популярностью, поддерживая интеграцию с большинством веб-инструментов. Присутствует возможность описывать конфигурацию как в отдельных файлах (JSON, YAML), так и непосредственно в коде (требует подключения плагина интеграции). Придерживаясь принципов разделения кода и единой ответственности, хранение конфигурации в файле YAML было предпочтительным.

В качестве системы контроля версий был выбран Git как наиболее часто используемое среди разработчиков решение, которое позволит гибко управлять версиями проекта и при необходимости организовывать командную работу. Также Git поддерживает популярный хостинг IT-проектов GitHub, на котором может быть опубликован разрабатываемый продукт в открытом доступе.

### *Описание программного продукта и его основных алгоритмов*

При проектировании API в соответствии с предполагаемым техническим заданием были выделены следующие эндпоинты:

- документация:
  - GET / — перенаправление на репозиторий проекта в GitHub;
  - GET /docs — документация в Swagger UI;
- непосредственно API:
  - GET /api:
  - без параметров — перенаправление на /docs;
- с параметрами — передача параметров в запрос к донорскому API (автоматически кешируется, если не превышен максимальный размер кеша);
  - PUT /api — обновление кешированного запроса (если он присутствует в кеше);
  - DELETE /api — удаление запроса из кеша (если он присутствует в кеше);
- управление кешем:
  - GET /cache — получение кешированных запросов;
  - GET /cache/maxsize — получение максимального размера кеша;
  - PUT /cache/maxsize — установка максимального размера кеша (не менее 1);
  - DELETE /cache/clear-all — полная очистка кеша.

Конечная структура проекта, реализующего API с таким функционалом, изображена на рисунке 5.

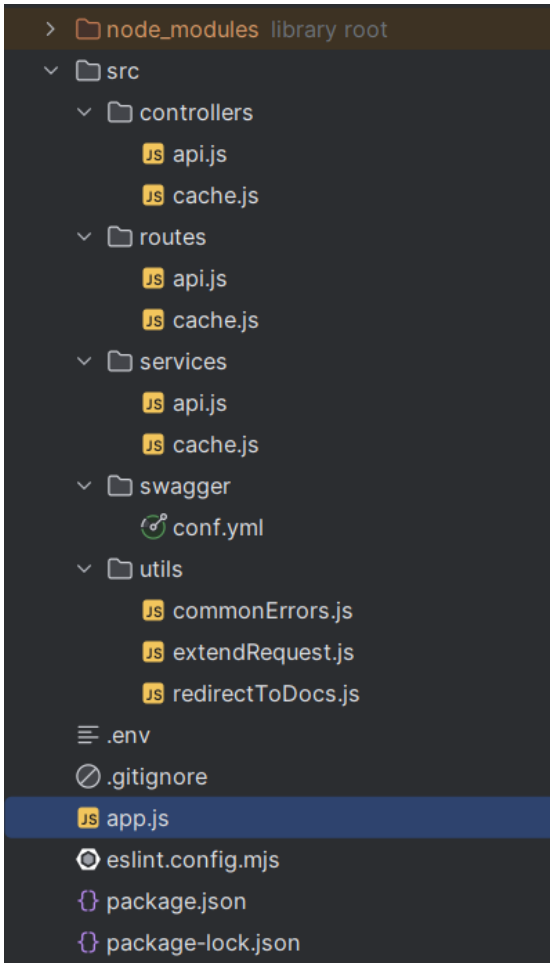



Рисунок 5 — Структура разработанного проекта

Рассмотрим ключевые компоненты системы. Важнейшим из них является package.json. В экосистеме Node.js он представляет собой центральный репозиторий настроек для инструментальных средств, используемых в проекте. В нём описываются, например, имя, версия и тип пакета, используемые в нём команды-скрипты, а также пакеты зависимостей и их версии. package-lock.json фиксирует полную информацию о зависимостях, включая адреса их загрузки, точные версии и хеши. Это нужно для того, чтобы на разных устройствах у разработчиков был один и тот же набор версий пакетов во избежание их конфликтов и несовместимостей. Сами зависимости скачиваются пакетным менеджером в папку node_modules, которую принято не включать в репозиторий, добавляя в .gitignore — служебный файл системы контроля версий Git, позволяющий указывать файлы и директории, которые Git не должен индексировать. Файл eslint.config.mjs содержит конфигурацию ESLint, использующую рекомендованные правила линтинга для JavaScript. Файл .env содержит необходимые для работы проекта переменные окружения. В нашем проекте целесообразно сохранить в этом файле хост и порт для URL-адреса API, ссылку на донорское API, а также репозиторий проекта на GitHub для перенаправления. Для загрузки переменных окружения в программу используется специальная библиотека dotenv.

Код, отвечающий непосредственно за реализацию поставленной задачи, хранится в папке src. Точкой входа в приложение является файл app.js, в котором импортируются необходимые модули и настраивается экземпляр Express. На рисунке 6 приводится содержание этого файла в разрабатываемом проекте.


Рисунок 6 — Содержание файла app.js

Во фреймворке Express существует несколько уровней архитектуры приложения, реализующих паттерны MVC и Middleware. Они помогают организовать код и разделить ответственность между различными компонентами. В работе использованы некоторые из них: роутеры (подключаются к экземпляру Express выше), контроллеры и сервисы и middleware. Ниже рассмотрим каждый элемент подробнее.

Роутеры в Express отвечают за определение маршрутов (routes) и обработку входящих HTTP-запросов. Они позволяют организовать и структурировать код, разделяя логику обработки запросов по различным URL-адресам. Роутеры могут быть созданы с помощью метода express.Router(), который позволяет группировать маршруты, относящиеся к определенной части приложения. В нашем примере используются два основных роутера: api (работа с запросами к донорскому API) и cache (управление кешем). На рисунке 7 приведён код первого из них.


Рисунок 7 — Код роутера api, реализующего работу с запросами к донорскому API

В роутерах определяются типы запросов и относительные пути обращения к ним, потом к ним привязываются действия — контроллеры. Контроллеры представляют собой функции или классы, которые содержат логику обработки запросов, поступающих на определенные маршруты. Они отвечают за взаимодействие с моделями данных, выполнение бизнес-логики и формирование ответов для клиента. Контроллеры помогают отделить логику обработки запросов от определения маршрутов, что делает код более чистым и поддерживаемым. В данной работе контроллеры созданы для каждого запроса из роутеров. На рисунке 8 представлен код контроллера updateItem, который используется роутером api для обновления ответа сервера на запрос в кеше.

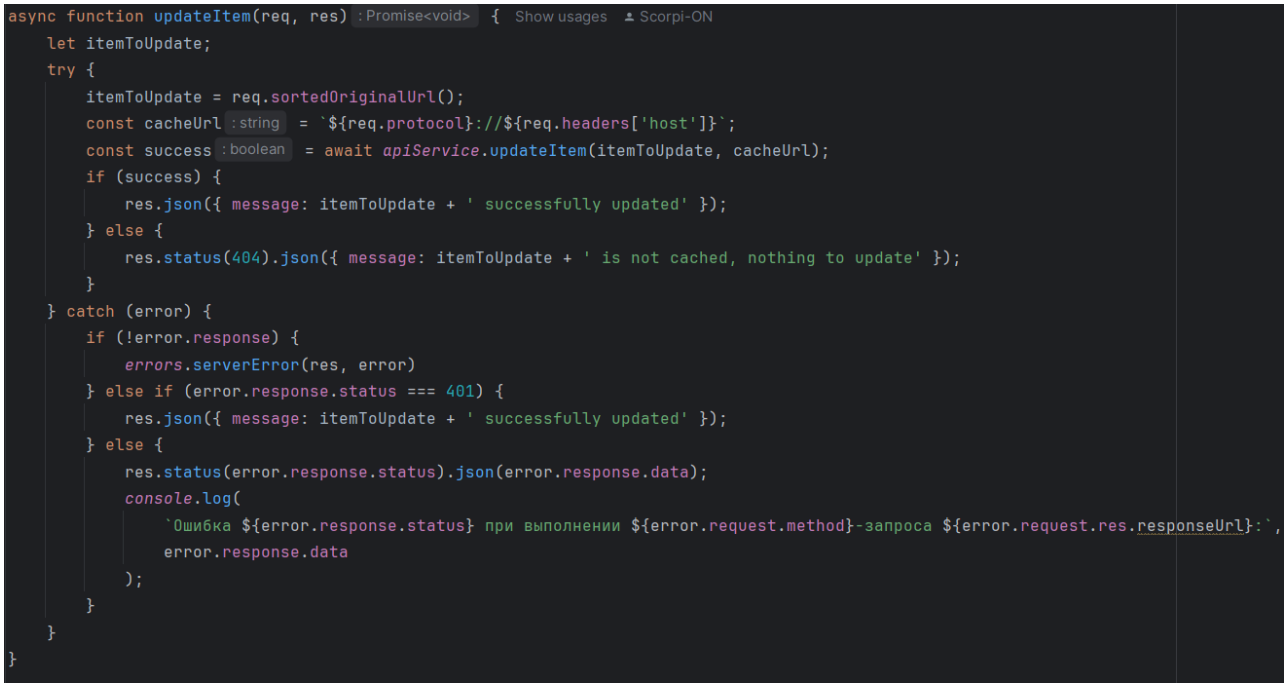



Рисунок 8 — Код контроллера updateItem, отвечающего за обновление ответа сервера на запрос в кеше

Сервисы представляют собой уровень абстракции, который используется для организации бизнес-логики и взаимодействия с внешними ресурсами, такими как базы данных или сторонние API. Сервисы помогают отделить логику работы с данными от контроллеров, что делает код более модульным и тестируемым. Они могут содержать функции для выполнения операций, таких как создание, чтение, обновление и удаление данных (CRUD). Рассмотрим подробнее сервис кеширования, код которого представлен на рисунке 9. Он представляет собой класс, модифицирующий библиотеку apicache под нужды разрабатываемого продукта. Помимо добавления необходимых свойств-геттеров, в проекте реализованы ограничение на размер кеша по количеству сохранённых записей и возможность регулировать это ограничение (свойство maxSize). Таких возможностей нет в оригинальной версии библиотеки apicache, поэтому использование собственного сервиса позволяет устранить связанные с этим неудобства.


Рисунок 9 — Код сервиса кеширования ApiCache, расширяющего возможности библиотеки apicache

Кроме того, в проекте присутствует директория utils, в которой находятся функции общего назначения для использования в различных местах кодовой базы. Наибольший интерес для нас представляет файл extendRequest.js, в котором реализована модификация поведения стандартного для Node.js класса IncomingMessage. Код данной утилиты приведён на рисунке 10. При её импорте модифицируется стандартный класс IncomingMessage модуля node:http, добавляя ему метод sortedOriginalUrl. Благодаря последнему можно получить URL-адрес запроса с отсортированными в алфавитном порядке запросами. Так как библиотека apicache кеширует ответы API именно по URL запроса, при кешировании существует возможность подменить его отсортированным вариантом (что и происходит в вышеописанном сервисе ApiCache).

```
const {IncomingMessage} = require("node:http");

Object.defineProperty(IncomingMessage.prototype, p: "sortedOriginalUrl", attributes: {
    value: function () :string {
        const sortedKeys :string[] = Object.keys(this.query).sort();
        const sortedParams :string = sortedKeys.map(
            key :string => `${key}=${this.query[key]}`
        ).join('&');
        return `${this.baseUrl}?${sortedParams}`;
    },
});
```

Рисунок 10 — Код утилиты extendRequest, модифицирующей сохранение порядка параметров в HTTP-запросе

В директории swagger находится YAML-конфиг conf.yaml, в котором приведено описание разрабатываемого API в нотации OpenAPI 3.1 [5]. Конфиг позволяет отобразить документацию к API в интерактивном веб-интерфейсе Swagger. На рисунках 11 — 13 приведены экранные формы разработанного интерфейса с описанием возможностей разработанного API.

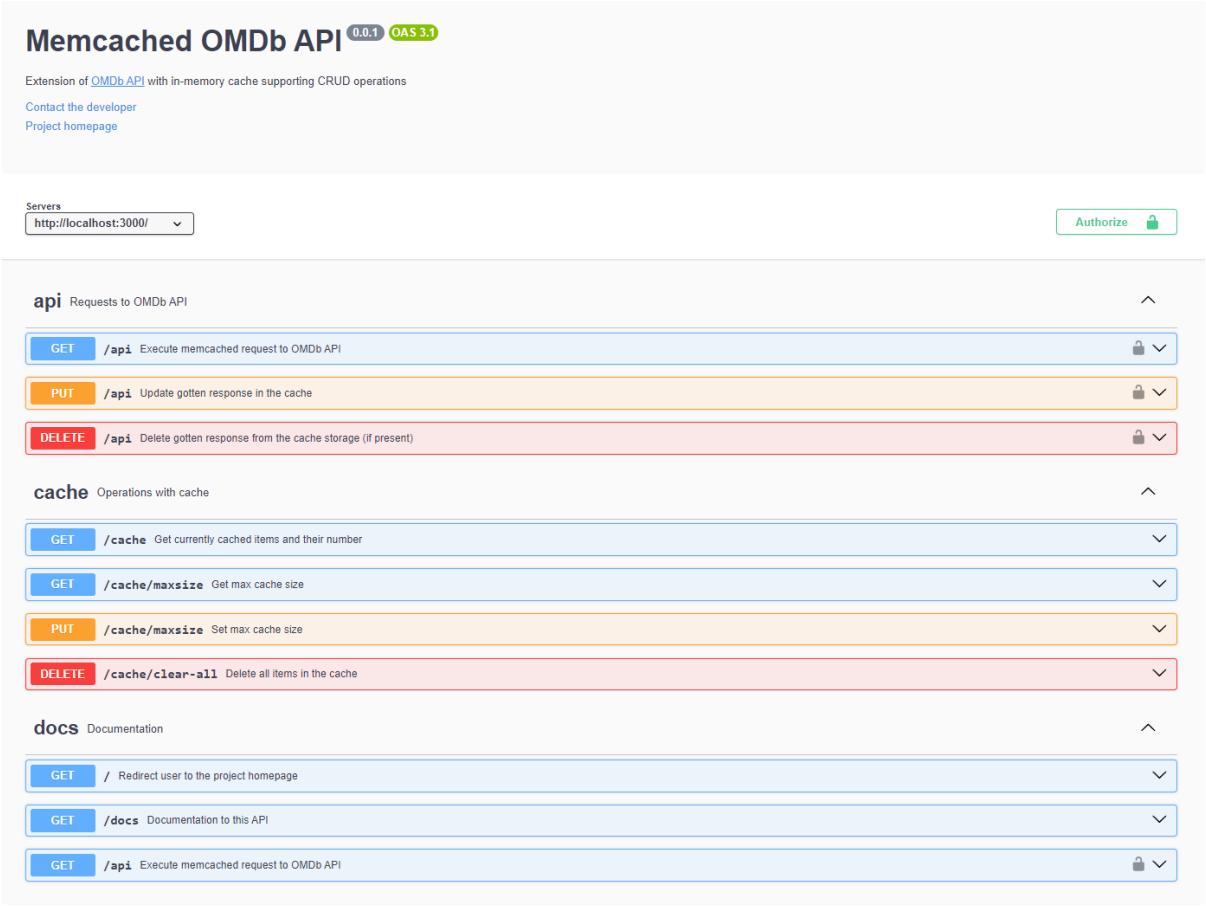



Рисунок 11 — Главная страница Swagger UI со всеми реализованными эндпоинтами

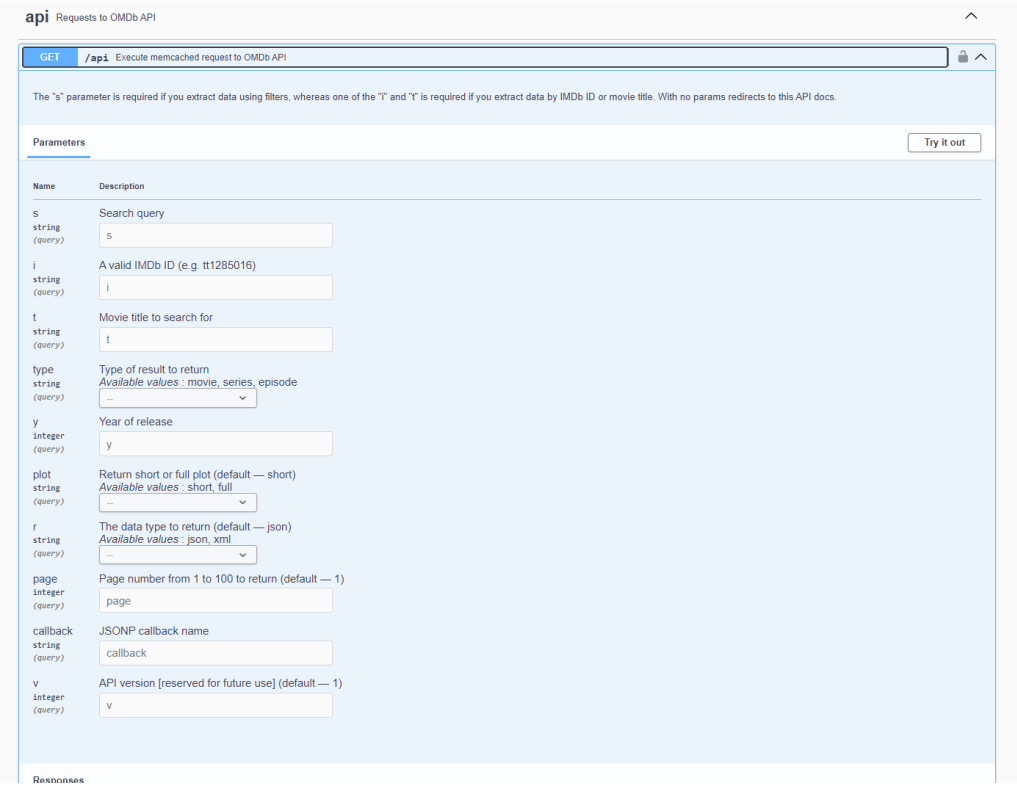



Рисунок 12 — Форма заполнения данных для тестирования запроса по маршруту /api

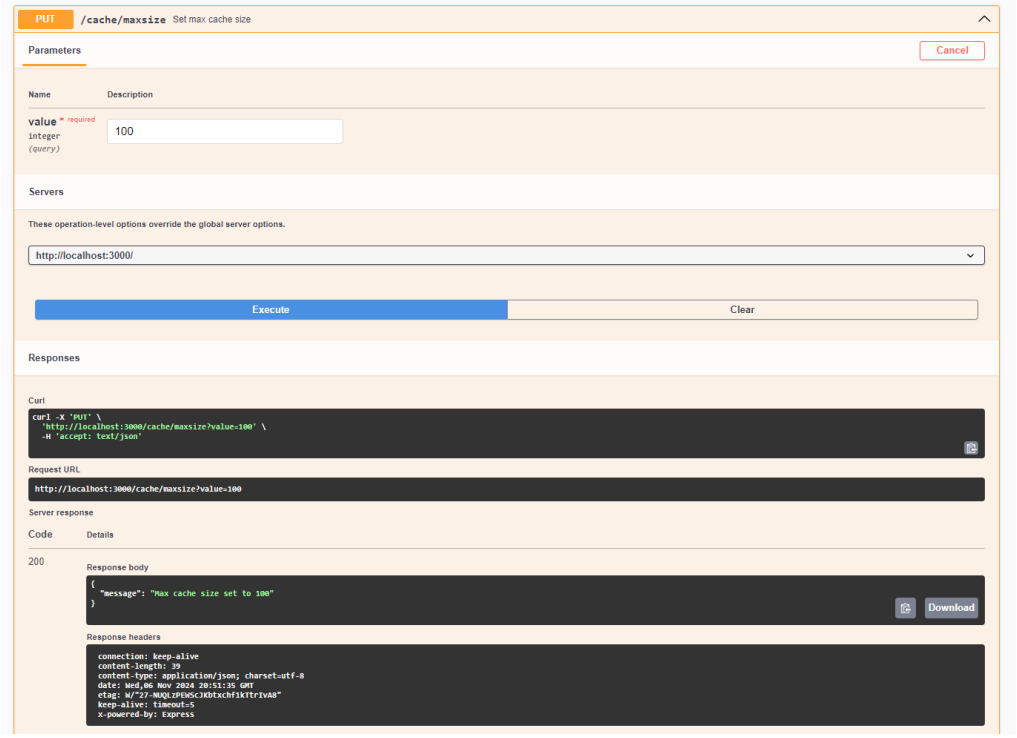



Рисунок 13 — Успешно выполненный PUT-запрос по маршруту /cache/maxsize с увеличением максимального размером кеша до 100 единиц

Таким образом, благодаря созданной документации можно быстро ознакомиться с форматом запросов к API и протестировать их выполнение с корректной передачей необходимых параметров вплоть до авторизации с

токеном. Это особенно полезно при дальнейшей разработке проекта и привлечении к работе над проектом других людей.

Разработанный проект имеет открытый исходный код и доступен на GitHub [6].

### *Вывод*

В рамках данной статьи был разработан проект API, который расширяет возможности публичного API сервиса IMDb, поддерживает сохранение ответов сервера в кеш, а также управление этим кешем. Разработка ориентирована на высокую масштабируемость и высокую скорость внедрения изменений.

В процессе разработки была подробно рассмотрена предметная область и нюансы поставленной задачи. Изучены и использованы такие технологии как платформа Node.js, фреймворк Express, библиотека apicache и набор инструментов Swagger.

### *Литература*


1. OMDb API - The Open Movie Database [Электронный ресурс] – Режим доступа: http://www.omdbapi.com/ – Загл. с экрана
2. [По полочкам] Кэширование [Электронный ресурс] – Режим доступа: https://habr.com/ru/articles/734660/ – Загл. с экрана
3. Express web framework (Node.js/JavaScript) [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs – Загл. с экрана
4. kwhitley / apicache [Электронный ресурс] – Режим доступа: https://github.com/kwhitley/apicache/tree/master – Загл. с экрана
5. Swagger (OpenAPI 3.0) [Электронный ресурс] – Режим доступа: https://habr.com/ru/articles/541592/ – Загл. с экрана
6. Scorpi-ON / Memcached-OMDb-API-entrance-exam [Электронный ресурс] – Режим доступа: https://github.com/Scorpi-ON/Memcached-OMDb-API-entrance-exam – Загл. с экрана



***Евтушенко С.Н., Дмитрюк Т.Г. Разработка прокси-API с поддержкой in-memory кеширования на Node.js.*** *В статье приведён системный анализ предметной области для разработки программного интерфейса, расширяющего возможности публичного API сервиса IMDb. Рассмотрено влияние кеширования на производительность API. Представлено обоснование выбора стека технологий для разработки, исходя из требований предполагаемого технического задания. Разработанное API поддерживает все возможности донорского API, CRUD-операции с кешем и управление его размером. Документация к API реализована при помощи набора инструментов Swagger.*



***Ключевые слова:*** *API, прокси, кеш, системный анализ, Node.js, JavaScript, Express, CRUD, Swagger.*



***Evtushenko S.N., Dmitriuk T.G. Development of proxy-API with in-memory caching support on Node.js.*** *The article presents a system analysis of the subject area for the development of a program interface that extends the capabilities of the public API of IMDb service. The influence of caching on API performance is considered. Justification of the choice of the technology stack for development based on the requirements of the proposed specification is presented. The developed API supports all features of the donor API, CRUD-operations with cache and cache size management. API documentation is implemented using Swagger toolkit.*



***Keywords:*** *API, proxy, cache, system analysis, Node.js, JavaScript, Express, CRUD, Swagger.*

УДК 004.825

# Исследование удалённого использования онтологического инжиниринга

**Д.А. Филипишин**[*1]

[*1] ассистент кафедры ПИ им. Л.П. Фельдмана, аспирант, Донецкий национальный технический университет, domaco@mail.ru

***Филипишин Д.А. Исследование удалённого использования онтологического инжиниринга.*** *Анализ современного объектно-ориентированного программирования показал всё большее движение к метапрограммированию и стремление к удалённому выполнению программного кода, а классификация используемых сущностей в той или иной мере стремится к структуре онтологии. Что позволило бы значительно упростить технологию клиентской части с помощью переноса вычислений и логики на серверную часть, а также внедрить логические операции при работе с объектами в программном коде. В связи с этим актуальной задачей является исследование удалённого использования онтологического инжиниринга с целью выявления его характерных особенностей.*

***Ключевые слова:*** *онтологии, онтологический инжиниринг, метаэвристика, GPRC, умные шаблоны, глитч.*

## *Введение*

Несмотря на достигнутые в последние годы успехи в безопасности и удобстве языков программирования для реализации программных систем и сервисов в различных предметных областях, по-прежнему футуристичной, кажется, идея частичной подгрузки или модификации «на лету» уже скомпилированных программ. Данная идея не нова, с ней знаком каждый, кто смотрел футуристические фильмы с участием передовых технологий звездолётов или иных программных систем будущего.

Основные трудности данного подхода, которые необходимо было бы преодолеть для реализации подобной программной системы или сервиса, это ограничения возможностей «столпов» программирования (минимум топ-10 популярных языков программирования). Например, расширить шаблонный механизм классов до уровня модификации шаблона класса и последующего вызова конструктора для создания его объекта (умные шаблоны), внедрить систему онтологического инжиниринга, в которой классы и создаваемые объекты систематизировать во внутренней онтологии путём генерации на этапе компиляции для возможности группировать обрабатываемые сущности на уровне машинного кода и выполнять логические операции над ними подобно существующим онтологическим системам.

В настоящее время большие перспективы в решении данных проблем связывают с применением технологии GRPC (Remote Procedure Calls) системой удалённого вызова процедур с открытым исходным кодом, метаэвристического подхода к программированию (выполнение кода на основе правил вывода, И-ИЛИ деревьев), онтологическим инжинирингом и так называемыми «глитчами» или ошибками в выполнении программного кода, вызываемые явным (вручную) или неявным способом.

Целью исследования является анализ использования удалённого использования онтологического инжиниринга на основе принципов метаэвристических методов, а также выявление преимуществ относительно технологии программирования на сегодняшний день.

## *Онтологический инжиниринг в ООП*

Существует такое направление, как онтологический инжиниринг (ОИ), который может рассматриваться как система формальных определений в конкретной предметной области и представленный в виде предлагаемого редактора онтологий с физической семантикой. Физическая семантика, прежде всего, свойственна задачам САПР, предполагающих наличие CAD/CAM/CAE подсистем [1].

Онтология (в информатике) – это попытка всеобъемлющей и детальной формализации некоторой области знаний с помощью концептуальной модели.

Онтологический анализ – разделение реального мира на составляющие и классы объектов, определение их онтологий или же совокупности фундаментальных свойств, которые определяют их изменения и поведение.

Интерес к разработке онтологий плавно нарастает как со стороны разработчиков интеллектуальных систем, так и со стороны бизнес-аналитиков. Усилия исследователей в основном были направлены на разработку технологических инструментов и примеров. Однако разработка практических онтологий в производстве, проектировании и менеджменте, особенно ИТ-менеджменте, остаётся скорее на уровне «искусства» [2].

Редакторами или конструкторами онтологий называют инструментальные программные средства, созданные специально для проектирования, редактирования и анализа онтологий. Основная функция любого редактора онтологий состоит в поддержке процесса формализации знаний и представлении онтологии как спецификации [3].

Редактор онтологий с физической семантикой представляет собой комплексную программную систему, в которой классы онтологии представлены программными классами подобно объектно-ориентированному языку программирования, например, C++, т.к. не подразумевается точка входа внутри одного из классов подобно C# или Java. Главным отличием редактора с физической семантикой от ныне существующих аналогов (см. рис. 1) является наличие у каждого из классов конструктора, методов и свойств, которые позволяют выполнять арифметические операции подобно реальным вычислениям моделируемого процесса с уже имеющимися сущностями (индивидуалами), а также сразу перед созданием сущности (индивидуала) вызывая конструктор.

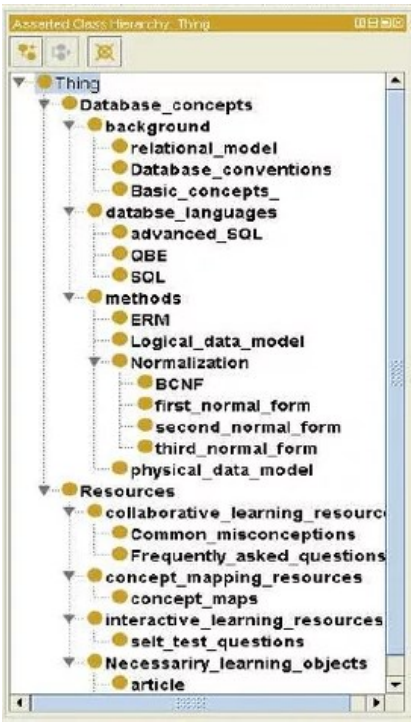



Рисунок 1 – Пример иерархии классов онтологии в Protege

Особым отличием от существующей парадигмы объектно-ориентированного программирования является новый механизм виртуального конструктора, который был частично реализован только в языке программирования Delphi, но не получил должного развития. Его суть в создании шаблона, инструменты которого позволяют модифицировать шаблон класса в зависимости от технического задания, после чего шаблон класса подобный template формирует необходимые параметры свойств и методов. И уже после этого вызывается конструктор для формирования сущности (индивидуала), который, в частности, можно применять для формирования массива сущностей, что невозможно для большей массы в ныне существующих редакторах онтологий (каждый индивидуал создаётся отдельно, после чего описывается класс или определение, которое присваивает ему необходимые свойства).

Данный редактор, в виду программной основы, описывает парадигму «закрытого мира», что позволяет значительно сузить логику работы с проектируемой онтологией и моделировать решение простых физических процессов в объектно-ориентированном виде на уровне физических токов и потенциалов.

Предложенный метод онтологического инжиниринга при внедрении в семантику одного из наиболее популярных языков программирования, например, C++, значительно расширил бы возможности по систематизации и обработке объектов и классовых типов данных (структура, класс, объединение, перечисление, пространство имён, интерфейс). Также модернизировать работу с указателями на объекты и ресурсы, их методы и свойства, путём генерации общего хранилища всех создаваемых текущим процессом объектов и получения адреса сущности по его признаку, например, имени класса или потомка, наличии константы или её значения в определённом диапазоне и т.п.

Данный подход позволил бы частично или полностью отказаться от нынешней системы управления указателями на объекты, перенеся всю логику в единое ссылочное хранилище (онтологию сущностей текущего процесса), в которое помещались бы метаданные каждого создаваемого объекта автоматически. Описанное преимущество потребовало бы создание нового инструмента, позволяющего работать с самим ссылочным хранилищем и преодоления всех последующих трудностей его внедрения, тестирования и обеспечения безопасности во все существующие программные системы и сервисы.

### *Метаэвристический подход в программировании*

Для развития онтологического инжиниринга в объектно-ориентированном программировании можно использовать метаэвристические методы, как инструмент дополнительной инкапсуляции при выполнении кода с целью соблюдения уровня конфиденциальности, а также как механизм самостоятельного выполнения технического задания системой с возможностью порождения необходимых сущностей и связей.

Метаэвристика — это независящая от задачи алгоритмическая структура высокого уровня [4], которая предоставляет набор рекомендаций или стратегий для разработки алгоритмов оптимизации. Существуют различные метаэвристические алгоритмы: генетические алгоритмы [5], оптимизация роя частиц [6], генетическое программирование [7], дифференциальная эволюция [8] и жадная рандомизированная адаптивная процедура поиска [9], и это лишь некоторые из них. Вот некоторые примеры недавних работ: алгоритмы DE и искусственной пчелиной колонии (ABC) использовались для определения кинетических параметров при гидрировании CO2 для производства углеводородов [10]; PSO использовался для калибровки параметров модели анаэробного сбраживания [11]; а GRASP использовался для прогнозирования маршрута после стихийного бедствия для многопериодной задачи планирования работы войск на примере города Порто-Пренс на Гаити после землетрясения 2010 года [12]. Дополнительную информацию о метаэвристических алгоритмах можно найти в следующих работах [13-16].

Наиболее подходящим метаэвристическим алгоритмом для поставленной задачи является генетическое программирование с применением И-ИЛИ-НЕ деревьев, что позволит инкапсулировать конфиденциальные данные от пользователей без необходимого уровня допуска, а также генерировать решения в соответствии техническому заданию порождая необходимые для вычислений сущности и связи без участия пользователя.

Генетическое программирование – автоматическое создание или изменение программ с помощью генетических алгоритмов. С помощью этой методологии «выращиваются» программы, всё лучше и лучше решающие поставленную задачу.

Основная идея состоит в построении программ путём применения генетических алгоритмов к некоторой модели вычисления. При этом разработчику программы остаётся лишь задать оценочную функцию, определяющую для каждого возможного результата вычисления в выбранной модели численное значение, называемое его пригодностью.

Операциями генетического программирования являются отбор наиболее приспособленных программ для размножения (кроссинговера), репликация и/или мутация в соответствии с заранее определённым показателем пригодности.

В качестве моделей вычисления в генетическом программировании чаще всего используют деревья, графы, команды процессора — «низкоуровневые» модели, имеющие ограниченный набор элементарных операций.

### *Глитч-эффекты и глитч-арт*

Манипулирование данными цифрового файла — это процесс целенаправленного изменения его содержимого, приводящий к искажениям, так называемым "глитчам". Эти искажения могут проявляться по-разному, в зависимости от типа файла и метода манипуляции. Один из способов создания глитчей – это так называемый "сбой смещения", возникающий при попытке открыть файл, предназначенный для одной программы, с помощью программы, предназначенной для работы с другим типом файлов. В этом случае программа некорректно интерпретирует данные, что и приводит к визуальным или звуковым артефактам.

Глитч (или глитч-эффект) — это явление непреднамеренного искажения цифрового изображения или звука в результате ошибки в работе электронного устройства. Такие глюки могут возникать в различных цифровых форматах, например, в фотографиях, видео, музыке или компьютерных программах.

Глитч-арт (искусство ошибки, цифровые помехи) — изобразительное искусство, выразительными средствами которого являются различные цифровые и аналоговые ошибки, например, такие как артефакты сжатия, баги, разрушение цифрового кода или физическое манипулирование электронными устройствами. Произведения глитч-арта демонстрируются на выставках, посвящённых цифровому искусству (см. рис. 2).

Рисунок 2 – Демонстрация глитч-арта

Таким образом, исходя из описания цифровой ошибки, а именно особенности подмены данных в связи с системными сбоями, можно применять этот механизм явно (с сервера), модифицируя работающую компьютерную программу налету.

На сегодняшний день понятие «глитч» проникло практически во все цифровые технологии.

Всё, чего в музыке, по идее, не должно быть – это глитч.

Звук зажеванной пленки, радиопомеха, даже диджейские звуки трения дисков. Истинно талантливый артист даже из пустоты сотворит искусство, вот музыканты и решили взять эти ошибки на вооружение, чтобы сконструировать нечто необычное.

Базовое понимание «музыки шума» было заложено еще в начале XXI века, но популярным глитч стал только в 90-е. Поначалу глитч создавался вручную: японский музыкант Ясунао Тоне наклеивал на CD-диски кусочки скотча, чтобы проигрыватель не мог спокойно прочитать записанную на них информацию. По мере развития технологий у музыкантов появилась возможность генерировать шумы самостоятельно и даже выстраивать из них полноценные композиции. Сейчас глитч используется во всех направлениях музыки: есть глитч-поп, глитч-рэп. Глитч-рок также называют нойз-роком. Чаще всего его можно услышать в электронике и её производных. Также сюда можно отнести понятие «перкуссия».

Перкуссия в музыке — это общее обозначение разных ударных инструментов, звук из которых извлекается посредством потряхивания или удара (рукой или палочкой). Обычно имеются в виду инструменты, не входящие в состав ударной установки: бонго, конги, бубны; маракасы и прочие шейкеры, различные трещотки, вудблок, ковбелл, колокольчики и многие другие. Перкуссия играет важную роль практически во всех жанрах музыки: от классической до тяжёлой рок-музыки. Она используется для создания узнаваемых ритмических украшений в музыке, управления темпом, обогащения звучания композиций, подчёркивания их ритмики

### *Фреймворк для удаленных вызовов процедур gRPC*

gRPC (gRPC Remote Procedure Call) — это современный высокопроизводительный фреймворк для удаленных вызовов процедур, разработанный Google. gRPC позволяет клиентам и серверам общаться напрямую, используя протокол HTTP/2 и Protocol Buffers (protobuf) в качестве языка описания интерфейсов (IDL). Эта технология предоставляет возможность эффективного взаимодействия между различными компонентами распределенных систем, независимо от языка программирования.

gRPC был создан в 2015м году как дополнение к фреймворку RPC для соединения многочисленных архитектур микросервисов, созданных с использованием различных техник. gRPC изначально был тесно связан с основной инфраструктурой Google, но в итоге был сделан с открытым исходным кодом и стандартизирован для использования широкой общественностью. Технология пришла на замену REST API, потому что последний создавал большое количество метаданных, хотя и предлагал усовершенствованный формат для работы со многими приложениями, что также привело к появлению GraphQL от Facebook [17].

Protobuf или Protocol Buffers — буферный протокол сериализации (бинарного преобразования) структурированных данных. Поскольку данные переводятся в двоичную форму, а зашифрованные сигналы становятся меньше, разбор с помощью Protobuf использует меньше CPU энергии для gRPC. Поэтому даже на компьютерах со слабым CPU, таких как сотовые телефоны, сообщения отправляются быстрее с помощью gRPC.

В gRPC клиентское приложение может напрямую вызывать метод серверного приложения на другом компьютере, как если бы это был локальный объект, что упрощает создание распределенных приложений и служб. Как и во многих системах RPC (см. рис. 3), gRPC основан на идее определения службы, определяя методы, которые могут быть вызваны удаленно с их параметрами и типами возвращаемых данных. На стороне сервера сервер реализует этот интерфейс и запускает сервер gRPC для обработки клиентских вызовов. На стороне клиента есть заглушка (называемая на некоторых языках просто клиентом), которая предоставляет те же методы, что и сервер.

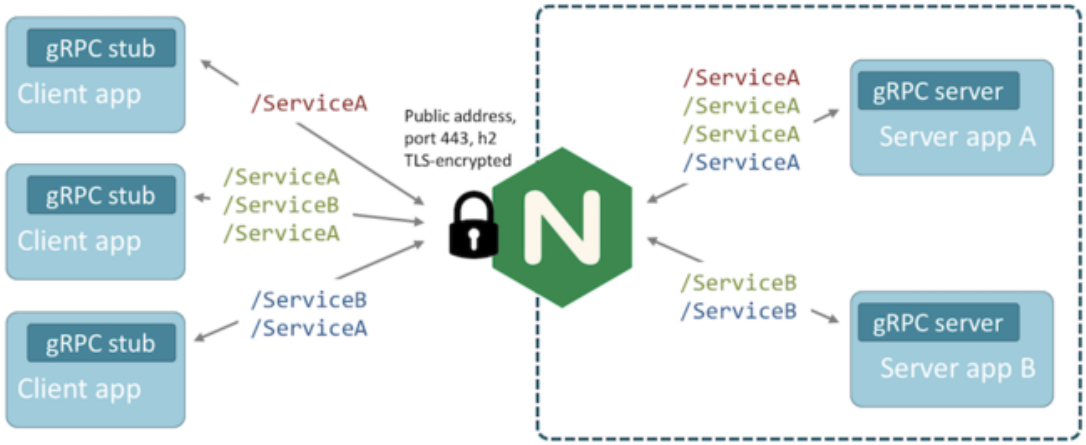



Рисунок 3 – Архитектура GRPC

Клиенты и серверы gRPC могут работать и взаимодействовать друг с другом в различных средах и могут быть написаны на любом из поддерживаемых языков gRPC. Это мощный инструмент для построения высокопроизводительных распределенных систем и микросервисов. Он обеспечивает эффективное общение между сервисами, поддерживает множество языков программирования и предлагает ясные и контрактно-ориентированные интерфейсы. Однако, как и любая технология, gRPC имеет свои недостатки и требует определенных усилий для освоения и интеграции.

Таким образом, в кооперативе с вышеописанными технологиями программисты фактически уже сегодня могут получить полностью динамическое программирование. Это когда структура исходного программного продукта написанного единожды может исполняться в несколько сотен или тысяч различных комбинаций с возможностью модификаций содержимого вовремя исполнения, а также порождать недостающие части по мере их необходимости.

### *Выводы*

В данной работе выполнен анализ удалённого использования онтологического инжиниринга на примере существующих программных инструментов. При выше выбранной комбинации технологий возможно уже сегодня значительно модернизировать парадигму объектно-ориентированного программирования с первостепенной целью качественно снизить порог вхождения для начинающих учеников и студентов, также увеличить сложность программных продуктов при этом уменьшив объём проектируемых модулей программы, которую необходимо описывать разработчику.

Онтологический инжиниринг позволяет визуально демонстрировать сущности и их связи, по примеру SADT/IDEF5 и диаграммы классов UML, графы. Ученикам и студентам позволяет «пощупать» или смоделировать понятия относительно друг друга с целью научиться понимать для чего в каждый конкретный момент будет создан тот или иной объект, что довольно тяжело сделать имеющимися на сегодняшний день учебными сервисами для учащихся и студентов со слабо развитым абстрактным мышлением, которые чаще всего нацелены либо на программный код, либо на визуальную составляющую.

Метод искусственных глитчей со стороны сервера или инициирующего фонового процесса позволяет отказаться от лицензионных dll или «программных бомб», которые заставляют пользователей приобрести лицензию или «тонко намекают» с помощью регулярно всплывающих сообщений. Метаэвристические алгоритмы позволяют разрабатывать программы, которые не только сами ищут решение, но и даже могут подсказывать специалисту ранее неизвестные пути решения в совершенно различных предметных областях.

Вышеописанные инструменты в кооперативе с технологией gPRC позволяют уже на сегодняшний день

приблизиться к программированию будущего, использовать процесс программирования и проектирования компьютерных программ подобно футуристическим фильмам, значительно усложнить программные комплексы и сервисы для более масштабных вычислений при этом упрощая труд разработчиков. Примером известных технологий, недоступных на сегодняшний день можно смело назвать: голограмма человека из фильмов о вселенной, телепортация или даже терминатор Т-1000 из фильма «терминатор 2», который получал карту визуального отображения всего лишь прикоснувшись к другому человеку (или по анализу ДНК проткнув человека, авторы досконально не раскрывают этот момент).

### *Литература*


1. Филипишин Д.А. Анализ применения редакторов онтологий с физической семантикой в педагогической деятельности вуза / Д.А. Филипишин, А.В. Григорьев, Е.И. Приходченко // Журнал «Информатика и кибернетика» №2 (32), 2023, Донецк, ДОННТУ. – с. 47 – 52.
2. Онтологический редактор Fluent Editor: учебно-методическое пособие к лабораторным работам / сост.: Н.М. Боргест, А.А. Орлова. – Самара: изд-во Самарского университета, 2017. – 44 с.
3. Шевченко, О.И. Технологии нестандартного обучения/ О. И. Шевченко, М. А. Волков, В. А. Леонов // Педагогика высшей школы – Казань, 2018 – № 3 (13) – С. 17-25.
4. Григорьев В.В. Вычислительная идентификация скоростей поверхностных реакций в масштабе пор: автореф. канд. ф-м наук, Якутск, 2022. – 142 с.
5. Goldberg D. E. Genetic algorithms. — Pearson Education India, 2006.
6. Wei Y., Qiqiang L. Survey on particle swarm optimization algorithm [j] // Engineering Science. — 2004. — Vol. 5, no. 5. — P. 87-94.
7. Sette S., Boullart L. Genetic programming: principles and applications // Engineering applications of artificial intelligence. — 2001.— Т. 14, № 6.— С. 727-736.
8. Storn R., Price K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces // Journal of global optimization. — 1997. — Т. 11, № 4. — С. 341-359.
9. Marques-Silva J. P., Sakallah K. A. GRASP: A search algorithm for propositional satisfiability // IEEE Transactions on Computers. — 1999. — Т. 48, №5.—С. 506-521.
10. Najari S., Grof G., Saeidi S. h gp. Modeling and optimization of hydrogenation of CO2: Estimation of kinetic parameters via Artificial Bee Colony (ABC) and Differential Evolution (DE) algorithms // International Journal of Hydrogen Energy. — 2019. — Т. 44, № 10. — С. 4630-4649.
11. Yang J., Lu L., Ouyang W. и др. Estimation of kinetic parameters of an anaerobic digestion model using particle swarm optimization // Biochemical Engineering Journal. — 2017. — Т. 120. — С. 25-32.
12. Barbalho T. J., Santos A. C., Aloise D. J. Metaheuristics for the work-troops scheduling problem // International Transactions in Operational Research. — 2020.
13. Liu Q., Li X., Liu H. и др. Multi-objective metaheuristics for discrete optimization problems: A review of the state-of-the-art // Applied Soft Computing. — 2020. — Т. 93. — С. 106382.
14. Karimi-Mamaghan M., Mohammadi M., Meyer P. и др. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art // European Journal of Operational Research. — 2022. — Т. 296, № 2. — С. 393-422.
15. Franco-Sepulveda G., Del Rio-Cuervo J. C., Pachon-Hernandez M. A. State of the art about metaheuristics and artificial neural networks applied to open pit mining // Resources Policy. — 2019. — Т. 60. — С. 125-133.
16. Mehrabi M., Pradhan B., Moayedi H. и др. Optimizing an Adaptive Neuro-Fuzzy Inference System for Spatial Prediction of Landslide Susceptibility Using Four State-of-the-art Metaheuristic Techniques // Sensors. — 2020. — Т. 20, № 6.
17. Индрасири Касун, Курупу Данеш gRPC: запуск и эксплуатация облачных приложений. Go и Java для Docker и Kubernetes. – СПб.: Питер, 2021. – 224 с.: ил. – (Серия «Бестселлеры O`Reilly»).



***Филипишин Д.А. Исследование удалённого использования онтологического инжиниринга.*** *Анализ современного объектно-ориентированного программирования показал всё большее движение к метапрограммированию и стремление к удалённому выполнению программного кода, а классификация используемых сущностей в той или иной мере стремится к структуре онтологии. Что позволило бы значительно упростить технологию клиентской части с помощью переноса вычислений и логики на серверную часть, а также внедрить логические операции при работе с объектами в программном коде. В связи с этим актуальной задачей является исследование удалённого использования онтологического инжиниринга с целью выявления его характерных особенностей*

***Ключевые слова:*** *онтологии, онтологический инжиниринг, метаэвристика, GPRC, умные шаблоны, глитч.*

***Filipishin D.A. Research on the remote use of ontological engineering.*** *The analysis of modern object-oriented programming has shown an increasing movement towards metaprogramming and the desire for remote execution of program code, and the classification of entities used tends to one way or another to the structure of ontology. This would significantly simplify the technology of the client side by transferring calculations and logic to the server side, as well as implement logical operations when working with objects in the program code. In this regard, an urgent task is to study the remote use of ontological engineering in order to identify its characteristic features.*

***Key words:*** *ontologies, ontological engineering, metaheuristic, GPRC, smart templates, glitch.*

УДК 004.06

# Исследование механизмов ускорения времени отклика современных web-приложений на основе микросервисной архитектуры

**В.В. Безуглый[*1], А.В. Боднар[*2]**

[*1] магистрант, Донецкий национальный технический университет,
vitaha.nik@mail.ru

[*2] доцент каф. ПИ, к.э.н., доцент, Донецкий национальный технический университет,
Linabykova13@ya.ru

***Безуглый В.В., Боднар А.В. Исследование механизмов ускорения времени отклика современных web-приложений на основе микросервисной архитектуры.*** *В статье проведён анализ популярных существующих принципов написания серверной части приложений, выполнено сравнение функционала, выполнено сравнение производительности решений.*

***Ключевые слова:*** *время отклика, микросервисы, web-приложения, производительность, асинхронность, конкурентность, балансировка нагрузки , масштабирование, кэширование, Redis, базы данных.*

## *Введение*

Современные веб-приложения требуют высокой производительности и устойчивости, особенно в условиях растущего количества пользователей и увеличения требований к функциональности. С появлением микросервисного подхода компании получили возможность создавать масштабируемые и гибкие системы, которые легко адаптируются под изменения. Go, как язык программирования, набирает популярность среди разработчиков микросервисов благодаря встроенной поддержке параллелизма, высокой производительности и легкости компиляции в статические бинарные файлы.Микросервисы часто связаны с проблемами коммуникаций, из-за которых возрастает время отклика. Изучение методов его ускорения помогает оптимизировать обмен данными между микросервисами и выбрать лучшие подходы, такие как кэширование, балансировка нагрузки и асинхронная обработка.Компании все чаще переходят к микросервисной архитектуре из-за ее гибкости и масштабируемости. Это особенно востребовано для крупных и быстро развивающихся приложений. В данной статье будет решена проблема выявления и решения проблем выбора средств разработки серверной части приложений.

Цель работы – исследовать основные механизмы и подходы для ускорения времени отклика микросервисных приложений на примере языка Golang, выявить преимущества и ограничения различных решений и предложить оптимальные практики для улучшения производительности.

Таким образом можно выделить задачи работы:

- провести глубокое изучение серверной части приложений;
- проанализировать существующие средства разработки серверной части приложения;
- провести анализ методик ускорения времени отклика клиентской части приложения;
- изучить кэширование запросов на примере популярных баз данных;
- разработать прототип серверной части приложения;
- выделить оптимальные средства разработки с целью ускорения работы приложения;

## *Асинхронность и конкурентность в Go*

Асинхронность и конкурентность – ключевые аспекты, влияющие на скорость работы микросервисов. Go использует концепцию конкурентности, реализованную с помощью горутин (goroutines). Конкурентность обеспечивает выполнение нескольких задач посредством переключения контекста [1].

В Go есть встроенный планировщик, который управляет горутинами и распределяет их выполнение между системными потоками. Это значит, что разработчику не нужно вручную управлять потоками — Go берет на себя оптимизацию параллельного выполнения кода. Быть кооперативным планировщиком означает, что планировщику нужны четко определенные события в пространстве пользователя, которые происходят в безопасных точках кода для принятия решений по планированию [2].

Горутины – это легковесные потоки выполнения в языке программирования Go, которые позволяют выполнять функции параллельно, не создавая при этом новые системные потоки. Они являются одной из главных особенностей Go, и их использование значительно упрощает написание конкурентного кода. Горутины занимают меньше ресурсов, чем обычные потоки. Вместо нескольких мегабайт, необходимых для обычных потоков, горутины используют стек размером всего в несколько килобайт, который может автоматически увеличиваться и уменьшаться в зависимости от потребностей.

### *Сравнение gRPC и REST API*

Для упрощения разработки серверной части используются средства для упрощенной передачи данных от сервера к клиенту, такие как gRPC и REST API. Сравнительная характеристика представлена в таблице 1.

Таблица 1 – Сравненительная характеристика gRPC и REST API

| Критерий сравнения | gRPC | REST API |
|---|---|---|
| Обработка запросов | Встроенная функция, несколько раз за вызов | Вручную, 1 раз за вызов |
| Скорость передачи | Сжатие в байтовый формат | Передача в виде читаемого JSON |
| Использование ресурсов системы | Оптимизированный подход | Классический подход |
| Трассировка | Встроенная функция, легко подключить | Вручную |
| Логирование | Встроенная функция, легко подключить | Вручную |
| Мониторинг | Встроенная функция, легко подключить | Вручную |
| Поддержка в языках программирования | Есть, но не в каждом | Нет |
| Простота разработки | Сложно | Просто |

Для повышения скорости межсервисной коммуникации в микросервисных системах часто используется gRPC, который обеспечивает более быструю и эффективную передачу данных по сравнению с традиционным REST (см. рис. 1). gRPC поддерживает двунаправленную потоковую передачу, что позволяет сократить задержки при работе с данными в реальном времени. Go предоставляет встроенную поддержку gRPC, что делает его подходящим для реализации высоконагруженных систем. И REST API, и RPC API отправляют и получают сообщения с использованием форматов обмена сообщениями JSON или XML [3]. Также стоит отметить использование современной версии HTTP/2 в gRPC.

В отличие от REST, gRPC - решение, которое не зависит от платформы и языка [4]. И благодаря HTTP/2 решает проблемы, связанные со скоростью передачи данных, их весом, в результате повышая эффективность обмена сообщениями.

| Method | IterationCount | Mean | Error | StdDev |
|---|---|---|---|---|
| RestGetSmallPayloadAsync | 100 | 11.832 ms | 0.2185 ms | 0.2044 ms |
| RestGetLargePayloadAsync | 100 | 1,151.514 ms | 6.1361 ms | 5.7398 ms |
| RestPostLargePayloadAsync | 100 | 1,529.500 ms | 19.1117 ms | 16.9421 ms |
| GrpcGetSmallPayloadAsync | 100 | 9.790 ms | 0.1912 ms | 0.1788 ms |
| GrpcStreamLargePayloadAsync | 100 | 1,504.844 ms | 11.8133 ms | 11.0502 ms |
| GrpcGetLargePayloadAsListAsync | 100 | 150.992 ms | 1.1345 ms | 0.9473 ms |
| GrpcPostLargePayloadAsync | 100 | 145.233 ms | 2.8594 ms | 2.6747 ms |
| RestGetSmallPayloadAsync | 200 | 23.167 ms | 0.3606 ms | 0.3373 ms |
| RestGetLargePayloadAsync | 200 | 2,212.474 ms | 8.9386 ms | 7.9239 ms |
| RestPostLargePayloadAsync | 200 | 3,094.323 ms | 22.3007 ms | 20.8601 ms |
| GrpcGetSmallPayloadAsync | 200 | 19.816 ms | 0.2869 ms | 0.2684 ms |
| GrpcStreamLargePayloadAsync | 200 | 3,017.144 ms | 24.9874 ms | 23.3732 ms |
| GrpcGetLargePayloadAsListAsync | 200 | 308.376 ms | 6.4603 ms | 12.4468 ms |
| GrpcPostLargePayloadAsync | 200 | 291.359 ms | 4.3825 ms | 4.0994 ms |

Рисунок 1 - Сравнение скорости обработки различных запросов к серверу REST и gRPC

### *Кэширование данных*

Кэширование данных помогает сократить время обработки запросов за счет хранения временных данных в оперативной памяти. В Go можно использовать Redis или Memcached для хранения данных в оперативной памяти и снижения нагрузки на базу данных. В Redis-е есть поддержка большого количества типов данных, среди которых строки, хэши, списки, множества и сортированные множества [5]. Иногда для локального кэширования используются библиотеки ristretto или bigcache, которые сохраняют данные в памяти, сокращая задержки при выполнении повторных запросов.

Самым популярным решением является база данных Redis. Это обусловлено тем, что управление этой базы данных не вызывает сложностей. Весь функционал доступен сразу же после установки образа Docker. Один из важных отличий Redis от других баз данных является то, что данные хранятся в энергозависимой памяти, что гарантирует потерю данных в случае перебоев в работе электросети. Для предотвращения потерь данных предусмотрены механизмы копирования данных в ПЗУ.

Пример кэширования с использованием Redis изображен на рисунке 2.

```go
package main

import (
    "context"
    "fmt"
    "github.com/redis/go-redis/v9"
)

var ctx = context.Background()  2 usages  new *

func main() {  Your Name *
    rdb := redis.NewClient(&redis.Options{
        Addr: "localhost:6379",
    })

    err := rdb.Set(ctx,  key: "key",  value: "value",  expiration: 0).Err()
    if err != nil {
        panic(err)
    }

    val, err := rdb.Get(ctx,  key: "key").Result()
    if err != nil {
        panic(err)
    }
    fmt.Println( a...: "key", val)
}
```

Рисунок 2 - Кэширование на примере RedisDB

Использование Redis помогает существенно ускорить работу приложения, избегая дополнительных обращений к базе данных для повторных запросов.

### *Балансировка нагрузки и масштабирование*

Для управления нагрузкой в микросервисной архитектуре используется Kubernetes, который позволяет легко масштабировать сервисы в зависимости от потребностей. Go прекрасно сочетается с Kubernetes, так как его исполняемые файлы легко контейнеризировать, а сам язык поддерживает масштабирование через использование Kubernetes позволяет централизованно управлять контейнерами, а значит, ускорить и упростить вывод новых продуктов на рынок, построить эффективный процесс разработки и тестирования [6]. Kubernetes Deployment и

Service. Kubernetes также поддерживает автоматическое масштабирование, что делает его подходящим решением для крупных систем с высоконагруженными микросервисами.

### *Результаты исследования и рекомендации*

Исследование показало, что использование асинхронности и конкурентности на Go, применение gRPC для межсервисной коммуникации и кэширование данных позволяют существенно снизить время отклика. Kubernetes с функцией авто-масштабирования позволяет управлять нагрузкой и поддерживать стабильную работу системы даже при высоких пиках нагрузки.

### *Разработка прототипа web-сервера с использованием gRPC и Redis*

Для разработки быстрой серверной части web-приложения были учтены ранее исследованные средства разработки и выявлены лучшие по производительности компоненты приложения. На рисунке 3 представлен сервер, состоящий из модулей доступа к базе данных Redis. Представлены функции Set и Get для удобного назначения и чтения данных из Redis.

```go
type CacheServer struct {  4 usages
    cache.UnimplementedCacheServiceServer
    redisClient *redis.Client
}

func NewCacheServer(redisAddr string) *CacheServer {  1 usage
    rdb := redis.NewClient(&redis.Options{
        Addr: redisAddr,
    })
    return &CacheServer{redisClient: rdb}
}

func (s *CacheServer) Set(ctx context.Context, req *cache.SetRequest) (*cache.SetResponse, error) {
    err := s.redisClient.Set(ctx, req.Key, req.Value,  expiration: 0).Err()
    if err != nil { return &cache.SetResponse{Success: false}, err }
    return &cache.SetResponse{Success: true}, nil
}

func (s *CacheServer) Get(ctx context.Context, req *cache.GetRequest) (*cache.GetResponse, error) {
    val, err := s.redisClient.Get(ctx, req.Key).Result()
    if err == redis.Nil { return &cache.GetResponse{Found: false}, nil } else if err != nil { return nil, err }
    return &cache.GetResponse{Value: val, Found: true}, nil
}

func main() {
    grpcServer := grpc.NewServer()
    cacheServer := NewCacheServer( redisAddr: "localhost:6379") // Redis на localhost
    cache.RegisterCacheServiceServer(grpcServer, cacheServer)

    listener, err := net.Listen( network: "tcp",  address: ":50051")
    if err != nil {
        log.Fatalf( format: "Failed to listen: %v", err)
    }
    fmt.Println( a...: "gRPC server listening on port 50051")
    if err := grpcServer.Serve(listener); err != nil {
        log.Fatalf( format: "Failed to serve: %v", err)
    }
```

Рисунок 3 - Серверная часть web-приложения с кэшированием

Клиентская часть web-приложения отправляет запросы к gRPC серверу. В случае успешной отправки данных в Redis, клиентское приложение выводит сообщение «Успешно» (см. рис. 4).

```go
type CacheServer struct {  4 usages
    cache.UnimplementedCacheServiceServer
    redisClient *redis.Client
}

func NewCacheServer(redisAddr string) *CacheServer {  1 usage
    rdb := redis.NewClient(&redis.Options{
        Addr: redisAddr,
    })
    return &CacheServer{redisClient: rdb}
}

func (s *CacheServer) Set(ctx context.Context, req *cache.SetRequest) (*cache.SetResponse, error) {
    err := s.redisClient.Set(ctx, req.Key, req.Value,  expiration: 0).Err()
    if err != nil : &cache.SetResponse{Success: false}, err ↵
    return &cache.SetResponse{Success: true}, nil
}

func (s *CacheServer) Get(ctx context.Context, req *cache.GetRequest) (*cache.GetResponse, error) {
    val, err := s.redisClient.Get(ctx, req.Key).Result()
    if err == redis.Nil : &cache.GetResponse{Found: false}, nil ↵ else if err != nil : nil, err ↵
    return &cache.GetResponse{Value: val, Found: true}, nil
}

func main() {
    grpcServer := grpc.NewServer()
    cacheServer := NewCacheServer( redisAddr: "localhost:6379") // Redis на localhost
    cache.RegisterCacheServiceServer(grpcServer, cacheServer)

    listener, err := net.Listen( network: "tcp",  address: ":50051")
    if err != nil {
        log.Fatalf( format: "Failed to listen: %v", err)
    }
    fmt.Println( a...: "gRPC server listening on port 50051")
    if err := grpcServer.Serve(listener); err != nil {
        log.Fatalf( format: "Failed to serve: %v", err)
    }
```

Рисунок 4 - Клиентская часть web-приложения

Разработанный backend в рамках исследования соответствует современным технологиям разработки приложений. Представленный отрывок из приложения является готовой основой для дальнейшей разработки. В будущем возможно расширение функционала с учетом растущих требований пользователей и бизнес-задач.

Система предполагает улучшение масштабируемости, чтобы эффективно работать под высокими нагрузками, а также развитие микросервисной архитектуры для повышения гибкости и облегчения обновления отдельных компонентов. Приложение возможно интегрировать с внешними API и сервисами, расширяя возможности взаимодействия с платежными системами, CRM, аналитическими платформами и другими решениями. Важной частью дальнейшей разработки станет усиление безопасности с внедрением более сложных механизмов аутентификации и авторизации, а также системы мониторинга для своевременного выявления и предотвращения угроз. С применением DevOps-практик и поддержкой CI/CD можно будет ускорить процесс развертывания и повысить стабильность эксплуатации, что сделает платформу гибкой и устойчивой к изменениям.

### *Заключение*

В данной статье были рассмотрены ключевые средства для ускорения времени отклика микросервисных приложений на Go. Использование Go позволяет добиться высокой производительности благодаря встроенной поддержке конкурентности и возможности работы с низкоуровневой коммуникацией через gRPC. Интеграция с Kubernetes и использование кэширования помогает улучшить отклик системы для высоконагруженных систем, обеспечивая гибкость и масштабируемость. В рамках исследования был разработан сервис для углубленного изучения современных актуальных средств разработки быстрых web-приложений. Также было уделено внимание важности правильной настройки сетевого взаимодействия, включая использование HTTP/2 и оптимизацию соединений, что значительно снизило задержки при передаче данных. Реализация систем кэширования, таких как

Redis, позволила сократить время доступа к часто запрашиваемым данным, что также положительно сказалось на времени отклика. Исследование помогло выделить основные механизмы повышения производительности веб-приложений на основе микросервисов и Go, а также определить, какие решения наиболее эффективны для ускорения времени отклика. В дальнейших исследованиях планируется исследовать современные технологии frontend с целью актуализации и выделения лучших фреймворков графической части web-приложений, чтобы обеспечить полную совместимость с разработанным backend и улучшить пользовательский интерфейс. Такой комплексный подход к frontend позволит создать интерфейс, который не только полностью интегрирован с backend, но и обеспечивает комфортный пользовательский опыт.

### *Литература*


1. Собеседование Golang разработчика (теоретические вопросы), Часть II. Что там с конкурентностью? [Электронный ресурс] – Режим доступа: https://habr.com/ru/articles/670974
2. Планирование в Go: Часть II — Планировщик Go [Электронный ресурс] – Режим доступа: https://habr.com/ru/articles/489862
3. gRPC vs REST, что выбрать для нового сервера? [Электронный ресурс] – Режим доступа: https://habr.com/ru/articles/565020
4. gRPC — альтернатива REST API от Google [Электронный ресурс] – Режим доступа: https://habr.com/ru/companies/oleg-bunin/articles/316652
5. Использование memcached и Redis в высоконагруженных проектах [Электронный ресурс] – Режим доступа: https://habr.com/ru/companies/oleg-bunin/articles/316652
6. Зачем крупным компаниям микросервисы, контейнеры и Kubernetes: как эти три технологии выводят IT на новый уровень [Электронный ресурс] – Режим доступа: https://cloud.vk.com/blog/mikroservisy-kontejnery-i-kubernetes



***Безуглый В.В., Боднар А.В. Исследование механизмов ускорения времени отклика современных web-приложений на основе микросервисной архитектуры.*** *В статье проведён анализ популярных существующих принципов написания серверной части приложений, выполнено сравнение функционала, выполнено сравнение производительности решений.*



***Ключевые слова:*** *время отклика, микросервисы, web-приложения, производительность, асинхронность, конкурентность, балансировка нагрузки , масштабирование, кэширование, Redis, базы данных.*



***Bezugly V.V., Bodnar A.V. Investigation of mechanisms for accelerating response time of modern web applications based on a microservice approach.*** *The article analyzes the popular existing principles of writing the backend of applications, compares the functionality, and compares the performance of solutions.*



***Key words:*** *response time, microservices, web applications, performance, asynchrony, competitiveness, load balancing, scaling, caching, Redis, databases.*

УДК 004.5

# Система отслеживания успеваемости студентов с уведомлением об изменениях

**Е.Д. Пачкин*, Д.С. Шидловская***
* студент, Кузбасского государственного технического университета имени Т.Ф. Горбачева,
addim111@yandex.ru
*ассистент, Кузбасского государственного технического университета имени Т.Ф. Горбачева
shidlovskajads@kuzstu.ru

***Пачкин Е.Д., Шидловская Д.С. Система отслеживания успеваемости студентов с уведомлением об изменениях.*** *В статье рассматривается система, предназначенная для оповещения студентов Кузбасского государственного технического университета (КузГТУ) об изменениях в их контрольных точках. Введение в систему контроля успеваемости студентов подчеркивает важность регулярного отслеживания учебного прогресса, особенно для первокурсников, которые впервые сталкиваются с этой концепцией. Обсуждаются проблемы недостаточной активности некоторых студентов в контроле своей успеваемости, что может привести к негативным последствиям для них.*

***Ключевые слова****: студент, КузГТУ, отслеживание, оповещение, статистика, успеваемость.*

### *Введение*

В образовательном процессе в Кузбасском государственном техническом университете, контрольные точки имеют значительное значение для оценки успеваемости студентов. В первый год обучения многие студенты впервые сталкиваются с этой концепцией, которая помогает им отслеживать свои достижения в течение учебного месяца. Регулярная проверка контрольных точек становится для них не только источником волнения, но и стимулом к учебе. Они с нетерпением ждут появления новой информации о своих успехах за прошедший месяц.

Тем не менее, не все студенты проявляют такую же активность и интерес к своей успеваемости. Некоторые из них могут не осознавать важность контроля своих результатов или просто не находят времени для регулярного мониторинга своей статистики. Это может привести к тому, что в конце контрольной недели они узнают о своих оценках слишком поздно, не имея возможности их улучшить.

Встроенных методов уведомлений об обновлениях информации в контрольных точках у КузГТУ не было обнаружено. Проведена консультация с центром информационных технологий университета и был сделан вывод, что данную систему реализовывать КузГТУ не собирается, в связи с явной проблемой была определена цель работы: создание собственной системы отслеживания академической успеваемости студентов, а также их уведомление при обнаружении изменений в ней.

### *Анализ разработки и реализация*

Эта система должна соответствовать некоторым требованиям: функциональным, нефункциональным, техническим, пользовательским. Функциональные требования включают отслеживание изменений, отправку уведомлений и просмотр успеваемости студентов. Нефункциональные требования касаются удобства использования, интуитивного интерфейса, мгновенного отклика, стабильности и безопасности. Технические требования охватывают обработку недоступности серверов КузГТУ, с которых берутся данные, оптимизацию обновлений и защиту от злоупотреблений, а пользовательские аспекты включают валидацию данных и настройку уведомлений.

В качестве платформы для реализации системы был выбран Telegram, что обусловлено его популярностью и надежностью, обеспечивающими широкий доступ для пользователей. Система будет функционировать при постоянном интернет-соединении для взаимодействия с серверами Telegram и КузГТУ, обеспечивая круглосуточный доступ, учитывая возможные технические перерывы. Она должна поддерживать одновременные обращения большого числа пользователей, особенно в периоды сессий, а также иметь механизмы для обработки сбоев серверов КузГТУ. Предусмотрено регулярное обновление информации о контрольных точках.

По вопросу персональных данных студентов была проведена консультация с юридическим отделом КузГТУ, который подтвердил, что в работе будет использоваться только та информация, которая публично

распространяется КузГТУ и на которую студенты подписывали согласие при поступлении или обучении в КузГТУ на обработку и распространение персональных данных, доступ к которым, как указано в согласии, может получить неограниченный круг лиц. Согласие на обработку включает имя, фамилию, отчество, номер академической группы, информацию о текущей успеваемости и промежуточной аттестации, а также другие данные, не затрагивающие цель работы. Все упомянутые данные могут быть получены неограниченным кругом лиц через официальный сайт КузГТУ или обучающимся и работниками КузГТУ через портал КузГТУ.

Для пользовательского интерфейса проекта используются возможности Telegram, реализованные на языке программирования Python с применением библиотеки aiogram. Эта библиотека активно развивается и полностью раскрывает потенциал Telegram, поддерживая асинхронное взаимодействие с ботом, что позволяет обрабатывать запросы от множества пользователей без задержек [1].

В процессе работы выполняется множество операций, включая проверку каждого действия пользователя, будь то обработка сообщений или частота нажатий на кнопки. При успешном получении данных от серверов КузГТУ анализируются изменения в контрольных точках, и в зависимости от предпочтений пользователя формируются соответствующие сообщения. Если пользователь не указал предпочтения, используются значения по умолчанию [2].

Для стабилизации нагрузки на серверы КузГТУ внедрена логика, регулирующая частоту обновления информации в зависимости от потребностей пользователей. База данных состоит из нескольких таблиц: пользователи, блокировки пользователей, кэш контрольных точек, оформленные подписки и логи.

Для реализации механизма уведомлений проводится регулярный анализ зачетных книжек с целью выявления изменений в контрольных точках. В зависимости от наличия изменений формируется сообщение, отправляемое пользователю в соответствии с его предпочтениями [3].

Также в работе применяется Docker для контейнеризации и развертывания системы независимо от операционной системы, что предотвращает потерю функциональности из-за особенностей конкретной ОС. Для разработки интуитивно понятных команд и меню взаимодействия с пользователем используется подход, основанный на понимании мышления среднестатистического студента и моделировании его поведения [4].

### *Взаимодействие пользователя с системой*

В начале взаимодействия с системой пользователь отправляет определенному telegram-боту[2] начальное сообщение, которое определяется самим telegram. При первом получении этого сообщения бот отвечает приветствием, в котором описываются функции системы, и закрепляет его для удобства пользователя. При последующих попытках отправки начального сообщения бот информирует, что для работы необходимо ввести номер зачетной книжки с примером ввода, а также предлагает кнопку для настройки персональных уведомлений (см. Рисунок 5). Важно отметить, что для получения информации о контрольных точках не обязательно каждый раз отправлять начальное сообщение и номер зачетной книжки; их можно отправлять в любом контексте взаимодействия с ботом, и обработка будет выполнена корректно.

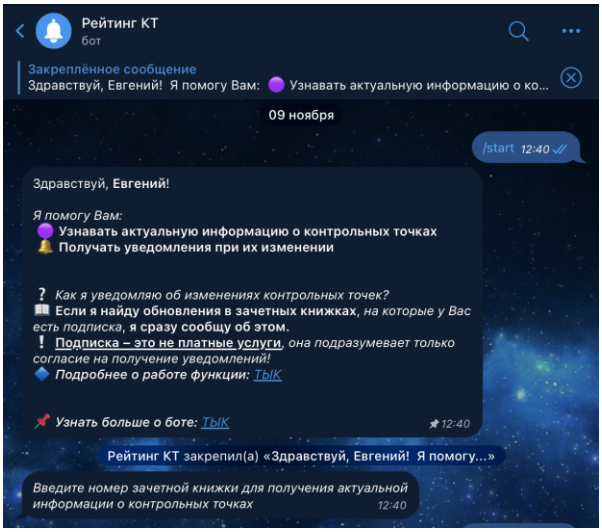



Рисунок 5 – Приветственное сообщение и запрос ввода номера зачетной книжки

[2] URL: https://t.me/KuzSTU_Rating_Bot

Далее пользователь отправляет сообщение с номером зачетной книжки, в ответ ему предоставляется временное сообщение, которое динамически изменяется, пока не будет создан форматированный и структурированный текст с актуальной информацией о контрольных точках за текущий семестр, который будет отправлен пользователю (см. Рисунок 6). Если пользователь ранее не запрашивал эту зачетную книжку, ему предлагается возможность получать уведомления об изменениях контрольных точек этой зачетной книжки. Если зачетная книжка уже была введена ранее, программа просто предоставляет актуальную информацию о контрольных точках без предложения подключения уведомлений.

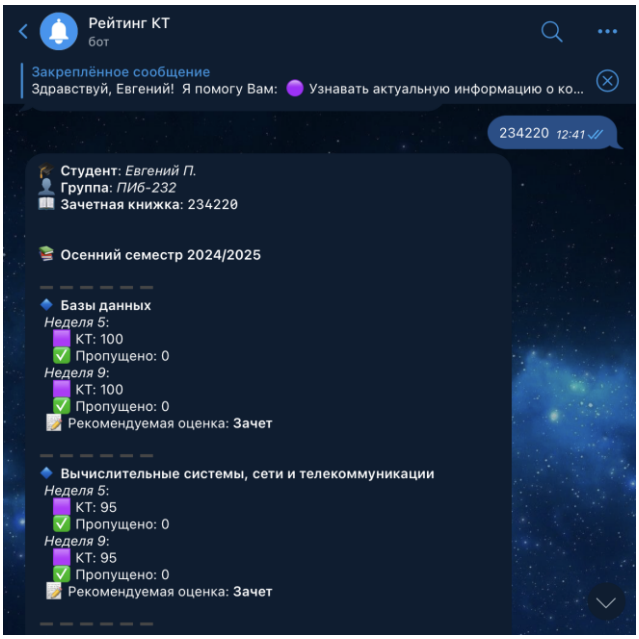



Рисунок 6 – Вывод информации о контрольных точках

Пользователь может отказаться от получения уведомлений и впоследствии самостоятельно включить их, нажав на специальную кнопку в командах бота, выбрав соответствующий пункт настроек и введя номер зачетной книжки. Бот уточнит, правильно ли найдена зачетная книжка, после чего пользователь может согласиться или отказаться от получения уведомлений. В случае случайной активации уведомлений пользователь может немедленно отказаться от них, как и в случае случайного отказа — активировать уведомления заново (см. Рисунок 7).

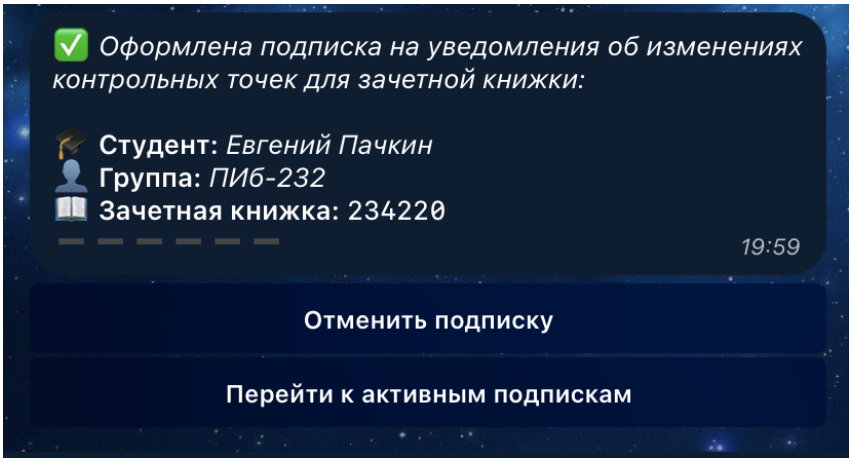



Рисунок 7 – Оформление подписки на уведомления

При обнаружении изменений контрольных точек зачетной книжки, все пользователи, отслеживающие изменения в указанной зачетной книжке, получат сообщения от бота с соответствующими изменениями. После получения уведомления пользователь сможет просмотреть не только изменения, но и полную статистику контрольных точек указанной зачетной книжки за текущий семестр, нажав соответствующую кнопку (см. Рисунок 8).

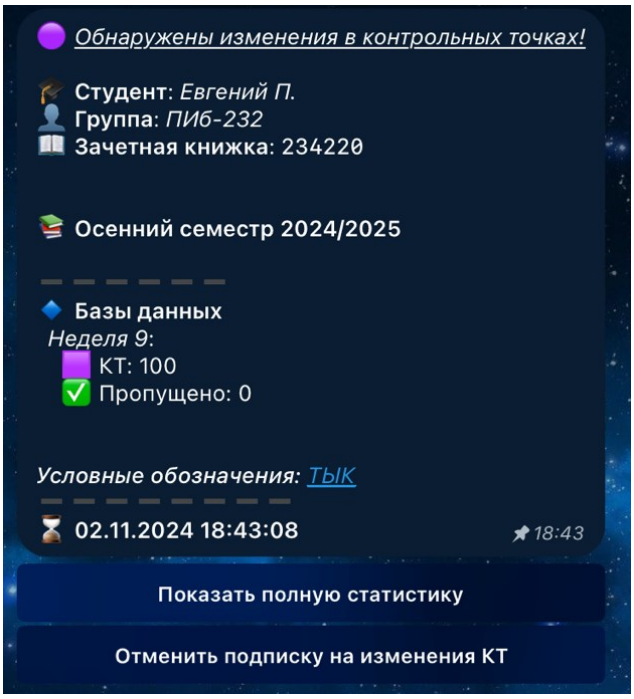



Рисунок 8 – Уведомление об изменении контрольной точки

Можно включить динамически выбираемый значок, который будет подбираться в зависимости от того, какая именно контрольная была изменена или выставлена, и добавляться в начало сообщения. При получении уведомления пользователь сможет, не заходя в telegram и не раскрывая полностью уведомление, предварительно узнать о состоянии успеваемости, просмотрев превью сообщения с уведомлением об изменении какой-либо контрольной точки.

Также в настройках можно выбрать, уведомления об изменениях каких контрольных точек в текущем контроле пользователь хочет получать. Например, если родителю важно только знать о том, что у их ребенка есть контрольные точки, отличные от низкой, он может установить соответствующую настройку.

При частом запросе актуальной информации о контрольных точках в течение короткого промежутка времени пользователю будет выдано предупреждение, а также уведомление для людей, сопровождающих проект. Важно отметить, что при обычном взаимодействии с ботом, например, при настройке персональных уведомлений, не возникает искусственной задержки в отклике системы; она возникает только при обращении к системе с целью получения информации от серверов КузГТУ. Эта задержка обеспечивает контроль отправки запросов и исключает злоупотребление системой с целью нанесения вреда серверам КузГТУ.

Если пользователь вводит неверный номер зачетной книжки или номер зачетной книжки студента, у которого нет контрольных точек за текущий семестр, ему будут отправлены соответствующие предупреждения о некорректном вводе.

### *Выводы*

Основные достижения проекта включают создание удобного интерфейса для получения актуальной информации о контрольных точках студентов, реализацию системы уведомлений об изменениях в текущем контроле с возможностью персональной настройки этих уведомлений, а также обеспечение высокой производительности, масштабируемости и доступности системы благодаря асинхронной обработке запросов, контейнеризации и надежной работе виртуального выделенного сервера.

Система позволяет учащимся оперативно отслеживать изменения в успеваемости, что способствует повышению мотивации к учебе и улучшению образовательных результатов. В результате, академическая успеваемость студентов, у которых оформлена подписка на уведомления об изменениях контрольных точек, значительно возросла благодаря своевременному уведомлению и возможности исправления контрольных точек в положенный срок.

В перспективе система может быть расширена следующими функциями: добавление возможности просмотра расписания занятий; реализация функции напоминаний о предстоящих дедлайнах и важных датах; внедрение системы рекомендаций по улучшению успеваемости на основе анализа данных; разработка web-

интерфейса средствами того же telegram для более удобного просмотра статистики; внедрение элементов геймификации для повышения мотивации студентов [5].

### *Литература*


1. Златопольский, Д.М. Основы программирования на языке Python. – Москва: ДМК Пресс, 2017. – 284 с.

2. **Федоров, Д.Ю. Программирование на языке высокого уровня Python** : учебное пособие для прикладного бакалавриата. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – URL: https://urait.ru/bcode/437489 (дата обращения: 10.11.2024).

3. Доусон, М. Программируем на Python. – Санкт-Петербург: Питер, 2014. – 416 с.

4. Сайбал, Г. Docker без секретов. Разработка и развертывание приложений с помощью Docker. – Санкт-Петербург: Издательство BHV, 2023. – 256 с.

5. Бабаскин, С.Я. Инновационный проект. Методы отбора и инструменты анализа рисков. – Москва: Издательский дом "Дело" РАНХиГС, 2018. – 240 с.



***Пачкин Е.Д., Шидловская Д.С. Система отслеживания успеваемости студентов с уведомлением об изменениях.*** *В статье рассматривается система, предназначенная для оповещения студентов Кузбасского государственного технического университета (КузГТУ) об изменениях в их контрольных точках. Введение в систему контроля успеваемости студентов подчеркивает важность регулярного отслеживания учебного прогресса, особенно для первокурсников, которые впервые сталкиваются с этой концепцией. Обсуждаются проблемы недостаточной активности некоторых студентов в контроле своей успеваемости, что может привести к негативным последствиям для них.*



***Ключевые слова****: студент, КузГТУ, отслеживание, оповещение, статистика, успеваемость.*



***Pachkin Eugene, Shidlovskaja Diana Student progress tracking system with notification of changes.*** *The article considers the development of a system designed to notify students of the Kuzbass State Technical University (KuzSTU) about their changes at control points. The introduction to the student performance monitoring system highlights the importance of regular monitoring of academic achievements, especially for first-year students who are encountering this concept for the first time. The problems of insufficient activity of some students in monitoring their academic performance are discussed, which can lead to negative consequences for them.*



***Key words****: student, KuzSTU, tracking, notification, statistics, academic performance.*

# СЕКЦИЯ 2. «ИНФОРМАЦИОННЫЕ СИСТЕМЫ, БАЗЫ ДАННЫХ, БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ»

УДК 004. 056.55

## Анализ эффективности протоколов передачи данных, использующих криптографические алгоритмы на эллиптических кривых

**А.В. Чернышова[*1], А.В. Коржов[*2]**

[*1] ст. преподаватель кафедры программной инженерии им. Л.П. Фельдмана,
Донецкий национальный технический университет,
chernyshova.alla@rambler.ru, SPIN-код: 3318-2066

[*2] магистрант кафедры программной инженерии им. Л.П. Фельдмана,
Донецкий национальный технический университет,
mr.demonsmert@mail.ru

***Чернышова А.В., Коржов А. В. Анализ эффективности протоколов передачи данных, использующих криптографические алгоритмы на эллиптических кривых.*** *В статье рассмотрено понятие эллиптической криптографии, рассмотрены основные современные протоколы передачи данных, которые применяют в своей основе эллиптическую криптографию, выявлены основные достоинства и недостатки данных протоколов. Также в статье приведён анализ основных уязвимостей к атакам современных протоколов передачи данных, использующих криптографию на эллиптических кривых.*

***Ключевые слова:*** *криптография, протокол, SSL/TLS, алгоритм, ECDHE, ECDSA, Signal Protocol, SSH, Bitcoin, IPSec, FIDO, WireGuard, преимущество, недостаток, уязвимость, атака.*

### *Введение*

С возникновением требований к безопасной передаче информации свое развитие получила целая наука — криптография. Изначально, криптография имела важное значение для государств мира и применялась для засекречивания важной информации, чтобы скрыть её от посторонних лиц. Однако, с развитием технологии Интернета, алгоритмы криптографической защиты данных стали важными для всех категорий граждан, поскольку все люди стремились обезопасить свою персональную и коммерческую информацию [1].

В настоящее время, при работе в Интернете, люди могут чувствовать себя безопасно, когда отправляют какую-либо информацию. Данная безопасность обусловлена тем, что существует множество протоколов передачи данных, которые основаны на различных уровнях защиты и применяют различные технологии для защиты информации. Однако, современный мир не стоит на месте и технологии в нём постоянно развиваются, что приводит к тому, что к средствам защиты информации, которые применяются в протоколах передачи данных, выдвигаются новые требования.

В современном мире особое место начинают занимать протоколы передачи данных, которые используют в своей основе криптографию на эллиптических кривых. Это обуславливается в первую очередь тем, что эллиптические кривые имеют более высокий уровень безопасности по сравнению с классическими криптографическими методами. Также, что не менее важно, в настоящее время протоколы на основе эллиптических кривых имеют более высокую эффективность передачи данных. Безусловно, у протоколов передачи данных, основанных на эллиптических кривых, есть свои неоспоримые достоинства, однако недостатки в современных реализациях подобных протоколов также выделяются. Проведя детальный анализ современных протоколов передачи данных с криптографической защитой на эллиптических кривых, можно будет сделать вывод о точной степени их эффективности применения в настоящее время.

### *Эллиптическая криптография*

В современном мире, эллиптическая криптография является отдельным разделом криптографии. Данный

раздел криптографии нацелен на изучении асимметричных криптосистем, которые основаны на эллиптических кривых над конечными полями.

Эллиптическая кривая — это кривая, которая основана на множестве точек и задается уравнением 3-й степени. Уравнение данной кривой приведено на рисунке 1.

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Рисунок 1 — Уравнение эллиптической кривой

Уравнение из рисунка 1 можно привести к форме Вейерштрасса, в том случае, когда характеристика поля не будет равной 2 или 3. В таком случае, мы получим сокращённую запись данного уравнения (см. рис. 2) [2].

$$y^2 = x^3 + ax + b$$

Рисунок 2 — Уравнение в форме Вейерштрасса

У эллиптических кривых выделяется три важных свойства:

- любая вертикальная прямая, параллельная оси Оу, никогда не пересекает кривую в форме Вейерштрасса или же пересекает строго два раза (при этом, точка касания будет считаться за двойное пересечение кривой);
- любая другая прямая способна пересекать кривую в форме Вейерштрасса строго один или три раза;
- множество точек эллиптической кривой составляет аддитивную абелевую группу в том случае, если к данному множеству добавлен нейтральный элемент, представленный в виде бесконечно удаленной точки [3].

К основным достоинствам эллиптической криптографии на фоне классических криптографических алгоритмов, можно отнести следующие три важных пункта:

- длина ключа значительно меньше, чем у алгоритмов, основанных на “классической” ассиметричной криптографии;
- скорость работы значительно выше, поскольку применяется меньший размер поля и структура, применяемая в эллиптической криптографии, является бинарной (структура бинарного конечного поля);
- возможность применять алгоритмы эллиптической криптографии в устройствах с ограниченными вычислительными ресурсами без потери надёжности защиты информации, благодаря маленькой длине ключа и более высокой скорости работы [4].

Таким образом, можно выделить то, что эллиптическая криптография значительно уменьшает размер ключей и повышает общую производительность системы, что крайне важно для широкого круга систем в современном мире. Особое место криптографические алгоритмы на основе эллиптических кривых нашли в протоколах передачи данных, что обусловлено компактностью ключей, применяемых в данных алгоритмах.

### *Анализ протоколов, использующих криптографию на эллиптических кривых*

В современном мире выделяется множество протоколов, которые применяют в своей основе криптографию на эллиптических кривых. Наиболее известные и широко применяемые протоколы с криптографией на основе эллиптических кривых следующие: SSL/TLS, Signal Protocol, SSH (Secure Shell), Bitcoin, FIDO (Fast Identity Online), IPSec, WireGuard.

Рассмотрим данные протоколы более подробно, а также выделим для чего конкретно применяется криптография на основе эллиптических кривых в данных протоколах.

SSL/TLS — криптографический протокол, который нацелен на то, чтобы обеспечивать защищённую передачу данных между узлами в сети Интернет.

Эллиптическая криптография нашла свое место в данном протоколе сразу в двух направлениях: генерация цифровых подписей и обеспечение безопасного канала передачи данных. Это обусловлено в первую очередь тем, что криптографические алгоритмы на основе эллиптических кривых позволяют снизить сложность вычислительных процессов для установки подключения, а также снизить максимальное время установки соединения. В протоколе SSL/TLS применяется сразу два алгоритма на основе эллиптических кривых: ECDHE и ECDSA [5].

ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) — алгоритм, который позволяет обеспечить безопасный канал для передачи данных. Алгоритм ECDHE крайне важен для SSL/TLS, поскольку он используется для согласования общего секретного ключа между клиентом и сервером, а также способен генерировать временные ключи для каждой сессии, обеспечивая свойство PFS (Perfect Forward Secrecy).

ECDSA (Elliptic Curve Digital Signature Algorithm) — данный алгоритм эллиптической криптографии нацелен на то, чтобы создавать и проверять электронные цифровые подписи. Его основное назначение — обеспечение целостности и подлинности данных. В TLS/SSL используется для генерации подписей,

аутентификации сервера и клиента, а также для подписи ключевых сообщений в процессе рукопожатия.

Signal Protocol — специальный криптографический протокол, который был создан с целью обеспечения сквозного шифрования голосовых вызовов, видеозвонков и мгновенных сообщений. В современном мире почти все популярные мессенджеры применяют Signal Protocol для реализации обмена сообщениями, совершения видеозвонков, а также голосовых вызовов. Однако, почти все мессенджеры модифицируют данный протокол, либо убирают некоторые функциональные возможности. Отличным примером модификации протокола Signal служит мессенджер Viber, который использует свой протокол Proteus. Proteus основан на протоколе Signal, однако использует немного другие криптографические средства защиты информации [6]. В основе Signal Protocol применяется две технологии, основанные на эллиптической криптографии — X3DH (Extended Triple Diffie-Hellman) и Double Ratchet.

X3DH — алгоритм, который был разработан с целью безопасного обмена ключами и данными. В Signal Protocol отвечает за генерацию ключей, подпись ключей на основе ECDSA, аутентификацию участников общения, их периодическое обновление и верификацию подлинности, а также за гарантию того, что ключ используется строго один раз.

Double Ratchet — алгоритм для безопасного и конфиденциального обмена сообщениями. Данный алгоритм основан на ECDHE и производит генерацию ключей к сообщениям, обновление цепочек ключей, а также создает общий секрет между участниками.

SSH (Secure Shell) — сетевой протокол сеансового уровня, который позволяет производить удалённое управление операционной системой и туннелирование TCP-соединений. Основное преимущество протокола SSH заключается в том, что он полноценно шифрует весь трафик, который через него передается [7].

Протокол SSH работает благодаря трём фазам: установка версии, согласование ключей, аутентификация. В каждой из данных фаз применяются алгоритмы криптографической защиты на основе эллиптических кривых. Таким образом, алгоритм ECDHE в данном протоколе служит для того, чтобы создавать общий секрет между участниками процесса обмена, генерировать сессионные ключи и производить их обновление. ECDSA, в свою очередь, применяется в протоколе SSH для того, чтобы создавать подписи ключей хоста, подписи аутентифицированных данных, а также для того, чтобы производить проверку подлинности сервера.

Bitcoin — считается первым протоколом технологии блокчейна (blockchain), а также является названием первой криптовалюты в мире, которая существует до сих пор. Стоит отметить, что данный протокол также считается сетевым протоколом. Протокол работает на основе блокчейна — распределённой базе данных, в которой информация о всех транзакциях хранится в цепочке блоков [8]. Следует отметить, что помимо протокола Bitcoin существуют и другие протоколы блокчейна, которые определяют другие криптовалюты мира. Считается, что протокол Bitcoin был первым протоколом, который применил криптографию на эллиптических кривых для повышения безопасности.

В основе протокола Bitcoin лежит четыре важных компонента:

- PoW (Proof of Work) — специальный механизм достижения консенсуса, используемый для подтверждения транзакций и создания новых блоков, который позволяет конкурировать системам майнинга (мощным ЭВМ) за завершение транзакций в сети и получение последующего вознаграждения [9];

- система транзакций и блокчейн — протокол Bitcoin производит запись всех транзакций в блоки, которые в последующем добавляются в общую цепочку блоков, при этом каждый новый блок содержит хэш-значение предыдущего блока, что позволяет создать неизменяемую цепочку данных и защитить сеть от подделок;

- механизм создания приватных и публичных ключей — благодаря применению ассиметричных криптографических алгоритмов достигается высокая безопасность протокола, поскольку каждый пользователь имеет свой приватный ключ (для подписи транзакций) и публичный ключ (для проверки подписи);

- алгоритм ECDSA — используется для генерации публичных и приватных ключей (используется кривая secp256k1), а также для создания и проверки подлинности цифровых подписей.

Благодаря применению эллиптической криптографии в виде алгоритма ECDSA, в протоколе Bitcoin обеспечивается подпись входных транзакций, верификация полномочий пользователя, а также подтверждается владение средствами. В связи с этим, можно сказать, что криптографические алгоритмы на основе эллиптических кривых позволяют обеспечить высокую безопасность и высокую производительность в протоколе Bitcoin, что важно для защиты активов пользователей и поддержания скорости операций в децентрализованных сетях.

FIDO (Fast Identity Online) — протокол аутентификации, позволяющий пользователям производить безопасный вход в различные онлайн-сервисы и приложения без необходимости ввода паролей. Вместо этого для проверки личности пользователя используются пары криптографических ключей. Сам протокол состоит из двух спецификаций:

- FIDO U2F — данная спецификация обеспечивает пользователям возможность проходить аутентификацию с помощью простого нажатия на физическое устройство, которое содержит ключ FIDO;

- FIDO UAF — данная спецификация протокола позволяет пользователям производить аутентификацию в сервисах с помощью применения биометрических факторов, или PIN-кодов, которые разблокируют ключи FIDO,

сохранённые на устройстве.

Обе спецификации протокола используют криптографию на эллиптических кривых для повышения уровня безопасности данных. Протокол применяет алгоритмы эллиптической криптографии в следующем:

- создание уникальной пары ключей для каждого сервиса — при регистрации устройства (первом входе в систему) создается уникальная пара публичного и приватного ключа благодаря алгоритму ECDSA, после чего публичный ключ отправляется на сервер, а приватный остается хранится в памяти устройства;

- создание подписи с применением ECDSA — при попытке входа пользователь должен пройти аутентификацию (например, использовать отпечаток пальца), после чего приватный ключ подписывается благодаря алгоритму ECDSA и отправляется на сервер, где происходит проверка подлинности пользователя.

Благодаря применению эллиптической криптографии в протоколе FIDO обеспечивается также PFS (Perfect Forward Secrecy), поскольку для каждого сервиса генерируется своя собственная пара ключей, что гарантирует то, что потеря доступа к одному сервису не приведёт к потере доступа к другим сервисам с личной информацией пользователя.

IPSec (Internet Protocol Security) — это комплекс протоколов сетевого уровня, которые в совокупности обеспечивают конфиденциальность данных. Данный протокол предоставляет пользователю механизмы шифрования и аутентификации IP-трафика с целью обеспечения безопасности сетевой коммуникации на уровне IP. В большинстве случаев, IPSec может быть настроен на двух уровнях — на уровне системы и на уровне приложений [10].

В основе IPSec лежит три основных протокола:

- протокол AH (Authentification Header) — данный протокол обеспечивает аутентификацию пакетов отправителя, а также контролирует целостность заголовка пакета, применяя механизмы хэширования данных;

- протокол ESP (Encapsulation Security Payload) — протокол обеспечивает проверку целостности пакетов данных и шифрование информации в пакете данных, при этом сам протокол способен работать сразу в двух режимах — транспортном (обеспечивает защиту пакетов протоколов более высокого уровня) и туннельном (обеспечивает связь между двумя шлюзами локальных сетей);

- протокол IKE (Internet Key Exchange) — данный протокол отвечает за установку соединения между сторонами, когда стороны договариваются о том, какие криптоалгоритмы будут использованы, а также когда стороны обмениваются сеансовыми ключами [11].

В настоящее время IPSec начинает переходить с применения классических криптографических алгоритмов на применение алгоритмов криптографической защиты на основе эллиптических кривых. Таким образом, в настоящее время эллиптическая криптография в данном комплексе протоколов обеспечивается в двух основных протоколах — ESP и IKE. В случае протокола ESP, эллиптическая криптография применяется для генерации ключей шифрования, аутентификации, а также векторов инициализации. Для обеспечения данных действий протокол ESP использует алгоритм ECDHE, который помимо генерации также обеспечивает PFS и защиту от компрометации предыдущих сессий. Помимо этого, в протоколе ESP эллиптическая криптография используется в комбинации с симметричными алгоритмами шифрования. В протоколе установки соединения IKE эллиптическая криптография применяется немного шире: алгоритм ECDHE применяется для безопасного обмена ключами и обеспечения FS (forward secrecy), алгоритм ECDSA используется для аутентификации узлов и обеспечения целостности сообщений.

Последний протокол, который стоит рассмотреть — WireGuard. Данный протокол относится к семейству VPN-протоколов. Особенность данного протокола заключается в том, что он имеет достаточно простую конфигурацию, высокую скорость работы и применяет самые современные методы криптографической защиты данных. В основе протокола WireGuard лежат несколько основных технологий:

- ChaCha20 — потоковый шифр, который применяется для реализации шифрования данных;

- Curve25519 — современный, безопасный алгоритм на эллиптических кривых, который позволяет создавать ключи и проводить обмен ими с высокой производительностью;

- Poly1305 — алгоритм аутентификации данных, который применяется в протоколе для проверки целостности и подлинности зашифрованных данных;

- BLAKE2s — криптографическая хэш-функция, которая применяется для генерации уникальных идентификаторов и ключей;

- HKDF — алгоритм, который производит генерацию ключей на основе полученного хэша [12].

В протоколе WireGuard эллиптическая криптография легла в основу работоспособности протокола. Поскольку при запуске протокола начальный этап идёт через генерацию ключей и установку соединения, то криптографический алгоритм Curve25519 играет ключевую роль для обеспечения высокой производительности работы протокола и повышения его уровня безопасности. После обмена ключами, в работу вступает алгоритм потокового шифрования ChaCha20 для шифрования данных между узлами, а алгоритм Poly1305 вычисляет код аутентификации и прикрепляет его к блоку данных. Для обновления ключей в процессе работы алгоритма используется HKDF, который создает производные ключи и повышает безопасность туннеля. В свою очередь,

BLAKE2s создает уникальные идентификаторы и хэши в рамках системы управления ключами, что повышает скорость процесса хэширования.

### *Анализ основных преимуществ и недостатков протоколов передачи данных, использующих криптографию на основе эллиптических кривых*

Рассмотренные выше протоколы передачи данных применяют в своих реализациях криптографию на эллиптических кривых, что повышает скорость их работы и степень защиты информации. Применение эллиптической криптографии безусловно является преимуществом данных протоколов, однако, как и любой существующий протокол, рассмотренные протоколы имеют свои недостатки.

Протокол SSL/TLS является важным и широко применяемым протоколом, однако корректное его внедрение может требовать дополнительных ресурсов, а некоторые старые системы могут не поддерживать его последние версии [13]. Основные преимущества и недостатки протокола SSL/TLS приведены в таблице 1.

Таблица 1 — Преимущества и недостатки протокола SSL/TLS

| Преимущества | Недостатки |
|---|---|
| Применение ECDHE позволяет снизить вычислительную сложность в момент установки подключения. | Необходимость тщательного выбора параметров эллиптической кривой для обеспечения высокой степени безопасности. |
| Сокращенное время установки соединения. | Повышенные требования к реализации для избежания уязвимостей. |
| Благодаря применению ECDHE обеспечивает свойство PFS (Perfect Forward Secrecy). | Потенциальная уязвимость к атакам по сторонним каналам при неправильной реализации. |
| ECDSA обеспечивает эффективную генерацию и проверку цифровых подписей. | Сложность поддержки работоспособности протокола на устаревших системах. |
| Протокол обеспечивает высокую степень безопасности при коротких длинах ключей. | |

Signal Protocol является хорошим протоколом для обеспечения сквозного шифрования голосовых вызовов, видеозвонков и мгновенных сообщений, не смотря на то, что в большей части применяются его модификации. В таблице 2 выделим основные преимущества и недостатки данного протокола.

Таблица 2 — Преимущества и недостатки Signal Protocol

| Преимущества | Недостатки |
|---|---|
| Обеспечение высокой эффективности процесса обмена ключами благодаря применению X3DH. | Высокая сложность реализации полного набора криптографических примитивов для обеспечения полной безопасности данных. |
| Надежная и безопасная аутентификация участников процесса обмена информации при помощи ECDSA. | При частом обновлении ключей повышаются требования к вычислительным ресурсам устройства. |
| Гарантия того, что все ключи, применяемые в процессе обмена информации, будут уникальными. | Необходимость дополнительного метода безопасного хранения долгосрочных ключей. |
| Эффективный процесс обновления ключей, благодаря применению Double Ratchet. | Существует возможность утечки метаданных. |
| Высокая скорость шифрования и малые требования к ресурсам. | |

SSH — сетевой протокол сеансового уровня. Одно из ключевых преимуществ данного протокола заключается в том, что он позволяет обеспечить защищенную передачу данных и предотвращает любую возможность несанкционированного подключения к каналу и перехвату из него данных [14]. Более подробно, преимущества и недостатки для данного протокола рассмотрены в таблице 3.

Таблица 3 — Преимущества и недостатки протокола SSH

| Преимущества | Недостатки |
|---|---|
| Защищенная передача данных по каналу связи, а также предотвращение несанкционированного подключения и перехвата данных. | Достаточно сильная зависимость от правильного выбора параметров эллиптической кривой для обеспечения высокой степени защиты данных. |
| Применение эллиптического алгоритма ECDHE для эффективной генерации сессионных ключей. | Меньшая совместимость с устаревшими системами. |
| Высокая степень надежности процесса аутентификации, благодаря применению алгоритма ECDSA. | Необходимость реализации дополнительного метода защиты приватных ключей. |
| Высокая степень оптимизации процесса создания и проверки цифровых подписей. | Существует возможность утечки метаданных. |

Протокол Bitcoin. Данный протокол работает на основе технологии блокчейна, что уже предполагает наличие высокой степени безопасности данных. Данный протокол считается достаточно новым и современным. После появления данного протокола началось активное развитие рынка криптовалют в мире, что привело к появлению ещё большего количество подобных протоколов, которые были названы в честь криптовалюты, для которой они применялись. Принято считать, что протокол Bitcoin (а также все протоколы на его основе) обладают крайне важным преимуществом — мгновенная скорость транзакций [15]. Мгновенная скорость транзакций говорит о том, что процесс верификации в данном протоколе обладает высокой эффективность. Более подробно преимущества и недостатки протокола Bitcoin приведены в таблице 4.

Таблица 4 — Преимущества и недостатки протокола Bitcoin

| Преимущества | Недостатки |
|---|---|
| Мгновенная скорость транзакций, которая обеспечивается благодаря высокой скорости верификации транзакций. | Невозможность изменения параметров кривой в эллиптическом алгоритме криптографии без хардфорка (выпуска такого обновления алгоритмов, при котором данное обновление будет полностью не совместимо ни с одной старой версией). |
| Благодаря применению кривой secp256k1 в алгоритме ECDSA обеспечивается высокая безопасность транзакций. | Высокая степень полной компрометации в случае утери приватного ключа. |
| Эффективная генерация публичных и приватных ключей. | Высокая сложность реализации методов для обеспечения безопасности приватных ключей. |
| Оптимальное соотношение размера ключей, применяемых в протоколе, и уровня безопасности протокола. | Потенциальная уязвимость к квантовым вычислениям. |
| Компактность блоков данных, благодаря малому размеру ключа. | |
| Высокая устойчивость к атакам, которые направлены на нарушение целостности данных. | |

Следующий протокол — FIDO (Fast Identity Online). Данный протокол, как и отмечалось ранее, позволяет пользователям безопасно входить в различные онлайн-сервисы и приложения без необходимости ввода паролей в данные сервисы. Данный протокол представлен в двух спецификациях и обладает, как рядом преимуществ, так и рядом недостатков. Более подробно это представлено в таблице 5.

Таблица 5 — Преимущества и недостатки протокола FIDO

| Преимущества | Недостатки |
|---|---|
| Эффективная генерация уникальных пар ключей для каждого сервиса. | Необходимость реализации методов безопасного хранения приватных ключей от сервисов на устройстве пользователя. |
| Обеспечение высокой эффективности процесса аутентификации, благодаря применению алгоритма ECDSA. | Высокая степень зависимости от корректности реализации протокола на стороне сервиса. |
| Обеспечивание свойства PFS (Perfect Forward Secrecy). | Достаточно высокая степень риска потери доступа ко всем сервисам в том случае, если пользователь потеряет свое устройство. |
| Оптимальное соотношение размера ключей, применяемых в протоколе, и уровня безопасности протокола. | Для спецификации FIDO UAF выделяются также уязвимости в биометрических методах защиты информации. |
| Высокая производительность и поддержка на мобильных устройствах. | |

IPSec (Internet Protocol Security) — протокол, предоставляющий пользователям механизмы шифрования и аутентификации IP-трафика. Данный протокол предоставляет полное шифрование трафика, передаваемых в туннеле, что является безусловным преимуществом данного протокола [16]. Выделим основные преимущества и недостатки применения эллиптической криптографии в данном протоколе, представив их в таблице 6.

Таблица 6 — Преимущества и недостатки IPSec

| Преимущества | Недостатки |
|---|---|
| Полное шифрование IP-трафика, который передается по туннелю. | Высокая сложность настройки и конфигурации протокола. |
| Алгоритм ECDHE предоставляет эффективный обмен ключами. | Высокие требования к вычислительным ресурсам устройства. |
| Обеспечение свойства PFS. | Возможные уязвимости при недостаточно сильных ключах или ошибках в реализации. |
| Эффективная работа в комбинации с симметричными алгоритмами шифрования. | Потенциальные проблемы совместимости, в случае полного перехода с классическим алгоритмов. |
| Высокая степень защиты от перехвата данных. | |

Последний протокол — WireGuard. Данный VPN-протокол обладает простой конфигурацией, высокой скоростью работы, а также применяет современные методы криптографии. Огромным преимуществом протокола является его защита от атак повторного использования, а также атак типа «отказ в обслуживании» [17]. Более подробно преимущества и недостатки протокола приведены в таблице 7.

Таблица 7 — Преимущества и недостатки протокола WireGuard

| Преимущества | Недостатки |
|---|---|
| Защита от атак повторного использования и атак типа «отказ в обслуживании». | Достаточно ограниченный выбор криптографический примитивов. |
| Высокая производительность благодаря применению алгоритма Curve25519. | Необходимость в регулярном процессе обновления ключей. |
| Достаточна простая, но крайне надёжная генерация ключей. | Высокая степень зависимости от корректной реализации HKDF, который применяется для обновления ключей. |
| Применение эллиптической криптографии и её эффективная интеграция с другими криптографическими алгоритмами. | Ограниченная совместимость с некоторыми операционными системами. |
| Оптимизация использования ресурсов системы. | Необходимость предоставления доверия к устройству, на котором хранится приватный ключ. |

Проанализировав преимущества и недостатки современных протоколов передачи данных, которые используют криптографию на эллиптических кривых, можно отметить, что все протоколы обладают сильными

достоинствами и применяют эллиптическую криптографию для повышения производительности процессов. Однако, рассмотренные протоколы также обладают рядом своих недостатков, которые иногда могут быть критичными и требовать значительных затрат ресурсов на своё устранение в будущем.

### *Анализ уязвимостей протоколов передачи данных, использующих криптографию на основе эллиптических кривых*

Рассмотренные протоколы передачи данных, с криптографией на основе эллиптических кривых, обладают своими уязвимостями. Среди таких уязвимостей можно выделить различные атаки: «человек по середине», повторное использование ключей, переполнения пакетов и др. Рассмотрим основные уязвимости протоколов передачи данных.

Протокол SSL/TLS подвержен нескольким типам атак, которые в первую очередь могут быть произведены в том случае, если алгоритм эллиптической криптографии настроен некорректно и содержит уязвимость в настройке. К протоколу SSL/TLS возможно применить следующие типы атак:

- атака типа «человек посередине» (Man-in-the-Middle, MITM) — вид атаки, когда злоумышленник способен тайно ретранслировать и изменять при необходимости связь между двумя сторонами, когда стороны считают, что они взаимодействуют друг с другом [18];
- атака по времени (timing attack) — это класс атак, который связан с исследованием времени операций обработки информации [19], в случае с эллиптической криптографией данная атака способна восстановить части приватного ключа, путём анализа времени выполнения операций с закрытым ключом ECDSA;
- атака с недопустимой кривой (invalid curve attack) — этот тип атаки предполагает использование математических свойств эллиптических кривых для того, чтобы получить информацию о приватном ключе сервера, что становится возможным если отсутствует механизм валидации точек;
- атака на повторное использование ключей — заключается в том, что при завершении сессии ключи не были корректно обнулены, что приводит к снижению безопасности данных, делая эти данные более уязвимыми для злоумышленников;

Signal Protocol имеет некоторые уязвимости, однако, в большинстве случаев, является достаточно защищённым протоколом. Самой опасной атакой для Signal Protocol является атака с неизвестным общим ключом (Unknown Key-Share Attack). Данная атака заключается в том, что злоумышленник устанавливает свой собственный общий ключ между двумя сторонами, при этом стороны уверены в том, что они установили общий ключ между собой. Эта атака возможна в том случае, когда происходит недостаточная валидация идентификаторов и требует активного вмешательства в процесс обмена ключами.

Протокол SSH подвержен двум типам атак: атака по сторонним каналам (slide-channel attack) и атака по времени (implementation-based timing attack). Рассмотрим данные атаки:

- атака по сторонним каналам (slide-channel attack) — вид атаки, который получает информацию из стороннего канала, при этом полученная информация может быть открытым текстом, зашифрованным текстом, либо ни одним из данных вариантов [20], а в случае протокола SSH с эллиптической криптографией может быть применима для анализа энергопотребления или электромагнитного излучения устройства, что позволит злоумышленнику извлечь приватный ключ алгоритма ECDSA;
- атака по времени (implementation-based timing attack) — атака, которая нацелена на анализ времени обработки операций с алгоритмом ECDSA, что позволяет восстановить части приватного ключа.

Протокол Bitcoin является на данный момент самым криптостойким. Данная его особенность заключается в том, что он использует цепочку блоков данных, где каждый новый блок основан на хэше предыдущего блока. Единственной потенциальной угрозой для данного протокола являются атаки с использованием квантовых компьютеров, которые в теории способны решить проблему дискретного логарифмирования и после этого полностью взломать алгоритм ECDSA. Выделяется несколько типов возможных квантовых атак, которые в будущем могут быть применимы для взлома [21].

Протокол FIDO, в свою очередь, подвержен атаке повторного воспроизведения (replay attack), которая предполагает повторное использование перехваченной подписи ECDSA, если в реализации протокола отсутствует механизм защиты от повторов. Ранее данная атака применялась при попытке подделать голос пользователя, с целью управления системами с помощью голосового управления и организовывалась путём скрытой записи положительной попытки получения доступа к системе, после чего запись воспроизводилась и происходила попытка получения доступа [22].

Протокол IPSec также подвержен атаке повторного воспроизведения (replay attack). Вторая атака, которой подвержен данный протокол, называется «атака на замену алгоритма» (algorithm substitution attack). Данная атака предполагает замену алгоритма эллиптической криптографии на более слабый алгоритм, который позволяет злоумышленнику получить необходимую информацию. Данная атака становится возможной в том случае, когда протокол не предполагает наличие строгой проверки используемых параметров.

WireGuard — протокол семейства VPN. Данный протокол, как и протокол SSH, подвержен атаке по сторонним каналам. В случае WireGuard данная атака используется с целью анализа реализации алгоритма Curve25519 и становится возможной только в том случае, когда арифметические операции не имеют оптимальной реализации. Недостаток атаки заключается в том, что злоумышленнику требуется прямой физический доступ к устройству пользователя.

Проанализировав уязвимости к атакам всех протоколов передачи данных, применяющих криптографию на эллиптических кривых, можно говорить о том, что большинство атак нацелены на то, чтобы получить приватные ключи пользователей, либо восстановить часть данных ключей. Некоторые атаки эффективнее, поскольку их можно производить удалённо от пользователя. В других случаях — требуется физический контакт с устройством пользователя, чтобы провести атаку. Наиболее безопасным в современном мире является протокол Bitcoin, который применяет эллиптическую криптографию и является криптостойким ко всем известным современным атакам.

### *Выводы*

В рамках данной статьи было представлено краткое описание понятия эллиптической криптографии, а также был произведен анализ современных протоколов передачи данных, которые используют в своей основе алгоритмы эллиптической криптографии.

Рассмотрены основные современные протоколы, такие как: SSL/TLS, Signal Protocol, SSH, протокол Bitcoin, FIDO, IPSec, WireGuard.

Для каждого протокола был проведён краткий анализ алгоритмов эллиптической криптографии, которые применяются в их основе и сделаны краткие выводы о том, для чего в протоколах применяются данные алгоритмы.

Также были выделены основные преимущества и недостатки протоколов передачи данных, применяющих криптографию на эллиптических кривых. Помимо преимуществ и недостатков, был проведён анализ протоколов передачи данных с точки зрения их уязвимости к различным видам атак.

Важным фактором является то, что все современные протоколы передачи данных применяют алгоритмы эллиптической криптографии с целью снижения длины ключей, повышения эффективности и безопасности работы протоколов.

В дальнейшем планируется более глубокое изучение архитектуры протоколов передачи данных, с целью подробного изучения алгоритмов эллиптической криптографии, которые применяются в данных протоколах, что позволит выявить достоинства и недостатки данных алгоритмов, а также провести их сравнительный анализ, чтобы сформировать требования к авторскому протоколу с криптографией на эллиптических кривых. Основным требованием для авторского протокола станет его безопасность и криптостойкость его алгоритма к современным атакам.

### *Литература*

1. Бондаренко, В. В. Повышение эффективности протоколов передачи данных, использующих криптографическую защиту / В. В. Бондаренко, А. В. Чернышова // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2022): Сборник материалов IV Международной научно-практической конференции, Донецк, 29–30 ноября 2022 года. Том 1. – Донецк: Донецкий национальный технический университет, 2022. – С. 66-73. – EDN AVLORE.

2. Завгородний, С. Д. Применение эллиптических кривых в криптографии / С. Д. Завгородний // Научные исследования и разработки студентов: Сборник материалов VI Международной студенческой научно-практической конференции, Чебоксары, 30 января 2018 года / Редколлегия: О.Н. Широков [и др.]. – Чебоксары: Общество с ограниченной ответственностью "Центр научного сотрудничества "Интерактив плюс", 2018. – С. 100-102. – EDN YRTXUE.

3. Трифонова, К. Н. Эллиптические кривые в криптографии / К. Н. Трифонова // Дни науки студентов Владимирского государственного университета имени Александра Григорьевича и Николая Григорьевича Столетовых: Сборник материалов научно-практических конференций, Владимир, 12 марта – 06 2018 года. – Владимир: Владимирский государственный университет им. Александра Григорьевича и Николая Григорьевича Столетовых, 2018. – С. 1130-1134. – EDN QJERQC.

4. Кухтинов, В. Н. Использование криптографии на эллиптических кривых / В. Н. Кухтинов // Наукосфера. – 2020. – № 5. – С. 114-118. – EDN THQNFK.

5. Макушев, В. С. Анализ применения эллиптических кривых в системах сертификации и управления ключами для сайтов / В. С. Макушев, С. М. Старолетов // Современные цифровые технологии : Материалы II Всероссийской научно-практической конференции, Барнаул, 01 июня 2023 года / Под общей редакцией А.А.

Беушев, А.С. Авдеев, Е.Г. Боровцов, А.Г. Зрюмова. – Барнаул: Алтайский государственный технический университет им. И.И. Ползунова, 2023. – С. 235-239. – EDN LUVJNG.

6. Сравнительный анализ систем мгновенного обмена сообщениями / Г. А. Положий, А. Р. Тосунова, О. А. Сафарьян, Л. В. Черкесова // Молодой исследователь Дона. – 2020. – № 4(25). – С. 59-63. – EDN CLWJYB.

7. Омельченко, М. В. Сравнительный анализ безопасной передачи данных с помощью протоколов TELNET и SSH / М. В. Омельченко, К. А. Плотникова, В. С. Мокшин // Инновационная наука. – 2020. – № 2. – С. 37-40. – EDN EMPOFR.

8. Кирилова, Д. А. Blockchain, как новая технология для разработки / Д. А. Кирилова, Н. С. Маслов, А. Д. Рейн // International Journal of Open Information Technologies. – 2019. – Т. 7, № 1. – С. 34-38. – EDN YSKGUX.

9. Дьяченко, С. В. Технологии обработки транзакций в blockchain / С. В. Дьяченко, Е. Г. Бойко // Естественно-гуманитарные исследования. – 2018. – № 20(2). – С. 6-10. – EDN XRGERN.

10. Хазимухаметов, И. Т. Использование криптографических методов в операционных системах Windows / И. Т. Хазимухаметов, С. Л. Перевалова // Тенденции развития науки и образования. – 2024. – № 108-12. – С. 169-173. – DOI 10.18411/trnio-04-2024-684. – EDN ZHBKKC.

11. Криптографические методы защиты информации и VPN в IP сетях / П. М. Мовсарова, Х. Р. Визирова, М. А. Бийсултанова [и др.] // Тенденции развития науки и образования. – 2019. – № 56-2. – С. 55-58. – DOI 10.18411/lj-11-2019-38. – EDN FHQSLT.

12. Черепанов, А. С. Современные протоколы для построения виртуальных частных сетей / А. С. Черепанов, Е. В. Шарлаев // Измерение, контроль, информатизация: Материалы XXIV Международной научно-технической конференции, Барнаул, 19 мая 2023 года / Под редакцией Л.И. Сучковой. – Барнаул: Алтайский государственный технический университет им. И.И. Ползунова, 2023. – С. 262-270. – EDN BIXKMN.

13. Исламгалеев, Д. Р. Протоколы TLS/SSL: безопасная передача данных в интернете / Д. Р. Исламгалеев // Научно-исследовательский центр "Technical Innovations". – 2023. – № 17. – С. 193-200. – EDN OLOCPB.

14. Горкавенко, В. С. Модель и проектные диаграммы системы централизованной защиты подсети хостов под управлением Unix-подобных операционных систем / В. С. Горкавенко, И. М. Ажмухамедов // Прикаспийский журнал: управление и высокие технологии. – 2019. – № 1(45). – С. 172-181. – EDN ZSAIWV.

15. Бутенко, Е. Д. Биткойн. Состояние и перспективы развития криптовалюты / Е. Д. Бутенко // Финансы и кредит. – 2014. – № 23(599). – С. 44-47. – EDN SESIXX.

16. Усков, А. В. Технологии обеспечения информационной безопасности корпоративных образовательных сетей / А. В. Усков, А. Д. Иванников, В. Л. Усков // Образовательные технологии и общество. – 2008. – Т. 11, № 1. – С. 472-479. – EDN IIYTJV.

17. Колпакова, А. А. Обзор технологий удаленного сетевого доступа / А. А. Колпакова // Интерэкспо Гео-Сибирь. – 2023. – Т. 7, № 1. – С. 253-260. – EDN NSCAJK.

18. Смирнов, Д. В. Обеспечение безопасности системы электронной почты от типа атак "человек посередине" на основе блокчейна / Д. В. Смирнов, А. И. Сотников, А. Ю. Асеев // Вестник Череповецкого государственного университета. – 2018. – № 4(85). – С. 31-38. – DOI 10.23859/1994-0637-2018-4-85-4. – EDN SJMLDX.

19. Минаев, В. А. Атаки по времени на информацию в недоверенных средах / В. А. Минаев, Е. В. Зеленцова, С. С. Петров // Моделирование, оптимизация и информационные технологии. – 2018. – Т. 6, № 4(23). – С. 456-468. – DOI 10.26102/2310-6018/2018.23.4.034. – EDN YZSOLB.

20. Жуков, А. Е. Криптоанализ по побочным каналам (Side Channel Attacks) / А. Е. Жуков // Защита информации. Инсайд. – 2010. – № 5(35). – С. 28-33. – EDN TMLOOT.

21. Квантовый хакинг: атаки на линии связи / Р. А. Белера, Д. Д. Деев, И. А. Смирнов [и др.] // Научное обозрение. Технические науки. – 2020. – № 4. – С. 7-13. – EDN AVTTCE.

22. Белослюдов, А. С. Обнаружение физических атак повторного воспроизведения речи с помощью легкой сверточной сети с графовым вниманием / А. С. Белослюдов, А. А. Лепендин, Я. А. Филин // Проблемы правовой и технической защиты информации. – 2023. – № 11. – С. 8-15. – EDN WFWMGU.

***Чернышова А.В., Коржов А. В. Анализ эффективности протоколов передачи данных, использующих криптографические алгоритмы на эллиптических кривых***. *В статье рассмотрено понятие эллиптической криптографии, рассмотрены основные современные протоколы передачи данных, которые применяют в своей основе эллиптическую криптографию, выявлены основные достоинства и недостатки данных протоколов. Также в статье приведён анализ основных уязвимостей к атакам современных протоколов передачи данных, использующих криптографию на эллиптических кривых.*

***Ключевые слова:*** *криптография, протокол, SSL/TLS, алгоритм, ECDHE, ECDSA, Signal Protocol, SSH, Bitcoin, IPSec, FIDO, WireGuard, преимущество, недостаток, уязвимость, атака.*

***Chernyshova A.V., Korzhov A.V. Effectiveness analysis of data transfer protocols using elliptic curve cryptographic algorithms.*** *This paper examines the concept of elliptic curve cryptography and analyzes major contemporary data transfer protocols based on elliptic curve cryptography, identifying their key advantages and disadvantages. The study also presents an analysis of primary vulnerabilities to attacks in modern data transfer protocols that implement elliptic curve cryptography.*

***Keywords:*** *cryptography, protocol, SSL/TLS, algorithm, ECDHE, ECDSA, Signal Protocol, SSH, Bitcoin, IPSec, FIDO, WireGuard, advantage, disadvantage, vulnerability, attack.*

УДК 519.866:336.748

# Методы и инструменты для прогнозирования курса валют

**Л.А. Лазебная[*1], Д.М. Зеленский[*2]**
[*1] к.т.н, доцент, Донецкий национальный технический университет,
L_Lazebnay@mail.ru, SPIN-код: 3146-9301, AuthorID: 847184
[*2] магистрант, Донецкий национальный технический университет,
dimitrizelenskiy@yandex.ru

***Лазебеная Л.А., Зеленский Д.М. Методы и инструменты для прогнозирования курса валют.*** *Статья посвящена комплексному исследованию методов и инструментов прогнозирования валютных курсов. В работе рассматриваются как классические подходы к прогнозированию, включая фундаментальный и технический анализ, так и современные методы, основанные на машинном обучении и технологиях больших данных. Особое внимание уделяется теоретическим основам валютного прогнозирования, факторам, влияющим на формирование курсов валют, и различным концептуальным моделям. Проведен сравнительный анализ эффективности традиционных и инновационных методов прогнозирования, оценка их достоверности и точности. Исследование представляет интерес как для специалистов в области финансовых рынков, так и для исследователей, занимающихся разработкой и применением современных технологий прогнозирования.*

***Ключевые слова:*** *прогнозирование, валютный курс, эконометрические модели, машинное обучение, нейронные сети, большие данные.*

## *Введение*

В современном мире валютный рынок играет ключевую роль в глобальной экономике. Колебания курсов валют влияют на множество аспектов экономической деятельности, от международной торговли до инвестиционных решений. В связи с этим, прогнозирование валютных курсов становится важнейшим инструментом для принятия обоснованных финансовых решений.

Актуальность исследования методов и инструментов прогнозирования курса валют обусловлена несколькими факторами. Во-первых, растущая нестабильность мировой экономики и геополитическая напряженность приводят к повышенной волатильности на валютных рынках. Это создает потребность в более точных и надежных методах прогнозирования.

Во-вторых, развитие технологий, в частности, искусственного интеллекта и машинного обучения, открывает новые возможности для анализа больших объемов данных и выявления сложных закономерностей в динамике валютных курсов. Это позволяет создавать более совершенные модели прогнозирования.

В-третьих, глобализация финансовых рынков и увеличение числа участников валютных операций повышают спрос на качественные прогнозы со стороны различных экономических агентов: от крупных корпораций и банков до частных инвесторов и трейдеров.

Цель данной работы заключается в систематизации и анализе существующих методов и инструментов прогнозирования курса валют, а также в оценке их эффективности в современных экономических условиях.

## *Понятие и факторы, влияющие на курс валют*

Курс обмена валют — это стоимость одной валюты, выраженная в единицах другой валюты. Это ключевой показатель, который отражает состояние экономики страны и её положение на международной арене.

На формирование курса обмена валют влияет множество факторов, которые можно разделить на несколько категорий:

Экономические факторы:

- Паритет покупательной способности.
- Состояние платёжного баланса страны.
- Уровень инфляции.

- Разница процентных ставок.
- Государственный долг.

Политические факторы:

- Стабильность политической системы.
- Геополитическая обстановка.
- Внешнеэкономическая политика государства.

Спекулятивные факторы:

- Действия крупных игроков на рынке валют.
- Ожидания участников рынка.

Технические факторы:

- Исторические тенденции и паттерны на графиках.

### *Основные подходы к прогнозированию финансовых рынков*

В области прогнозирования финансовых рынков, включая валютный рынок, выделяют три основных подхода:

1.Фундаментальный анализ: основан на изучении макроэкономических показателей, политических событий и других факторов, влияющих на экономику страны и, следовательно, на курс ее валюты. Этот метод предполагает анализ экономических отчетов, новостей и заявлений официальных лиц.

2.Технический анализ: базируется на исследовании исторических данных о движении цен и объемов торгов. Технические аналитики используют различные индикаторы и графические паттерны для предсказания будущих движений курса.

3.Сентимент-анализ: изучает настроения и ожидания участников рынка. Этот подход учитывает психологические факторы, влияющие на принятие решений трейдерами и инвесторами. [1]

### *Прогнозирование валютных курсов: ключевые концепции и модели*

В области прогнозирования валютных курсов существует несколько ключевых теорий и моделей:

Теория паритета покупательной способности (ППС) предполагает, что обменные курсы между валютами будут стремиться к уровню, при котором покупательная способность каждой валюты в разных странах будет одинаковой.

Модель непокрытого паритета процентных ставок утверждает, что разница в процентных ставках между двумя странами должна быть равна ожидаемому изменению обменного курса их валют.

Монетарная модель связывает изменения валютного курса с изменениями в предложении денег, реальном доходе и процентных ставках в соответствующих странах.

Модель платёжного баланса рассматривает влияние торгового баланса и потоков капитала на валютный курс.

Портфельная модель учитывает влияние международных инвестиционных потоков на формирование валютных курсов.

Модели временных рядов используют статистические методы для анализа исторических данных и прогнозирования будущих значений курса.

Модели на основе машинного обучения применяют алгоритмы искусственного интеллекта для выявления сложных закономерностей в движении валютных курсов.

Каждая из этих теорий и моделей имеет свои преимущества и ограничения. На практике часто используется комбинация различных подходов для повышения точности прогнозов. Важно отметить, что ни одна модель не может гарантировать абсолютно точный прогноз из-за сложности и динамичности валютного рынка.

### *Фундаментальный анализ*

Фундаментальный анализ – это метод прогнозирования валютного курса, основанный на изучении экономических, политических и социальных факторов, влияющих на стоимость валюты. Этот подход предполагает тщательное исследование макроэкономических показателей и событий, которые могут оказать влияние на валютный рынок.

Основные элементы фундаментального анализа включают:

1.Анализ экономических показателей: ВВП, инфляция, уровень безработицы, промышленное производство, розничные продажи.

2.Оценка монетарной политики: изучение решений центральных банков, изменений процентных ставок, объемов денежной массы.

3.Анализ платежного баланса: исследование торгового баланса, потоков капитала, прямых иностранных инвестиций.

4.Политические факторы: оценка стабильности правительства, геополитических рисков, изменений в законодательстве.

5.Анализ рыночных настроений: изучение ожиданий инвесторов, настроений потребителей и бизнеса.

Преимущество фундаментального анализа заключается в его способности учитывать широкий спектр факторов, влияющих на валютный курс. Однако он требует глубоких знаний экономики и постоянного мониторинга большого объема информации. [2]

### *Технический анализ*

Технический анализ – это метод прогнозирования, основанный на изучении исторических данных о движении цен и объемов торгов. Этот подход предполагает, что все фундаментальные факторы уже учтены в цене, и будущее движение курса можно предсказать, анализируя графики и статистические данные.

Основные инструменты технического анализа:

1.Графические паттерны: фигуры продолжения и разворота тренда.

2.Индикаторы тренда: скользящие средние, MACD, ADX.

3.Осцилляторы: RSI, Stochastic, CCI.

4.Уровни поддержки и сопротивления.

5.Волновая теория Эллиотта и числа Фибоначчи.

Технический анализ позволяет быстро обрабатывать большие объемы данных и выявлять краткосрочные тенденции. Однако он может быть менее эффективен при долгосрочном прогнозировании и не учитывает фундаментальные факторы, влияющие на рынок.

### *Эконометрические модели*

Методы эконометрического анализа — это математические инструменты, которые используются для прогнозирования курсов валют на основе анализа статистических данных.

1.Одним из самых простых методов является линейная регрессия. В этой модели курс валюты рассматривается как функция от одной или нескольких переменных. Например, $Y = \beta0 + \beta1X1 + \beta2X2 + \ldots + \beta nXn + \varepsilon$, где Y — курс валюты, Xi — независимые переменные (экономические показатели), βi — коэффициенты, ε — случайная ошибка.

2.Модели ARIMA (AutoRegressive Integrated Moving Average) используются для анализа временных рядов и учитывают автокорреляцию в данных. Общая форма модели ARIMA (p, d, q): $Yt = c + \varphi1Yt-1 + \ldots + \varphi pYt-p + \theta1\varepsilon t-1 + \ldots + \theta q\varepsilon t-q + \varepsilon t$, где p — порядок авторегрессии, d — порядок интегрирования, q — порядок скользящего среднего.

3.Векторные авторегрессионные модели (VAR) используются для анализа взаимосвязей между несколькими временными рядами. Каждая переменная моделируется как функция прошлых значений всех переменных в системе.

4.Модели с условной гетероскедастичностью (GARCH) применяются для моделирования волатильности курсов валют, учитывая изменение дисперсии ошибок во времени.

Методы эконометрического анализа позволяют определить, как различные факторы влияют на стоимость валюты в количественном выражении. Также они дают возможность проверять теории с помощью статистических тестов. Однако для применения этих методов необходимо большое количество исторических данных, и они могут быть чувствительны к изменениям в экономической структуре.

Каждый из этих классических методов имеет свои преимущества и недостатки. На практике часто наиболее эффективным оказывается сочетание различных методов, чтобы получить больше информации и повысить точность прогнозов. [3]

### *Методы машинного обучения в прогнозировании*

Машинное обучение (МО) представляет собой подход, при котором алгоритмы автоматически улучшают свою производительность с опытом. В контексте прогнозирования валютных курсов методы МО способны обрабатывать большие объемы данных и выявлять сложные нелинейные зависимости.

Основные методы МО, применяемые в прогнозировании валютных курсов:

1.Деревья решений и случайные леса: Эти методы строят иерархическую структуру правил для принятия решений. Случайные леса, состоящие из множества деревьев решений, особенно эффективны для обработки нелинейных зависимостей в данных.

2.Метод опорных векторов (SVM): SVM ищет оптимальную гиперплоскость для разделения классов

данных. В задачах прогнозирования SVM может использоваться для классификации направления движения курса или для регрессии при прогнозировании конкретных значений.

3.Градиентный бустинг: Этот метод последовательно обучает множество слабых моделей, каждая из которых корректирует ошибки предыдущих. Алгоритмы вроде XGBoost и LightGBM часто показывают высокую точность в задачах прогнозирования.

4.Кластерный анализ: Методы кластеризации, такие как K-средних или DBSCAN, могут использоваться для выявления групп схожих рыночных состояний, что помогает в прогнозировании будущего поведения курса.

Преимущество методов МО заключается в их способности автоматически адаптироваться к изменяющимся рыночным условиям и обрабатывать сложные взаимосвязи между множеством факторов. [4]

### *Нейронные сети и глубокое обучение*

Нейронные сети и методы глубокого обучения представляют собой подмножество машинного обучения, вдохновленное структурой и функционированием человеческого мозга. Эти методы особенно эффективны при работе с большими объемами данных и сложными паттернами.

Основные типы нейронных сетей, применяемых в прогнозировании валютных курсов:

1.Многослойные персептроны (MLP): Классические полносвязные нейронные сети, способные моделировать сложные нелинейные зависимости между входными данными и прогнозируемым курсом.

2.Сверточные нейронные сети (CNN): Изначально разработанные для обработки изображений, CNN также применяются для анализа временных рядов, выявляя локальные паттерны в данных о валютных курсах.

3.Рекуррентные нейронные сети (RNN) и LSTM: Эти архитектуры специально разработаны для обработки последовательных данных. LSTM (Long Short-Term Memory) особенно эффективны для моделирования долгосрочных зависимостей в финансовых временных рядах.

4.Автоэнкодеры: Используются для уменьшения размерности входных данных и выделения наиболее информативных признаков, что может улучшить качество прогнозирования.

Глубокое обучение позволяет создавать модели, способные автоматически извлекать сложные признаки из сырых данных, что особенно ценно при работе с разнородной информацией, влияющей на валютные курсы.

### *Алгоритмы с применением больших данных (Big Data)*

Технологии Big Data позволяют обрабатывать огромные объемы структурированных и неструктурированных данных, что открывает новые возможности для прогнозирования валютных курсов.

Основные направления применения Big Data в прогнозировании:

1.Анализ новостных потоков: Алгоритмы обработки естественного языка применяются для анализа новостей, социальных медиа и других текстовых источников, оценивая их влияние на валютный рынок.

2.Анализ рыночных микроструктур: Обработка данных о каждой транзакции (тиковых данных) позволяет выявлять краткосрочные паттерны в поведении рынка.

3.Интеграция разнородных источников данных: Объединение экономических показателей, технических индикаторов, новостных событий и других факторов для создания комплексных прогностических моделей.

4.Распределенные вычисления: Использование технологий типа Hadoop и Spark для обработки больших объемов данных и обучения сложных моделей на распределенных кластерах.

Применение технологий Big Data позволяет учитывать широкий спектр факторов, влияющих на валютный курс, и обрабатывать данные в режиме реального времени, что повышает точность и актуальность прогнозов.

Современные инструменты прогнозирования, основанные на машинном обучении, нейронных сетях и технологиях Big Data, обладают высоким потенциалом для улучшения точности прогнозов валютных курсов. Однако их эффективное применение требует значительных вычислительных ресурсов, наличия качественных данных и глубокого понимания как финансовых рынков, так и алгоритмов машинного обучения. [5]

### *Преимущества и недостатки классических методов*

Классические методы прогнозирования курса валют, такие как фундаментальный анализ, технический анализ и эконометрические модели, до сих пор активно используются.

Фундаментальный анализ учитывает макроэкономические факторы, такие как ВВП, инфляция, процентные ставки и торговый баланс, и глубоко связан с реальной экономикой. Он требует детального понимания экономики и доступа к актуальной информации, но плохо справляется с краткосрочными колебаниями курса, ориентируясь на долгосрочные тенденции.

Технический анализ использует исторические данные о движении цены, графические модели, индикаторы и осцилляторы, и доступен для любых финансовых инструментов. Он прост в освоении, но не учитывает внешние экономические факторы и не всегда предсказывает сильные ценовые колебания.

Эконометрические модели, такие как ARIMA и модели временных рядов, используют статистические методы для выявления закономерностей и требуют значительных вычислительных ресурсов и качественных данных для построения точных прогнозов. Они объективны и научно обоснованы, но не всегда возможны в условиях волатильного валютного рынка.

### *Анализ достоверности современных методов*

Современные методы прогнозирования валютных курсов стали возможны благодаря развитию технологий машинного обучения и искусственного интеллекта. Они позволяют более точно и оперативно предсказывать изменения курсов валют, но их эффективность зависит от применяемой методики, объёма данных и условий на рынке.

Основные инструменты, используемые для прогнозирования, включают нейронные сети, модели глубокого обучения, алгоритмы машинного обучения и гибридные системы.

Нейронные сети обучаются на исторических данных и выявляют сложные зависимости между макроэкономическими и техническими факторами. Они способны адаптироваться и корректировать прогнозы по мере изменения рыночной ситуации.

Рекуррентные нейронные сети (RNN) и LSTM (Long Short-Term Memory) эффективны в краткосрочном прогнозировании, но зависят от настройки модели и объёма данных. При недостатке данных и переобучении модели могут давать неверные прогнозы.

Модели глубокого обучения, особенно в сочетании с алгоритмами обработки больших данных, показали высокую точность в прогнозировании валютных курсов. Они учитывают множество факторов, включая макроэкономические индикаторы, рыночные настроения, технические индикаторы и новостные потоки. Однако для обучения моделей глубокого обучения требуются значительные вычислительные мощности и данные.

Алгоритмы машинного обучения, такие как случайные леса (Random Forest), градиентный бустинг (XGBoost) и метод опорных векторов (SVM), гибкие и адаптируются к различным типам данных. Они способны выявлять как линейные, так и нелинейные зависимости и минимизировать риск ошибок переобучения. Однако они требуют качественных данных и сложной интерпретации.

Гибридные модели объединяют преимущества различных подходов, компенсируя недостатки отдельных методов для повышения точности. Они эффективны в условиях высокой волатильности, но не учитывают неожиданные изменения.

Точность современных методов высока в краткосрочной перспективе, но требует доработки для долгосрочных прогнозов и учёта редких событий. [6]

### *Анализ и оценка точности реальных прогнозов*

В качестве иллюстрации применения современных методов прогнозирования можно привести прогноз курса евро к доллару на 2020 год, выполненный с использованием нейронных сетей. На основе анализа макроэкономических данных и технических индикаторов модель предсказала укрепление евро по отношению к доллару в первом квартале. Прогноз оказался верным: евро действительно вырос на 3%, что подтверждает высокую точность краткосрочных прогнозов, полученных с помощью современных методов.

Другим примером может служить прогнозирование курса рубля с помощью эконометрических моделей в 2014 году, когда введение санкций привело к резким колебаниям на валютном рынке. Классические модели не смогли учесть столь значительное воздействие внешнеполитических факторов, что привело к ошибкам в прогнозировании. В результате рубль значительно обесценился по отношению к доллару, что не было предсказано заранее.

Эти примеры демонстрируют, что, несмотря на более высокую точность, современные методы не всегда способны полностью предсказать поведение валют на долгосрочной перспективе, особенно в условиях возникновения неожиданных экономических и политических событий.

### *Литература*

1.Агеев А.И., Глазьев С.Ю., Митяев Д.А., Золотарева О.А., Переслегин С.Б. Построение модели прогноза курса валют на долгосрочном и краткосрочном горизонтах // Экономические стратегии. 2022. № 6 (186). С. 16 - 25. DOI: 10.33917/es-6.186.2022.16-25. EDN: CREHGS. URL: https://www.researchgate.net/publication/370633353_Postroenie_modeli_prognoza_kursa_valut_na_dolgosrocnom_i_kratkosrocnom_gorizontah (дата обращения: 05.11.2024)

2.Братюхин К. В. Сравнение теоретических методов анализа и прогнозирования динамики валютного курса. // Научный альманах Центрального Черноземья. Учредители: Цифровая академия. — С. 124–135. URL: https://www.elibrary.ru/item.asp?id=54078502 (дата обращения: 05.11.2024)

3.Погосян Э. А. Прогнозирование валютных курсов с использованием методов статистики // Научные записки молодых исследователей. 2015. №1. URL: https://cyberleninka.ru/article/n/prognozirovanie-valyutnyh-kursov-s-ispolzovaniem-metodov-statistiki (дата обращения: 05.11.2024).

4.Луппова В. В. Инструментальные императивы в моделировании валютных курсов // Наука, образование и культура. 2016. №7 (10). URL: https://cyberleninka.ru/article/n/instrumentalnye-imperativy-v-modelirovanii-valyutnyh-kursov (дата обращения: 05.11.2024).

5.Иванов М. А. и др. Разработка модели прогнозирования валютного курса USD/RUB на основе новостного фона с использованием искусственных нейронных сетей: магистерская диссертация по направлению подготовки: 38.04. 01 - Экономика. – 2020. URL: https://vital.lib.tsu.ru/vital/access/services/Download/vital:11598/SOURCE01 (дата обращения: 05.11.2024).

6.Люкшин А. М. Прогнозирование валютного курса в современных экономических условиях // Финансовая жизнь. – 2018. – №. 4. – С. 78-84. URL: https://elibrary.ru/item.asp?id=36579902 (дата обращения: 05.11.2024).

***Лазебеная Л.А., Зеленский Д.М. Методы и инструменты для прогнозирования курса валют.*** *Статья посвящена комплексному исследованию методов и инструментов прогнозирования валютных курсов. В работе рассматриваются как классические подходы к прогнозированию, включая фундаментальный и технический анализ, так и современные методы, основанные на машинном обучении и технологиях больших данных. Особое внимание уделяется теоретическим основам валютного прогнозирования, факторам, влияющим на формирование курсов валют, и различным концептуальным моделям. Проведен сравнительный анализ эффективности традиционных и инновационных методов прогнозирования, оценка их достоверности и точности. Исследование представляет интерес как для специалистов в области финансовых рынков, так и для исследователей, занимающихся разработкой и применением современных технологий прогнозирования.*

***Ключевые слова:*** *прогнозирование, валютный курс, эконометрические модели, машинное обучение, нейронные сети, большие данные.*

***Lazebenaya L.A., Zelensky D.M. Methods and tools for currency exchange rate forecasting.*** *The article is devoted to a comprehensive study of methods and tools for forecasting currency exchange rates. The paper examines both classical forecasting approaches, including fundamental and technical analysis, and modern methods based on machine learning and big data technologies. Special attention is paid to the theoretical foundations of currency forecasting, factors affecting exchange rate formation, and various conceptual models. A comparative analysis of the effectiveness of traditional and innovative forecasting methods, assessment of their reliability and accuracy is conducted. The research is of interest both for financial market specialists and researchers involved in the development and application of modern forecasting technologies.*

***Keywords:*** *forecasting, exchange rate, econometric models, machine learning, neural networks, big data.*

# СЕКЦИЯ 3. «МЕТОДЫ, ТЕХНОЛОГИИ И СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

УДК 007.5+519.816

## Формирование образов в окружении выбора мобильного робота и их классификация с использованием нейронных сетей и алгоритмов логико-лингвистической классификации

**А.Ю. Кучмин [*1], И.Л. Тарасова [*2]**

[*1] д.т.н, Институт проблем машиноведения РАН,
kayu@ipme.ru, OrcID: 0000-0003-0699-6112, SPIN-код: 6671-3874

[*2] к.т.н., Институт проблем машиноведения РАН,
til@ipme.ru, OrcID: 0000-0002-2282-150X, SPIN-код: 6680-8214

***Кучмин А.Ю., Тарасова И.Л. Формирование образов в окружении выбора мобильного робота и их классификация с использованием нейронных сетей и алгоритмов логико-лингвистической классификации.*** *В статье представлен краткий обзор алгоритмов формирования решающих правил для построения классификационных моделей и приведены рекомендации по их применению. Рассмотрен алгоритм формирования атрибутов образов в окружении выбора мобильного робота, на основе которых построены классификационные модели с использованием нейронных сетей и алгоритмов логико-лингвистической классификации. Машиннные эксперименты позволили провести оценку точности классификации различных изображений, которые показали, что при уровне помехи выше 80% и больших выборках лучше использовать нейросетевые алгоритмы классификации, а алгоритмы логико-лингвистической классификации более эффективны при малых выборках и уровне помехи 50%-60%.*

***Ключевые слова****: нейронные сети, фаззификация, логико-лингвистическая классификация, логико-вероятностная классификация, классифицируемые изображения.*

### *Введение*

Для правильного принятия решения при управлении мобильными роботами [1], в том числе и беспилотными летательными аппаратами (БПЛА), особенно важно повышение точности и скорости классификации образов в окружении выбора. Это во многом определяет качество управления безаварийных ситуаций при выполнении поставленных задач: обнаружение целей, их идентификация (классификация), выбор метода, средств уничтожения и ликвидация [2]. Наиболее сложной из них и, в наибольшей степени, влияющей на достижение цели, является задача обнаружения и классификация образов (целей).

Управление такими объектами осуществляется, на основе информации, полученной от многочисленных средств измерений и измерительных модулей. Однако большое количество объектов функционирует в условиях неопределенности. Эта неопределенность может быть обусловлена рядом причин, одной из которых является погрешность или неточность измерений. С целью минимизации погрешности проводятся многократные повторные измерения изображений в одинаковых условиях с последующей обработкой полученных данных. К сожалению, провести многократные повторные измерения изображений для БПЛА, от многочисленных средств измерений, оказывается не всегда возможной. Это приводит к неполной и противоричивой информации, которая влияет на формирование и классификацию образов в окружении выбора. Классификация образов связана с поиском закономерностей на основе построения правил, способных объяснить имеющиеся факты и предсказывать новые или недостающие, присущие классифицируемым изображениям. Поэтому целью анализа полученных данных является построение классификационных моделей классов, и создание решающих правил отнесения рассматриваемых изображений к какому-либо классу образов [1,3,4].

Задачи формирования изображений в системах технического зрения описаны, например, в [5,6]. В результате работы указанных систем на основе анализа полученных данных об окружении выбора робота в базе данных формируется множество изображений $G = \{g_1,g_2,\ldots,g_n\}$. Кроме того, в базе данных робота обычно имеется множество эталонных образов (изображений) $O = \{o_1,o_2,\ldots, o_m\}$. Указанные множества являются упорядоченными, а их элементами являются логические переменные (ЛП), принимающие значение 0 или 1 [1].

При этом каждые изображение $g_i$ и эталонный образ $o_j$ характеризуются наборами признаков (атрибутов). Количество атрибутов фиксировано, значения атрибутов могут быть числовыми, логическими и символьными.

В настоящее время для решения этой проблемы чаще всего обращаются к использованию искусственных нейронных сетей (НС).

Нейросети позволяют создавать значения из большого количества неточных или сложных значений, а также аппроксимировать, классифицировать и распознавать более точно и быстро в сравнении с классическими алгоритмами. Хотя во многих задачах человеческий мозг превосходит возможности существующих на сегодня НС, их преимущества нельзя игнорировать, поэтому они имеют широкое применение. К достоинствам нейросетей можно отнести: самообучаемость, фильтрацию шумов и входных данных, адаптацию к новым входным данным, отказоустойчивость при повреждении некоторых нейронов, скорость работы и т.д.

### *1. Обобщенное описание задачи формирования классификационных моделей*

В общем случае построение классификационных моделей может быть следующим.

На первом шаге среди множества изображений $G$ выделяют множество изображений $g_{O1}$, относящихся к классу $o_{1,}$ которому присваивается имя этого класса. Затем среди множества $O^1=O\backslash g_{O1}$ выделяют множество изображений $g_{O2}$, относящихся к классу $o_2$, после чего среди множества оставшихся изображений $O^2 = O^1\backslash g_{O2}$ выделяют изображения $g_{O3}$, относящиеся к классу $o_{3,}$ и присваивается имя этого класса. Этот процесс будет продолжаться до тех пор, пока не будут исчерпаны все изображения из базы данных, т.е. $((((O\backslash g_{O1})\backslash g_{O2})\ldots)g_{Ok})=O^k$. Если будет так, что все изображения и эталонные образы данных исчерпаны, но останутся не проклассифицированные изображения, т.е. $O^k\neq\emptyset$, то этим изображениям присваивается имя нового класса ($k$+1) и они временно заносятся в базу данных с этим именем. Таких изображений может быть много. В этом случае может быть поставлена задача разбиения ($k$+1) класса на новые эталонные образы и соответственно, отнесение этих изображений к вновь введенным эталонам.

Обычно для выделения класса изображения используются нейронные сети [7, 8] с обучением сформированной выборки. Для выделения $O_q$-го класса из изображений можно построить обучающую выборку $K_q = K_q^+\cup K_q^-$, где $K_q^+\subset O_q$ и $K_q^-\subset G_q$. На основании обучающей выборки $K_q$ можно построить правило, сопоставляющее положительные (эталонные) и отрицательные образы (классифицируемые) обучающей выборки. Решающее правило является корректным, если оно в дальнейшем успешно распознаёт образы, не вошедшие первоначально в обучающую выборку.

### *2. Алгоритмы формирования решающих правил*

Известен ряд алгоритмов, формирующих решающие правила в виде дерева решений [9], или набора продукционных правил. Прежде всего, это алгоритмы UD3 и C4.5 [10,11], алгоритм CN2 [12] и ряд других.

Результаты исследований и экспериментов показали преимущества и недостатки этих алгоритмов. В частности, при уровне шума до 25% при различных обучающих выборках точность классификации по алгоритму UD3 лежала в пределах от 78% до 85%, а по алгоритму C4.5 – в пределах от 75% до 80%. При более высоком уровне шума подобные алгоритмы становятся не эффективными. Для преодоления этих ограничений предлагается использовать так называемые структурно-лингвистические методы и искусственные нейронные сети. Структурно-лингвистический метод лег в основу алгоритмов логико-вероятностной (ЛВК) и логико-лингвистической классификаци (ЛЛК), который использует, например, отбор признаков на основе вероятностного подхода перекрестной энтропии [13,14]. В [1] предлагаются логико-вероятностные алгоритмы решения задачи классификации, в которых классификация осуществляется за счет вычисления минимальной суммы квадратов разностей значений вероятностей элементов строк атрибутов эталонов и классифицируемых изображений.

Однако в реальных условиях эксплуатации БПЛА получение данных об атрибутах образов и определение их вероятностей в окружении выбора не всегда реализуемо. Значительно проще и быстрее вычислять значения функций принадлежности $\mu(.)$ атрибутов путем их фаззификации [15, 16]. Поэтому для классификации образов в окружении выбора БПЛА целесообразно использовать логико-лингвистические решающие правила, построенные на основе нечеткой или неполной информации [17] и соответствующие алгоритмы ЛЛК. При небольших объемах информации алгоритмы ЛВК и ЛЛК позволяют достигать высокую скорость классификации при точности классификации порядка 80% и уровне шума до (30-50)%.

В случае увелечения уровня шума более, чем на 50% от уровня сигнала можно обучить нейроную сеть. Но одним из основных недостатков НС является ее проблема подготовки выборки для обучения в связи с недостаточным количеством доступных материалов. Кроме того, реальные условия эксплуатации БПЛА, показывают, что получение данных об атрибутах образовов является сложной задачей и носит ограниченный характер. Поэтому машинное обучение НС может привести к тупику, а сам процесс будет занимать длительное время.

Решить эту задачу можно исскуственно, путем генерирования новых атрибутов образов, базирующихся на собранных данных, которые позволяют восполнять недостающую информацию об атрибутах и формировать новые образы, которые в дальнейшем используются при машинном обучении в нейросетях.

## *3. Компьютерное моделирование*

### *3.1. Входные данные*

При формировании образов рассматривались следующие группы: $s_1$- окружающая среда; $s_2$- наземная техника; $s_3$- плавсредства; $s_4$ - летательные аппараты; $s_5$ - люди.

В группе $s_1$ использовались эталоны следующих образов: $O_1(s_1)$ – Пустыня; $O_2(s_1)$ - Степь; $O_3(s_1)$ Море; $O_4(s_1)$ – Озеро; $O_5(s_1)$ – Река; $O_6(s_1)$ – Пруд; $O_7(s_1)$ – Горы; $O_8(s_1)$ - Скалы; $O_9(s_1)$ - Холмы; $O_{10}(s_1)$ - Лиственный лес; $O_{11}(s_1)$ - Хвойный лес; $O_{12}(s_1)$ - Смешанный лес; $O_{13}(s_1)$ - тропический лес.

В группе $s_2$ использовались эталоны следующих образов: $O_{14}(s_2)$ - Строительно-дорожный автомобиль; $O_{15}(s_2)$ - Грузовой автомобиль; $O_{16}(s_2)$ - Легковой автомобиль; $O_{17}(s_2)$ - Бронеавтомобиль; $O_{18}(s_2)$ - Бронетранспортер; $O_{19}(s_2)$ - Танк; $O_{20}(s_2)$ - САУ.

В группе $s_3$ использовались эталоны следующих образов: $O_{21}(s_3)$ - Яхта или катер; $O_{22}(s_3)$ - буксир или тральщик; $O_{23}(s_3)$ – Эсминец; $O_{24}(s_3)$ - Подводная лодка; $O_{25}(s_3)$ – Авианосец.

В группе $s_4$ использовались эталоны следующих образов: $O_{26}(s_4)$ – Беспилотник; $O_{27}(s_4)$ - Вертолет; $O_{28}(s_4)$ - Истребитель; $O_{29}(s_4)$ - Бомбардировщик.

В группе $s_5$ использовались эталоны следующих образов: $O_{30}(s_5)$ – Мужчина; $O_{31}(s_5)$ - Женщина; $O_{32}(s_5)$ - Старик; $O_{33}(s_5)$ - Старуха; $O_{34}(s_5)$ - Ребенок.

При этом все образы характеризовались следующим набором видов атрибутов:

Вид $Y_1$ *«Размер контура»*: $y_{11}$ - Совсем маленький; $y_{12}$ - Очень маленький; $y_{13}$ – Маленький; $y_{14}$ – Средний; $y_{15}$ – Большой; $y_{16}$ - Очень большой; $y_{17}$ - Совсем большой.

Вид $Y_2$ *«Соотношение сторон контура»*: $y_{21}$ - Длина гораздо больше ширины; $y_{22}$ - Длина больше ширины; $y_{23}$ - Длина незначительно больше ширины; $y_{24}$ - Длина равна ширине; $y_{25}$ - Высота (глубина) гораздо больше ширины; $y_{26}$ - Высота (глубина) больше ширины; $y_{27}$ - Высота (глубина) равна ширине.

Вид $Y_3$ *«Неровность поверхности»*: $y_{31}$ – Ровная; $y_{32}$ - Очень незначительная; $y_{33}$ - Очень небольшая; $y_{343}$ - Небольшая; $y_{35}$ - Большая; $y_{36}$ - Очень большая; $y_{37}$ - Очень значительная.

Вид $Y_4$ *«Скорость передвижения»*: $y_{41}$ – Нулевая; $y_{42}$ - Очень маленькая; $y_{43}$ – Незначительная; $y_{44}$ – Средняя; $y_{45}$ – Большая; $y_{46}$ - Очень большая; $y_{47}$ - Сверх большая.

Вид $Y_5$ *«Цвет поверхности»*: $y_{51}$ – Серый; $y_{52}$ – Голубой; $y_{53}$ – Синий; $y_{54}$ – Зеленый; $y_{55}$ – Салатный; $y_{56}$ – Оранжевый; $y_{57}$ – Желтый.

Вид $Y_6$ *«Температура поверхности»*: $y_{61}$ - Очень низкая; $y_{62}$ – Низкая; $y_{63}$ – Средняя; $y_{64}$ – Небольшая; $y_{65}$ – Большая; $y_{66}$ – Высокая; $y_{67}$ - Очень высокая.

Вид $Y_7$ *«Неравномерность температуры»*: $y_{71}$ – Равномерная; $y_{72}$ - Немного неравномерная; $y_{73}$ - Средне неравномерная; $y_{74}$ - Много неравномерная; $y_{75}$ - Сильно неравномерная; $y_{76}$ - Очень сильно неравномерная; $y_{77}$ - Значительно неравномерная.

Вид $Y_8$ *«Громкость звука»*: $y_{81}$ - Нет или неопределенная; $y_{82}$ - Очень маленькая; $y_{83}$ - Маленькая; $y_{84}$ – Средняя; $y_{85}$ – Переменная; $y_{86}$ – Большая; $y_{87}$ - Очень большая.

Вид $Y_9$ *«Тональность звука»*: $y_{91}$ – Неопределенная; $y_{92}$ – Переменная; $y_{93}$ - Очень высокая; $y_{94}$ – Высокая; $y_{95}$ – Средняя; $y_{96}$ – Низкая; $y_{97}$ - Очень низкая.

Вид $Y_{10}$ *«Запах»*: $y_{101}$ - Нет или неопределенный; $y_{102}$ - Немного хвойный; $y_{103}$ – Хвойный; $y_{104}$ - Немного цветочный; $y_{105}$ – Цветочный; $y_{106}$ - Немного бензиновый; $y_{107}$ – Бензиновый.

### *3.2. Формирование атрибутов образов*

В общем виде строки значений функций принадлежности атрибутов можно представить, как

$O_1 /\mu_{11}\ \mu_{12}\ \mu_{13}\ \ldots\ \mu_{1N} /,\ O_2 /\mu_{21}\ \mu_{22}\ \mu_{23}\ \ldots\ \mu_{2N} /,\ O_3/\mu_{31}\ \mu_{32}\ \mu_{33}\ \ldots.\ \mu_{3N} /,\ \ldots\ O_K\ /\mu_{K1}\ \mu_{K2}\ \mu_{K3}\ \ldots\ldots\ldots\ \mu_{KN} /$, где: $O_i$ – $i$-ый эталонный образ, $y_{ij}$ – $ij$ –ый атрибут образа, $K$-общее количество эталонных образов ($K$=34), $N$ – общее число атрибутов для каждого образа ($N$=70), $\mu_{i1,}\ \mu_{i2,}\ \mu_{i3,}\ \ldots\ \mu_{iN,}$ - априорные значения функции принадлежности каждого атрибута, устанавливаемые экспертами.

Значения функций принадлежности атрибутов нового изображения формировались из значений функций принадлежности атрибутов эталонов путем добавления к ним случайного шума, уровень которого лежал в пределах эталонного сигнала и менялся вплоть до получения уровня шума в 100% от уровня сигнала.

Алгоритм формирования нового изображения $G_i$ был следующим:

1. Формирование БД строк значений функций принадлежности атрибутов эталонных изображений $O_1(s_1)$, … $O_{34}(s_5)$;

$O_1(s_1)$ = / 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0.1 0.2 0.5 0.2 0 0 0 0.9 0.1 0 0 0 0 0 0.3 0 0 0 0 0.3 0.4 0 0 0.2 0.5 0.3 0 0 0.3 0.5 0.2 0 0 0 0 0.5 0.5 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 / пустыня;

$O_2(s_1)$ = / 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0.1 0 0.4 0.5 0 0 0 0 1 0 0 0 0 0 0 0.1 0 0 0 0.3 0.3 0 0 0.3 0 0 0.2 0.5 0.3 0 0 0 0.4 0.6 0 0 0 0 0 0 0.4 0.6 0 0 0 0 0 0 0.6 0.4 0 0 0 0 0 0 0 0 0 0.4 0.6 0 0 / степь;

$O_3(s_1)$ =/ 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0.2 0.5 0.3 0 0 0 0 0.1 0.4 0.5 0 0 0 0 0 0.4 0.4 0.2 0 0 0 0 0 0.1 0.3 0.6 0 0 0 0.7 0.3 0 0 0 0 0 0 0 0.1 0.2 0.4 0.2 0.1 0 0 0 0.3 0 0 0.6 0.1 0 1 0 0 0 0 0 0 / море;

$O_4(s_1)$ = / 0 0 0 0 0 0 1 0 0 0 0 0.2 0.8 0 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0.2 0.8 0 0 0 0 0 0 0.3 0.4 0.3 0 0 0 0 0 0 0.2 0.3 0.5 0 0 0 0.6 0.3 0.1 0 0 0 0 0 0 0.4 0.6 0 0 0 0 0 0.6 0.4 0 0 0 0 0 0 1 0 0 0 0 0 0 / озеро;

$O_5(s_1)$ = / 0 0 0 0 0 1 0 0 0 0.7 0.3 0 0 0 0 0 0 0 0 0.1 0.5 0.3 0.2 0 0 0 0.1 0 0 0.9 0 0 0 0 0 0.4 0.4 0.2 0 0 0 0 0 0 0 0.2 0.4 0.4 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0 0 0.4 0.6 0 0 0 0 0 0 0 0 0.7 0.3 0 0 1 0 0 0 0 0 0 / река;

$O_6(s_1)$ = / 0 0 0 1 0 0 0 0 0 0 0.3 0.7 0 0 0 0 0.1 0.5 0.3 0.1 0 0 0 0 0.4 0.6 0 0 0 0 0 0 0.4 0.2 0.1 0.3 0 0 0 0 0 0.2 0.2 0.6 0 0 0 0.7 0.3 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0.8 0 0 0 0 0.2 0 0 / пруд;

$O_7(s_1)$ =/ 0 0 0 0 0 0 0 0.6 0.4 0 0 0 0.5 0.5 0.6 0.4 0 0 0 0 0 0 0.6 0.4 0 0 0.1 0.6 0 0 0.3 0 0 0 0 0.5 0.3 0 0 0.2 0 1 0 0 0.1 0.2 0.5 0.2 0 0 0 0 0 0 0.3 0.7 0 0 0 0 0 0 0.4 0.6 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0.8 0 0 0.2 0 0 0 / горы;

$O_8(s_1)$ =/ 0 0 0 0 0 0 0 0.6 0.4 0 0 0 0.2 0.3 0.5 0.7 0.3 0 0 0 0 0 0 0 0 0.6 0.4 0.4 0.6 0 0 0 0 0 0 0.5 0 0 0 0 0 0.3 0.2 0 0 0.3 0.4 0.3 0 0 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0.4 0 0 0.6 0 0 0 1 0 0 0 0 0 0 / скалы;

$O_9(s_1)$ = / 0 0 0 0 1 0 0 0 0 0 0.2 0.8 0 1 0 0 0 0 0 0 0.7 03 0 0 0.2 0.6 0 0 0.2 0 0 0 0.1 0 0 0 0.5 0.2 0.2 0 0 0 0.2 0.2 0.6 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0.5 0.5 0 0 0 0 1 0 0 0 0 0 0 0 0 0.3 0 0 0.7 0 0 0 / холмы;

$O_{10}(s_1)$ = / 0 0 0 0 0 0 0.8 0.2 0 0 0 0 0.4 0.6 0 0 0 0 0 0.1 0 0 0.6 0.3 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0.7 0 0 0 0.3 0 0 0.2 0.6 0.2 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0.2 0.5 0.3 0 0 0 0 0 0.4 0 0 0.6 0 0 1 0 0 0 0 0 0 / лиственный лес;

$O_{11}(s_1)$ = / 0 0 0 0 0 0.7 0.3 0 0 0 0 0.3 0.7 0 0 0 0 0 0.1 0 0 0.5 0.4 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0.2 0.5 0.3 0 0 0 0 0.6 0.3 0.1 0 0 0 0 0 0.1 0.5 0.4 0 0 0 0 0 0.3 0 0 0.7 0 0 0 0 1 0 0 0 0 / хвойный лес;

$O_{12}(s_1)$ = / 0 0 0 0 0 0 0.6 0.4 0 0 0 0 0.4 0.6 0 0 0 0 0 0.1 0 0 0.6 0.3 0 0 0 0 1 0 0 0 0 0 0 0 0 0.7 0 0 0.3 0 0 0 0.2 0.6 0.2 0 0 0 0 0.6 0.4 0 0 0 0 0 0 0.2 0.5 0.3 0 0 0 0 0 0.4 0 0 0.6 0 0 0 1 0 0 0 0 0 0 / смешанный лес;

$O_{13}(s_1)$ = / 0 0 0 0 0 0 0.5 0.5 0 0 0 0 0.3 0.7 0 0 0 0 0 0.1 0 0 0.7 0.2 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0.2 0.4 0.4 0 0 0 1 0 0 0 0 0 0 0 0 0.2 0.4 0.4 0 0 0 0 0 0 0.4 0 0 0 0.6 0 0 0 0 0 0 1 0 0 0 / тропический лес;

$O_{14}(s_2)$ = / 0 0 0 0.3 0.7 0 0 0 0 0 0.3 0.7 0 0 0 0.7 0.3 0 0 0 0 0 1 0 0 0 0 0 0 0.4 0.6 0 0 0 0 0.6 0 0 0 0.2 0 0 0 0.2 0 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0 0.4 0.6 0 0 0 0 0 0 0 0 0 0.6 0.4 0 0 0 0.4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 / строительно-дорожный автомобиль;

$O_{15}(s_2)$ = / 0 0 0 0.3 0.7 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0.4 0.6 0 0 0 0 0 0.2 0.3 0.5 0 0 0 0.2 0 0 0.5 0 0 0.1 0.2 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0 0.3 0.5 0.2 0 0 0 0.4 0 0 0 0.6 0 0 0 0 0 0 0 0 0 1 /грузовой автомобиль;

$O_{16}(s_2)$ = / 0 0 0.3 0.7 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0.3 0.5 0.2 0 0 0 0 0 0.2 0.2 0.4 0.2 0 0 0 0.2 0.2 0.1 0 0.1 0.2 0.2 0 0 0 0.2 0.6 0.2 0 0 0 0 0 0.4 0.5 0.1 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0 0 0 0 1 / легковой автомобиль;

$O_{17}(s_2)$ = / 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0.7 0.3 0 0 0 0 0 0.3 0.3 0.4 0 0 0 0.4 0 0 0 0.3 0 0 0 0.3 0 0 0 0 0.4 0.6 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0.2 0.5 0.3 0 0 0 0 0.3 0 0 0 0 0.7 0 0 0 0 0 0 0 0 1 / бронеавтомобиль;

$O_{18}(s_2)$ = / 0 0 0 0.7 0.3 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0.6 0.4 0 0 0 0 0 0.4 0.6 0 0 0 0 0.4 0 0 0 0.3 0 0 0 0.3 0 0 0 0 0 0.5 0.5 0 0 0 0 0 0.2 0.5 0.3 0 0 0 0 0 0 0 0 0.5 0.5 0 0 0 0 0.3 0 0 0 0 0 0.4 0 0.3 0 0 0 0 0 0 0 0 1 / бронетранспортер;

$O_{19}(s_2)$ = / 0 0 0 0.6 0.4 0 0 0 0 0.1 0.9 0 0 0 0 0 1 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0.2 0.6 0.2 0 0 0 0.4 0 0 0 0.3 0 0 0 0.3 0 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 / танк;

$O_{20}(s_2)$ = / 0 0 0 0.5 0.5 0 0 0 0 0.1 0.9 0 0 0 0 0.1 0.9 0 0 0 0 0.3 0.6 0.1 0 0 0 0 0.4 0.4 0.2 0 0 0 0.4 0 0 0 0.3 0 0 0 0.3 0 0 0 0.1 0.5 0.4 0 0 0 0 0 0 0.6 0.4 0 0 0 0 0 0.2 0.5 0.3 0 0 0 0 0.2 0 0 0 0.8 0 0 0 0 0 0 0 0 1 / САУ;

$O_{21}(s_3)$ = / 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0.8 0.2 0 0 0 0 0.4 0.6 0 0 0 0 0 0.6 0.3 0.1 0 0 0 0 0.4 0.4 0 0 0 0.2 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0.6 0.4 0 0 0 0 0 0 0 0 0.8 0.2 0 0 0 0 0 0 0 1 0 /яхта и катер;

$O_{22}(s3)$ = / 0 0 0 0.4 0.6 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0.5 0.3 0.2 0 0 0 0 0 0.7 0.3 0 0 0 0.5 0.3 0 0 0.2 0 0 0 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0.5 0.5 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0 0 0 1 / буксир;

$O_{23}(s_3)$ = / 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0.4 0.3 0.3 0 0 0 0 0.2 0.5 0.2 0.1 0 0.4 0.3 0 0 0.3 0 0 0 0 0 0.8 0.2 0 0 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0 1 0 / эсминец;

$O_{24}(s3)$ = / 0 0 0.4 0.6 0 0 0 0 0 1 0 0 0 0 0 1 0.1 0 0 0.9 0 0 0 0 0 0 0.6 0.3 0.1 0 0.5 0.5 0 0 0 0 0 0 0.1 0.8 0.1 0 0 0 0 0.4 0.3 0.2 0.1 0 0 0 0.6 0.4 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 / подводная лодка;

$O_{25}(s_3)$ = / 0 0 0 0 0 1 0 0 0 1 0 0 0 0.2 0.8 0 0 0 0.5 0.3 0.2 0 0 0 0.4 0.4 0.2 0 0 0.3 0.4 0 0 0.3 0 0 0 0 0.1 0.8 0.1 0 0 0 0 0.2 0.5 0.3 0 0 0 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0.4 0.6 0 0 0 0 0 0 1 0 / авианосец;

$O_{26}(s_4)$ = / 0.5 0.5 0 0 0 0 0 0 0 0.2 0.8 0 0 0 1 0 0 0 0 0.8 0.2 0 0 0 0 0 0.2 0.4 0.3 0.1 0 0 0.4 0.4 0 0 0.2 0 0 0 0 0 0.2 0.3 0.3 0.2 0 0 0 0 0.1 0.5 0.2 0.2 0 0 0 0.2 0.5 0.3 0 0 0 0 0 0 0.8 0.2 0 0 0 0 0 0 0 0 1 0 / беспилотник;

$O_{27}(s_4)$ = / 0 0 1 0 0 0 0 0 0 0.2 0.3 0.5 0 0 1 0 0 0 0 0.2 0.8 0 0 0 0 0.2 0.4 0.3 0.1 0 0 0.4 0.4 0 0 0 0.2 0 0 0 0 0 0.3 0.5 0.2 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0 0 0.6 0.4 0 0 0 0 0 0.4 0.6 0 0 0 0 0 0 0 0 1 / вертолет;

$O_{28}(s_4)$ = / 0 0 0 0.6 0.4 0 0 0 0 0.2 0.8 0 0 0 0 0 0 0.1 0 0 0.5 0.3 0.1 0 0 0 0 0 0 0 0 0.2 0.8 0.3 0.5 0 0 0 0.2 0 0 0 0 0 0 0.1 0.1 0.4 0.4 0 0 0 0 0 0.2 0.3 0.5 0 0 0 0 0 0 0 1 0 0 0 0.6 0.4 0 0 0 0 0 0 0 0 0 1 0 / истребитель;

$O_{29}(s_4)$ = / 0 0 0 0.4 0.6 0 0 0 0 0.1 0.9 0 0 0 0 0 0 0 0.1 0 0 0.4 0.4 0.1 0 0 0 0 0 0 0 0.1 0.2 0.7 0.3 0.5 0 0 0 0.2 0 0 0 0 0 0 0.1 0.1 0.5 0.3 0 0 0 0 0.1 0.2 0.3 0.4 0 0 0 0 0 0 0.3 0.7 0 0 0.5 0.5 0 0 0 0 0 0 0 0 0 1 0 / бомбардировщик;

$O_{30}(s_5)$ = / 0 1 0 0 0 0 0 0 0 0 0.4 0.6 0.6 0.4 0 0 0 0 0.5 0.4 0.1 0 0 0 0 0.1 0.7 0.2 0 0 0 0 0.4 0.2 0 0 0 0.2 0.2 0 0 0 1 0 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0 0 0.8 0.2 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 / мужчина;

$O_{31}(s_5)$ = / 0 1 0 0 0 0 0 0 0 0 0.5 0.5 0 1 0 0 0 0.4 0.4 0.2 0 0 0 0.2 0.7 0.1 0 0 0 0 0.2 0.2 0.1 0.2 0.2 0.1 0 0 1 0 0 0 0 0.6 0.4 0 0 0 0 0 0 0.1 0.7 0.2 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 / женщина;

$O_{32}(_{s5})$ = / 0.3 0.7 0 0 0 0 0 0 0 0 0.6 0.4 0 1 0 0 0 0 0.5 0.4 0.1 0 0 0 0 0.5 0.5 0 0 0 0 0 0.4 0.3 0 0 0 0.3 0 0 0 0 1 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0.3 0 0 0 0 0.7 0 1 0 0 0 0 0 0 / старик;

$O_{33}(s_5)$ = / 0.4 0.6 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0.4 0.5 0.1 0 0 0 0 0.6 0.4 0 0 0 0 0 0 0.2 0.2 0.1 0.2 0.2 0.2 0.1 0 0 1 0 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0 0 0.6 0.4 0 0 0 0 0 0 0 0.4 0 0 0 0 0.6 0 0 0 0 1 0 0 0 / старуха;

$O_{34}(s_5)$ = / 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0.4 0.4 0.2 0 0 0 0 0.1 0.5 0.4 0 0 0 0 0 0 0.2 0.2 0.1 0.2 0.1 0.2 0 0 1 0 0 0 0 0 0.6 0.4 0 0 0 0 0 0 0 0 0.7 0.3 0 0 0 0 0 0 0.5 0 0 0 0.5 0 0 1 0 0 0 0 0 0 0 / ребенок.

2. Выбор одной из строк значений функции принадлежности атрибутов эталонов $O_i$, например, для образа «легковой автомобиль »:

$O_{16}(s_2)$ = / 0 0.3 0.7 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0.3 0.5 0.2 0 0 0 0 0 0 0.2 0.2 0.4 0.2 0 0 0 0.2 0.2 0.1 0.1 0.2 0.2 0 0 0 0 0.2 0.6 0.2 0 0 0 0 0 0.4 0.5 0.1 0 0 0 0.3 0.5 0.2 0 0 0 0 0 0 0.5 0.5 0 0 0 0 0 0 0 0 0 1 /

3. Задание погрешностей значений функции принадлежности ***r***.

4. Формирование $N$ ($N$=70) атрибутов строки значений функции принадлежности атрибутов близкого изображения к эталоному.

- получение $n$ ($n$=70) случайных чисел $R_{in}$ из генератора случайных чисел, распределенных по равномерному закону в пределах от 0 до 1.

- вычисление $n$ случайных чисел с учетом заданной погрешности $R_{Gin}=R_{in}*r$.

-вычисление значений функции принадлежности каждого атрибута нового изображения:

$G_i$ ($\mu_{ik}$)= abs($O_i(\mu_{ik})$– $R_{Gin}$))*$q_k$,

где $q_k$ - коэффициенты значимости.

5. Запись в БД нового изображения $G_i$.

В приведенном алгоритме коэффициенты значимости $q_k$ устанавливались для различных групп атрибутов путем голосования по большинству поданных голосов экспертов. Значения коэффициентов выбирались в виде целого числа от 1 до 10. Например, для изображения «легковой автомобиль» были установлены следующие коэффициенты значимости атрибутов:

$O_{16}(s_2)$ «легковой автомобиль»: $q_1$=3; $q_2$=4; $q_3$=6; $q_4$=7; $q_5$=4; $q_6$=7; $q_7$=6; $q_8$=7; $q_9$=2; $q_{10}$=10;

С помощью приведенного алгоритма была дополнена база данных до 68000 образов, которая далее использовалась для машинного обучения нейросети и тестирования алгоритмов ЛЛК.

***3.3. Использование нейросети и алгоритма ЛЛК для классификации образов***

Для распознавания образов была использована нейронная сеть прямого распространения, состоящая из 34 параллельных каналов, обучающихся независимо с помощью процедуры обратного распространения ошибки. Данная НС спроектирована для распознавания 34 различных образов. Структура нейронной сети представлена на рисунке 1. Все эталонные образы пронумерованы от 1 до 34, каждому каналу присваивается номер, кодирующий соответствующий образ. Входом в НС является вектор характеристик **In**, кодирующий образ с помощью 70-ью атрибутов, которые поступают на входы каналов НС одновременно. Структура каналов одинаковая, каждый из них состоит из двух слоёв нейронов. В первом слое находится 34 нейрона с весами $\mathbf{W}_{i,1}$, смещениями $\mathbf{B}_{i,1}$ и линейными функциями активации $f_1$. Второй слой состоит из одного решающего нейрона с весами $\mathbf{W}_{i,2}$, и смещением $B_{i,1}$ и сигмоидной функцией активации $f_2$. Идеальным случаем распознавания образа будет выход решающего нейрона $P_i$ равный 1, если образ соответствует номеру канала, кодирующему данный образ, или 0, если образ не соответствует номеру канала. При наличии помехи и образа отличающегося от эталона, выход решающих нейронов каналов находится в диапазоне от 0 до 1. Тогда выбирается номер канала с максимальным значением выхода решающего нейрона, который и кодирует распознанный образ.

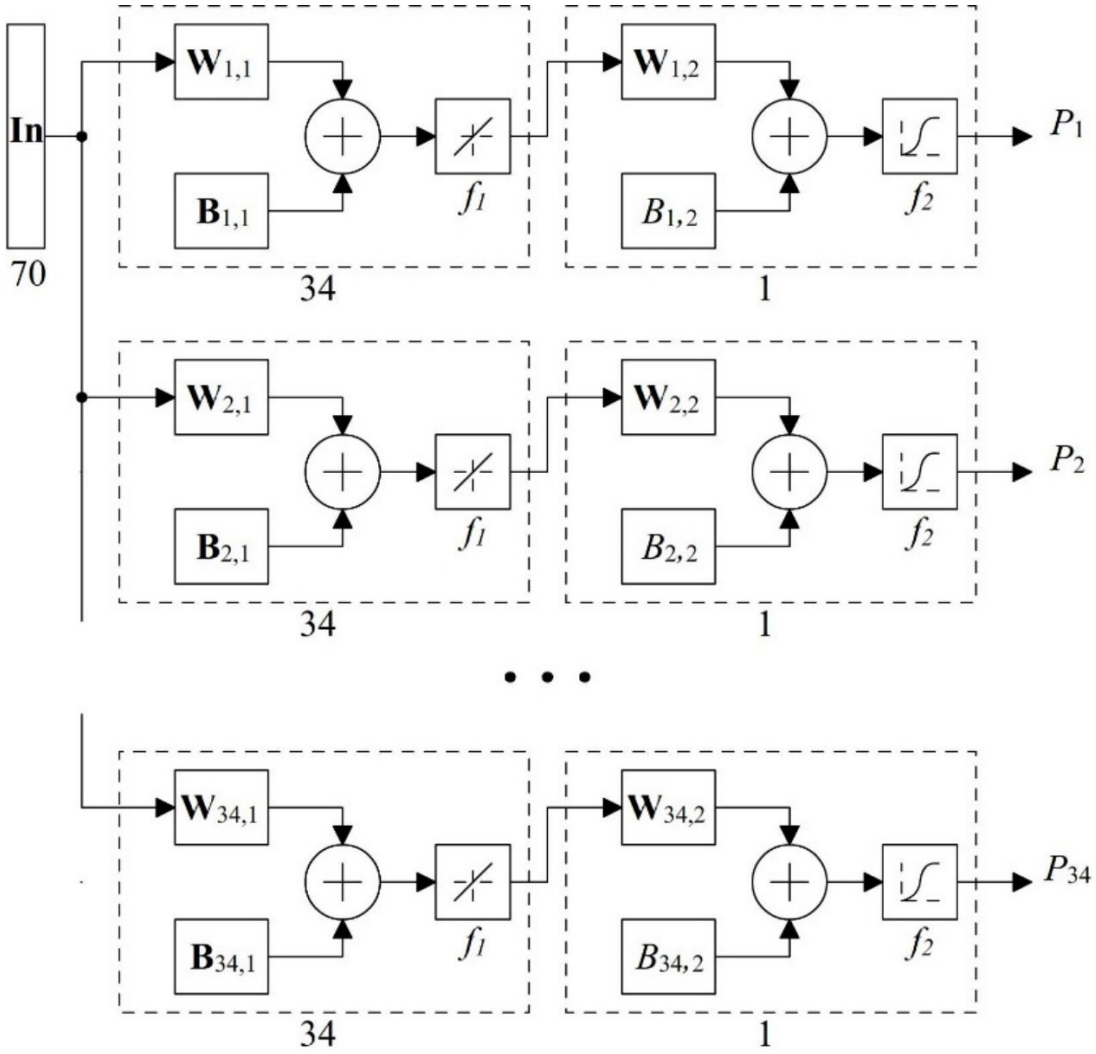



Рисунок 1 – Структура нейросети

Для реализации НС в качестве компьютерной модели использовался пакет программ в среде MATLAB. Предлагаемая архитектура позволяет увеличить количество распознаваемых НС образов путём добавления новых параллельных каналов. В ходе моделирования и обучения НС было установлено, что минимальный размер обучающей выборки, для которой получено устойчивое распознавание, составляет 6800 обучающих образов. С увеличением выборки уменьшается процент неправильных распознаваний.

Результаты компьютерного эксперимента приведены на рисунке 2 для 6800 изображений и рисунке 3 для 68000 изображений. На рисунке 2 по оси абцисс указаны отношения уровня помехи к уровню сигнала в %, а по оси ординат – точность алгоритма в %.

Для сравнения с результатами распознавания с помощью НС было протестирована созданная база данных алгоритмами ЛЛК. Результаты тестирования приведены на рисунке 4 (680 изображений), на рисунке 5 (6800 изображений) и на рисунке 6 (6800 изображений). По оси абцис указаны отношения уровня помехи к уровню сигнала в %, а по оси ординат – точность алгоритма в %.

По результатам тестовых компьютерных испытаний нейросети и алгоритма ЛЛК можно вычислить интегральные оценки эффективности:

$$I = \int_0^{100} l(x)dx\,,$$

где: $l(x)$ - функция точности (см. рис. 2, 3, 4, 5, 6), а $x$ – помеха ($0 \le x \le 100$).

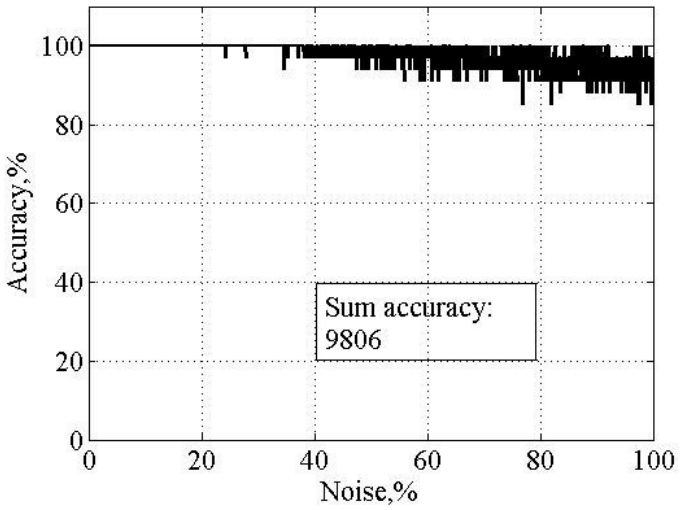



Рисунок 2 – График изменения точности классификации в зависимости от уровня помех для 6800 изображений при использовании НС

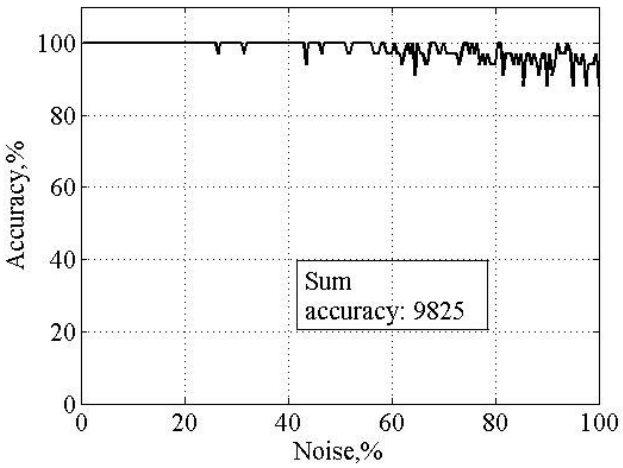



Рисунок 3 – График изменения точности классификации в зависимости от уровня помех для 68000 изображений при использовании НС.

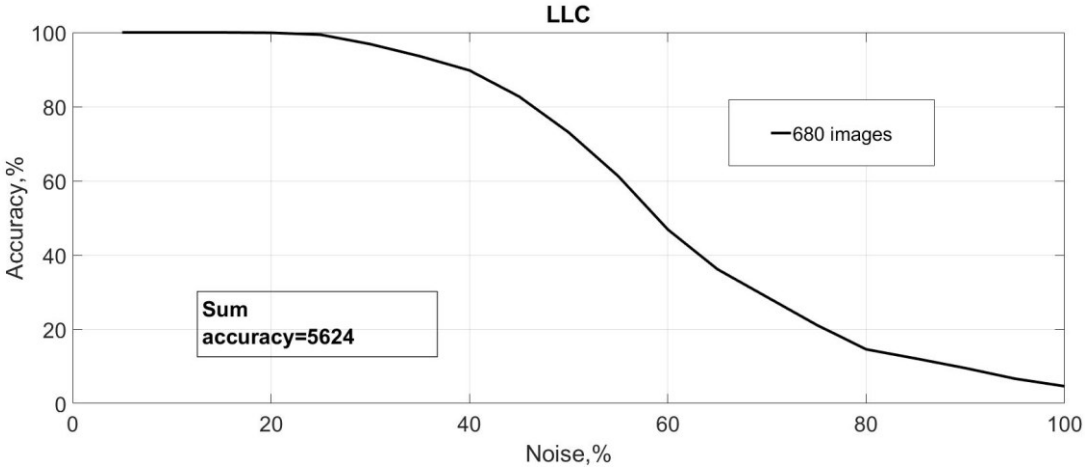



Рисунок 4 – График изменения точности классификации в зависимости от уровня помех для 680 изображений при использовании алгоритма ЛЛК

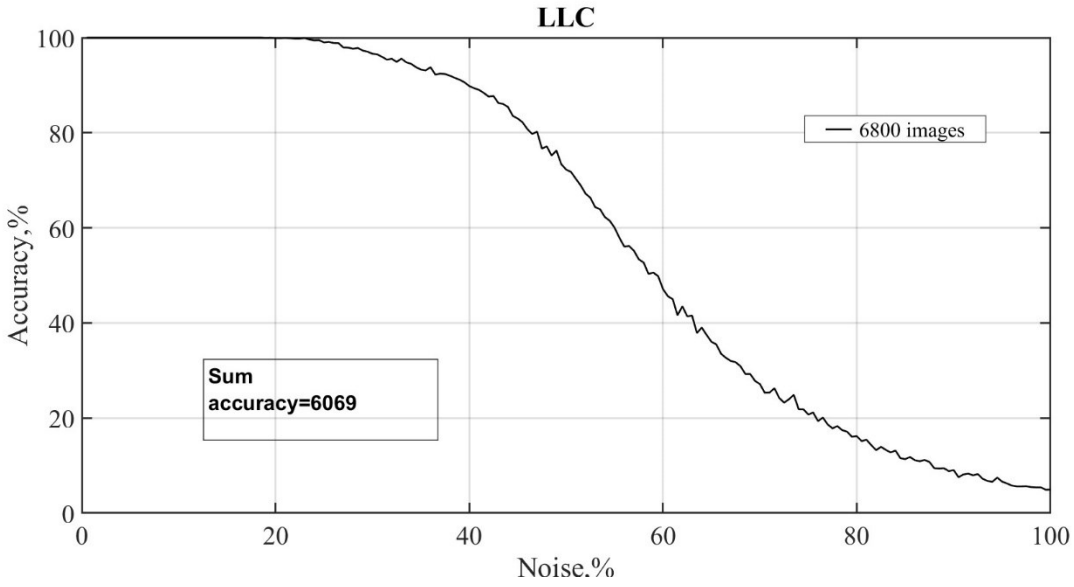



Рисунок 5 – График изменения точности классификации в зависимости от уровня помех для 6800 изображений при использовании алгоритма ЛЛК

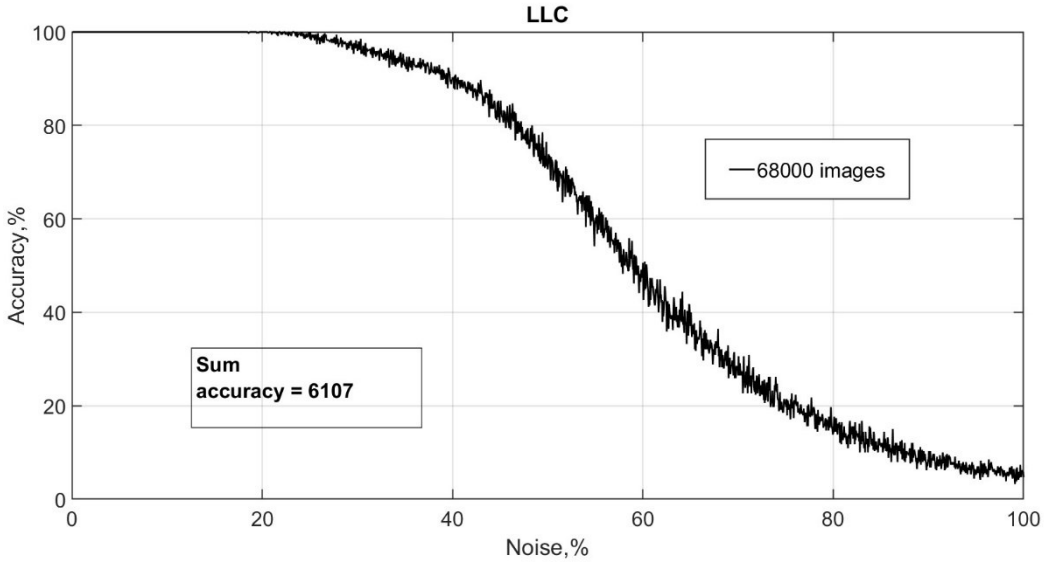



Рисунок 6 – График изменения точности классификации в зависимости от уровня помех для 68000 изображений при использовании алгоритма ЛЛК

В табл.1 приведены входные и выходные данные тестовых испытаний и интегральные оценки эффективности.

Таблица 1 – Входные и выходные данные комьютерного тестирования

| Кол-во изображений (вх. данные) | | 680 | 6800 | 68000 |
|---|---|---|---|---|
| Кол-во шагов помехи | | 20 | 200 | 2000 |
| Пределы изменения помехи от уровня сигнала,% | | 5 -100 | 0.5-100 | 0.05-100 |
| Шаг помехи,% | | 5 | 0.5 | 0.05 |
| Интегральная оценка эффективности, *I* | ЛЛК | 5624 | 6069 | 6107 |
| | НС | - | 9806 | 9825 |

При выборке 680 изображений не удалось обучить нейронную сеть, поэтому отсутствует результат в таблице 1. Из таблицы 1 видно, что при высоком уровне помехи полученные средние интегральные оценки эффективности при обучении нейросети при выборке 6800 и 68000 изображений больше, чем при использовании алгоритмов ЛЛК.

### *Заключение*

Решение задачи формирования образов и их классификацию на основе сенсорных данных позволяет строить классификационные модели объектов в окружении выбора мобильного робота с помощью нейросетевых алгоритмов и алгоритмов логико-лингвистической классификации.

Повышение точности и скорости классификации образов в окружении выбора особенно важно для мобильных роботов, например, БПЛА, перемещающихся в условиях не полной определенности, так как способствует эффективному принятию решений при ситуационном управлении движением, исключающим аварийные ситуации.

При уровне шума выше 80% и при больших выборках образов нейросетевые алгоритмы классификации работают более эффективно, а алгоритмы логико-лингвистической классификации менее эффективны. При уровне шума 50% - 60% и малых выборках использование алгоритмов логико-лингвистической классификации более эффективно, по сравнению с нейросетевыми алгоритмами. Это обьясняется тем, что большие выборки предполагают использование значительных компьютерных ресурсов: память, время и т.д.

***Благодарности***

Работа поддержана Минобрнауки Российской Федерации (проект госзадания 124041500008-1)

### *Литература*

1. Smart Electromechanical Systems: Decision making. Eds. Andrey E. Gorodetskiy, Irina L. Tarasova. Studies in Systems, Decision and Control, Vol. 352, Springer International Publishing, 2021, 270p.
2. Evaluating Classification Models. [электронный ресурс] /Ishaan Dey, Evan Heitman, Jagerynn T. Verano. Электрон. данные: 24.05.2019. Режим доступа https://github.com/ishaandey/Classification_Evaluation_Walkthrough
3. Макаренко С. И., Тимошенко А. В., Васильченко А. С. Анализ средств и способов противодействия беспилотным летательным аппаратам. Часть 1. Беспилотный летательный аппарат как объект обнаружения и поражения. // Системы управления, связи и безопасности. - 2020, - No. 1. С. 109-146.
4. Достоверный и правдоподобный вывод в интеллектуальных системах. /Под ред. В. Н. Вагина, Д. А. Поспелова. – Изд. 2-е, М.: ФизМатЛит, – 2008. - 712 с.
5. Ботуз С. П. Методы и средства графоаналитического анализа многоспектральных изображений в человеко-машинных системах технического зрения. // Техническое зрение. – 2020. – No. 5. http://magazine.technicalvision.ru/2020/03/
6. Алпатов Б. А., Бабаян П. В. Технологии обработки и распознавания изображений в бортовых системах технического зрения. // Вестник Рязанского государственного радиотехнического университета. -2017. – N0. 2 (60). - С. 34-44.
7. Сикорский О. С. Обзор сверточных нейронных сетей для задачи классификации изображений. // Новые информационные технологии в автоматизированных системах. – 2017. – No. 20. – С. 37–42.
8. Карасиков М. Е., Стрижов В. В. Классификация временных рядов в пространстве параметров порождающих моделей. // Информатика и ее применение. – 2016. – Том 10. – No. 4. – С. 121–131.
9. Quinlan J. R. Induction of Decision Trees. Machine Learning. 1986. Vol. 1, Issue 1, pp. 81 – 106.
10. Quinlan J. R. Improved Use of Continuous Attributes in C 4.5. // Journal of Artifical Intelligence Research. – 1996. - Vol. 4. – Issue 1. – pp. 77-90.
11. Clark P., Niblett T. The CN2 Induction Algorithm. // Machine Learning. – 1989. – Vol. 3, – Issue 4, –pp. 261–283.
12. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning: with Applications in R. Springer-Verlag New York. – 2013. – 426 p.
13. Abellán J., Castellano J. G. Improving the Naive Bayes Classifier via a Quick Variable Selection Method Using Maximum of Entropy. // Entropy. – 2017. – Vol.19, – No. 6.
14. Дубнов Ю. А. Метод отбора признаков на основе вероятностного подхода перекрестной энтропии на примере задачи распознавания изображений. // Искусственный интеллект и принятие решения. – 2020. – No. 2. – С. 78-85.
15. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений. - М.: Мир, 1976. - 168 с.

16. Зак Ю. А. Принятие решений в условиях нечетких и размытых данных: Fuzzy-технологии. М.: ЛИБРОКОМ, 2013. - 352 с.

17. Ногин В. В. Многокритериальный выбор на основе нечеткой информации. // Искусственный интеллект и принятие решени. – 2019. – No. 1

***Кучмин А.Ю., Тарасова И.Л. Формирование образов в окружении выбора мобильного робота и их классификация с использованием нейронных сетей и алгоритмов логико-лингвистической классификации.*** *В статье представлен краткий обзор алгоритмов формирования решающих правил для построения классификационных моделей и приведены рекомендации по их применению. Рассмотрен алгоритм формирования атрибутов образов в окружении выбора мобильного робота, на основе которых построены классификационные модели с использованием нейронных сетей и алгоритмов логико-лингвистической классификации. Машиннные эксперименты позволили провести оценку точности классификации различных изображений, которые показали, что при уровне помехи выше 80% и больших выборках лучше использовать нейросетевые алгоритмы классификации, а алгоритмы логико-лингвистической классификации более эффективны при малых выборках и уровне помехи 50%-60%.*

***Ключевые слова:*** *нейронные сети, фаззификация, логико-лингвистическая классификация, логико-вероятностная классификация, сенсорные данные, классифицируемые изображения.*

***Kuchmin A.Yu., Tarasova I.L. Formation of images in the environment of a mobile robot's choice and their classification using neural networks and logical-linguistic classification algorithms.*** *The article presents a brief overview of algorithms for forming decision rules for constructing classification models and provides recommendations for their application. An algorithm for forming image attributes in the environment of a mobile robot's choice is considered, on the basis of which classification models are built using neural networks and logical-linguistic classification algorithms. Machine experiments made it possible to assess the accuracy of classifying various images, which showed that at a noise level above 80% and large samples, it is better to use neural network classification algorithms, and logical-linguistic classification algorithms are more effective with small samples and a noise level of 50%-60%.*

***Key words:*** *neural networks, fuzzification, logical-linguistic classification, logical-probabilistic classification, classified images.*

УДК 004.089

# Применение биоинспирированных алгоритмов для повышения эффективности массового рекрутинга

**О.В. Булыгина[*1], М.Ю. Воротилова[*2]**

[*1] к.э.н., доцент, филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске,
baguzova_ov@mail.ru, OrcID: 0000-0001-6890-2842, SPIN-код: 5000-4428

[*2] магистрант, филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске,
rita.vorotilova@mail.ru

***О.В. Булыгина, М.Ю. Воротилова Применение биоинспирированных алгоритмов для повышения эффективности массового рекрутинга.*** *В статье рассмотрена проблема повышения эффективности массового рекрутинга с использованием биоинспирированных алгоритмов. Представлена классификация биоинспирированных алгоритмов и обоснован выбор алгоритма косяка рыб (Fish School Search, FSS) для решения поставленной задачи. Разработана процедура применения FSS для оптимизации массового рекрутинга, включающая поиск «наилучшего» набора кандидатов в многомерном пространстве с использованием операторов движения косяка рыб.*

***Ключевые слова:*** *массовый рекрутинг, биоинспирированные алгоритмы, оптимизация, алгоритм косяка рыб, автоматизация подбора персонала, критерии отбора кандидатов, Fish School Search.*

## *Введение*

В современном мире организации часто сталкиваются с необходимостью одновременного привлечения большого количества новых сотрудников. Данный процесс, известный как массовый рекрутинг, особенно актуален для динамично развивающихся компаний, находящихся в стадии активного роста, а также для организаций с высокой текучестью кадров. Успешная реализация массового подбора персонала требует существенных затрат времени, финансов и человеческих ресурсов, которые необходимы для обработки огромного количества поступающих резюме, оценки профессиональных и личных качеств кандидатов, а также выбора наиболее подходящих из них. Большой приток соискателей вызывает ряд сложностей, связанных с «качественным» анализом входящих резюме. Проведение данного процесса вручную является довольно трудоемким и длительным, что может привести к снижению качества отбора и увеличению вероятности принятия «неправильных» кадровых решений. Таким образом, низкая эффективность массового рекрутинга без использования автоматизированных инструментов может отрицательно сказаться на производительности труда и привести к дополнительным расходам на обучение и замену персонала.

Сегодня существуют различные инструменты автоматизации процесса массового рекрутинга. Такое средство, как системы управления кандидатами, позволяет хранить и систематизировать информацию о соискателях, автоматически сортировать и фильтровать резюме по заданным критериям [1]. Текстовая аналитика дает возможность извлекать ключевые сведения из неструктурированных данных в резюме и сопроводительных письмах, определять соответствие кандидатов требованиям вакансии. Машинное обучение применяется для автоматизации отбора кандидатов, построения прогнозных моделей успешности и эффективности сотрудников на основе исторических данных. Тем не менее, данные решения имеют ограничения, связанные с необходимостью ручной настройки критериев фильтрации резюме и трудностями при обработке неструктурированной информации. В связи с этим актуальной становится задача разработки нового подхода к автоматизации массового рекрутинга, который обеспечит гибкость в определении критериев отбора кандидатов и будет способствовать принятию обоснованных кадровых решений.

Одним из перспективных решений является применение биоинспирированных алгоритмов, которые основаны на моделировании происходящих в природе процессов, в частности эволюции, поведении живых организмов, физических и химических явлениях. Такие алгоритмы обладают способностью находить «наилучшие» решения в сложных и многомерных пространствах поиска. Применение биоинспирированных алгоритмов может обеспечить гибкость и адаптивность в определении критериев отбора кандидатов, автоматизировать процесс принятия кадровых решений и улучшить «качество» подбора персонала. Таким

образом, целью данной работы является разработка нового подхода к автоматизации процесса массового подбора персонала с применением биоинспирированных алгоритмов оптимизации.

### *Классификация биоинспирированных алгоритмов оптимизации*

Согласно классификации, предложенной в работе [2], существует более 250 различных биоинспирированных алгоритмов, которые могут быть разделены на две основные категории в зависимости от особенностей их поведения и механизмов создания новых решений. Авторы этой классификации выделяют алгоритмы с прямым формированием решений (Solution Creation) и алгоритмы с дифференциальным движением векторов (Differential-Vector Movement) (см. рис.1).

Алгоритмы с прямым формированием решений основаны либо на комбинировании решений (например, генетический алгоритм, Genetic Algorithm, GA), либо на непрямом взаимодействии между решениями через изменение окружающей среды (например, алгоритм муравьиной колонии, Ant Colony Optimization, ACO).

Алгоритмы второй категории генерируют новые решения путем смещения или мутации существующих, в то время как алгоритмы первой категории создают новые решения путем комбинирования нескольких существующих решений или с помощью других механизмов. Категория с дифференциальным движением векторов включает алгоритмы, в которых поиск осуществляется с учетом информации о популяции, только значимых решений (лучших и/или худших) или с использованием малых групп (подпопуляций или окрестностей). Наиболее известными алгоритмами в этой категории являются алгоритм роя частиц (Particle Swarm Optimization, PSO) и дифференциальная эволюция (Differential Evolution, DE).

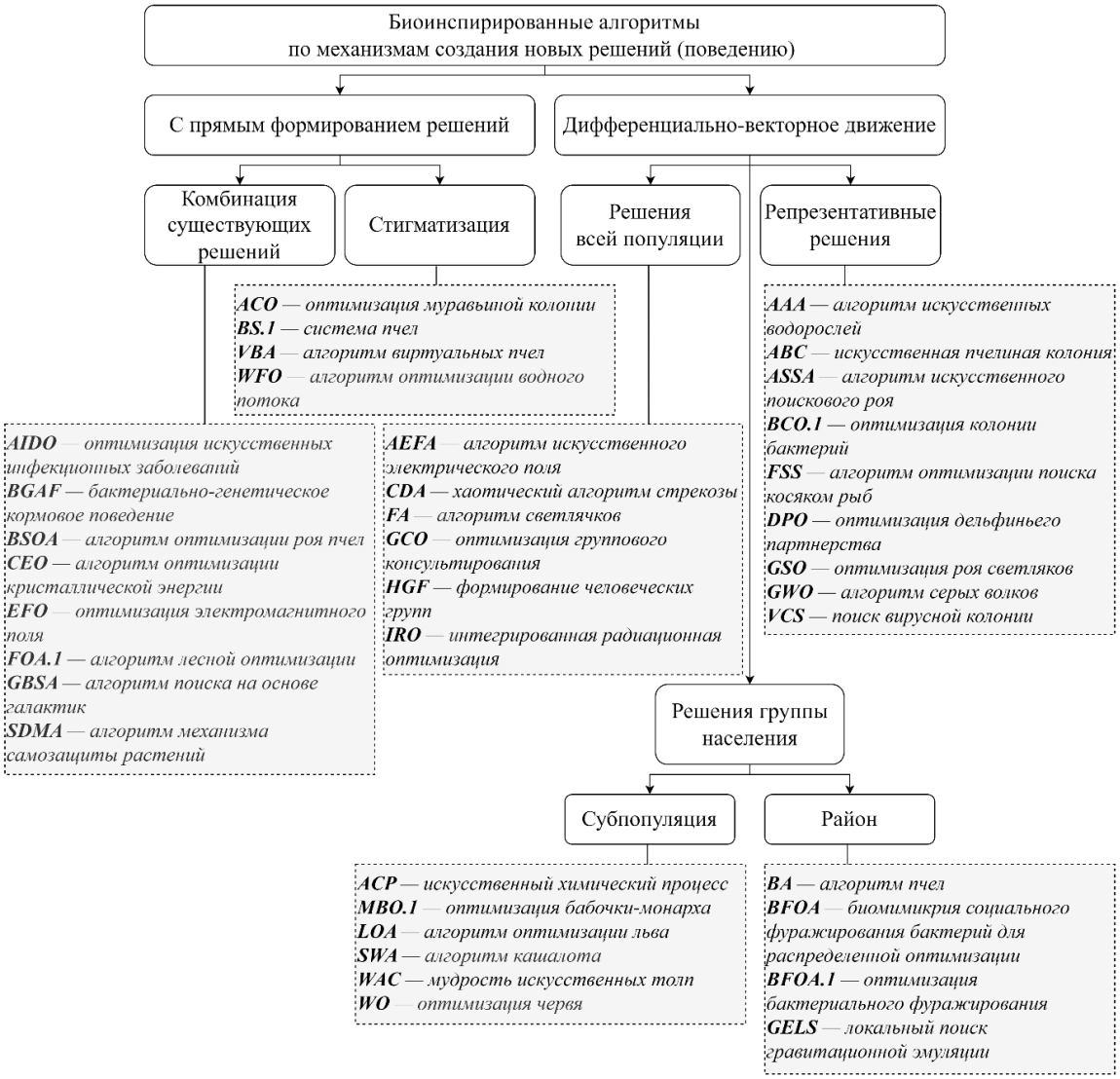



Рисунок 1 – Классификация биоинспирированных алгоритмов по механизмам создания новых решений

### *Применение алгоритма косяка рыб для оптимизации массового отбора кандидатов*

Для решения задачи оптимизации массового рекрутинга предлагается использовать алгоритм косяка рыб (Fish School Search, FSS), относящийся к категории алгоритмов с дифференциальным движением векторов. Данный алгоритм моделирует процесс коллективного поиска решений, где каждая рыба представляет собой потенциальное решение [3], характеризующееся набором критериев отбора кандидатов. Ключевыми преимуществами данного алгоритма являются способность эффективно исследовать пространство поиска, адаптироваться к специфике решаемой задачи и избегать застревания в локальных оптимумах. Процедура применения FSS для оптимизации массового рекрутинга представлена на рисунке 2.

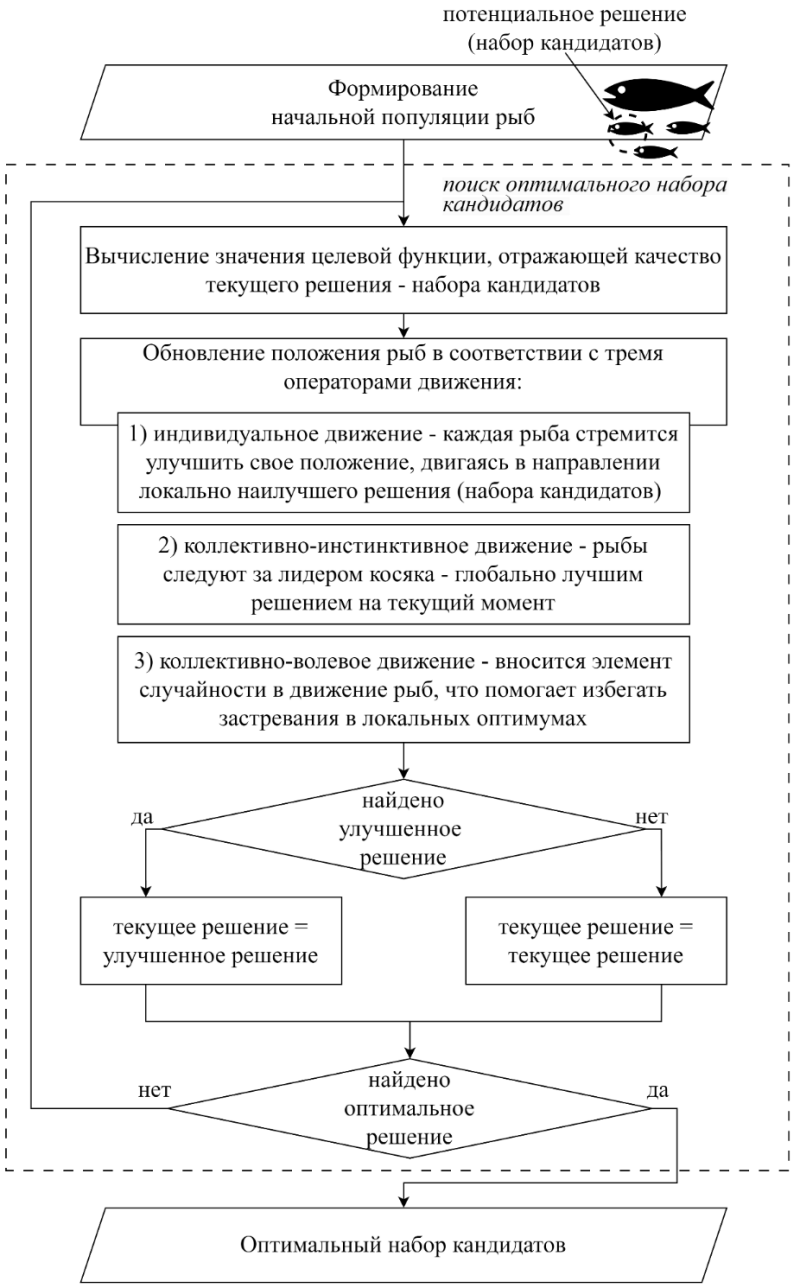



Рисунок 2 - Блок-схема алгоритма оптимизации массового отбора кандидатов на основе FSS

С помощью алгоритма FSS осуществляется поиск «наилучшего» набора кандидатов в многомерном пространстве решений. Алгоритм начинается с инициализации популяции рыб, каждое n-мерное местоположение рыбы представляет собой возможное решение оптимизационной задачи (набор кандидатов, характеризующийся критериями отбора). Далее в итеративном процессе происходит поиск «наилучшего» набора

кандидатов. Позиция рыбы представляет собой вектор весов различных критериев отбора кандидатов. На каждой итерации вычисляется значение целевой функции для каждой рыбы, отражающее качество текущего решения. Движения помогают найти «наилучшее» сочетание весов критериев для отбора наиболее подходящих кандидатов. Движение рыб в косяке определяется тремя основными операторами:

- индивидуальным движением: каждая рыба случайным образом исследует окрестности своей текущей позиции в поиске лучшего решения (кандидата или набора критериев) (см. формулу 1).

$$x_i(t+1) = x_i(t) + step_{ind} * rand(-1,1), \quad (1)$$

где $x_i(t)$ - текущая позиция i-той рыбы, $x_i(t+1)$ - новая позиция i-той рыбы, $step_{ind}$ - размер шага индивидуального движения (параметр алгоритма), rand(-1,1) - случайное число в диапазоне [-1,1]. Новая позиция $x_i(t+1)$ принимается только в том случае, если при изменении положения улучшается приспособленность рыбы i. Если это не так то $x_i(t)$ остается прежним, а $x_i(t+1) = x_i(t)$.

- коллективно-инстинктивным движением (тенденция следовать за лидером косяка): отражает тенденцию рыб двигаться в направлении успешных членов косяка (см. формулу 2). Движение взвешивается по величине улучшения целевой функции.

$$I = \frac{\sum_{i=1}^{N} \Delta x_i * \Delta f_i}{\sum_{i=1}^{N} \Delta f_i}, \quad (2)$$

где I - средневзвешенное значение перемещений каждой рыбы, $\Delta x_i$ - изменение позиции i-той рыбы на предыдущей итерации, $\Delta f_i$ - изменение значения целевой функции i-той рыбы. Это означает, что рыба, которая испытала большее улучшение, будет привлекать других рыб в свою текущую позицию. После вычисления вектора I каждая рыба будет побуждена к перемещению (см. формулу 3):

$$x_i(t+1) = x_i(t) + I, \quad (3)$$

- коллективно-волевым движением (внесение элемента случайности для избегания локальных оптимумов) (см. формулу 3): направление движения определяется относительно центра масс всего косяка. Коллективно-волевое движение обеспечивает сплоченность косяка, направляя рыб к общему центру масс, что помогает избежать чрезмерного рассеивания решений в пространстве поиска. Прежде всего, барицентр B(t) рассчитывается на основе положения $x_i$ и веса $W_i$ каждой рыбы (см. формулу 4):

$$B(t) = \frac{\sum_{i=1}^{N} \Delta x_i(t) * W_i(t)}{\sum_{i=1}^{N} W_i(t)}, \quad (4)$$

Если общий вес, определяемый суммой весов всех N рыб $\sum_{i=1}^{N} W_i(t)$, увеличился с прошлой по текущую итерацию, рыбы притягиваются к барицентру (см. формулу 5). Если суммарный вес не увеличился, рыбы удаляются от барицентра (см. формулу 6):

$$x_i(t+1) = x_i(t) + step_{vol} * rand(0,1) * \frac{x_i(t) - B(t)}{distance(x_i(t), B(t))}, \quad (5)$$

$$x_i(t+1) = x_i(t) - step_{vol} * rand(0,1) * \frac{x_i(t) - B(t)}{distance(x_i(t), B(t))}, \quad (6)$$

где $step_{vol}$ - размер шага волевого движения, $\mathrm{distance}(x_i(t), B(t))$ - евклидово расстояние между положением рыбы i и барицентром школы, rand (0,1) - равномерно распределенный массив случайных чисел с той же размерностью, что и B, и значениями, изменяющимися от 0 до 1.

Помимо операторов движения определен оператор кормления, используемый для обновления веса каждой рыбы (см. формулу 7):

$$W_i(t+1) = W_i(t) + \frac{\Delta f_i}{\max(|\Delta f_i|)}, \quad (7)$$

где $W_i(t)$ - весовой параметр для рыбы i, $\Delta f_i$ - изменение приспособленности между последней и новой позициями, $\max(|\Delta f_i|)$ - максимальное абсолютное значение изменения приспособленности среди всех рыб. Параметры $step_{ind}$ и $step_{vol}$ линейно убывают вместе с итерациями.

Положения рыб обновляются в соответствии с операторами индивидуального, коллективно-инстинктивного и коллективно-волевого движения. Обновленные решения оцениваются, и процесс продолжается до достижения условия останова. В результате формируется ранжированный список кандидатов, наиболее соответствующих требованиям вакансии.

Для эффективного применения алгоритма FSS в задаче оптимизации массового рекрутинга необходимо тщательно продумать представление решений, определение целевой функции и настройку параметров алгоритма. Представление решений должно включать все релевантные критерии отбора кандидатов, такие как образование, опыт работы, профессиональные навыки, личностные качества и т.д. Целевая функция должна адекватно отражать качество набора кандидатов и учитывать приоритеты организации в отношении каждого критерия. Параметры алгоритма, такие как размер популяции, коэффициенты индивидуального и коллективного движения, должны быть настроены с учетом специфики решаемой задачи и объема обрабатываемых данных.

Важным аспектом применения алгоритма FSS является предварительная обработка и структурирование данных о кандидатах. Резюме и сопроводительные письма должны быть проанализированы с помощью методов

текстовой аналитики для извлечения ключевой информации и приведения ее к единому формату. Это позволит алгоритму эффективно оперировать данными и находить «наилучшие» решения. Кроме того, необходимо предусмотреть механизмы обратной связи и корректировки критериев отбора на основе результатов собеседований и дальнейшей работы нанятых сотрудников. Это позволит постоянно совершенствовать процесс массового рекрутинга и адаптировать его к меняющимся потребностям организации.

### *Преимущества применения FSS в массовом рекрутинге*

Применение алгоритма FSS для оптимизации массового рекрутинга имеет ряд преимуществ и положительных сторон:

- алгоритм позволяет одновременно учитывать множество критериев, характеризующих кандидатов (образование, опыт, навыки, личностные качества и т.д.), и определять их относительную значимость для каждой конкретной вакансии (это обеспечивает комплексную оценку соискателей и способствует выбору наиболее подходящих кандидатов);
- FSS позволяет автоматизировать процесс обработки резюме, оценки кандидатов и формирования «наилучшего» набора соискателей (это значительно сокращает временные и трудовые затраты рекрутеров на ручной просмотр резюме и освобождает их для более важных задач, таких как проведение собеседований и адаптация новых сотрудников);
- формирует ранжированный список кандидатов, наиболее соответствующих требованиям вакансий, что повышает вероятность найма действительно подходящих сотрудников (это способствует снижению текучести кадров, повышению производительности труда и улучшению общих результатов деятельности организации);
- автоматизация отбора кандидатов с помощью FSS снижает влияние субъективных факторов, присущих ручному отбору, таких как личные предубеждения и ошибки рекрутеров (это повышает объективность и беспристрастность процесса найма, обеспечивая равные возможности для всех кандидатов);
- алгоритм способен эффективно обрабатывать большие объемы данных о кандидатах, что особенно актуально в условиях массового рекрутинга (FSS может быть применен как для отдельных вакансий, так и для масштабных кампаний по подбору персонала, обеспечивая стабильность и эффективность процесса независимо от количества соискателей).

### *Выводы*

Таким образом, использование алгоритма FSS является перспективным подходом к автоматизации массового рекрутинга, который позволяет учитывать множество критериев отбора кандидатов, повышать качество подбора персонала и сокращать временные и трудовые затраты на ручной просмотр резюме. Реализация предложенного алгоритма в рамках информационной системы управления кадрами способствует оптимизации процесса найма, снижению издержек и улучшению кадрового потенциала организации. Дальнейшие исследования в данной области могут быть направлены на интеграцию алгоритма FSS с другими методами интеллектуального анализа данных для повышения эффективности массового рекрутинга. Важным аспектом является разработка удобных пользовательских интерфейсов и средств визуализации, обеспечивающих эффективное взаимодействие рекрутеров с системой и принятие обоснованных кадровых решений.

Исследование выполнено за счет гранта Российского научного фонда №22-11-00335, https://rscf.ru/project/22-11-00335/.

### *Литература*

1. Что такое система отслеживания кандидатов (ATS) [Электронный ресурс] // SAP. – Электрон. дан. – Режим доступа: https://www.sap.com/central-asia-caucasus/products/hcm/recruiting-software/what-is-an-applicant-tracking-system.html

2. Taxonomy of bio-inspired algorithms [Электронный ресурс] / D.Molina [et al.] // Cognitive Computation. – Электрон. дан. – 2020. – P. 1-61. – Режим доступа: https://arxiv.org/pdf/2002.08136v1

3. Fish School Search Algorithm for Constrained Optimization [Электронный ресурс] / J.B. Monteiro-Filho [et al.] // Computational intelligence research group - Polytechnical School of Pernambuco. – Электрон. дан. – 2022. – P. 1-12. – Режим доступа: https://arxiv.org/pdf/1707.06169v1

4. Пучков, А. Ю. Нейро-нечеткий классификатор состояния технологического процесса / А. Ю. Пучков, М. И. Дли, Е. И. Лобанева // Известия Санкт-Петербургского государственного технологического института (технического университета). – 2021. – № 57. – С. 105-110.

5. Булыгина, О. В. Направления гибридизации алгоритмов роевого интеллекта и нечеткой логики для решения оптимизационных задач в социально-экономических системах / О. В. Булыгина, Д. Д. Ярцев, Н. Н. Прокимнов, Е. К. Верейкина // Прикладная информатика. – 2024. – Т.19, №5. – С.45-67.

***О.В. Булыгина, М.Ю. Воротилова Применение биоинспирированных алгоритмов для повышения эффективности массового рекрутинга.*** *В статье рассмотрена проблема повышения эффективности массового рекрутинга с использованием биоинспирированных алгоритмов. Представлена классификация биоинспирированных алгоритмов и обоснован выбор алгоритма косяка рыб (Fish School Search, FSS) для решения поставленной задачи. Разработана процедура применения FSS для оптимизации массового рекрутинга, включающая поиск «наилучшего» набора кандидатов в многомерном пространстве с использованием операторов движения косяка рыб.*

***Ключевые слова:*** *массовый рекрутинг, биоинспирированные алгоритмы, оптимизация, алгоритм косяка рыб, автоматизация подбора персонала, критерии отбора кандидатов, Fish School Search.*

***Bulygina O.V., Vorotilova M.Y. Application of bioinspired algorithms to improve mass recruiting efficiency.*** *The article deals with the problem of increasing the efficiency of mass recruitment using bio-inspired algorithms. The classification of bio-inspired algorithms is presented and the choice of the Fish School Search (FSS) algorithm for solving the problem is substantiated. A procedure for applying FSS to optimize mass recruitment has been developed, including the search for the best set of candidates in a multidimensional space using the operators of fish school movement.*

***Key words:*** *mass recruitment, bio-inspired algorithms, optimization, Fish School Search algorithm, recruitment automation, candidate selection criteria, Fish School Search.*

УДК 004.89

# Возможности применения биоинспирированных алгоритмов при формировании портфеля проектов

**О.В. Булыгина[*1], В.А. Дружинина[*2]**

[*1] к.э.н., доцент, филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске,
baguzova_ov@mail.ru, OrcID: 0000-0001-6890-2842, SPIN-код: 5000-4428

[*2] магистрант, филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске,
valeriedruzhinina@mail.ru

***О.В. Булыгина, В.А. Дружинина Возможности применения биоинспирированных алгоритмов при формировании портфеля проектов.*** *В статье рассмотрены современные подходы к формированию портфеля проектов. Основное внимание уделяется разработке методов, оптимизирующих выбор проектов с учетом экономической целесообразности и социальной значимости. Анализируются традиционные методы и биоинспирированные алгоритмы, в частности, алгоритм «стаи серых волков» (GWO), который использует принципы коллективного поведения для поиска оптимальных решений. Авторами предложен комплексный метод, сочетающий биоинспирированный алгоритм GWO и нечеткую кластеризацию, что позволяет улучшить точность и гибкость процесса формирования портфеля проектов.*

***Ключевые слова:*** *формирование портфеля проектов, биоинспирированные алгоритмы, оптимизация, алгоритм «стаи серых волков», системы поддержки принятия решений, управление портфелем проектов, Grey Wolf Optimizer.*

## *Введение*

В современных условиях одним из ключевых способов преодоления последствий санкционного давления на российскую экономику выступает стимулирование создания производственных мощностей, ориентированных на выпуск импортозамещающей продукции. На данный момент существует широкий спектр государственных программ, направленных на финансовую и регуляторную поддержку разработчиков этой продукции. В этой связи актуальной становится задача формирования портфеля проектов, претендующих на такую поддержку.

Данная задача относится к классу условной оптимизации, поскольку предполагает отбор проектов, которые одновременно будут экономически целесообразны и социально значимы, а их оценка будет проводиться по множеству параметров (включая доступность ресурсов, уровень рисков и т.д.) с учетом бюджетных ограничений. В специализированной литературе можно найти множество традиционных методов, применяемых при формировании портфеля проектов [1].

Например, метод линейного программирования использует математические модели для определения оптимального набора проектов с учётом ресурсных ограничений. Его основное преимущество — это возможность получать математически обоснованный результат. Однако данный метод сложен в построении моделей и имеет ограничения при использовании нечётких факторов, что снижает его применимость в условиях, когда необходимо учитывать сложные внешние условия.

Метод анализа выгод и затрат (Cost-Benefit Analysis, CBA) предполагает детальную оценку ожидаемых выгод и затрат для каждого проекта. Этот метод считается простым и интуитивно понятным, что облегчает выбор проектов на основе анализа их экономической эффективности. Но CBA затрудняется при количественной оценке нематериальных выгод и рисков, что делает его менее подходящим для проектов с долгосрочными результатами. Кроме того, данный метод усложняется при учёте внешних факторов.

Метод анализа рисков и неопределённости оценивает риски для каждого проекта, используя сценарный анализ и моделирование. Например, метод Монте-Карло позволяет анализировать проекты с высокой степенью неопределённости и учитывать различные возможные исходы. Однако он требует сложных вычислений, а высокая степень неопределённости может снижать точность прогноза. Данный метод также не всегда применим в условиях жёстких ограничений.

Можно выделить следующие общие недостатки приведенных традиционных методов формирования портфеля проектов: сложности учета неопределенности, ограниченные возможности адаптации к динамично изменяющимся условиям выполнения проектов, а также высокая степень субъективности при оценке неметрических или квазиметрических параметров. Кроме того, перечисленные методы зачастую не

предусматривают механизмов учета взаимосвязей между проектами. Итак, несмотря на разнообразие методов портфельного управления, ни один из них не является универсальным решением. Для ряда задач целесообразно разрабатывать «специализированные» инструменты. Например, метаэвристические методы способствуют нахождению решений, близких к глобальному оптимуму, за приемлемое время даже при недостатке точных данных о пространстве поиска. Наибольшее внимание сегодня привлекают биоинспирированные подходы, которые основываются на моделировании коллективного поведения живых организмов.

### *Классификация биоинспирированных алгоритмов оптимизации по природным процессам и явлениям, лежащим в их основе*

Согласно [2] существует более 270 биоинспирированных алгоритмов, которые могут быть разделены на несколько основных категорий в зависимости от природных процессов или явлений, лежащих в их основе. Среди них: эволюция популяций; роевой интеллект; алгоритмы, основанные на физических и химических процессах; социальные алгоритмы; алгоритмы, основанные на биологических процессах растений; прочие (рис.1).


Рисунок 1 – Классификация биоинспирированных алгоритмов по природным процессам и явлениям, лежащим в их основе

Рассмотрим представленные классы. Популяционные алгоритмы вдохновлены эволюцией в природе. Каждый индивидуум в таких алгоритмах является решением задачи и имеет значение приспособленности. Оптимальное решение находится прохождением процессов размножения и выживания, в результате чего популяция решений эволюционирует, то есть данный класс основан на концепции эволюции популяции (без учета других действий индивидуумов).

Алгоритмы роевого интеллекта образуют крупнейшую группу, охватывающую алгоритмы, которые имитируют коллективное поведение живых организмов, таких как млекопитающие, насекомые, птицы и рыбы. В этот класс входят подкатегории, основанные на поведении водных, наземных, летающих животных, а также микроорганизмов, что демонстрирует разнообразие моделей для различных типов задач оптимизации и классификации, опирающихся на коллективное поведение.

Основная идея социальных алгоритмов заключается в том, что решения можно улучшить через взаимодействие между отдельными агентами, которые обмениваются информацией и адаптируют свое поведение на основе социального опыта. Социальные алгоритмы особенно полезны при решении многокритериальных задач оптимизации, в которых требуется одновременно учитывать несколько противоречивых критериев. В таких задачах поиск компромиссных решений становится сложным, так как улучшение одного критерия может ухудшить другой. Социальные алгоритмы решают эту проблему за счет взаимодействия агентов, каждый из которых может стремиться к оптимизации различных аспектов задачи.

В следующую группу входят алгоритмы, основанные на физических процессах (гравитация, электрические поля) и химических реакциях. Эти алгоритмы моделируют процессы, такие как теплопередача, движение молекул, реакция химических веществ, колебания или волновые явления, и применяют эти принципы для поиска оптимальных решений в сложных задачах. Главная особенность таких алгоритмов заключается в их способности находить равновесие и стабильные состояния, что полезно для решения задач, требующих глобальной оптимизации.

Небольшую категорию, включающую алгоритмы, которые моделируют биологические процессы растений, такие как опыление и оптимизация корневых систем, часто применяют в задачах, связанных с ростом и распределением ресурсов в условиях ограничений.

### *Биоинспирированный алгоритм «стаи серых волков»*

Оптимизация методом «стаи серых волков» (GWO) представляет собой метаэвристический алгоритм поиска наилучшего решения, использующий поведенческие принципы животных для решения сложных задач. В GWO используется иерархическая структура стаи серых волков, а также стратегия их охоты.

Целью алгоритма является определение ближайшего волка до добычи и оптимальное перестроение остальных волков таким образом, чтобы они все оказались ближе всего к цели. В основе лежит факт, что относительно ближайшего к добыче волка, перестраиваются остальные волки, образуя нечто похожее на кольцо. Данное «перестроение» повторяется до тех пор, пока волки не соберутся в стаю вокруг добычи, что и будет являться оптимальным моментом для атаки с минимальным расстоянием до цели. Рассмотрим процесс движения волков к точке с оптимальным расстоянием для нападения поэтапно.

Этап 1. Инициализация исходной популяции волков.

Каждому волку соответствует вектор $x_i = (x_{i1}, x_{i2},\ldots, x_{in})$, его позиция. Добыче, на которую охотятся волки, соответствует вектор $\theta^* = (\theta_1, \theta_2,\ldots, \theta_n)$. Позиции членов стаи случайны.

Этап 2. Нахождение ближайших трех волков.

Формируется список трех волков, которые ближе всего находятся к добыче (с наилучшими значениями целевой функции). Самому близкому придается имя – альфа (α), второму – бета (β), третьему – дельта (δ). Отбор происходит на основе минимизации целевой функции

$$f_{(x1\ldots xn)} = min\{(\theta_1 - x_{11})^2 + (\theta_2 - x_{12})^2 + \cdots + (\theta_n - x_{1n})^2\} \quad (1)$$

Этап 3. Вычисление обновленных координат волка в стае при смещении его к α, β или δ.

Вычисляются векторы $D_\alpha$, $D_\beta$, $D_\delta$ и векторы $Y_1$, $Y_2$, $Y_3$.

$$\vec{D}_\alpha = |\vec{C}_1 * X_\alpha - X|,\ \vec{D}_\beta = |\vec{C}_2 * X_\beta - X|,\ \vec{D}_\delta = |\vec{C}_3 * X_\delta - X| \quad (2)$$

$$\vec{Y}_1 = X_\alpha - A_1 * (\vec{D}_\alpha),\ \vec{Y}_2 = X_\beta - A_2 * (\vec{D}_\beta),\quad \vec{Y}_3 = X_\delta - A_3 * (\vec{D}_\delta) \quad (3)$$

Векторы A и C вычисляются по формулам (4) и (5):

$$\vec{A} = 2\,\vec{a} * \overrightarrow{r_1} - \vec{a} \quad (4)$$

$$\vec{C} = 2 * \overrightarrow{r_2} \quad (5)$$

Вектор $a$ линейно уменьшается от 2 до 0 в течение итераций, что позволяет волкам постепенно сужать область поиска, приближаясь к оптимальному решению. Формулы (4) и (5) задают коэффициенты случайного расширения и сжатия вокруг лидеров, что способствует исследованию пространства поиска.

$r_1$ *и* $r_2$– генерирующиеся случайным образом случайные вектора, элементы которых лежат в диапазоне [0,1], а *a* – вектор, вычисляемый по формуле (6):

$$\vec{a} = 2 - 2 * (\frac{itr\ maxitr}{maxitr}), \quad (6)$$

где *itr* – номер итерации; *maxitr* – общее количество повторений.

Этап 4. Вычисление новых координат для волка.

Для вычисления новых координат волка находят среднее арифметическое к координатам если бы волк бежал к волкам α, β и δ.

$$\vec{x}(i,1) = \frac{y_1 + y_2 + y_3}{3} \qquad (7)$$

Этап 5. Сравнение номера итерации с максимальным количеством итераций. Если они не совпадают, то все возвращаются к шагу 2.

Этап 6. После завершения всех итераций находится волк с наилучшим значением целевой функции, который и считается ближайшим к добыче.

### *Особенности применения GWO при формировании портфеля проектов*

Для формирования оптимального состава портфеля проектов предлагается применять сочетание методов GWO и нечеткой кластеризации. GWO поможет находить лучшие комбинации проектов, минимизируя или максимизируя целевую функцию (например, совокупные затраты или риски), в то время как нечеткая кластеризация будет использована для группировки пространства проектов по схожим характеристикам (рискам, выгодам или срокам реализации) с целью сокращения времени выполнения алгоритма.

На рисунке 2 показана схема предложенного алгоритма формирования портфеля проектов, основанного на комплексном использовании методов роевого интеллекта (GWO) и нечеткой кластеризации.

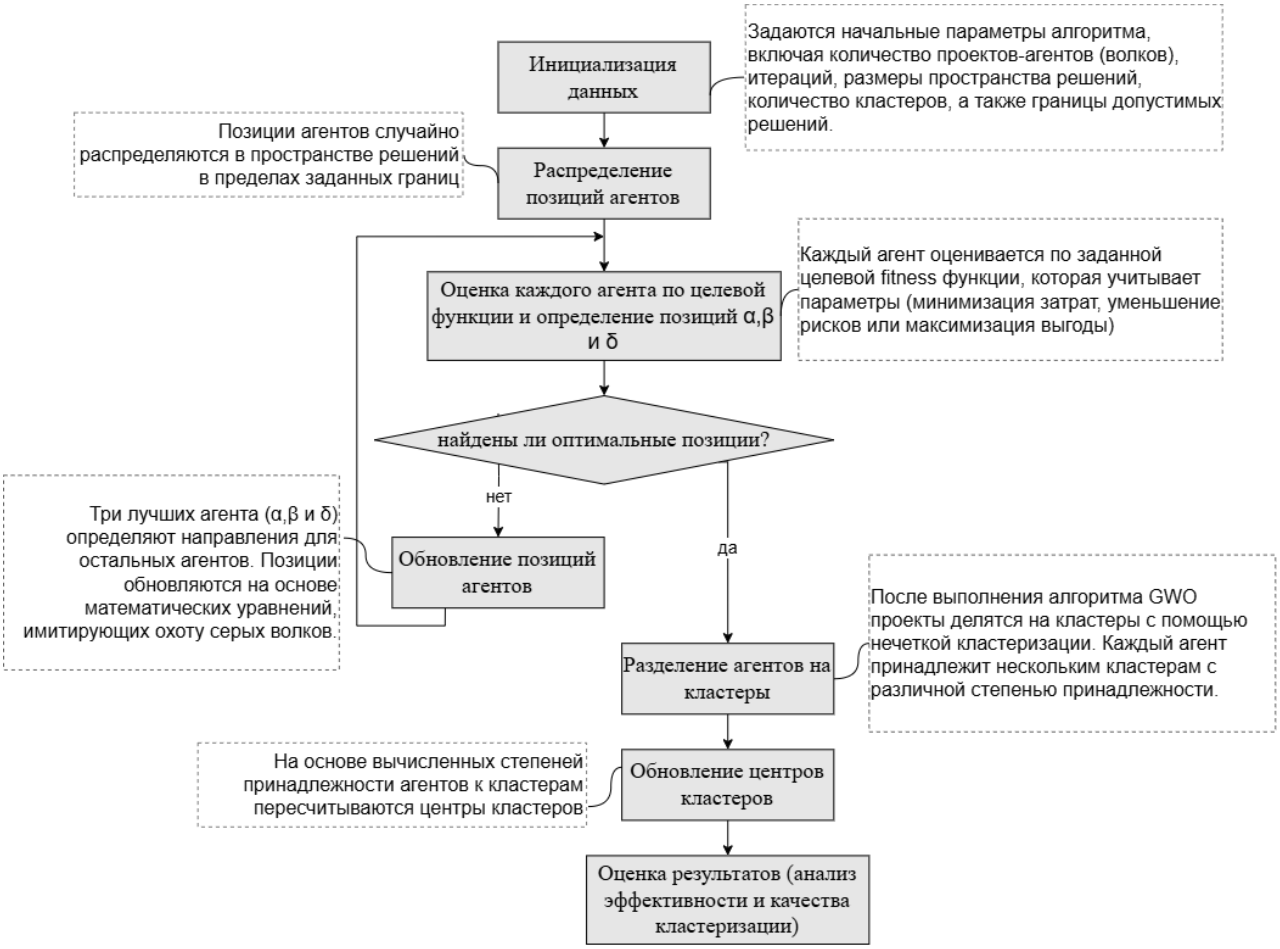



Рисунок 2 – Блок-схема комплексного использования методов GWO и нечеткой кластеризации

Среди преимуществ использования комплексного метода можно выделить следующие [4]:

- использование GWO позволяет эффективно искать «наилучшие» решения за счёт механизма распределения проектов в пространстве решений. GWO имитирует социальное поведение волков в природе, где позиции агентов играют ключевые роли в поиске оптимума, что улучшает эффективность алгоритма и уменьшает вероятность попадания в локальные минимумы;

- в задачах управления проектами часто присутствует высокий уровень неопределённости, связанный с

рисками, ресурсами и временными ограничениями. Нечёткая логика позволяет учитывать эти неопределённости, моделируя их в форме лингвистических переменных и правил, что обеспечивает гибкость в оценке и ранжировании проектов, а также позволяет более точно оценивать относительную приоритетность проектов;

- сочетание GWO и нечеткой логики обеспечивает высокую адаптивность и устойчивость алгоритма. Метод GWO способен динамически корректировать позиции агентов в зависимости от текущих условий поиска, а нечеткая логика позволяет гибко учитывать изменения приоритетов и условий в реальном времени. Это особенно важно для многокритериальных задач, характерных для портфельного управления, где необходимо учитывать баланс между разнообразными критериями.

### *Выводы*

Таким образом, использование биоинспирированных алгоритмов является перспективным направлением в автоматизации процесса управления портфелем проектов. Интеграция методов GWO и нечеткой логики при формировании портфеля обеспечивает повышение точности и гибкости поиска решений, адаптируя метод к изменяющимся условиям и ограничивающим факторам. Это делает данный метод предпочтительным для задач, требующих высокого уровня устойчивости к неопределённости и способности находить сбалансированные решения в условиях ограниченных ресурсов. Дальнейшие исследования в этой области могут быть сосредоточены на повышении адаптивности и точности предложенного комплексного метода, а также на интеграции его с другими алгоритмами многокритериальной оптимизации для более сбалансированного отбора проектов. Кроме того, исследования могут включать разработку новых нечетких моделей для управления рисками, специфичными для различных отраслей, что позволит формировать более устойчивые портфели в условиях неопределенности.

Исследование выполнено за счет гранта Российского научного фонда №22-11-00335, https://rscf.ru/project/22-11-00335/.

### *Литература*

1. Методические аспекты формирования портфеля проектов в инновационной экосистеме [Электронный ресурс] / Толстых Т.О., Гамидуллаева Л.А., Шмелева Н.В. // Модели, системы, сети в экономике, технике, природе и обществе. – 2020. – №1. – Режим доступа: https://cyberleninka.ru/article/n/metodicheskie-aspekty-formirovaniya-portfelya-proektov-v-innovatsionnoy-ekosisteme

2. Taxonomy of bio-inspired algorithms [Электронный ресурс] / D. Molina [et al.] // Cognitive Computation. – Электрон. дан. – 2020. – P. 1-61. – Режим доступа: https://arxiv.org/pdf/2002.08136v1

3. Grey Wolf Optimizer [Электронный ресурс] / S. Mirjalili [et al.] // Advances in Engineering Software. – 2014. – P. 46-61. – Режим доступа: https://www.researchgate.net/publication/260010809_Grey_Wolf_Optimizer

4. Булыгина О. В. Направления гибридизации алгоритмов роевого интеллекта и нечеткой логики для решения оптимизационных задач в социально-экономических системах / О. В. Булыгина, Д. Д. Ярцев, Н. Н. Прокимнов, Е. К. Верейкина // Прикладная информатика. – 2024. – Т.19. – №5. – С.45-67.

5. Булыгина О.В., Тюкаев Д.А., Яшин Е.С. Оптимизация портфеля проектов импортозамещения наукоемкой продукции с использованием модифицированного алгоритма пчелиных колоний // Бизнес. Образование. Право. – 2024. – №2. – С.125-131

6. Дли М.И., Стоянова О.В. Способы представления экспертных данных в системах поддержки принятия решений по управлению сложными проектами // Нейрокомпьютеры: разработка, применение. – 2016. – № 7. – С. 21-28.

***О.В. Булыгина, В.А. Дружинина Возможности применения биоинспирированных алгоритмов при формировании портфеля проектов.*** *В статье рассмотрены современные подходы к формированию портфеля проектов. Основное внимание уделяется разработке методов, оптимизирующих выбор проектов с учетом экономической целесообразности и социальной значимости. Анализируются традиционные методы и биоинспирированные алгоритмы, в частности, алгоритм «стаи серых волков» (GWO), который использует принципы коллективного поведения для поиска оптимальных решений. Авторами предложен комплексный метод, сочетающий биоинспирированный алгоритм GWO и нечеткую кластеризацию, что позволяет улучшить точность и гибкость процесса формирования портфеля проектов.*

***Ключевые слова:*** *формирование портфеля проектов, биоинспирированные алгоритмы, оптимизация, алгоритм «стаи серых волков», системы поддержки принятия решений, управление портфелем проектов, Grey Wolf Optimizer.*

***Bulygina O.V., Druzhinina V.A. Application of Bio-Inspired Algorithms in Project Portfolio Formation.*** *The article examines modern approaches to forming a project portfolio. It focuses on the development of methods that optimize project selection based on economic feasibility and social significance. Traditional methods and bio-inspired algorithms are analyzed, particularly the Gray Wolf Optimization (GWO) algorithm, which utilizes principles of collective behavior to find optimal solutions. The authors propose a combined method that integrates the GWO bio-inspired algorithm and fuzzy clustering, enhancing the accuracy and flexibility of the project portfolio formation process.*

***Key words:*** *project portfolio formation, bio-inspired algorithms, optimization, Gray Wolf Optimization algorithm, decision support systems, project portfolio management, Grey Wolf Optimizer.*

УДК 004.89

# Анализ современных методов оптимизации параметров процесса обеспечения качества топливных брикетов

**О.В. Булыгина[*1], К.В. Хлусович[*2]**

*1 к.э.н., доцент, Филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске
baguzova_ov@mail.ru

*2 студент, Филиал ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске,
khlusovich_kristina@mail.ru

***Булыгина О.В., Хлусович К.В. Анализ современных методов оптимизации параметров процесса обеспечения качества топливных брикетов.*** Статья посвящена обзору современных методов многокритериальной оптимизации параметров процесса обеспечения качества топливных брикетов, таких как калорийность, влажность, зольность и механическая прочность. Особое внимание уделено метаэвристическим алгоритмам, способным находить оптимальные решения при учете противоречивых требований к качеству и изменяющихся условий.

***Ключевые слова:*** альтернативные источники энергии, многокритериальная оптимизация, качество продукции, метаэвристические алгоритмы, алгоритм режима потока, алгоритм поиска состояний вещества, оптимизация движения ионов

## *Введение*

С ростом спроса на альтернативные источники энергии топливные брикеты наращивают свою популярность благодаря экологичности, высокой калорийности и удобству. Для расширения доли рынка и выхода на новые сегменты, необходимо постоянно следить за качеством продукции, уделяя особое внимание таким потребительским свойствам, как калорийность, влажность, зольность, механическая прочность и цена. Для этого нужно обеспечить «наилучшие» значения этих параметров в условиях различных ограничений. Данная задача относится к многокритериальной оптимизации, которая предполагает поиск «наилучшего» решения по нескольким критериям (целевым функциям) [1]. На сегодняшний день существуют множество подходов к решению таких задач, примем каждый из них имеет свои преимущества и особенности применения.

## *Современные подходы к проведению многокритериальной оптимизации*

Наиболее известным методом является метод взвешенных критериев (взвешенной суммы), который присваивает каждому критерию вес в зависимости от его важности. Итоговая оценка для каждой альтернативы определяется как сумма произведений значений критериев на их веса [2]. В качестве «лучшего» решения выбирается то, у которого наибольшая итоговая оценка. Метод прост, но весовые коэффициенты значительно влияют на результат, что требует тщательного анализа.

Пусть:

1) $x$ – альтернативное решение, которое оценивается;
2) $f_i(x)$ – значение $i$-го критерия для решения $x$ (например, калорийность);
3) $w_i$ – вес i-го критерия, отражающий его значимость в общей оценке (например, $\sum w_i = 1$).

Тогда итоговая оценка $S(x)$ для решения $x$ определяется формулой:

$$S(x) = \sum_{i=1}^{n} w_i * f_i(x), \quad (1)$$

где $n$ – общее число критериев и $w_i$ – весовой коэффициент критерия $i$, где обычно $w_i > 0$ и $\sum_{i=1}^{n} w_i = 1$.

Например, оценка качества топливных брикетов происходит по двум критериям калорийность $f_1(x)$ и влажность $f_2(x)$. Если важность калорийности вдвое выше важности влажности, то можно присвоить веса $w_1 = 0{,}67$ и $w_2 = 0{,}33$. Тогда оценка для данного решения будет $x$ будет: $S(x) = 0.67 * f_1(x) + 0.33 * f_2(x)$.

Метод предполагает, что чем выше итоговая оценка $S(x)$, тем лучше решение $x$. Однако выбор весов $w_i$ критичен и должен быть обоснован: изменение весов может сильно повлиять на результат и, следовательно, требует тщательного анализа и экспертизы, чтобы веса корректно отражали приоритеты.

Метод Парето оптимизирует решения так, чтобы улучшение одного критерия не ухудшало другие. Такие решения образуют «фронт Парето» – набор оптимальных вариантов, что важно при балансировке взаимозависимых показателей [2]. Для набора решений $x$ с несколькими целевыми функциями $f_1(x)$, $f_2(x)$, …, $f_n(x)$, где каждое $f_i(x)$ описывает один из показателей качества (например, калорийность или влажность), решение $x^*$ называется Парето-оптимальным, если не существует другого решения $x'$, которое улучшает хотя бы один критерий без ухудшения других.

Пусть:

1) $x \in X$ – пространство всех возможных решений;

2) $f(x) = (f_1(x), f_2(x), \ldots, f_n(x))$ – вектор целевых функций, характеризующий каждое решение $x$ в $R^n$;

3) $x^*$ – потенциально оптимальное решение.

Тогда $x^*$ является Парето-оптимальным, если не существует другого решения $x' \in X$, такого что:

$$f_i(x') \le f_i(x^*) \text{ для всех } i \in \{1, 2, \ldots, n\},$$
$$\text{и } f_i(x') < f_i(x^*) \text{ для хотя бы одного } j \in \{1, 2, \ldots, n\}. \quad (2)$$

Решения на Парето-фронте позволяют организациям выбрать такие параметры, которые соответствуют предпочтениям в текущих условиях, например, снизить влажность при минимальном снижении калорийности или наоборот.

Метод TOPSIS оценивает решения по их близости к идеальной и удаленности от неидеальной точек [3]. Идеальная точка соответствует наилучшим значениям всех критериев, а неидеальная – худшим. Лучшее решение максимально близко к идеальной точке и максимально далеко от неидеальной, что позволяет учитывать как положительные, так и отрицательные характеристики.

Пусть $X = \{x_{ij}\}$ – это матрица решений, где $x_{ij}$ – это значение j-го критерия для i-го решения. Сначала проводится нормализация матрицы решений, чтобы все критерии были измерены в одинаковых единицах. Это обычно выполняется с использованием евклидовой нормы:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}, \quad (3)$$

где $m$ – количество альтернатив, а $j$ – количество критериев.

Если каждый критерий имеет свой вес $w_j$, то получаем взвешенную нормализованную матрицу $v_{ij}$:

$$v_{ij} = w_j * r_{ij}. \quad (4)$$

Идеальная точка (максимальные значения для всех критериев): $v_j^+ = \max(v_{ij})$ для всех $j$.

Неидеальная точка (минимальные значения для всех критериев): $v_j^- = \min(v_{ij})$ для всех $j$.

Для каждой альтернативы вычисляются следующие расстояния:

1) Расстояние до идеальной точки (положительное расстояние):

$$D_i^+ = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^+)^2}, \quad (5)$$

2) Расстояние до неидеальной точки (положительное расстояние):

$$D_i^- = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)^2}. \quad (6)$$

Наконец, для каждой альтернативы вычисляется коэффициент близости к идеальной точке $C_i$, который определяется так:

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-}, \quad (7)$$

где $D_i^+$ и $D_i^-$ – расстояние до идеально и неидеальной точки соответственно.

«Лучшее» решение – это решение с наибольшим значением коэффициента $C_i$, то есть находящееся максимально близко к идеальной точке и максимально далеко от неидеальной.

Коэффициент близости $C_i$ = 1 означает, что альтернатива является идеальной, а $C_i$ = 0 – альтернатива является неидеальной. Лучшее решение будет иметь максимальное значение $C_i$, так как оно будет максимально близким к идеальной точке и максимально удаленным от анти-идеальной.

Метод анализа иерархий (AHP) разделяет задачу на несколько уровней: цель (верхний уровень), критерии (средний уровень) и альтернативы (нижний уровень), и предполагает попарное сравнение вариантов [3]. Этот метод удобен для задач, требующих учета сложных взаимосвязей.

Для того чтобы определить важность каждого элемента, AHP использует шкалу сравнений, где каждому сравнению присваивается определенный балл:

1 – равное значение,

3 – умеренное преимущество одного элемента,

5 – значительное преимущество,

7 – очень значительное преимущество,

9 – экстремальное преимущество, и значения для промежуточных баллов (2, 4, 6, 8).

Например, если критерий 1 важнее критерия 2 в два раза, то присваивается значение 2.

Для каждого уровня иерархии строится ***матрица попарных сравнений***. Например, для попарного сравнения *n* критериев матрица будет *n* x *n*, где элемент $a_{ij}$ – это оценка важности критерия *i* относительно критерия *j*.

Для попарного сравнения альтернатив для каждого критерия матрица будет выглядеть аналогично. Если $a_{ij}$ – это оценка важности критерия *i* относительно критерия *j*, матрица попарных сравнений будет симметричной, то есть $a_{ij} = 1/a_{ij}$, а элементы на диагонали всегда равны 1 (например, $a_{ij} = 1$).

На основе матрицы попарных сравнений рассчитываются веса критериев и альтернатив с использованием метода ***собственного вектора***. Этот процесс можно описать следующим образом:

1) Каждую колонку матрицы нормализуют, деля каждый элемент на сумму всех элементов в соответствующей колонке:

$$\widehat{a_{ij}} = \frac{a_{ij}}{\sum_{i=1}^{n} a_{ij}}. \quad (8)$$

2) После нормализации для каждой строки матрицы вычисляется среднее значение, которое и будет являться весом для соответствующего критерия (или альтернативы):

$$w_i = \frac{1}{n}\sum_{j=1}^{n} \widehat{a_{ij}}. \quad (9)$$

3) Важно проверить консистентность полученных попарных сравнений с использованием коэффициента консистентности $C_r$. Коэффициент $C_r$ вычисляется как:

$$C_r = \frac{\lambda_{max} - n}{(n-1) * CI'}, \quad (10)$$

где $\lambda_{max}$ – наибольшее собственное значение матрицы попарных сравнений, а *n* – размер матрицы.

Если коэффициент $C_r$ < 0,1, то попарные сравнения считаются консистентными.

После вычисления весов для всех критериев и альтернатив, можно составить итоговую оценку для каждой альтернативы. Эта оценка рассчитывается как сумма произведений весов критериев и весов альтернатив для каждого критерия:

$$S_i = \sum_{j=1}^{n} w_j * v_{ij}, \quad (11)$$

где $S_i$ – итоговая оценка для альтернативы *i*, $v_{ij}$ – оценка альтернативы *i* по критерию *j*, $w_j$ – вес критерия *j*.

Альтернатива с наибольшей итоговой оценкой $S_i$ будет наилучшей альтернативой, которая соответствует заданной цели и удовлетворяет всем критериям.

Метод компромиссного программирования минимизирует отклонения от идеала по каждому критерию, стремясь найти сбалансированное решение с минимальными потерями.

Помимо традиционных алгоритмов, для решения задач многокритериальной оптимизации могут использоваться методы искусственного интеллекта, включая машинное обучение. Например, нейронные сети могут обучаться на данных, что позволяет предсказывать им качество продукции и оптимизировать параметры производства, адаптируясь к изменениям условий [4]. Также следует выделить генетические алгоритмы, которые имитируют естественный отбор, применяя скрещивание и мутации для улучшения решений с каждым поколением (подходит для задач с большим числом параметров и сложными взаимосвязями) [4].

### *Метаэвристические методы многокритериальной оптимизации*

Перспективным способом решения задач многокритериальной оптимизации сегодня считаются метаэвристики. Например, традиционные методы, основанные на градиенте, не справляются с решением сложных оптимизационных задач с высокоразмерными пространствами и нелинейными функциями. Это привело к развитию метаэвристических алгоритмов, вдохновленных биологией, в частности эволюцией и коллективным поведением живых организмов. Помимо биоинспирированных алгоритмов, сегодня активно развиваются методы, основанные на принципах физики и химии.

В последние годы новые алгоритмы, основанные на нелинейных физических процессах, продемонстрировали высокую эффективность и надежность в глобальной оптимизации, предлагая альтернативу методам на базе эволюции и коллективного поведения. Данный подход можно успешно применять для многокритериальной оптимизации параметров сложных производственных и бизнес-процессов.

Такие алгоритмы могут использоваться для многокритериальной оптимизации параметров процесса обеспечения качества топливных брикетов благодаря способности эффективно работать с множеством взаимозависимых показателей, таких как теплотворная способность, плотность, влажность и зольность. Эти алгоритмы находят компромиссные решения между конфликтующими параметрами, создавая набор Парето-оптимальных решений, что позволяет выбрать «наилучший» вариант в зависимости от приоритетов качества. Кроме того, они гибко адаптируются к изменениям производственного процесса, например, при смене сырья или изменении требований клиентов, и могут быстро перенастраиваться, поддерживая стабильное качество продукции. Их высокая вычислительная эффективность позволяет решать задачи с большими объемами данных без значительных затрат ресурсов. Таким образом, метаэвристические алгоритмы помогают снизить отклонения от стандартов, повышая общее качество и конкурентоспособность продукции, что делает их перспективным способом многокритериальной оптимизации параметров процесса обеспечения качества топливных брикетов.

С учетом [5] можно выделить следующие виды метаэвристических алгоритмов, подходящих для решения задачи обеспечения качества топливных брикетов (табл. 1).

Таблица 1 – Метаэвристические алгоритмы оптимизации

| Название алгоритма | Описание | Виды |
|---|---|---|
| 1. Алгоритмы механики жидкостей (Fluid Mechanics based Metaheuristics) | Способны адаптироваться к сложным взаимодействиям между параметрами, также обеспечивают эффективное чередование глобального поиска и локальной эксплуатации, что помогает находить компромиссные решения между такими параметрами, как плотность, зольность и теплотворная способность. | 1.1 Алгоритм поиска вихрей (Vortex Search Algorithm)<br>1.2 Алгоритм режима потока (Flow Regime Algorithm)<br>1.3 Алгоритм оптимизации по принципу Архимеда (Archimedes' Optimization Algorithm) |
| 2. Термодинамические алгоритмы (Thermodynamics based Metaheuristics) | Управляют параметрами с помощью механизмов передачи и распределения энергии, также позволяют находить решения, обеспечивающие баланс между показателями качества брикетов, моделируя различные состояния для учета противоречивых целей. | 2.1 Оптимизация теплообмена (Thermal Exchange Optimization)<br>2.2 Поиск состояний вещества (States of Matter Search)<br>2.3 Оптимизация газовых молекул (Kinetic Gas Molecules Optimization)<br>2.4 Оптимизатор растворимости газа Генри (Henry Gas Solubility Optimizer) |
| 3. Электромагнитные алгоритмы | Они используют силы притяжения и отталкивания, что эффективно при многокритериальной оптимизации, | 3.1 Алгоритм искусственного электрического поля (Artificial Electric Field Algorithm) |

| | | |
|---|---|---|
| (Electromagnetism based Metaheuristics) | где требуется учитывать несколько независимых параметров. | 3.2 Оптимизация, вдохновленная магнитным полем (Magnetic Inspired Optimization)<br>3.3 Алгоритм электромагнитных полей (Electromagnetic Fields Algorithm)<br>3.4 Оптимизация движения ионов (Ions Motion Optimization) |

Из класса Fluid Mechanics based Metaheuristics можно детально рассмотреть алгоритм режима потока (Flow Regime Algorithm, FRA). Он подходит для многокритериальных задач, так как сочетает глобальный и локальный поиск через моделирование турбулентного и ламинарного режимов потока. Турбулентный режим используется для глобального поиска, что помогает эффективно исследовать пространство решений и находить варианты, подходящие для различных параметров, таких как плотность, теплотворная способность и зольность. Ламинарный режим, напротив, ориентирован на локальный поиск, что позволяет FRA находить точные значения вблизи найденных компромиссных решений и улучшать качество брикетов по конкретным критериям [6].

Использование подходов, подобных числу Рейнольдса, позволяет FRA динамически управлять балансом между исследованием и эксплуатацией. Это особенно важно при оптимизации нескольких параметров качества, так как дает гибкость в реагировании на противоречивые цели, характерные для производства топливных брикетов. Формула для расчета числа Рейнольдса выглядит следующим образом:

$$Re = \frac{\rho * \mathrm{v} * \mathrm{L}}{\mu}$$

где $\rho$ – плотность жидкости (кг/м$^3$), v – скорость потока (м/с), L – характерный линейный размер (м), μ - динамическая вязкость жидкости (Па*с).

Число Рейнольдса помогает определить, будет ли поток ламинарным или турбулентным:

- ламинарный поток: $Re < 2000$;
- переходный режим: $2000 < Re < 4000$;
- турбулентный поток: $Re > 4000$.

В контексте алгоритма оптимизации на основе режима потока, число Рейнольдса можно применять для настройки алгоритма, чередуя глобальный и локальный поиск в зависимости от значений, аналогичных *Re*, где турбулентный режим активирует исследование, а ламинарный — эксплуатацию.

Из термодинамических алгоритмов (Thermodynamics based Metaheuristics) наиболее подходящим для рассматриваемой задачи будет алгоритм 4.2 Поиск состояний вещества (States of Matter Search, SMS). Данный алгоритм основан на моделировании переходов между физическими состояниями вещества.

Классически вещество существует в трех состояниях – газообразном, жидком и твердом. Эти состояния различаются по силе межмолекулярного взаимодействия: твердые тела имеют самую сильную силу притяжения между молекулами, за ними следуют жидкости, где силы притяжения несколько слабее, и газы, где эти силы очень слабы. Соответственно, плотность упаковки молекул уменьшается от твердых тел к газам, что приводит к различной степени взаимодействия молекул в каждом состоянии.

Молекулы газа обладают большей свободой движения по сравнению с жидкостями и твердыми телами, что моделируется как фаза исследования (глобальный поиск). Для твердых тел, где межмолекулярное взаимодействие почти отсутствует, моделируется эксплуатационная фаза (локальный поиск). Жидкости, где молекулы обладают умеренной подвижностью, представляют баланс между исследованием и эксплуатацией. В SMS агенты (молекулы) начинают с газообразного состояния (чистый поиск), а алгоритм постепенно изменяет интенсивность исследования и эксплуатации, пока не достигнет твердого состояния (чистая эксплуатация).

Основными причинами выбора алгоритма поиска состояний вещества являются:

1) Баланс между исследованием и эксплуатацией: SMS моделирует переходы между физическими состояниями вещества, что позволяет алгоритму адаптироваться к различным условиям, обеспечивая как глобальный поиск, так и локальную оптимизацию. В контексте обеспечения качества, это важно для нахождения оптимальных параметров, поскольку необходимо как исследовать новые параметры, так и эффективно использовать уже найденные.

2) Адаптивность к изменениям: Алгоритм начинает с «газообразного» состояния, что подразумевает высокий уровень свободы для поиска новых решений, и постепенно переходит к «твердому» состоянию, что позволяет детализировать и улучшать уже найденные решения. Это может быть полезно в ситуации, когда параметры качества необходимо постоянно корректировать на основе новых данных.

3) Моделирование молекулярных взаимодействий: SMS учитывает силу межмолекулярного взаимодействия, что может быть метафорически применимо для анализа различных параметров обеспечения

качества, где взаимодействие между различными факторами (например, состав материалов, условия производства) критически важно.

И, наконец, из электромагнитных (Electromagnetism based Metaheuristics) алгоритмов наиболее актуальным для задачи многокритериальной оптимизации параметров обеспечения качества топливных брикетов является 5.4 Оптимизация движения ионов (Ions Motion Optimization, IMO). Алгоритм оптимизации движения ионов (Ions Motion Optimization, IMO), предложенный Джавайди и соавторами [6], вдохновлен природой сил, действующих между положительно заряженными (катионами) и отрицательно заряженными (анионами) ионами. Алгоритм имитирует притяжение и отталкивание между ионами для достижения оптимального решения.

В IMO ионы перемещаются в пространстве поиска на основе зарядов, которые им присваиваются. Анионы притягиваются к катионам, и наоборот, что обеспечивает как исследование, так и эксплуатацию пространства поиска. В процессе оптимизации взаимодействие между ионами позволяет находить глобальные оптимальные решения за счет баланса притяжения и отталкивания, характерного для электростатических взаимодействий между заряженными частицами [7].

### *Заключение*

В условиях роста спроса на альтернативные источники энергии топливные брикеты становятся востребованным продуктом. Для обеспечения их конкурентоспособности необходимо уделять особое внимание многокритериальной оптимизации потребительских свойств, таких как калорийность, влажность, зольность и механическая прочность. Существует множество подходов к оптимизации, включая методы взвешенной суммы, анализа Парето, TOPSIS и иерархий, а также современные методы на основе искусственного интеллекта и генетических алгоритмов. Особый интерес в задаче многокритериальной оптимизации представляют метаэвристические алгоритмы. Такие подходы, как Fluid Mechanics, Thermodynamics и Electromagnetism based Metaheuristics, позволяют эффективно работать с нелинейными задачами и множеством взаимозависимых параметров, что важно для поддержания качества топливных брикетов. Например, алгоритмы на основе режимов потока и состояний вещества сочетают глобальный и локальный поиск, адаптируясь к изменяющимся условиям производства, что позволяет находить компромиссные решения между различными показателями качества. Внедрение метаэвристических методов в процесс обеспечения качества топливных брикетов может повысить их устойчивость производственных процессов к изменениям параметров сырья, улучшить стабильность характеристик и повысить конкурентоспособность продукции.

*Работа выполнена в рамках государственного задания, проект №FSWF-2023-0012.*

### *Литература*

1. Муллина Э.Р., Мишурина О.А., Бессонова Ю.А., Басков В.А. Анализ факторов, влияющих на качество топливных брикетов // Вестник МГТУ им. Г. И. Носова. 2022. №3. URL: https://cyberleninka.ru/article/n/analiz-faktorov-vliyayuschih-na-kachestvo-toplivnyh-briketov.

2. Богданова П.А., Сахаров Д.М., Васильева Т.В. Обзор методов многокритериальной оптимизации в задачах принятия решений // Инновационные аспекты развития науки и техники. 2021. №6. URL: https://cyberleninka.ru/article/n/obzor-metodov-mnogokriterialnoy-optimizatsii-v-zadachah-prinyatiya-resheniy.

3. Пипия Г.Т. Целевые критерии для многокритериальной модели условной оптимизации оценки качества продукции // Системный анализ в проектировании и управлении: сборник научных трудов XXII Международной научно-практической конференции. СПБ: Санкт-Петербургский политехнический университет Петра Великого, 2018. С. 184-190.

4. Фам Куанг Хиеп, Квятковская И.Ю. Решение задач многокритериальной оптимизации для оценки качества объектов с неоднородными признаками // Вестник СГТУ. 2014. №1. URL: https://cyberleninka.ru/article/n/reshenie-zadach-mnogokriterialnoy-optimizatsii-dlya-otsenki-kachestva-obektov-s-neodnorodnymi-priznakami.

5. Soumitri Chattopadhyay, Aritra Marik, Rishav Pramanik A Brief Overview of Physics-inspired Metaheuristic Optimization Techniques [Электронный ресурс] – Режим доступа: https://arxiv.org/abs/2201.12810.

6. Щербина О.А. Метаэвристические алгоритмы для задач комбинаторной оптимизации (обзор) // ТВИМ. 2014. №1 (24). URL: https://cyberleninka.ru/article/n/metaevristicheskie-algoritmy-dlya-zadach-kombinatornoy-optimizatsii-obzor.

7. Булыгина О.В., Ярцев Д.Д., Прокимнов Н.Н., Верейкина Е.К. Направления гибридизации алгоритмов роевого интеллекта и нечеткой логики для решения оптимизационных задач в социально-экономических системах // Прикладная информатика. 2024. Т.19. №5. С.45-67.

***Булыгина О.В., Хлусович К.В. Анализ современных методов оптимизации параметров процесса обеспечения качества топливных брикетов.*** *Статья посвящена обзору современных методов многокритериальной оптимизации параметров процесса обеспечения качества топливных брикетов, таких как калорийность, влажность, зольность и механическая прочность. Особое внимание уделено метаэвристическим алгоритмам, способным находить оптимальные решения при учете противоречивых требований к качеству и изменяющихся условий.*

***Ключевые слова:*** *альтернативные источники энергии, многокритериальная оптимизация, качество продукции, метаэвристические алгоритмы, алгоритм режима потока, алгоритм поиска состояний вещества, оптимизация движения ионов*

***Bulygina O.V., Khlusovich K.V. Analysis of modern methods for optimizing the parameters of the process of ensuring the quality of fuel briquettes****. The article is devoted to a review of modern methods of multi-criteria optimization of the parameters of the process of ensuring the quality of fuel briquettes, such as calorific value, moisture, ash content and mechanical strength. Particular attention is paid to metaheuristic algorithms capable of finding optimal solutions taking into account conflicting quality requirements and changing conditions.*

***Keywords:*** *alternative energy sources, multi-criteria optimization, product quality, metaheuristic algorithms, Flow Regime Algorithm, States of Matter Search, Ions Motion Optimization*

УДК 004.932.72

# К вопросу об извлечении знаний из мультимодальных моделей для задачи распознавания объектов

**Я.С. Пикалёв[*1], Р.С. Хакимов[*2]**

[*1] к.т.н, с.н.с., Федеральное государственное бюджетное научное учреждение
«Институт проблем искусственного интеллекта»,
pikaliov@gmail.com

[*2] м.н.с., Федеральное государственное бюджетное научное учреждение
«Институт проблем искусственного интеллекта»,
khakimov.ru@mail.ru

***Пикалёв Я.С., Хакимов Р.С. К вопросу об извлечении знаний из мультимодальных моделей для задачи распознавания объектов.*** *В статье рассматриваются современные методы и подходы к извлечению знаний из мультимодальных моделей для решения задачи распознавания объектов. Проанализированы ключевые преимущества мультимодального подхода, включая интеграцию различных типов данных (изображения, текст, видео) и повышение точности распознавания за счёт межмодального выравнивания. Представлены методики обучения и дистилляции знаний. Особое внимание уделено перспективам применения мультимодальных моделей в задачах, связанных с анализом и обработкой мультимедиа.*

***Ключевые слова:*** *мультимодальные модели, распознавание объектов, дистилляция знаний, интеграция данных, искусственный интеллект, компьютерное зрение.*

## *Введение*

Извлечение (дистилляция) знаний (Knowledge Distillation, KD) – это метод, при котором меньшая модель (ученик) обучается воспроизводить поведение более крупной и сложной модели (учителя). В контексте мультимодальных моделей KD позволяет уменьшить размер и вычислительную сложность моделей, сохраняя их производительность в задачах, связанных с несколькими модальностями (текст, изображения, видео, и т. д.).

Дистилляция знаний из мультимодальных моделей позволяет передать способность извлечения взаимосвязей между различными модальностями, такими как текст и изображения, в упрощённую модель. В контексте задачи распознавания объектов это означает, что компактная модель может не только эффективно обнаруживать объекты, но и учитывать контекст, например, текстовые подсказки или пространственные связи, которые учитель извлекает из мультимодальных данных.

Цель данной работы – проанализировать современные подходы и методы извлечения знаний из мультимодальных моделей в контексте задачи распознавания объектов. Особое внимание уделено выявлению преимуществ мультимодального подхода, описанию ключевых методик интеграции данных разных модальностей. Рассматриваются перспективы их использования в различных областях, требующих высокого уровня интерпретации данных.

Дистилляция знаний для задачи распознавания объектов направлена на улучшение производительности компактных моделей-учеников, которые работают быстрее и требуют меньше ресурсов, сохраняя при этом высокую точность и эффективность детекции.

Основные цели дистилляции можно разделить на несколько категорий:

1. Повышение точности ученика

– Сохранение знаний учителя: Компактная модель-ученик обучается воспроизводить результаты сложной модели-учителя, что позволяет ей наследовать высокий уровень точности.

– Учет сложных связей: Учитель передаёт ученику способность распознавать объекты с учётом контекста, пространственных и семантических зависимостей.

2. Сокращение вычислительных затрат

– Оптимизация для реального использования: Компактные модели могут эффективно работать на устройствах с ограниченными ресурсами (мобильные устройства, встроенные системы).

3. Уменьшение размера модели

– Экономия памяти: Ученик значительно меньше по размеру, что облегчает его использование в системах с ограниченными ресурсами.

– Ускорение работы: Упрощённые модели обеспечивают более быстрое выполнение задач без значительных потерь в точности.

4. Сохранение мультимодальной информации

• Перенос межмодальных связей: в случае мультимодальных моделей учитель передаёт ученику способности учитывать зависимости между текстом, изображением и другими модальностями.

• Контекстное понимание: Для задач, где важны текстовые подсказки (например, описания объектов), ученик наследует способность обрабатывать их на высоком уровне.

5. Повышение устойчивости к разнообразию данных

• Обучение на сложных примерах: Ученик получает знания о редких или сложных объектах, которые учитель уже умеет детектировать.

• Устойчивость к шуму: Знания учителя помогают ученику сохранять точность при работе с неидеальными или зашумлёнными данными.

6. Универсальность модели

• Адаптация к новым задачам: Дистилляция позволяет ученику применять знания учителя в смежных задачах (например, сегментация или классификация).

• Гибкость в различных сценариях: Ученик сохраняет способность работать в условиях ограниченной доступности одной из модальностей (например, текста)[1].

В контексте задачи распознавания объектов с использованием мультимодальных моделей, учитель и ученик играют ключевые роли. Учитель – это сложная и мощная модель, обученная на больших наборах данных, которая служит источником знаний. Ученик – более компактная модель, которая извлекает знания из учителя, сохраняя основные возможности при меньшей вычислительной стоимости.

***Архитектура учителя. Основные компоненты:***

1. Энкодеры модальностей:
   - Текстовый энкодер:
     - Трансформеры (например, GPT или BERT).
     - Преобразуют текстовые данные в эмбеддинги.
   - Визуальный энкодер:
     - ResNet, Vision Transformers (ViT).
     - Извлекают пространственные признаки из изображений.
2. Модуль межмодальной интеграции:
   - Обеспечивает объединение признаков из разных модальностей.
   - Например, кросс-внимания для текста и изображения.
3. Детектор объектов:
   - Использует мультимодальные признаки для локализации и классификации объектов.
   - Часто основан на архитектурах Faster R-CNN или DETR.

***Архитектура ученика. Основные компоненты:***

1. Визуальный энкодер:
   - Упрощённая версия CNN или ViT.
   - Извлекает только визуальные признаки.
2. Детектор объектов:
   - Использует визуальные признаки для предсказания координат рамок и классов.
   - Упрощённая структура на основе моделей, таких как YOLO.

3. Простая интеграция знаний – включает механизмы для воспроизведения мультимодальных представлений учителя.

Таблица 1 – Основные отличия между учителем и учеником

| Характеристика | Учитель | Ученик |
|---|---|---|
| Сложность | Глубокая модель с большим количеством параметров | Компактная модель с минимальными ресурсами |

| | | |
|---|---|---|
| Мультимодальность | Обрабатывает несколько модальностей (текст, изображение) | Может работать с одной модальностью (например, изображением) |

Продолжение таблицы 1

| | | |
|---|---|---|
| Цель | Извлечение сложных межмодальных зависимостей | Упрощённое воспроизведение знаний учителя |
| Использование ресурсов | Высокие требования к памяти и вычислениям | Эффективная работа на ограниченных устройствах |
| Применение | Научные исследования, большие серверные системы | Реальное использование: мобильные устройства, встроенные системы |

### *Взаимодействие учителя и ученика*

Во время обучения ученик получает знания от учителя через:

1. Предсказания: Ученик учится воспроизводить выходы модели учителя (рамки объектов, классы).
2. Промежуточные представления: Ученик воспроизводит эмбеддинги, которые учитель генерирует на промежуточных этапах.
3. Карты внимания: Ученик копирует механизм, который использует учитель для фокусировки на ключевых частях изображения.
4. Межмодальные связи: Ученик перенимает способность учителя учитывать текстовые подсказки для улучшения детекции.

### *Основные методы извлечения знаний*

1. Дистилляция предсказаний (Logit Distillation)

Дистилляция предсказаний – это один из наиболее базовых и широко используемых методов передачи знаний от модели-учителя к модели-ученику. Этот подход фокусируется на переносе информации о выходных распределениях модели-учителя, таких как вероятности классов и параметры ограничивающих рамок (bounding boxes), которые учитель генерирует.

Основная идея – модель-учитель передаёт модели-ученику "мягкие" предсказания — вероятностные распределения, которые содержат больше информации, чем просто метки классов. Эти предсказания помогают ученику лучше понять сложные взаимосвязи между данными, включая относительную уверенность в предсказаниях.

Компоненты метода:

1) Вероятности классов (Class Probabilities):

- Учитель передаёт распределение вероятностей по всем классам, а не только правильный класс.
- Пример: для объекта, который принадлежит классу "кошка", учитель может предсказать [0.7, 0.2, 0.1], что означает 70% уверенности в классе "кошка", 20% – "собака", 10% – "птица".

2) Ограничивающие рамки (Bounding Boxes):

- Учитель передаёт координаты ограничивающих рамок объектов, включая такие параметры, как центр рамки, ширина и высота.
- Например, координаты ($x$, $y$, $\omega$, $h$), где $x$, $y$ – координаты центра, $\omega$, $h$ – ширина и высота.

Функция потерь

– Для классов объектов:

$$L_{\text{class}} = \text{KL}(p_{\text{teacher}} || p_{\text{student}}),$$

где KL – дивергенция Кульбака-Лейблера между распределениями вероятностей,

$p_{\text{teacher}}$ – распределение вероятностей учителя,

$p_{\text{student}}$ – распределение вероятностей ученика.

– Для координат рамок:

$$L_{\text{bbox}} = \|\text{BBox}_{\text{teacher}} - \text{BBox}_{\text{student}}\|^2,$$

где BBox – параметры рамок, такие как ($x$, $y$, $\omega$, $h$).

– Общая функция потерь:

$$L = \alpha \cdot L_{\text{class}} + \beta \cdot L_{\text{bbox}},$$

где α и β – веса для задач классификации и предсказания рамок соответственно.

Преимущества метода:

– Простота реализации.

– Снижение требований к данным: ученик может эффективно обучаться даже на небольших наборах данных, используя знания учителя.

Ограничения

– Чувствительность к качеству учителя: Если учитель недостаточно точен, ученик наследует его ошибки.

– Ограниченная сложность: Не передаются глубокие взаимосвязи или пространственные отношения, которые можно изучить с помощью других методов, таких как дистилляция признаков.

2. Дистилляция признаков (Feature Distillation)

Дистилляция признаков – это метод передачи знаний, при котором модель-учитель делится с модельюучеником промежуточными представлениями (фичами), извлечёнными из данных. Этот подход позволяет ученику обучаться воспроизводить представления, которые формируются на скрытых слоях учителя, сохраняя важную пространственную и семантическую информацию.

В отличие от дистилляции предсказаний, которая фокусируется на выходных распределениях, дистилляция признаков работает с промежуточными признаками, извлечёнными учителем. Эти признаки могут включать пространственные карты, высокоуровневые представления объектов и другие признаки, которые помогают ученику лучше понимать структуру данных.

Компоненты метода:

1) Извлечение признаков:

– Учитель генерирует представления на скрытых слоях (например, пространственные карты или эмбеддинги) на промежуточных этапах обработки данных.

– Примеры признаков: карты внимания, карты активации, тензоры с информацией об объектах.

2) Передача знаний:

– Ученик обучается повторять эти представления, минимизируя различия между своими признаками и признаками учителя.

Функция потерь:

$$L_{\text{feature}} = \|h_{\text{teacher}} - h_{\text{student}}\|^2,$$

где $h$ – промежуточные представления (тензоры признаков).

Преимущества метода

– Сохранение контекста: Передаются пространственные и семантические взаимосвязи, важные для распознавания сложных объектов.

– Универсальность: Подходит для большинства задач компьютерного зрения, включая классификацию, детекцию, сегментацию.

Ограничения

– Чувствительность к различиям архитектур: Если учитель и ученик имеют существенно разные структуры, передача признаков может быть неэффективной.

– Необходимость согласования: Признаки учителя и ученика могут иметь разные размеры или формы, что требует дополнительных методов для их выравнивания.

3. Дистилляция карт внимания (Attention Distillation)

Дистилляция карт внимания – это метод передачи знаний, при котором модель-учитель обучает модельученика воспроизводить своё распределение внимания. Картами внимания называют пространственные или активационные карты, которые показывают, на какие области входных данных (например, изображения) модель фокусируется при выполнении задачи.

Карта внимания отражает важные для модели регионы данных, которые она использует для предсказаний. Дистилляция карт внимания помогает ученику сконцентрироваться на тех же областях, что и учитель, ускоряя его обучение и повышая точность.

Компоненты метода:

1) Извлечение карт внимания:

– Учитель генерирует карты внимания из своих промежуточных активаций.

– Примеры карт:

  – Карты, показывающие силу активаций в разных регионах изображения.

  – Карты на основе весов внимания в трансформерах.

2) Сравнение карт учителя и ученика:

– Карты внимания ученика сравниваются с картами учителя, и потери вычисляются на основе их сходства.

3) Передача знаний:

– Модель-ученик минимизирует разницу между своими картами внимания и картами учителя.

Функция потерь:

$$L_{\text{attention}} = \|A_{\text{teacher}} - A_{\text{student}}\|^2,$$

где $A$ – карты внимания, полученные из скрытых слоёв модели.

Преимущества метода

– Сохранение пространственных отношений: Карты внимания сохраняют информацию о пространственном распределении объектов.

– Улучшение интерпретируемости: Карты внимания делают модель более интерпретируемой, показывая, где она "смотрит".

Ограничения

– Чувствительность к выбору карт: Разные слои учителя могут генерировать карты разного качества. Неправильный выбор слоя может снизить эффективность.

– Высокая вычислительная сложность: Работа с картами внимания требует значительных вычислительных ресурсов.

– Разные архитектуры: Для учителя и ученика с разными архитектурами требуется адаптация карт.

4. Дистилляция отношений между объектами (Relation Distillation)

Дистилляция отношений между объектами – это метод передачи знаний, при котором акцент делается на обучении модели-ученика воспроизводить семантические и пространственные связи между объектами, которые выявляет модель-учитель. Этот подход особенно эффективен в задачах, где важны не только характеристики объектов, но и их взаимосвязь, например, в детекции объектов, сегментации сцен, визуальном вопросно-ответном модуле (VQA) и других задачах компьютерного зрения.

Объекты на изображении часто взаимодействуют друг с другом, и эти взаимодействия содержат важную информацию для понимания сцены. Например:

– Расположение чашки относительно стола.

– Взаимодействие людей в группе.

– Пространственные отношения, такие как «слева от», «над», «рядом с».

Модель-учитель извлекает эти связи и передаёт их ученику, помогая последнему лучше понять контекст сцены.

Компоненты метода:

1) Извлечение отношений – учитель генерирует представления отношений между объектами на основе их признаков и пространственных характеристик. Это могут быть векторы, матрицы или графовые представления.

2) Передача знаний – ученик обучается воспроизводить эти отношения, минимизируя различия между своими представлениями и представлениями учителя.

3) Используемые типы отношений:

– Пространственные: взаимное расположение объектов (например, расстояние или относительные координаты).

– Семантические: смысловые связи между объектами (например, объект A принадлежит категории "мебель", а объект B – "посуда").

– Контекстуальные: как объекты влияют на общий смысл сцены.

Функция потерь:

$$L_{\text{relation}} = \|R_{\text{teacher}} - R_{\text{student}}\|,$$

где $R$ – матрица отношений объектов.

Преимущества:

– Улучшает понимание сложных сцен с большим количеством объектов.

– Повышает точность за счёт учёта контекста.

Ограничения

– Сложность вычислений: извлечение и сопоставление отношений между объектами требует значительных ресурсов.

– Чувствительность к качеству учителя: ошибки учителя в определении отношений могут негативно повлиять на обучение ученика.

– Проблемы со шкалированием: в сценах с большим количеством объектов число пар или графовых связей быстро растёт.

5. Многозадачная дистилляция (Multi-Task Distillation)

Многозадачная дистилляция – это метод передачи знаний, при котором модель-учитель обучает модель-ученика решать несколько задач одновременно. Этот подход позволяет эффективно использовать ресурсы и улучшать качество выполнения всех задач, комбинируя знания, извлечённые из различных аспектов данных.

Вместо того чтобы обучать отдельные модели для каждой задачи, многозадачная дистилляция передаёт знания от одной мощной модели-учителя, способной решать несколько задач, к одной или нескольким моделям-ученикам. Ученикы могут быть:

– Универсальными, выполняющими те же задачи, что и учитель.

– Узкоспециализированными, ориентированными на одну задачу, но с учётом знаний из других задач.

Компоненты многозадачной дистилляции

1) Обучение учителя – учитель обучается решать все задачи одновременно, создавая представления, которые содержат общую и специфическую информацию.

2) Экстракция знаний – учитель передаёт ученику:

– Предсказания для каждой задачи (логиты, вероятности).

– Промежуточные представления (признаки, карты внимания).

3) Обучение ученика – ученик минимизирует различия между своими предсказаниями и представлениями и результатами учителя.

Функция потерь:

$$L = \alpha \cdot L_{\text{detection}} + \beta \cdot L_{\text{caption}},$$

где $L_{\text{detection}}$ – потери для распознавания, $L_{\text{caption}}$ – потери для генерации описаний.

Преимущества:

- Позволяет ученику быть универсальной моделью.
- Снижает потребность в нескольких специализированных моделях.

Ограничения

- Балансировка задач – неправильная настройка коэффициентов может привести к тому, что одна задача доминирует над другими.
- Взаимная зависимость – если одна задача плохо решается учителем, это может негативно повлиять на обучение ученика.

6. Дистилляция межмодального выравнивания (Cross-Modal Distillation)

Дистилляция межмодального выравнивания – это метод передачи знаний, при котором одна модальность (например, текст, изображение, звук) используется как учитель для другой модальности (ученика). Цель – улучшить понимание и взаимодействие между различными модальностями за счёт обучения модели-ученика извлекать и интерпретировать взаимосвязанные данные, даже если они изначально принадлежат разным доменам.Функция потерь:

$$L_{\text{alignment}} = \left\| S_{\text{teacher}}(x_{\text{text}}, x_{\text{image}}) - S_{\text{student}}(x_{\text{text}}, x_{\text{image}}) \right\|,$$

где $S(x_{\text{text}}, x_{\text{image}})$ – функция оценки соответствия между текстом и изображением.

Подходы к межмодальной дистилляции:

– Односторонняя дистилляция – учитель обучает ученика в одной модальности. Пример: текстовая модель учит визуальную модель понимать текстовые подсказки.

– Взаимная дистилляция – учитель и ученик обучаются друг у друга. Пример: текстовая и визуальная модели совместно выравниваются.

– Дистилляция через общий латентный слой – модели обеих модальностей преобразуют данные в общее векторное пространство. Пример: представления изображений и текста проецируются в один и тот же латентный вектор.

Преимущества:

– Повышает производительность в задачах, где используются текстовые подсказки.

– Сохраняет контекстную информацию из текстов.

Ограничения

– Качество учителя – если учитель плохо обучен, это может снизить производительность ученика.

– Выравнивание доменов – разные модальности имеют разные распределения данных, что усложняет их выравнивание[3].

Таким образом, мультимодальное извлечение знаний имеет критическое значение, поскольку оно позволяет моделям решать задачи с большей точностью, универсальностью и эффективностью, чем использование одной модальности. Вот ключевые причины, почему это важно:

1. Богатство данных для более глубокого анализа. Разные модальности предоставляют уникальную информацию, которую сложно или невозможно получить из одного источника. Например:

– Изображения: содержат пространственные и визуальные признаки, такие как форма, цвет и текстура.

– Текст: добавляет семантический и контекстный уровень, поясняя, что изображено.

– Видео: вводит временную составляющую, раскрывая динамику объектов.

Сочетание этих модальностей позволяет создать более полное представление об объектах, их связях и контексте.

2. Устранение неоднозначностей. Иногда визуальная информация недостаточна для точной идентификации объектов. Примеры:

– Однотипные объекты: яблоко и помидор могут выглядеть похоже, но текстовые подсказки (например, "фрукт" или "овощ") устраняют неопределённость.

– Контекст: текст может указать, что объект используется в определённой ситуации, например, "кофейная чашка на рабочем столе".

3. Обобщение и переносимость. Мультимодальные модели лучше обобщают информацию:

– Они могут успешно работать с данными, которые редко встречаются в обучающем наборе.

– Благодаря мультимодальности, модель учится связывать понятия из одной модальности (например, текст) с другой (например, изображение), что облегчает применение знаний на новых доменах.

4. Улучшение пользовательского опыта. В реальных приложениях мультимодальное извлечение знаний делает системы более интуитивными и эффективными:

– Визуальные подсказки: текст может уточнять, на что именно обращать внимание при анализе изображения.

– Генерация описаний: автоматическое описание изображения текстом делает его доступным для пользователей с ограничениями по зрению.

5. Снижение требований к разметке данных. Традиционные модели для распознавания объектов требуют больших размеченных наборов данных. Мультимодальные модели используют межмодальные связи для:

– Заполнения пробелов в данных одной модальности за счёт информации из другой.

– Улучшения обучения на основе данных без разметки (self-supervised learning).

6. Улучшение моделей в реальных условиях. В реальных сценариях объекты могут быть:

– Частично закрыты (occluded);

– Неясно видны из-за плохого освещения или качества изображения.

Мультимодальные данные помогают компенсировать эти ограничения:

– Текст, связанный с изображением, предоставляет дополнительное описание.

– Видео добавляет информацию о движении и последовательности событий.

7. Синергия между модальностями. Мультимодальные модели позволяют выявлять связи между различными модальностями, что создаёт:

– Синергетический эффект: объединение модальностей улучшает результаты, чем если бы каждая использовалась отдельно.

– Модели общего представления: объекты из разных модальностей (например, изображение и текст) можно сопоставить в едином латентном пространстве, что облегчает их обработку.

8. Преимущества в создании компактных моделей. Благодаря методам дистилляции знаний, мультимодальные модели могут передавать знания компактным (моно-)модальным моделям. Это даёт:

– Снижение вычислительных затрат.

– Возможность использования мощных моделей на устройствах с ограниченными ресурсами, таких как мобильные телефоны.

9. Будущее мультимодальных систем. С развитием технологий взаимодействие с системами становится всё более мультимодальным:

– Чат-боты: текстовый запрос может быть усилен изображениями для более точного ответа.

– AR/VR: комбинируют визуальные данные с аудио и текстом для создания иммерсивного опыта.

– Генеративные модели: мультимодальные системы, такие как DALL-E или Stable Diffusion, используют текст для создания изображений[4].

Извлечение знаний из мультимодальных моделей играет центральную роль в создании более умных, универсальных и эффективных систем для распознавания объектов. Это направление решает широкий круг задач: от улучшения точности и интерпретации до создания компактных моделей, которые работают в условиях ограниченных ресурсов. В будущем мультимодальные подходы продолжат находить новые применения в самых разных областях, формируя основу для интеллектуальных систем нового поколения.

В данной работе были рассмотрены основные подходы по дистилляции знаний из мультимодальных моделей, их ключевые свойства, преимущества и недостатки. Данная работа может быть полезна ученым и исследователям в сфере машинного обучения для уменьшения количества параметров в нейросетевых архитектурах.

### *Литература*

1. Hinton G. Distilling the Knowledge in a Neural Network / Hinton G., Vinyals O., Dean J. – 2015. – С.1–9.

2. Cui H. Open Visual Knowledge Extraction via Relation-Oriented Multimodality Model Prompting / Cui H.,

Fang X., Zhang Z., Xu R., Kan X., Liu X., Yu Y., Li M., Song Y., Yang C. // Advances in Neural Information Processing Systems – 2023. – T. 36 – № NeurIPS – C.1–21.

3. Detail A.I. Contrastive Language-Image Pre-Training with Knowledge Graphs – Supplementary Material / Detail A.I. – 2022. – № NeurIPS – C.1–6.

4. Bianchi F. Contrastive Language-Image Pre-training for the Italian Language / Bianchi F., Attanasio G., Pisoni R., Terragni S., Sarti G., Balestri D. // CEUR Workshop Proceedings – 2023. – T. 3596.

***Пикалёв Я.С., Хакимов Р.С. К вопросу об извлечении знаний из мультимодальных моделей для задачи распознавания объектов.*** *В статье рассматриваются современные методы и подходы к извлечению знаний из мультимодальных моделей для решения задачи распознавания объектов. Проанализированы ключевые преимущества мультимодального подхода, включая интеграцию различных типов данных (изображения, текст, видео) и повышение точности распознавания за счёт межмодального выравнивания. Представлены методики обучения и дистилляции знаний. Особое внимание уделено перспективам применения мультимодальных моделей в задачах, связанных с анализом и обработкой мультимедиа.*

*Ключевые слова: мультимодальные модели, распознавание объектов, дистилляция знаний, интеграция данных, искусственный интеллект, компьютерное зрение.*

***Pikalyov Yaroslav, Khakimov Renat On the issue of knowledge distillation from multimodal models for object detection.*** *The article explores modern methods and approaches to knowledge distillation from multimodal models for solving object recognition tasks. Key advantages of the multimodal approach are analyzed, including the integration of various data types (images, text, video) and improved recognition accuracy through cross-modal alignment. Training and knowledge distillation techniques are presented, with particular attention given to the prospects of applying multimodal models to tasks related to multimedia analysis and processing.*

***Keywords:*** *multimodal models, object detection, knowledge distillation, data integration, artificial intelligence, computer vision.*

УДК 004.896

# Структура метаданных сегментов карт навигации дрона

**Б.В.Павленко**[*1]

[*1] аспирант, ФБГНУ «Институт проблем искусственного интеллекта»
bogdanpavl2000@mail.ru

***Павленко Б.В., «Структура метаданных сегментов карт навигации дрона»***

*Числовые и текстовые метаданные визуальной информации очень важны для обучения моделей, если необходимо задать дополнительную контекстную информацию разного формата. В контексте задачи определения местоположения по распознаванию местности рассматривается вопрос о структуре метаданных для сегментов карт, используемых беспилотными летательными аппаратами (БПЛА) в условиях частичной или полной потери сигнала. Рассматривается подход, в котором визуальные данные и текстовые описания интегрируются для повышения точности геолокализации. В основе работы лежит деление карт на ячейки с фиксированными метаданными, включая GPS-координаты, высоту съемки, угол наклона камеры, а также внутренние индексы и массив окружающих ячеек. Уникальной особенностью предложенной структуры является использование динамического пересчета локальной ориентации объектов на основе глобальной карты, что позволяет эффективно адаптировать метаданные без их избыточного хранения.*

***Ключевые слова:*** *БПЛА, геолокализация, карты, распознавание изображений, метаданные.*

## *Введение*

На сегодняшний день беспилотные летательные аппараты (БПЛА) широко применяются в различных областях, таких как спасательные операции, разведка, военные разработки, точное земледелие и мониторинг. Параллельно с этим происходит бурное развитие технологий искусственного интеллекта и растет тенденция на его повсеместное внедрение.

Технологии интеллектуального анализа данных, такие как распознавание изображений являются объектом интереса с точки зрения их применения в задачах полуавтономной и беспилотной навигации. Важной научной и практической задачей является обеспечение надежной автономной навигации дронов, особенно в условиях частичной или полной потери сигнала управления, а также недоступность навигации по GPS.

Задача ориентирования БПЛА на местности без доступа к навигационным системам сводится к решению задачи геолокализации на основе сопоставления перекрестных видов и известного набора размеченных областей карты. Формирование специализированного набора данных для решения этой задачи с помощью глубокого обучения поднимает вопрос использования дополнительных данных, характеризующих местность и входную визуальную информацию – метаданных. Также метаданные описывают больший контекст, что делает их важным элементом для качественного обучения мультимодальной модели.

Исходя из вышесказанного, целью работы является разработка структуры метаданных для сегментов карт, которая позволила бы модели эффективно адаптировать метаданные и хранящуюся в них, в том числе, контекстную информацию, например такую, как относительное положение объектов, изменяющихся при изменении ориентации дрона.

## *Проблема потери сигнала*

Потеря сигнала управления БПЛА в следствие естественных причин, таких, как погодные условия, ограниченность радиуса действия беспроводного сигнала, или целенаправленных воздействий, таких как средства радиоэлектронной борьбы или взлом для перехвата – важные проблемы практического характера. В независимости от характера выполняемых работ, будь то задачи быстрого реагирования спасательных служб, например пожарные дроны [1], боевые задачи (разведчики, ударные, боевые БПЛА [2]), управление и потенциальные ограничения остаются общими.

БПЛА в основном управляются оператором или имеют автономный режим полета до заданной цели. В устройстве удаленного управления дроном используются определенные протоколы, определенные рабочие диапазоны частот и специализированные контроллеры со своими радиоканалами, что, однако, вовсе не устраняет возможность потенциального взлома БПЛА или глушения GPS-сигнала [3][4].

Помимо глушения, серьезную проблему представляет наличие проработанных атак на разные чипы и протоколы, как например взлом WiFi с WEP-шифрованием и использование уязвимостей чипов и контроллеров, как например с Xbee – распространенными модулями беспроводной связи. Неисправленные уязвимости позволяют довольно легко взламывать дроны, перехватывать сигналы, управление, видеопоток с камеры, загружать вредоносное ПО или просто выводить устройство из строя.

Можно предположить также, что в отдельных случаях во избежание взлома или перехвата может возникнуть необходимость отключения любых передающих сигнал компонентов. Из-за этого дрону может полагаться только на внедренные в его память карты и сопоставление видимой им местности с известными метками загруженной карты, если предполагать, что он оснащен соответствующей моделью.

### *Используемый подход*

Как известно, задача геолокализации заключается в определении местоположения объекта или области на основе входных данных, таких как изображения. Для оценки потенциала дронов в такого рода задачах авторы работы «University-1652 A Multi-view Multi-source Benchmark for Drone-based Geo-localization» [5] разработали новый многовидовой и бенчмарк под названием University-1652, в котором объединили данные с трех платформ: изображений с дронов, спутников и наземных камер для 1652 университетских зданий по всему миру.

Авторы утверждают, что большинство наборов данных для данной задачи используют сопоставление двух видов – наземного и спутникового, что несет в себе такие проблемы, как обусловленные особенностью местности случайные перекрытия видов на наземных снимках, например деревья или другие здания. Это усложняет модели поиск признака, инвариантного к точке обзора. Соответственно, они предложили использовать дроны, как источник сбора подходящих для данной задачи снимков, способных охватить виды с множества направлений без проблем, присущих наземным снимкам. University-1652 был первым набором данных, позволяющим решать одновременно две новые задачи: локализацию цели с использованием изображений дрона и его навигацию.

Представленный в работе [5] набор данных в последствии упоминался авторами работы Orientation-Guided Contrastive Learning for UAV-View Geo-Localisation [6], в которой предлагается методика обучения модели для решения задачи предсказания меток позиции за счет извлечения признаков из изображений со спутника и дрона (см.рис.1).

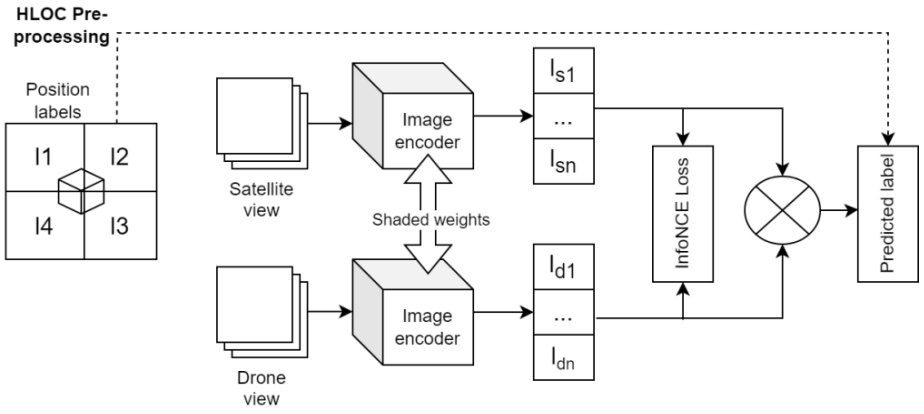



Рисунок 1 – Схема конвейера обучения модели на основе подхода из работы Orientation-Guided Contrastive Learning for UAV-View Geo-Localisation

В представленном методе используется метод иерархической локализации (HLOC) [7] для оценки ориентации видов, а также общие веса энкодеров, что часто применяется в рамках контрастивного обучения и служит для синхронизации представлений между разными модальностями данных. Если данные представляют собой дополнительное текстовое описание контекста и метаданные, то для корректной связи управляющей программы-автопилота и нейронной сети может потребоваться четкая структура хранения этих данных.

### *Карта и структура метаданных*

Последующие предложения основываются на концепции, что по умолчанию дрон имеет в памяти некоторую область карты. В период времени со старта $t_0$ до времени $t_x$, когда он может потерять сигнал, он получает сигналы управления и данные GPS.

Предположим, начальная карта имеет размерность $HxW$ км. Для удобства работы с ней предлагается разделить ее на сетку, где у каждой ячейки будет свой набор метаданных. Первая группа данных – это «внешние»

данные: координаты GPS, высота съемки, угол наклона камеры. Вторая группа – это «внутренние» данные: индексы текущей ячейки карты в виде пары *(i, j)*, и массив окружающих клеток карты на удалении равном 1.

Предположим, каждая клетка хранит структуру о положении окружающих ячеек относительно двух слоев, глобальной карты и локальной ориентации. Эти структуры описываются массивом из строк и столбцов, где ячейки равноудалены от центральной, соответствующей текущему положению. Они хранят глобальные индексы ячеек и коды направлений, составленные из первых букв сокращений TOP, BOTTOM, CENTER, LEFT, RIGHT (см.рис.2).Обозначения сторон для ячеек следующие:

- TL – сверху слева;
- TF – сверху спереди;
- TR – сверху справа;
- CL – центр слева;
- CR – центр справа;
- BL – снизу слева;
- BF – снизу прямо;
- BR – снизу справа.

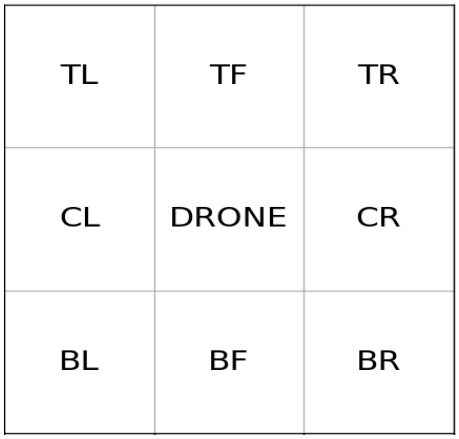



Рисунок 2 – Окружающие ячейки на удалении равном 1

Ячейки глобальной карты всегда имеют одно направление согласно сторонам света и не меняются в зависимости от поворота дрона, в отличие от массива ячеек локальной ориентации, которые меняются соответственно направлению взгляда (см.рис.3, 4). Слева на рисунках показаны изменяемые локальные коды расположений.

Допустим, что дрон «смотрит» в глобальном направлении TF, предположим, что в зоне его видимости в ячейке TL находятся объекты. Тогда описание относительно глобальной карты даст следующие направления: объект слева спереди (TL), что соответствует локальным направлениям до его поворота. После поворота на 90 градусов вправо, объекты, находящиеся в ячейке глобальной карты TL, для дрона будет уже слева сзади, или же в ячейке локальной карты BL.

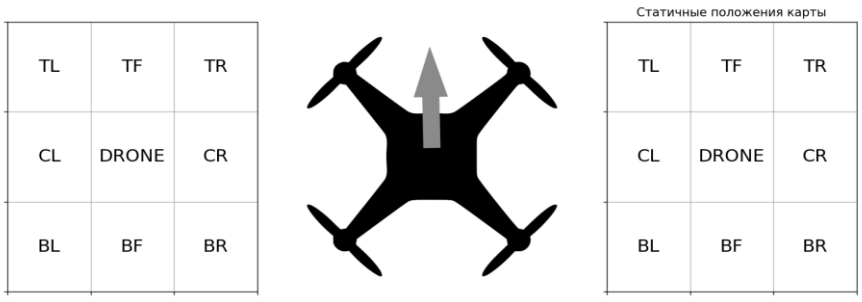



Рисунок 3 – Окружающие ячейки на удалении равном 1

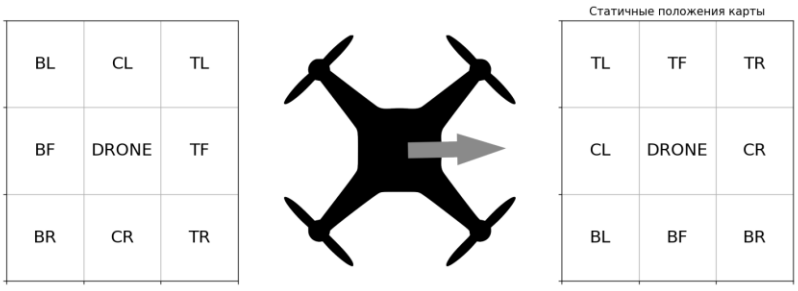



Рисунок 4 – Окружающие ячейки на удалении равном 1 при повороте на 90 градусов вправо

При таком подходе, теоретически, нет необходимости хранить в данных ячейки все возможные вариации положений объекта, так как они могут быть получены простым поворотом матрицы направлений в зависимости от направления взгляда, что позволяет на ходу адаптировать и эффективно использовать метаданные карты. Достаточно взять данные о наличии объекта из соответствующей ячейки глобальной карты, а текущие коды положения – из локального набора ячеек.

Это может быть использовано не только для адаптации данных пространственного расположения окружающих объектов, записанных в глобальной карте, но и о наличии и состоянии сигнала до момента возникновения проблем или полной его потери. Например, можно отслеживать последние *k* ячеек до исчезновения сигнала, чтобы мочь оценить границы зоны поражения средствами РЭБ или перехвата. Информацию об этом можно кодировать простыми числовыми кодами (см.рис.5), которые могут быть использованы в качестве ключей словаря, где значениями могут быть различные запрограммированные маршруты, действия и т.п. На рисунке 5 ячейки с отсутствием сигнала закрашены темным цветом.

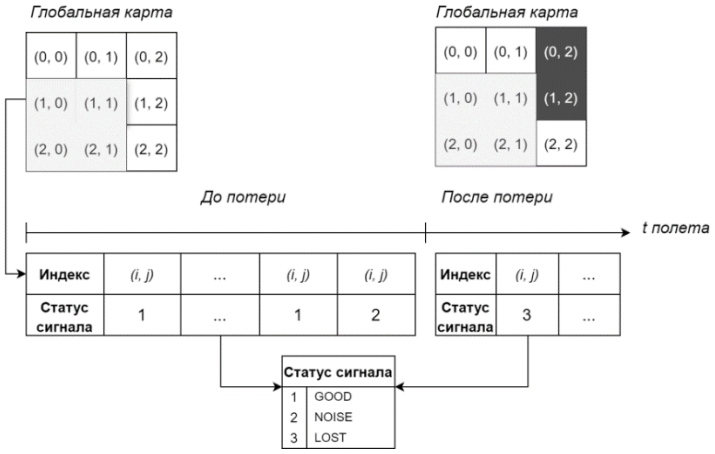



Рисунок 5 – Схема работы кодирования статуса сигнала в каждой ячейке

Данных об индексах последней ячейки, где статус связи управления или GPS был хорошим также теоретически можно использовать для построения пути автономного возврата домой или как минимум для возврата в зону, где при возобновлении сигнала дрон может снова быть направляем оператором. В автономном режиме эти данные могут быть использованы нейронной сетью для принятия решений, если предусмотрены такие функции.

В общем виде, каждая ячейка карты хранит в себе следующую структуру:

1. Внешние данные:
    a. $i_{global}$, $j_{global}$ – индекс ячейки глобальной (загруженной) карты;
    b. *GPS*-координаты*;*
    c. *h* – высота съемки*;*
    d. *angle* – угол наклона камеры;
    e. *description* – текстовое описание находящихся объектов;
2. Внутренние данные в виде массива ячеек вокруг текущей вида: $\{(i_{global}, j_{global}, d, s)_1, \dots (\dots)_n\}$, где *d, s* – коды направления и состояния сигнала соответственно.

### *Выводы*

В данной работе рассмотрена проблема автономной навигации беспилотных летательных аппаратов в условиях потери сигнала управления или GPS. Приведен краткий обзор существующего подхода решения задачи. Из приведенных работ указано на проблемы, связанные с использованием исключительно спутниковых и наземных снимков, такие как случайные перекрытия и недостаточную устойчивость к изменениям точек обзора. Рассмотрены предлагаемые решения в виде использования БПЛА в качестве источника многоракурсных снимков, а также контрастивное обучение модели для более точного сопоставления мультимодальных данных, таких как изображения разного вида ракурсов, текст, гео-метки.

Предложена методика хранения карты на летательном устройстве и структура метаданных для ее сегментов, основанная на использовании статической карты с метаданными о расположении объектов, и динамической локальной с данными о текущем относительном положении. Предложена структура хранимой карты и метаданных ее сегментов для адаптации используемых данных об окружении без необходимости хранения их вариантов для каждого направления. Это оставляет необходимость подобного увеличения числа данных только для набора обучающего данных, что предполагает использование техник аугментации.

Приведены примеры использования данных в такой структуре в ситуациях наличия сигнала и его полной потери. Метаданные ячеек могут хранить как статичные, так и динамически изменяющиеся данные, например, о наличии и состоянии сигнала. В перспективе рассмотреть размерность хранимых ячеек.

### *Литература*


1. Топ-4 дрона для пожаротушения от Hongfei и Ehang — RCMonsteRU [Электронный ресурс] / RCMonsteRU. – Режим доступа: https://rcmonste.ru/blog/article/vysokie-tehnologii-protiv-ognya-top-4-drona-dlya-pozharotusheniya-ot-hongfei-i-ehang. - Загл. с экрана.
2. Дроны [Электронный ресурс] / topwar.ru. – Режим доступа: https://topwar.ru/armament/drones/
3. PHD VI: как у нас угнали дрона [Электронный ресурс] / Securitylab. – Режим доступа: https://www.securitylab.ru/analytics/482548.php. - Загл. с экрана.
4. Сложно ли угнать коптер? Несколько уже реализованных способов перехвата управления / Хабр [Электронный ресурс] / Habr. – Режим доступа: https://habr.com/ru/articles/398603/
5. University-1652: A Multi-view Multi-source Benchmark for Drone-based Geo-localization [Электронный ресурс] / ResearchGate. – Режим доступа: https://www.researchgate.net/publication/372858597_Orientation-Guided_Contrastive_Learning_for_UAV-View_Geo-Localisation. - Загл. с экрана.
6. Deuser, F., Habel, K., Oswald, N., Werner, M. Orientation-Guided Contrastive Learning for UAV-View Geo-Localisation [Электронный ресурс] / arXiv. — Режим доступа: https://arxiv.org/pdf/2308.00982. - Загл. с экрана.
7. He, K., Girshick, R., Dollár, P. Rethinking ImageNet Pre-training [Электронный ресурс] / arXiv. — Режим доступа: https://arxiv.org/pdf/1812.03506. - Загл. с экрана.



***Павленко Б.В. Структура метаданных сегментов карт навигации дрона.*** *В статье представлен анализ структуры метаданных для сегментов карт, используемых в задачах геолокации беспилотных летательных аппаратов (БПЛА). Рассматривается подход, в котором визуальные данные и текстовые описания интегрируются для повышения точности навигации в условиях частичной или полной потери сигнала. Основой решения является разделение карты на ячейки с фиксированными метаданными, включая GPS-координаты, высоту съемки, угол наклона камеры, а также динамический пересчет локальной ориентации объектов на основе глобальной карты. Это позволяет эффективно адаптировать метаданные и уменьшить избыточность хранимых данных. Предложенная структура открывает перспективы для более надежной навигации дронов в сложных условиях эксплуатации.*



***Ключевые слова:*** *БПЛА, геолокализация, карты, распознавание изображений, метаданные.*



***Pavlenko Bohdan. The metadata structure of the segments of the drone navigation maps.*** *The article analyzes the metadata structure for map segments used in UAV-based geolocation tasks. It explores an approach that integrates visual data with textual descriptions to enhance navigation accuracy in scenarios involving partial or complete signal loss. The solution is based on dividing the map into cells with fixed metadata, including GPS coordinates, shooting altitude, and camera tilt angle, along with dynamic recalculation of objects' local orientation based on the global map. This enables efficient metadata adaptation while reducing storage redundancy. The proposed structure offers promising opportunities for more reliable UAV navigation in challenging operational environments.*



***Key words:*** *UAV, geolocation, maps, image recognition, metadata.*

УДК 004.4

# Машинное обучение в задаче компьютерного моделирования параметрического анализа когнитивных динамических систем

**И.В. Чернядьев**[*1]
[*1] инженер – исследователь, Институт проблем искусственного интеллекта, г. Донецк,
магистрант, Донецкий государственный университет
chernyadev-i@mail.ru

***Чернядьев И.В., Машинное обучение в задаче компьютерного моделирования параметрического анализа когнитивных динамических систем.*** *В данной работе разработан модуль анализа онтологий текста для автоматического распознавания эмоционального состояния сотрудников предприятия. Целью исследования является создание инструмента для диагностики настроений внутри коллектива, что актуально для психологии и смежных областей. Использована современная архитектура BERT, а также собран объемный набор данных для настройки параметров модели. Таксономии текста делятся на три категории: нейтральный, позитивный, негативный. Модель способна улавливать контекст в предложениях и на его основе выдавать правильные ответы, что говорит о ее высокой точности. Перспективные направления исследований включают интеграцию в системы поддержки принятия решений, использование в медицине и социологии.*

***Ключевые слова:*** *нейронная сеть, распознавание эмоций, обработка естественного языка, глубокое обучение, искусственный интеллект.*

## *Введение*

Стремительный прогресс наукоемких прорывных технологий сопровождается разрушительными стрессогенными факторами. В этой связи особую актуальность представляют фундаментальные научные исследования, которые ведутся в ФГБНУ «ИПИИ» г. Донецка (директор Иванова С.Б.).

Применение нейронных сетей для определения психоэмоционального состояния может оказать значительное воздействие на множество областей, таких как социология и медицина [1]. Для того, чтобы автоматизировать анализ сообщений от пользователей системы, был реализован модуль, задачей которого является обработка естественного языка, а конкретно – учет настроений коллектива внутри предприятия. Тематика сообщений включает в себя три категории: 1) мнение о коллективе: модуль может эффективно выявлять эмоциональную окраску текстов, связанных с межличностными отношениями; 2) замечания и предложения: модуль может определять уровень удовлетворенности сотрудников и ключевые аспекты сообщений; 3) мнение о музыке: у модуля нет ограничений на тематическую область текста, что позволяет применять её для любого содержания, включая искусство.

Все категории подразумевают под собой задачи классификации эмоционального окраса текста. В данной работе была использована одна из самых эффективных и популярных в мире языковых моделей в задачах обработки текста – BERT, а конкретно - rubert-tiny. RuBERT — это адаптация оригинального BERT для русского языка. Она была обучена на русскоязычных текстах, что делает её более подходящей для анализа сообщений на русском языке, где важен учёт грамматики, морфологии и синтаксиса. RuBERT превосходит универсальные модели, такие как английская версия BERT, в задачах, связанных с обработкой русского языка.

## *Анализ современного состояния исследования*

В статьях [1] и [2] исследуется применение нейросетевых технологий для управления детекцией различных факторов в эмоциональных системах ИИ. Авторы предлагают методологию, которая позволяет эффективно идентифицировать и анализировать как экзогенные, так и эндогенные факторы, влияющие на адаптивные системы.

Статья [3] представляет собой фундаментальную работу в области глубокого обучения и обработки естественного языка, в которой представлена модель Transformer. Эта модель значительно изменяет способ передачи последовательностей, полностью полагаясь на механизмы self-attention, устраняя необходимость в рекуррентных или сверточных слоях, которые были в предыдущих архитектурах, таких как RNN и CNN.

В публикации [4] представлен всесторонний обзор эволюции и текущего состояния предварительно обученных языковых моделей на основе Transformer в области NLP. Он служит ценным справочником для понимания основополагающих концепций, методологий и будущих направлений в этой области искусственного интеллекта.

В работе [5] приводится подробное исследование кросс-энтропийных и связанных с ними функций потерь. Эта категория включает в себя несколько функций потерь, обычно используемых в машинном обучении, таких как логистические потери, обобщенная перекрестная энтропия и средняя абсолютная ошибка.

### *Основное содержание*

В реализации этого проекта использовалась облегченные версия rubert-tiny, которая предлагает архитектуру модели с существенным уменьшением числа параметров. Это не единственная подходящая модель, однако, ее выбор обоснован следующими факторами: 1) вычислительная эффективность: у модели rubert-tiny меньше параметров (12M), что снижает требования к вычислительным ресурсам. С более крупными моделями, например, rubert-base-cased, в которой уже 180M параметров, работать нет возможности, поскольку ресурсов машины не хватает даже на то, чтобы загрузить модель на устройство для дальнейшей работы; с учётом оборудования, такая модель позволяет выполнять обучение и инференс (анализ текстов) быстрее и без перегрузки памяти GPU; 2) Достаточная точность: несмотря на упрощённую архитектуру, rubert-tiny сохраняет конкурентную точность для задач классификации и анализа эмоциональной окраски текста (позитивный, негативный, нейтральный) rubert-tiny показывает хорошие результаты (90% точности на валидационной выборке).

Почему мы вообще используем предобученную модель? Дело в том, что обучение такой модели, как BERT, с нуля потребовало бы сбор огромного корпуса текстов, содержащих миллиарды предложений, подготовки данных и значительных вычислительных ресурсов. Это трудоёмкий и дорогостоящий процесс. Поэтому для нашей задачи, как и для многих других, целесообразно использовать уже заранее обученную модель.

Хотя предварительно обученная модель BERT мощна, это обобщенный инструмент. Он понимает язык, но не приспособлен для какой-либо конкретной задачи. Тонкая настройка, по сути, является актом адаптации этого обобщенного инструмента для специализированной работы. В результате требуется гораздо меньше времени для обучения тонко настроенной модели — как будто мы уже обучили нижние слои нашей сети и нам нужно только аккуратно настроить их, используя их вывод в качестве признаков для нашей задачи классификации. Кроме того, из-за предварительно обученных весов этот метод позволяет нам точно настроить нашу задачу на гораздо меньшем наборе данных, чем это потребовалось бы в модели, созданной с нуля.

Описание задачи обработки текста:

Пусть у нас есть текстовый вход $X = \{x_1, x_2, \dots, x_n\}$, где $x_i$ – это токены (слова или их части). Цель модели BERT — вычислить контекстные представления токенов $H = \{h_1, h_2, \dots, h_n\}$, где каждый вектор $h_i$ содержит информацию о токене $x_i$ в контексте всего текста X.

Описание работы модели BERT в модуле можно поделить на несколько этапов:

**1. Входные данные и их преобразование**

Модель BERT принимает на вход последовательности токенов: $X = \{x_1, x_2, \dots, x_n\}$ — входная последовательность длиной n, где $x_i$ — это токены; каждому $x_i$ сопоставляется векторное представление через эмбеддинги:

$$E = \{e_1, \dots, e_n\},$$

где $e_i \epsilon R^d$, d – размерность эмбеддингов.

Эмбеддинг для каждого токена $e_i$ состоит из: Embedding - Вектор, соответствующий конкретному токену; Positional Embedding - вектор, который добавляет информацию о позиции токена в последовательности; Segment Embedding - вектор, указывающий, к какому предложению относится токен [3]. В итоге получим представление:

$$e_i = \text{Embedding}(x_i) + \text{PositionalEmbedding}(i) + \text{Segment Embedding}(s_i)$$

**2. Механизм внимания (Self-Attention)**

Ключевая идея трансформеров, на основе которых был создан BERT — механизм self-attention, который позволяет каждому токену взаимодействовать с другими токенами в последовательности. Каждый токен обращается ко всем остальным токенам, чтобы взвешивать их значимость.

Для каждого эмбеддинга $e_i$ вычисляются три вектора: Query (запрос): $q_i = W_Q e_i$; Key (ключ): $k_i = W_K e_i$; Value (значение): $v_i = W_V e_i$, где $W_Q, Q_K, W_V \in R^{d \times d_k}$ — обучаемые параметры, а $d_k$ — размерность пространств запросов, ключей и значений. Векторы запросов, ключей и значений объединяются в соответствующие матрицы Q, K, V. Коэффициенты внимания определяются следующим образом:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

где: $QK^T$ – скалярное произведение векторов запросов и ключей для оценки степени их взаимосвязи; деление на $\sqrt{d_k}$ стабилизирует градиенты при обучении; softmax гарантирует, что сумма всех получившихся весов для каждого эмбеддинга равна 1, что позволяет рассматривать их как распределение вероятностей; у**множение на** V применяет вероятности для взвешивания информации, которую нужно передать.

Для улучшения представлений используется несколько "голов" внимания:

$$\mathrm{Multihead}(Q, K, V) = \mathrm{Concat}(\mathrm{head1}, \mathrm{head2}, \ldots, \mathrm{head_h})W_O,$$

где: $\mathrm{head_i} = \mathrm{Attention}(Q_i, K_i, V_i)$; $W_O \in R^{hd_k \times d}$ – матрица с обучаемыми параметрами. Она гарантирует, что вектор после прохода через Multihead Attention (MHA) вернется в свою исходную размерность [3].

Многоголовое внимание позволяет модели фокусироваться на разных аспектах информации в последовательности.

**3. Нормализация и остаточные связи**

После слоя внимания применяются:

1. Skip-connection (остаточные связи). Векторы, полученные после прохода через слой MHA, складываются с векторами до прохода через этот слой. Выглядит это вот так:

$$h_i = z_i + e_i,$$

где $z_i$ – выходной вектор после MHA, $e_i$ – исходное представление.

С остаточными соединениями можно строить более глубокие сети, не теряя при этом способность к обучению. Это помогает избежать проблемы исчезающих градиентов [3].

2. Нормализация по слоям (Layer Normalization). Вот так реализована нормализация по слоям в pytorch:

$$\mathrm{LayerNorm}(x) = \frac{x - \mu}{\sigma} * \gamma + \beta$$

где $\mu$ и $\sigma$ — среднее и стандартное отклонение по входу, а $\gamma$ и $\beta$ — обучаемые параметры [3].

**4. Слой Feed-Forward Network (FFN)**

FFN представляет собой комбинацию двух линейных слоев с функцией активации ReLU между ними [3].

$$\mathrm{FFN}(x) = W2 * \mathrm{ReLU}(x * W1 + b1) + b2,$$

где W1, W2, b1, b2 – обучаемые параметры; W1, W2 – это матрицы весов двух линейных слоев нейронной сети. Они определяют, как входные данные трансформируются и передаются между слоями сети. Первый слой W1 расширяет размерность представления (в среднем в 4 раза), позволяя модели учитывать больше признаков и извлекать более сложные закономерности. Слой W2 затем уменьшает размерность, возвращая вектор к требуемому размеру; b1, b2 – это смещения (bias) двух последовательных слоев. Они позволяют сдвигать активацию нейронов, чтобы они лучше соответствовали входным данным.

**5. Обучение BERT**

BERT анализирует каждый токен с учётом всего контекста: слева-направо и справа-налево одновременно. Это достигается за счёт обработки всей последовательности токенов в каждой итерации attention. Модель обучается с использованием двух задач:

1. Masked Language Model (MLM): Некоторый процент токенов (например, 15%) заменяется на [MASK]; Модель должна предсказать исходный токен $x_i$ на основе контекста [4].

$$\mathcal{L}_{\mathrm{MLM}}^{(x)} = -\frac{1}{|M_x|} \sum_{i \in M_x} \log P(x_i | x_{\setminus M_x})$$

где $x_{\setminus M_x}$ представляет замаскированную версию x, а $M_x$ представляет набор замаскированных позиций токенов в x.

2. Next Sentence Prediction (NSP): Модель обучается предсказывать, следуют ли два предложения друг за другом [4]. Целевая функция:

$$\mathcal{L}_{\mathrm{NSP}\ \mathrm{NSP}}^{(x,y)} = -\log P(d/x, y),$$

где $d \in \{1, 0\}$ представляет, являются ли предложения последовательными или нет.

В итоге получаем общий алгоритм обработки текста языковой моделью BERT: Вход: Последовательность токенов x; Слои: обработка последовательности через L слоёв трансформера (механизм self-attention, остаточные соединения, нормализация, FFN); Выход: Контекстуальные представления для каждого токена $h_i$.

Для задач классификации (как в нашем случае) используется токен [CLS], который представляет собой агрегированное представление всей последовательности. Выходной вектор токена [CLS] подаётся в полносвязный слой:

$$P(y|X) = \mathrm{softmax}(h_{[CLS]} W_c + b_c)$$

$P(y|X)$ – это условная вероятность выходного класса y, X – данные входной последовательности. Вычисление этой вероятности происходит с помощью функции softmax; $h_{[CLS]}$ – это скрытое состояние специального токена [CLS], который используется для представления всей последовательности; $W_c$ и $b_c$ – это обучаемые веса и смещения линейного слоя, который проецирует представление [CLS] в вектор вероятностей классов [5].

Модель должна научиться предсказывать правильные вероятности классов на основе обобщенного представления всей входной последовательности, кодируемого в токене [CLS]. Обучение направлено на максимизацию вероятности выбора правильного класса.

Для реализации данной задачи применялись следующие инструменты: фреймворк машинного обучения для языка Python — PyTorch и ряд Python-библиотек. Предобученная модель и датасеты взяты на платформе **Hugging Face**.

Для того, чтобы модель выдавала корректный результат, необходимо ее обучить на правильно размеченном наборе данных. Русскоязычные датасеты, которые лежат в открытом доступе, обладают существенными недостатками: недостаточное разнообразие, несбалансированность категорий. Для того, чтобы решить эту проблему, были выбраны наиболее подходящие датасеты и приведены к общему виду: текст, категория (0 – нейтральный окрас, 1 – положительный, 2 – негативный). Далее данные были собраны в один большой набор и затем еще прошли балансировку, чтобы всех категорий в датасете было поровну. В итоге получился набор, содержащий в себе примерно 240 000 записей.

После того, как были сформированы данные, на которых будет обучаться модель, можно приступать к реализации самого обучения. Первым делом идет загрузка и разделение данных из датасета на обучающую (80%) и тестовую (20%) выборки. После этого шага добавляем токенизацию текста при помощи BertTokenizer. BertTokenizer – это токенизатор из библиотеки Hugging Face, который загружается для модели rubert-tiny. Токенизатор преобразует текст в числовые представления, которые можно подавать на вход модели. Крайне важно использование совместимых токенизатора и модели. Далее загружается модель rubert-tiny для задачи классификации текста. Модель переносится на устройство cuda (если доступна видеокарта с поддержкой CUDA) или на cpu. Для подачи данных в модель используются объекты DataLoader из PyTorch. Они создаются на основе обучающего и тестового датасетов. DataLoader упрощает работу с данными, автоматически обрабатывая батчи и перемешивая данные. Обучение модели (рис. 1) проходит в цикле из несколько эпох. В каждой эпохе:

- Модель переводится в режим обучения;
- Для каждого батча:
    - Обнуляются градиенты оптимизатора;
    - Данные перемещаются на устройство;
    - Получаем выход модели (outputs), результат содержит предсказания модели и значение функции потерь;
    - Вычисляются градиенты методом обратного распространения ошибки;
    - После вызова loss.backward() оптимизатор использует градиенты для корректировки параметров модели в направлении уменьшения ошибки;
    - Накапливается значение потерь для мониторинга процесса обучения;
- После завершения эпохи отображается среднее значение потерь по всем батчам;

```
for epoch in range(num_epochs):
    model.train()
    running_loss = 0.0
    for batch in tqdm(train_loader):
        optimizer.zero_grad()
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        labels = batch['labels'].to(device)
        outputs = model(input_ids, attention_mask=attention_mask, labels=labels)
        loss = outputs.loss
        loss.backward()
        optimizer.step()
        running_loss += loss.item()
    print(f'Epoch {epoch + 1} loss: {running_loss / len(train_loader)}')
```

Рисунок 1 – обучение модели

В качестве оптимизатора был выбран Adam. Он автоматически подстраивает шаг обучения для каждого параметра на основе величины градиента и накопленной информации о моменте. Это важно при работе с глубокими нейросетями, где распределение градиентов практически всегда является неравномерным. Для скорости обучения выбрано число 2e-5. Такое низкое значение обусловлено тем, что BERT уже предобучен на огромных объемах данных. Во время дообучения на специфической задаче цель — слегка подстроить параметры, сохраняя основную структуру. Слишком большой шаг может привести к нестабильности обучения, а слишком маленький — к замедлению процесса.

Размер батча определяет, сколько примеров обрабатывается одновременно. Выбор размера 16 обусловлен компромиссом между эффективностью обучения и ограничением видеопамяти устройства.

Используется **кросс-энтропийная функция потерь,** подходящая для многоклассовой классификации. Она измеряет насколько хорошо модель предсказывает правильные значения на основе входных данных.

Пример графического представления изменяющихся в ходе обучения показателей представлен на рис. 2. На первом графике отображается изменение значения функции потерь в процессе обучения по эпохам. Падение потерь указывает на то, что модель становится более точной в предсказаниях на обучающих данных [6]. Второй график показывает изменение точности модели на тестовых данных по эпохам.

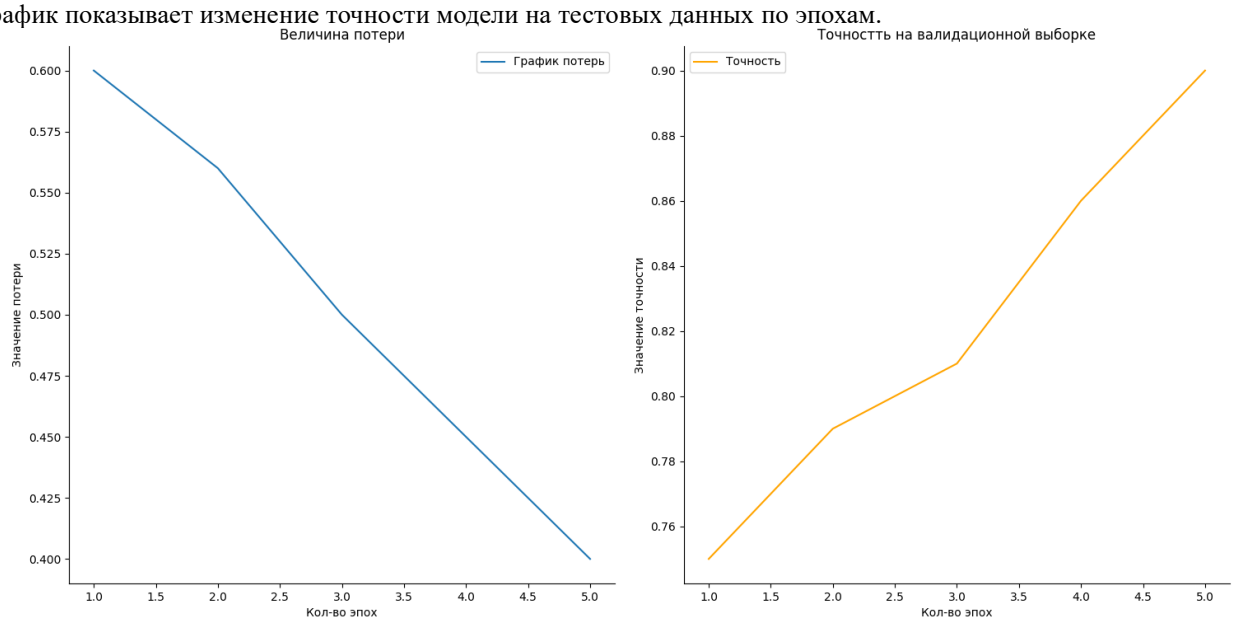



Рисунок 2 - Графики функции потерь и изменения точности

На графике матрицы путаницы (рис. 3) представлены результаты предсказаний модели по всем классам эмоций. Матрица путаницы — это инструмент измерения производительности для решения задач классификации машинного обучения. Она представляет собой таблицу, которая позволяет визуализировать производительность алгоритма путём сравнения фактических и прогнозируемых результатов. Это позволяет увидеть, как часто модель ошибается, классифицируя эмоции, и в каких случаях. Диагональ матрицы отражает случаи, когда предсказанный и реальный класс совпадают [6].

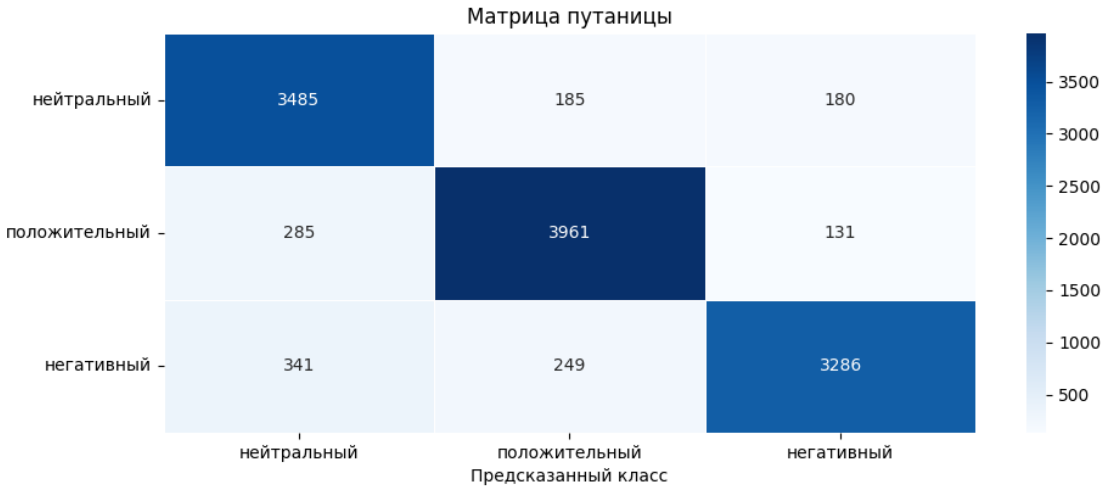



Рисунок 3 – Матрица путаницы

Достижение такой высокой точности за малое количество эпох обучения подтверждает, что модель успешно справляется с задачей извлечения значимых особенностей из текста. Эти результаты подчеркивают потенциал использования глубокого обучения для автоматизации анализа эмоциональных состояний.

### *Выводы*

В рамках проекта была реализована система автоматического анализа эмоционального окраса текста, основанная на использовании предобученной языковой модели rubert-tiny. Использование облегчённой версии модели позволило решить задачи классификации текста с хорошей точностью, при этом сохранив вычислительную эффективность. Благодаря меньшему количеству параметров модель успешно работала на оборудовании с ограниченными ресурсами, что позволило избежать проблем с перегрузкой GPU. Для обучения модели был создан сбалансированный датасет, включающий три категории эмоциональной окраски (нейтральный, положительный, негативный). Это повысило качество обучения и снизило вероятность смещения модели в сторону одного из классов. Внедрение разработанного модуля открывает возможности для улучшения взаимодействия пользователей с системой, а также для мониторинга и анализа психоэмоционального состояния сотрудников. В дальнейшем можно рассмотреть расширение функционала, использование более мощных моделей и углублённое исследование качества обработки текстов на нестандартных данных.

### *Литература*


1. Бондарчук, В. В. Нейросетевая методология управления процессом детекции экзогенных и эндогенных факторов адаптивной системы эмоционального искусственного интеллекта [Текст] / В. В. Бондарчук, Н. М. Кравченко // Международный электронный научный журнал «Евразийский Союз Ученых. Серия: технические и физико-математические науки». – № 11(114)/2023 Том 1 – С. 14–23. – ISSN: 2413-9335 (электронный вариант): 2782-246X. – DOI: 10.31618/ESU.2413-9335.2023.1.114.1.1917 (Дата публикации: 03.01.2024 г.).
2. Bondarchuk, V. V. Indirect criteria algorithm for the control of a cognitive neural network [Text] / V. V. Bondarchuk, N. M. Kravchenko, Ya. S. Pikalyov // Proceeding of the International University Scientific Forum «Practice Oriented Science: UAE – Russia – India» (07.01.2024). – P. 92–102. – ISBN 978-5-905695-87-2. - DOI 10.34660/INF.2024.32.41.030.
3. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin: Attention Is All You Need, 2017, pp. 2-13.
4. Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha: A Survey of Transformer-based Pretrained Models in Natural Language Processing, 2021, pp. 9-15.
5. Anqi Mao, Mehryar Mohri, Yutao Zhong: Cross-Entropy Loss Functions: Theoretical Analysis and Applications, 2023, pp. 7-8.
6. Чернядьев И.В., Бондарчук В.В., Системный анализ программного обеспечения концептуальной нейросетевой модели системы классификации эмоций на изображениях [Текст] // Евразийский Союз Ученых. Серия: технические и физико-математические науки. – 2024. – с. 6-14.



***Чернядьев И.В., Машинное обучение в задаче компьютерного моделирования параметрического анализа когнитивных динамических систем.*** *В данной работе разработан модуль по анализу онтологий текста для автоматического распознавания эмоционального состояния сотрудников предприятия. Целью исследования является создание инструмента для диагностики настроений внутри коллектива, что актуально для психологии и смежных областей. Использована современная архитектура BERT, а также собран объемный набор данных для настройки параметров модели. Таксономии текста делятся на три категории: нейтральный, позитивный, негативный. Модель способна улавливать контекст в предложениях и на его основе выдавать правильные ответы, что говорит о ее высокой точности. Перспективные направления исследований включают интеграцию в системы поддержки принятия решений, использование в медицине и социологии*.



***Ключевые слова:*** *нейронная сеть, распознавание эмоций, обработка естественного языка, глубокое обучение, искусственный интеллект.*



***Chernyadev I.V., Machine learning in the problem of computer modeling of parametric analysis of cognitive dynamic systems.*** *In this paper, a text ontology analysis module has been developed for automatic recognition of the emotional state of employees of the enterprise. The aim of the study is to create a tool for diagnosing moods within the team, which is relevant for psychology and related fields. The modern BERT architecture was used, and a voluminous data set was collected to configure the model parameters. The taxonomies of the text are divided into three categories: neutral, positive, and negative. The model is able to capture the context in sentences and give correct answers based on it, which indicates its high accuracy. Promising areas of research include integration into decision support systems, use in medicine and sociology.*



***Keywords****: neural network, emotion recognition, NLP, deep learning, artificial intelligence.*

УДК 519.178

# Муравьиный алгоритм кликого покрытия

**Хуссейн Ф.А.**
Аспирант, Южный Федеральный Университет,
firas94mecha@gmail.com

***Хуссейн Ф.А., Муравьиный алгоритм кликого покрытия.*** *В данной статье рассматривается проблема поиска субоптимального разбиения клик графа с использованием Муравьиного алгоритма. Предлагается нижняя граница для числа разбиений клик с учетом геометрических характеристик задачи. Экспериментальные результаты показывают, что метод дает оптимальные решения для конкретных конфигураций док-станций (с 10 вертикальными и 1–3 горизонтальными рядами) и субоптимальные решения для более общих случаев, все в пределах относительно короткого времени вычислений. Для решетчатого графа алгоритм выдал решение с 8 кликами менее чем за 7 секунд.*

***Ключевые слова:*** *Муравьиный алгоритм, Задача о покрытии кликой, Разделение клики, Теория графов*

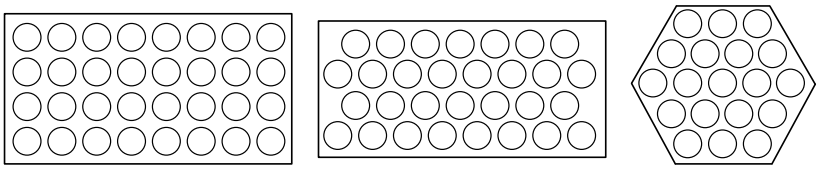
## *Введение*

Автономные необитаемые подводные аппараты (АНПА) всё чаще применяются в различных областях благодаря своей способности выполнять задачи, которые могут быть опасными или труднодоступными для человека. Групповое использование АНПА представляет собой актуальное направление в развитии технологий, позволяющее повысить эффективность и безопасность выполнения задач. Групповое применение АНПА позволяет координировать действия нескольких аппаратов для выполнения сложных задач, таких как обследование больших морских территорий или выполнение совместных операций по обнаружению и нейтрализации угроз. В групповой конфигурации АНПА могут выполнять разные функции, что позволяет увеличить общую эффективность. Например, один аппарат может заниматься сбором данных, другой — проведением анализа, а третий — передачей информации, увеличивая при этом вероятность успешного завершения операции, так как в случае поломки одного из аппаратов другие могут продолжить выполнение задачи.

Разработка и внедрение многоагентных систем управления (группа АНПА является многоагентной системой, где каждая единица АНПА определяется как агент) становятся всё более сложными и интеллектуальными. Непосредственному использованию группы АНПА, как правило, предшествует этап формирования строя из подводных аппаратов. Формирование строя – это подготовительный этап, в течение которого группа АНПА еще не может выполнять групповую миссию. Он является обязательным, но затратным по времени, поэтому основное требование к этому этапу – он должен быть осуществлен по возможности за кратчайшее время.

Для удобства транспортировки и последующего погружения в воду все АНПА закрепляют в некоторой конструкции, называемой доковой станцией (Рис. 1). В доковой станции все аппараты находятся в отсеках, каждый из которых имеет форму прямого кругового цилиндра. Отсеки параллельны друг другу и образуют, как правило, некоторую регулярную структуру. В наиболее простом случае это может быть структура решетки, заполняющей прямоугольную доковую станцию (Рис. 1, а)). В более сложных случаях –решетки с более плотной упаковкой (Рис. 1, б)) или регулярные структуры в доковой станции, имеющей круглое или иное сечение (Рис. 1, в)).

* Исследование выполнено при поддержке гранта Российского научного фонда № 24-19-00063 «Теоретические основы и методы группового управления безэкипажными подводными аппаратами», https://rscf.ru/project/24-19-00063 / на базе Федерального государственного образовательного учреждения высшего образования «Южный федеральный университет».



а) б) в)

Рис. 1. Конструкции доковых станций.

Для определенности будем в дальнейшем считать, что отсеки образуют прямоугольную решетку (Рис. 1, а)). С целью удобства транспортировки, хранения и перемещений доковую станцию следует делать как можно меньше, а, следовательно, отсеки для АНПА располагать как можно ближе.

Формирование строя АНПА начинается после погружения доковой станции с аппаратами на заданную глубину, которая во многом определяется групповой задачей и особенностями акватории, где будет формироваться строй. При этом выход подводных аппаратов из доковой станции для формирования строя не может быть произвольным из-за особенностей их первоначального движения в водной среде. Это связано с тем, что при выходе из доковой станции подводные аппараты находятся в зоне неустойчивого движения, при котором могут наблюдаться непредсказуемые процессы рыскания, тангажа и крена. В результате этого аппараты, находящиеся близко друг к другу, могут столкнуться и получить повреждения. Зона безопасного начального положения АНПА при выходе из доковой станции, как правило, значительно превышает расстояние между соседними отсеками в доковой станции. Поэтому близко расположенные в доковой станции аппараты не могут осуществлять выход из станции одновременно из-за требуемой безопасности на начальном этапе движения.

С другой стороны, последовательный выход аппаратов из доковой станции с временным интервалом «безопасности» (10 с. и более), учитывая большое количество аппаратов, приводит к неприемлемому затягиванию процесса формирования строя. Компромисс между безопасностью и длительностью выхода аппаратов из станции находится путем формирования подгрупп АНПА, расположенных достаточно далеко друг от друга внутри доковой станции; аппараты, входящие в одну группу, могут покидать станцию одновременно. Формирование таких групп гарантирует безопасность первоначального движения аппаратов, а количество таких групп определяет общее время, которое требуется для выхода всех АНПА из доковой станции. Чем меньше групп, тем быстрее происходит формирование строя. Определение минимального числа таких «безопасных» групп является основной задачей управления на данном предварительном этапе перед началом формирования строя.

### *Формальная постановка задачи*

Группа агентов ( $mn$ ) в начальный момент времени находится в доковой станции. Агенты в определенном порядке выпускаются из доковой станции и должны достичь заданные позиции. Пусть отсеки доковой станции образуют равномерную прямоугольную решетку с $m$ строками и $n$ столбцами. Под расстоянием между двумя соседними по горизонтали и вертикали цилиндрическими отсеками будем понимать расстояние между их осями. Будем считать, что расстояние между соседними строками и соседними столбцами одинаково и равно $d$. Если $r$ – радиус максимального перпендикулярного к продольной оси сечения АНПА, то, очевидно, должно выполняться условие $d > 2r$.

С учётом графа $G = (X, U)$, в котором $X$ – множество вершин, $U$ – множество рёбер между ними, клика $C$ определяется как подмножество вершин $C \subseteq X$, что каждые две различные вершины смежны. Кликовое покрытие или разбиение на клики неориентированного графа — это разбиение вершин на клики, подмножества вершин, внутри которых каждые две вершины смежны.

Таким образом, любая подгруппа АНПА, которой в графе $G$ соответствует *полный подграф*, может быть выведена из доковой станции одновременно, поскольку между любой парой аппаратов из этой подгруппы расстояние изначально будет не меньше, чем $s$, т.е. выполняются условия безопасного выхода из станции. Соответственно, покрытие множества вершин кликами называется *кликовым покрытием* или, если требуется попарная непересекаемость клик, *кликовым разбиением* [1-2].

Во многих случаях полезно знать нижние и верхние оценки минимального размера кликового покрытия, что позволяет соответственно получить оценки времени, необходимого для выведения всех АНПА из доковой станции. Наименьшее количество клик графа, покрывающих множество его вершин, называется *числом кликового покрытия* [2]. Известные оценки снизу для этой характеристики получаются на основании оценок *числа внутренней устойчивости* (максимальное количество вершин в графе $G$, попарно не смежных между собой) для графов общего вида, и являются в большинстве случаев очень грубыми. Более точные оценки можно

получить, если учесть особенности получаемого в задаче графа, например, как в рассматриваемом случае, учесть регулярность размещения отсеков, соответствующих вершинам $X$ графа $G$.

По условию в одну клику не могут попасть никакие две вершины, если расстояние между соответствующими отсеками в доковой станции меньше, чем $s$. Следовательно, если на доковую станцию наложить круг радиуса $s/2$, то никакие две вершины, соответствующие отсекам внутри этого круга, не могут попасть в одну клику, а значит, количество таких вершин может служить нижней оценкой числа кликового покрытия. Более полезной и простой оказывается нижняя оценка, которая может быть получена на основе теоремы Блихфельда [3]. Ее доказательство основано на идеях геометрической теории чисел: если площадь плоской фигуры, расположенной на целочисленной решетке, больше некоторого целого числа $b$, то эту фигуру можно сдвинуть так, чтобы она покрыла не менее $b+1$ узлов. Важно отметить, что утверждение теоремы Блихфельда формулируется на решетке с единичным шагом, а расположение фигуры никак не связано с узлами решетки. Кроме того, результат этой теоремы распространяется на любые, в том числе невыпуклые фигуры. Применительно к кругу сразу получаем, что число целых точек, покрываемых кругом радиуса $s/2$, будет не менее, чем $\left[\frac{\pi s^2}{4d^2}\right]+1$, где $[\bullet]$ обозначает взятие целой части. Таким образом, это число является нижней оценкой числа кликового покрытия (и числа кликового разбиения) графа $G$.

### *О субоптимальных алгоритмах кликового разбиения*

Известно, что задача о минимальном кликовом покрытии, а, следовательно, и минимальном кликовом разбиении произвольного графа относится к числу NP–полных проблем [4]. Наивный алгоритм, пробующий все возможные комбинации клик способен найти оптимальное решения, но за счет очень больших временных затрат. Поэтому для достаточно большого $|X|$ найти такое покрытие за приемлемое время проблематично. В связи с этим предлагаются и применяются различные алгоритмы для приближенного, но быстрого решения этой задачи [5], которые позволяют за приемлемое время найти решение близкое к оптимальному. Данная статья предлагает новый метод решения на основания мета-эвристики муравьиного алгоритма.

### *Муравьиный алгоритм кликового разбиения*

Муравьиный алгоритм [6] основан на том, что при поиске пищи муравьи демонстрируют сложное совместное поведение, определенное выделяемыми ими феромонами – собирательное название веществ, выделяемых некоторыми видами животных и обеспечивающих коммуникацию между особями одного вида. Феромоны привлекают других муравьев и указывают путь к источнику пищи. Детальное муравьиного алгоритма можно найти в работе [6].

Входным параметром алгоритма является список координат $(x, y)$ всех АНПА в доковой станции, соответствующих вершинам графа $G$, которые должны быть покрыты. Рассчитывается матрица $D = \|d_{ij}\|$ расстояний $d_{ij}$ от каждого узла до остальных узлов; вместе с этим рассчитываются параметры эвристики муравьиного алгоритма в форме матрицы $\|\eta_{ij}\| = \|1/d_{ij}\|$.

Для решения задачи кликового разбиения необходимо ввести функцию стоимости, которая будет минимизировать количество клик, необходимых для покрытия графа. В контексте муравьиного алгоритма эта функция стоимости будет направлять муравьев к нахождению решений, которые требуют минимального количества клик $C_1, C_2, ..., C_z$. Функция стоимости для муравья (решение) k задается следующим образом:

$$C = \left|\left\{C_1^k, C_2^k, ...C_z^k\right\}\right|.$$

Каждый муравей строит решение, пытаясь минимизировать количество клик. В каждой итерации каждый муравей пытается добавить максимальное количество вершин в текущую клику, соблюдая условие клики (все вершины должны быть попарно связаны). В случае невозможности добавления новой вершины в данной клике эта клика сохраняется, и муравей начинает наращивать следующую клику с исключением тех вершин, которые уже добавлены в предыдущих кликах. Муравей совершает поиск, когда не остаётся вершины в списке $X$.

### *Результаты моделирования*

Для исследования эффективности данного алгоритма была написана программа на языке Python.

Проводилось исследование с различными размерами доковой станции. Рассматривались доковые станции с 10 вертикальными рядами и $1 \div 19$ горизонтальными рядами. На рис. 7 показано количество клик в найденном кликовом разбиении, а на рис. 8 – время расчета решения в секундах в зависимости от числа горизонтальных рядов с АНПА в доковой станции.

Данный метод обеспечивает оптимальные решения для определённых размеров доковых станций (10 вертикальных и 1, 2, 3 горизонтальных рядов), а также субоптимальные решения для более общих случаев за относительно короткое время. Для решёточного графа алгоритм муравьиной колонии нашёл решение с 8 кликами, затратив на расчёт не более 7 секунд. Это позволяет заключить, что предложенный метод быстро решает поставленную задачу с хорошим приближением к оптимальному решению и требует дальнейшего изучения. Перспективным направлением исследований может стать влияние гиперпараметров муравьиного алгоритма на качество решения и время его вычисления.

С целью сравнения был реализован наивный алгоритм, генерирующий все возможные комбинации клик [7]. Для доковой станции размером $1 \times 10$ (один горизонтальный ряд и 10 вертикальных рядов) алгоритм смог найти оптимальное решение с числом кликового покрытия 3, однако на это он потратил 16 секунд. Для доковой станции размером $2 \times 10$ наивный алгоритм проработал сутки и не завершил поиск решения.

Также был исследован жадный алгоритм [7], он показал очень высокое быстродействие, но полученные результаты оказались очень далеки от оптимальных. На рис. 2 приведено количество клик в кликовом разбиении, а на рис. 3 – время решения в секундах в зависимости от количества горизонтальных рядов с АНПА в доковой станции.

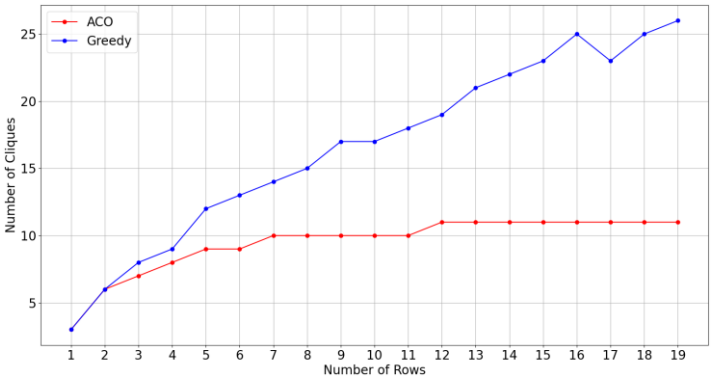



Рис. 2. Количество клик графов с различными количествами узлов.

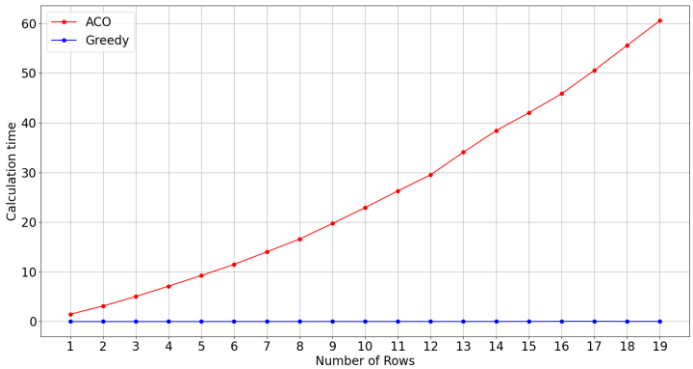



Рис. 3. Время расчета решения для графов с различными количествами узлов.

### *Заключение*

В работе решалась проблема поиска субоптимального кликового разбиения графа, для решения использовался подход на основе муравьиного алгоритма. Приведена нижняя оценка числа кликового разбиения, учитывающая геометрические особенности решаемой задачи. Экспериментально было показано что данный метод даёт оптимальные решения для некоторых размеров доковых станций (10 вертикальных и 1, 2, 3 горизонтальных рядов) и субоптимальные решения для более общих случаев за достаточно короткий промежуток времени. Для решёточного графа $4 \times 10$ муравьиный алгоритм дал решение с 8 кликами, при этом время расчета решения составило не более 7 секунд. Исходя из этого можно сказать, что предложенный метод решает поставленную задачу быстро с хорошим приближением к оптимальному решению и заслуживает дальнейшего

исследования. В качестве объекта исследования может быть влияние гиперпараметров муравьиного алгоритма на качество решения и время расчета.

### *Литература*


1. Левин М.Ш. О разбиении графа на основе клик. Информационные процессы, Т. 22, № 4, 2022, с. 308–318.
2. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. – М.: Наука, 1990, 384 с.
3. Энциклопедия элементарной математики. Книга пятая – Геометрия. – М.: Наука, 1966. – 624 с.
4. Karp R. Proceedings of a Symposium on the Complexity of Computer Computations / R. E. Miller, J. W. Thatcher. – Plenum Press, 1972, p. 85–103.
5. Marcos O.R., Fast constructive and improvement heuristics for edge clique covering / Jens Gramm et al., Data Reduction, Exact, and Heuristic Algorithms for Clique Cover.
6. Dorigo, M., Birattari, M. (2011). Ant Colony Optimization. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_22
7. https://github.com/farjasju/CliqueCover



***Хуссейн Ф.А., Муравьиный алгоритм кликого покрытия.*** *В данной статье рассматривается проблема поиска субоптимального разбиения клик графа с использованием Муравьиного алгоритма. Предлагается нижняя граница для числа разбиений клик с учетом геометрических характеристик задачи. Экспериментальные результаты показывают, что метод дает оптимальные решения для конкретных конфигураций док-станций (с 10 вертикальными и 1–3 горизонтальными рядами) и субоптимальные решения для более общих случаев, все в пределах относительно короткого времени вычислений. Для решетчатого графа алгоритм выдал решение с 8 кликами менее чем за 7 секунд.*



***Ключевые слова:*** *Муравьиный алгоритм, Задача о покрытии кликой, Разделение клики, Теория графов*



***Hussein F.A., Ant algorithm for clique coverage.*** *This paper addresses the problem of finding a suboptimal partition of graph cliques using an ant colony optimization algorithm. A lower bound on the number of partitions of cliques is proposed, taking into account the geometric characteristics of the problem. Experimental results show that the method yields optimal solutions for specific docking station configurations (with 10 vertical and 1–3 horizontal rows) and suboptimal solutions for more general cases, all within a relatively short computation time. For a lattice graph, the algorithm yielded a solution with 8 cliques in less than 7 seconds.*



***Keywords****: Ant colony Optimization, Clique covering problem, Clique partitioning, Graph theory*

УДК 004.855.5

# Определение семантической эквивалентности текстов требований предприятий и рабочих программ дисциплин

**Н.В. Мелещенко[*1], О.И. Федяев[*2]**

[*1] аспирант, Донецкий национальный технический университет,
meleshenko.nikolay@mail.ru

[*2] к.т.н, доцент, Донецкий национальный технический университет,
olegfedyayev@mail.ru, OrcID: 0000-0001-6822-7306, SPIN-код: 7777-3791

***Мелещенко Н.В., Федяев О.И. Определение семантической эквивалентности текстов требований предприятий и рабочих программ дисциплин.*** *Формализован процесс сопоставления текста требований предприятия и рабочей программы дисциплины. Поставленная задача решена путём компьютерной обработки текстов требований предприятий на естественном языке методами машинного обучения. Построен конвейер обработки естественного языка для получения смысловой оценки текста. Исследовано влияние параметров модели на определение схожести текстов по смыслу. Изучено влияния порядка слов в документах на их смысл в реализованной модели. Экспериментальные исследования проведены при помощи разработанной программной реализации. Обосновано применения данного подхода для определения схожести по смыслу текстов требований предприятий и рабочих программ дисциплин.*

***Ключевые слова:*** *кафедра университета, требования предприятий, рабочие программы дисциплин, определение схожести текстов, машинное обучение.*

## *Введение*

В настоящее время одной из важных проблем системы высшего образования в России является несоответствие между компетенциями, получаемыми выпускниками в учебных заведениях, и необходимыми профессиональными умениями для трудоустройства [1,2]. Таким образом возникает необходимость в регулярной и профессионально-ориентированной кооперации учебных заведений и предприятий для решения данной проблемы [3,4].

Как известно, уровень профессиональной подготовки выпускников во многом определяется рабочими программами дисциплин (РПД). Содержание РПД инспектирует учебно-методическая комиссия кафедры на предмет их соответствия государственному стандарту по направлению подготовки. Важной функцией учебно-методической комиссии является своевременная актуализация содержания рабочих программ дисциплин, которая должна способствовать подготовке настоящих профессионалов, востребованных на рынке труда. Рабочие программы при актуализации должны периодически обновляться в соответствии с требованиями современной цифровой экономики и согласовываться с работодателями. Учебно-методическая комиссия кафедры анализирует требования работодателей и даёт рекомендации соответствующим лекторам на корректировку рабочих программ, которых они затрагивают. Участники этого процесса (предприятия, методическая комиссия кафедры, лекторы) образуют распределённую систему, для которых характерна территориальная удалённость, автономность и функциональная неоднородность (см. рис. 1). Взаимодействие участников в этой системе осуществляется на уровне смыслового анализа текстовых документов (рекомендации предприятий, рабочие программы дисциплин). Эта задача решается методами обработки естественного языка (NLP), образующих одно из направлений машинного обучения [5].

Автоматизация решения этой задачи является целью данной работы. Функционал системы автоматизации должен реализовывать интеллектуальный анализ требований предприятий и по критерию смысловой близости определять рабочую программу дисциплины. Для достижения поставленной цели выполнена формализация указанной задачи, разработан алгоритм ее решения и проведен ряд экспериментальных исследований.

### *Постановка задачи*

Пусть имеется множество рабочих программ дисциплин $P$, где каждый элемент $p \in P$ является текстовым документом, определяющим содержание и структуру учебного процесса по изучению конкретной учебной дисциплины с целью получения профессиональных знаний.

Пусть $R$ будет множеством требований от предприятий, где каждый элемент $r \in R$ является также текстовым документом, в котором представлены необходимые для работы предприятия компетенции.

Цель заключается в установлении соотношения между требованиями и рабочей программой дисциплины в виде специальной функции $sim: R \rightarrow P$, которая каждому требованию $r \in R$ ставит в соответствие наиболее близкую в профессиональном плане рабочую программу дисциплины $p \in P$.

Таким образом, данная задача предполагает для известного множества рабочих программ $P$ и предложенного текстового документа с требованиями предприятия $r \in R$ определение наиболее подходящей дисциплины p_sim ∈ P.

Дальнейшей задачей является определение функции $sim$ соотнесения требований предприятия с одной из рабочих программ.

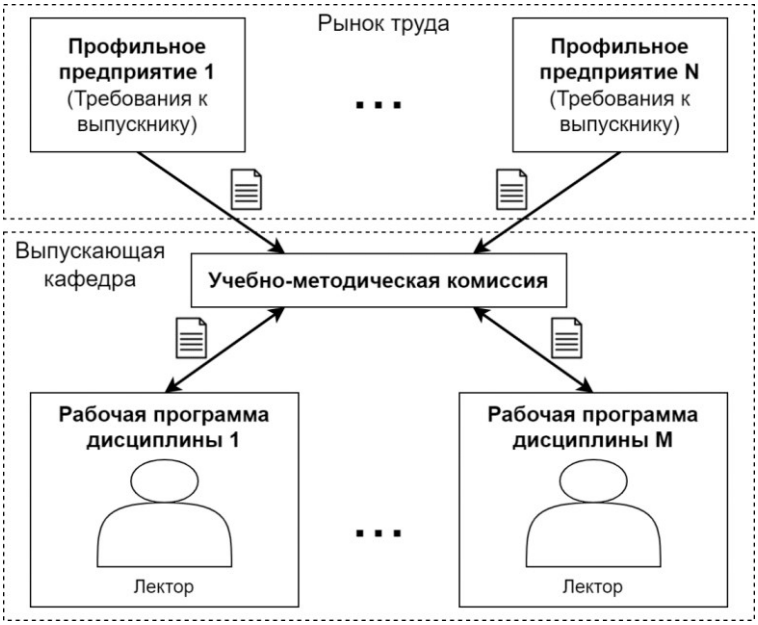



Рисунок 1 – Схема взаимодействия кафедры с профильными предприятиями по обновлению содержания рабочих программ дисциплин

### *Сопоставление требований предприятия и рабочей программы дисциплины*

Определим метрику $M(r,p)$, которая для каждого требования $r \in R$ и каждой рабочей программы дисциплины $p \in P$ будет отражать их схожесть по смыслу в виде вещественного числа с областью значений $E\big(M(r,p)\big) = [-1;1]$. Тогда задача соотнесения требований предприятия с рабочей программой будет состоять в том, чтобы найти такую дисциплину $p$, для которой значение метрики $M(r,p)$ будет максимально, т.е.:

$$\forall\, r \in R\ \langle M(r, sim(r)) = \max_{p \in P} M(r,p) \rangle.$$

Для реализации данной метрики были использованы методы обработки естественного языка (NLP), т.к. они применяются для оценки схожести текстовых документов. С помощью этих методов можно преобразовать тексты в числовое представление, используя модели векторного представления текста. Для вычисления метрики в данном случае достаточно оценить косинусное расстояние между вектором документа требований и рабочей программы дисциплины:

$$M(r,p) = \frac{vec(r) * vec(p)}{\|vec(r)\| * \|vec(p)\|},$$

где $vec$ – функция преобразования текстового документа в вектор заданной длины.

Для преобразования текстового документа в вектор рассматривались модели, основанные на количественных характеристиках слов в тексте, такие как BOW, TF-IDF, и модели вложений слов, такие как Word2Vec, Doc2Vec, FastText и Glove [5, 6]. Для решения данной задачи была выбрана модель вложений слов

Doc2Vec, потому что она позволяет учитывать контекст и смысл слов. Для неё разработан алгоритм, который позволяет создавать векторное представление документа любой длины. Для программной реализации функции $vec$ был построен конвейер NLP (см. рис. 2).

Основными этапами конвейера NLP являются: разделение исходного текста документа на токены с помощью регулярных выражений, фильтрация стоп-слов и слишком коротких или длинных слов, лемматизация, преобразование полученных токенов в семантический вектор текста документа с помощью модели Doc2Vec.

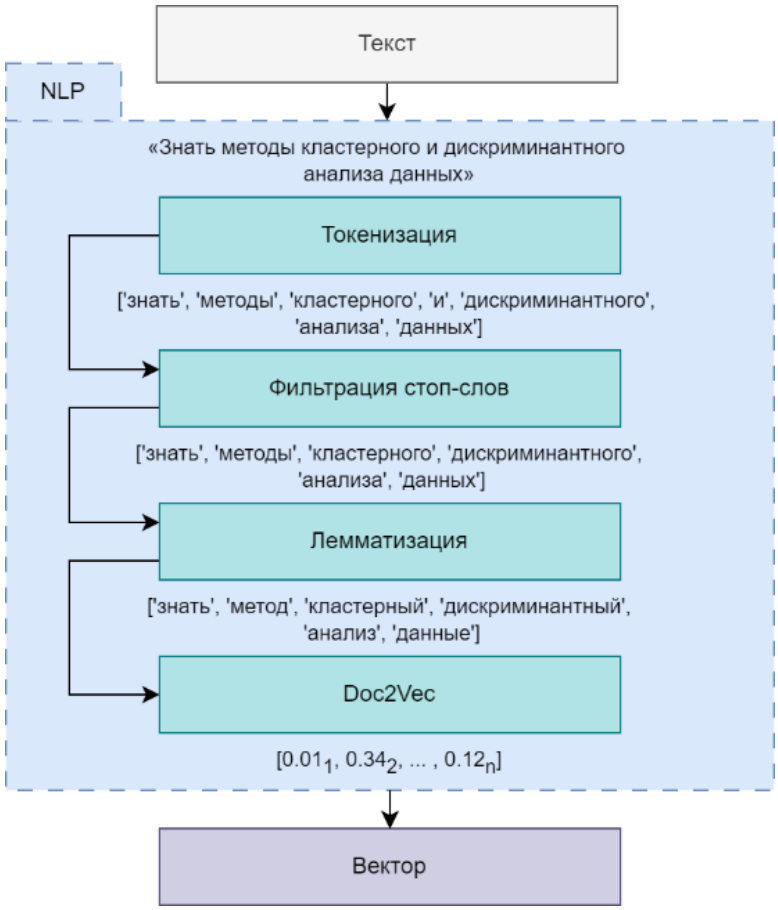



Рисунок 2 — Конвейер NLP для преобразования текстового документа в вектор фиксированного размера n

### *Определение сходства текстов*

Программная реализация модели выполнена с помощью языка программирования Python, библиотек Gensim и spaCy. Экспериментальные исследования разработанной программной модели проведены на примере 3-х рабочих программ дисциплин (информационная безопасность, нейро-нечёткие системы и интеллектуальный анализ данных) и 3-х требований (рекомендаций) предприятий.

В первом эксперименте показано тематическое сходство текстов требований от разных предприятий и рабочих программ дисциплин кафедры. Значения метрики $M$ для требований и рабочих программ дисциплин представлены в таблице 1. Числовые значения, выделенные жирным шрифтом, правильно подчёркивают наибольшую тематическую связь требований с профилем рабочей программы.

Таблица. 1 — Результаты оценки метрики $M(r,p)$

| $r$ \ $p$ | **ИБ** | **ННС** | **ИАД** |
|---|---|---|---|
| Требования к специалистам по НС | 0.58 ± 0.2 | **0.72 ± 0.5** | 0.61 ± 0.05 |
| Требования к специалистам по ИАД | 0.51 ± 0.3 | 0.29 ± 0.4 | **0.98 ± 0.1** |
| Требования к специалистам по ИБ | **0.97 ± 0.3** | 0.34 ± 0.5 | 0.48 ± 0.2 |

***Влияние параметров модели Doc2Vec на сходство текстов по смыслу***

Исходные параметры модели Doc2Vec следующие:
– количество эпох обучения = 50;
– количество эпох обучения для новых документов = количество эпох обучения * 2;
– размер контекстного окна = 5;
– размер вектора документа = 100.

Исследуем влияние некоторых параметров на оценку сходства документов требований предприятий и рабочих программ дисциплин по смыслу. Для эксперимента возьмем требования для специалиста по интеллектуальному анализу данных (ИАД): *«Кандидат должен владеть технологиями анализа больших данных, уметь применять нейронные сети для анализа данных. Иметь представление о принципах обработки естественного языка, языковых моделях, методах поиска и генерации текстов. Знать возможности чат-бота ChatGPT. Знать методы извлечения знаний из разных источников знаний. Кандидат должен знать о возможностях пакетов Data Mining и Text Mining. Знать методы дискриминантного и кластерного анализа данных. Иметь представление о деревьях решений, критериях расщепления, знать алгоритм CART для построения деревьев решений. Знать смешанную модель авторегрессии»*. Результат эксперимента по изменению количества эпох обучения продемонстрирован на рисунке 3.

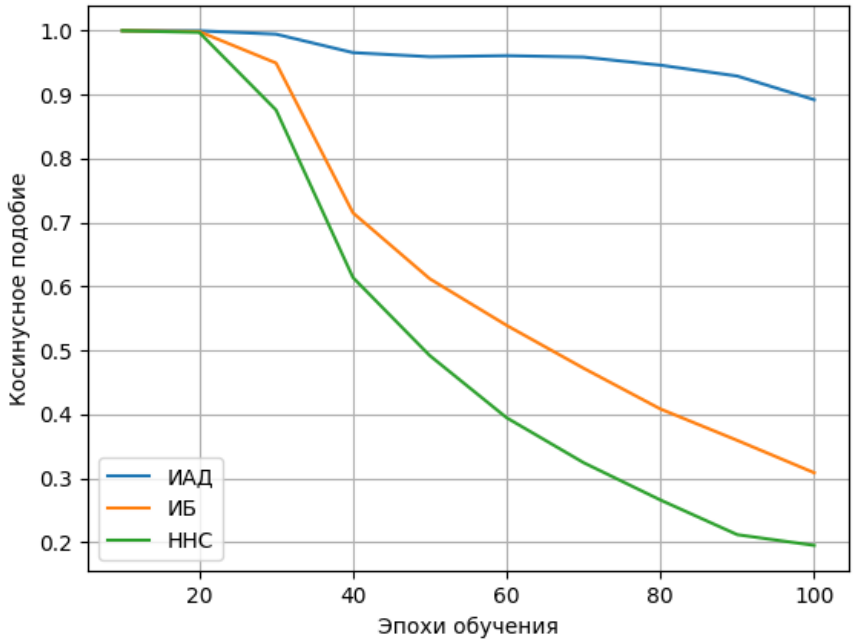



Рисунок 3 — График влияния количества эпох обучения на косинусное подобие вектора требований предприятия к специалисту по ИАД и РПД

На рисунке видно, что увеличение количества эпох обучения приводит к увеличению точности модели, но это также приводит к переобучению модели, но в нашей задаче это скорее полезно, так как документов РПД намного меньше чем документов требований. По итогу экспериментов можно заключить, что наиболее подходящее количество эпох находится в пределах 60-80, чтобы модель не успела сильно переобучиться на документах РПД. Результат эксперимента по изменению размера вектора Doc2Vec продемонстрирован на рисунке 4.

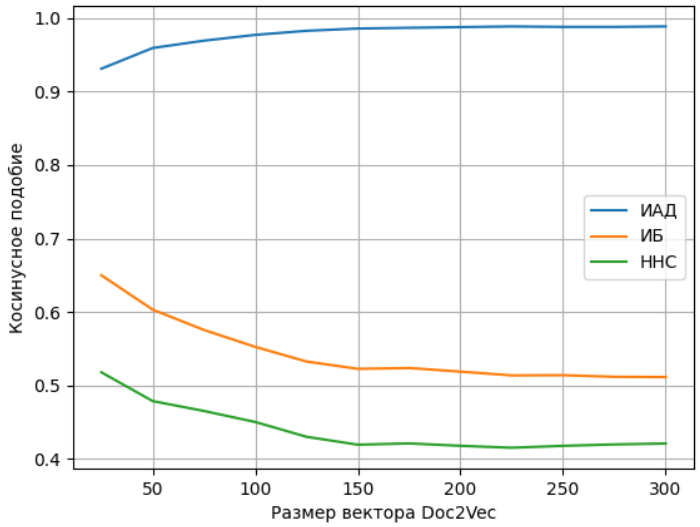



Рисунок 4 — График влияния размера вектора Doc2Vec на косинусное подобие вектора требований предприятия к специалисту по ИАД и РПД

Из рисунка видно, что размер вектора оказывает достаточно малое влияние на определение схожести при большом размере вектора, следовательно, он может быть сокращен для уменьшения количества вычислений и экономии вычислительных ресурсов. Исходя из экспериментальных данных, наилучшим вариантом является размер вектор 150. Результат эксперимента по изменению размера контекстного окна Doc2Vec продемонстрирован на рисунке 5.

На рисунке видно, что слишком малый размер контекстного окна и слишком большой размер показывают схожие результаты. Также видно, что при увеличении размера окна до 2 слов резко повышается ложная схожесть с другими РПД, а затем, по мере увеличение размера окна, она плавно уменьшается, а значит наилучшую точность можно получить при большем значении размера окна. По итогам экспериментов подходящее значение находится в пределах от 8 и более. Такой большой размер связан с тем, что средняя длина предложения в документах РПД составляет в среднем 8 слов, при этом среднеквадратичное отклонение составляет приблизительно 15 слов, таким образом, чтобы покрыть все это предложение так, чтобы слово посередине имело в качестве контекста все предложение, нужен размер контекстного окна 11 ((8 + 15) / 2 = 11.5).

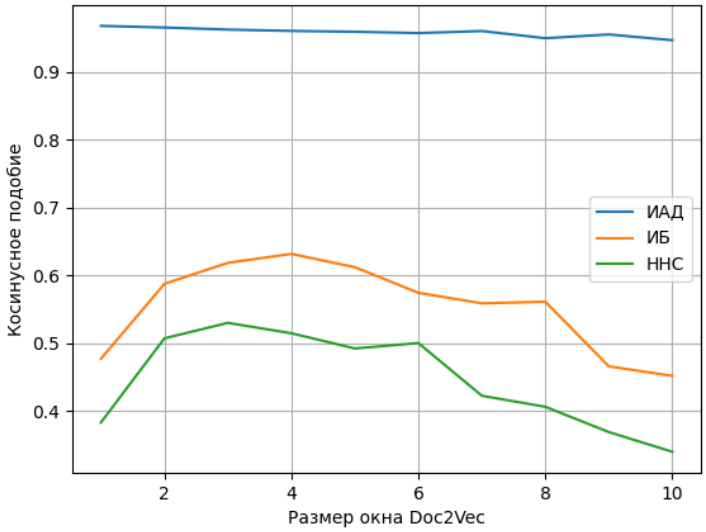



Рисунок 5 — График влияния размера окна Doc2Vec на косинусное подобие вектора требований предприятия к специалисту по ИАД и РПД

### *Влияние слов с высокой повторяемостью на сходство текстов по смыслу*

Любой корпус документов содержит слова, которые есть в большинстве документов, и поэтому они не несут в себе много информации о документах. Исследуем определение схожести документов по смыслу в зависимости от удаления слов с высокой повторяемостью в различных документах. Результат эксперимента по исключению из документов слов с высокой частотой в корпусе продемонстрирован на рисунке 6.

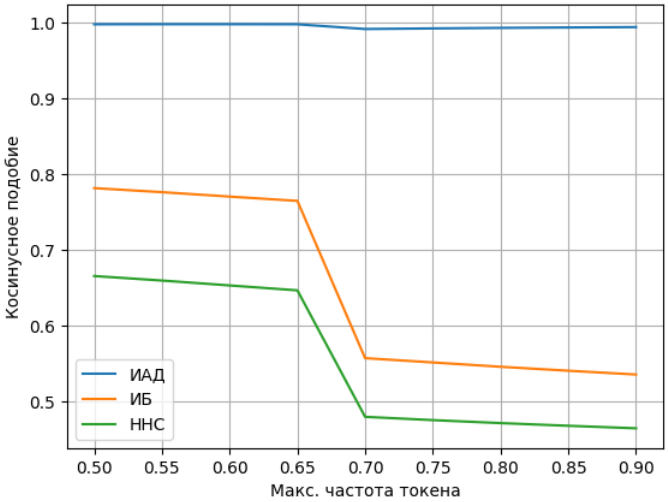



Рисунок 6 — График влияния частоты токенов в корпусе на косинусное подобие вектора требований предприятия к специалисту по ИАД и РПД

На рисунке видно, что при удалении из текстов токенов, которые содержаться в более чем половине документов корпуса (частота 0.5), падает точность определения схожести документов по их косинусному подобию, а при увеличении частоты документов — увеличивается, так как падает ложная схожесть. Поэтому следует отказаться от удаления токенов с высокой частотой, так как это позволяет не терять информацию необходимую для разделения документов по смыслу.

### *Влияние изменения порядка слов на сходство текстов*

Так как Doc2Vec определяет вектор документа путем изучения контекста каждого слова [7], включенного в него, то необходимо оценить влияние перестановок слов в документах на результат обучения модели Doc2Vec. Для этого исследуем определение схожести в зависимости от порядка предложений в тексте. Результат эксперимента по изменению порядка предложений продемонстрирован на рисунке 7. Было проведено 10 экспериментов, в которых предложения в текстах РПД переставлялись случайным образом.

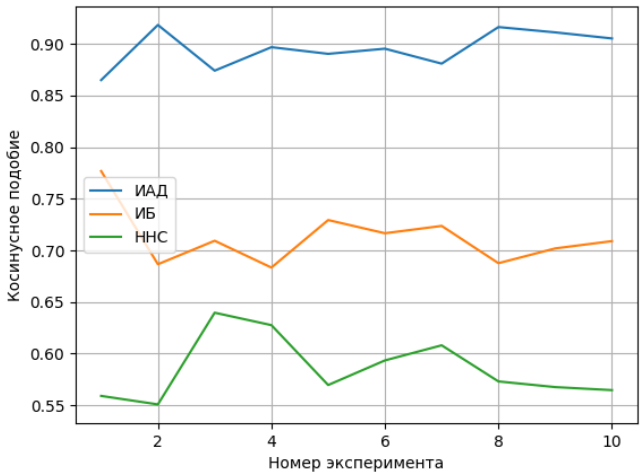



Рисунок 7 — Изменение косинусного подобия вектора требований предприятия к специалисту по ИАД и РПД

при случайных перестановках предложений в РПД

На рисунке видно, что отклонение подобия документов составляет приблизительно 0.05, это значит, что перестановки предложений в тексте оказывает сравнительно небольшое влияние на конечное подобие векторов. Можно сделать вывод, что если размер окна Doc2Vec меньше средней длины предложения (5 < 8), то перестановки предложений не оказывают значительного влияния на компоненты вектора документа, так как меняется лишь порядок обучающих примеров для Doc2Vec. Далее проведем эксперимент с перестановками слов в предложениях, его результаты показаны на рисунке 8. Было проведено 10 экспериментов, в которых слова в предложениях текстов РПД переставлялись случайным образом.

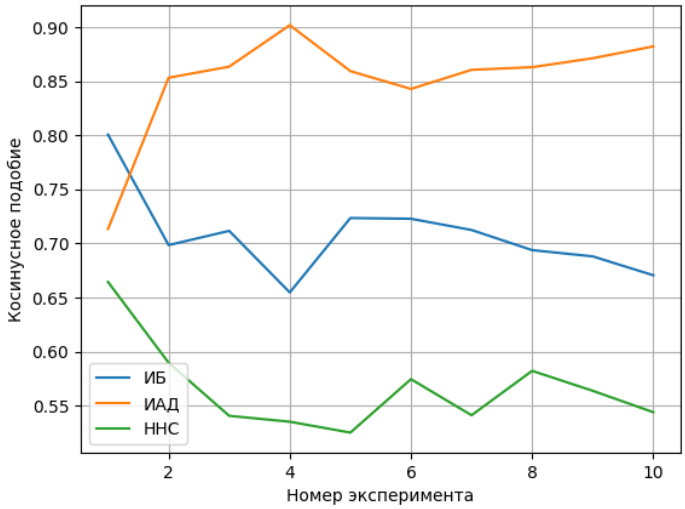



Рисунок 8 — График изменения косинусного подобия вектора требований предприятия к специалисту по ИАД и РПД при случайных перестановках слов в предложениях РПД

Из рисунка следует, что перестановки слов в предложениях оказывают большее влияние на результат, чем перестановки предложений. Это также объясняется тем, что размер окна Doc2Vec меньше средней длины предложений в текстах РПД, таким образом меняются уже сами обучающие примеры, а не их порядок. Однако при данных исходных параметрах влияние мало, что видно на рисунке.

Таким образом, было исследовано влияние изменения порядка слов на определение сходства документов требований предприятий и РПД при помощи косинусного подобия векторов документов, вычисленных моделью Doc2Vec. Эксперименты показали, что данный подход хорошо показал себя в нашей задаче, так как позволяет свести к минимуму влияние словесных формулировок в текстовых документах при определении их схожести по смыслу.

### *Выводы*

Разработанная система интеллектуального анализа текстов эффективно сопоставляет требования предприятий с рабочими программами дисциплин. Применение методов обработки естественного языка и высокоэффективных алгоритмов позволяет регулярно актуализировать РПД, улучшая взаимодействие между учебными заведениями и работодателями. Это способствует повышению уровня подготовки квалифицированных специалистов, соответствующих современным требованиям рынка труда, и открывает новые направления для автоматизации образовательных процессов.

### *Литература*

1. Грязнов С.А., Кузнецов М.И. Проблемы взаимодействия вузов и работодателей в сфере информационных технологий // Международный журнал гуманитарных и естественных наук. - 2024. - № 8-1(95). - С. 66-68.

2. Климова Ю.О., Устинова К.А. Несоответствие уровня подготовки ИТ-кадров требованиям работодателей: проблемы и пути их преодоления // Экономические и социальные перемены: факты, тенденции, прогноз. - 2021. - №5. - С. 202-219.

3. Миронова Д.Ю., Киселева П.С., Баранов И.В. Кооперация вузов и предприятий в контексте новых

вызовов современного инженерного образования // Вестник Омского университета. Серия: Экономика. - 2023. - №1. - С. 60-70.

4. Вадова Л.Ю. Система взаимодействия вуза и работодателей в подготовке будущих специалистов // Международный журнал прикладных и фундаментальных исследований. - 2016. - №5-2. - С. 311-315.

5. Лейн Х., Хапке Х., Ховард К. Обработка естественного языка в действии. - СПб.: Питер, 2020.

6. Логунова Т.В., Щербакова Л.В., Васюков В.М., Шимкун В.В. Анализ алгоритмов классификации текстов // Universum: технические науки. - 2023. - №2-2 (107).

7. Le, Q.V., Mikolov, T. Distributed Representations of Sentences and Documents // International Conference on Machine Learning. – 2014.

***Мелещенко Н.В., Федяев О.И. Определение семантической эквивалентности текстов требований предприятий и рабочих программ дисциплин.*** *Формализован процесс сопоставления текста требований предприятия и рабочей программы дисциплины. Поставленная задача решена путём компьютерной обработки текстов требований предприятий на естественном языке методами машинного обучения. Построен конвейер обработки естественного языка для получения смысловой оценки текста. Исследовано влияние параметров модели на определение схожести текстов по смыслу. Изучено влияния порядка слов в документах на их смысл в реализованной модели. Экспериментальные исследования проведены при помощи разработанной программной реализации. Обосновано применения данного подхода для определения схожести по смыслу текстов требований предприятий и рабочих программ дисциплин.*

***Ключевые слова:*** *кафедра университета, требования предприятий, рабочие программы дисциплин, определение схожести текстов, машинное обучение.*

***Meleshchenko N.V., Fedyaev O.I. Definition of semantic equivalence of texts of requirements of enterprises and work programs of disciplines.*** *The process of comparing the text of the requirements of the enterprise and the work program of the discipline has been formalized. The task was solved by computer processing of texts of enterprise requirements in natural language using machine learning methods. A natural language processing pipeline has been built to obtain a semantic assessment of the text. The influence of model parameters on determining the similarity of texts in meaning is investigated. The influence of the word order in documents on their meaning in the implemented model is studied. Experimental studies were carried out using the developed software implementation. The application of this approach is justified to determine the similarity in the meaning of the texts of the requirements of enterprises and work programs of disciplines.*

***Key words:*** *department of the university, requirements of enterprises, work programs of disciplines, definition of similarity of texts, machine learning.*

УДК 004.93

# Сиамские свёрточные нейронные сети для распознавания лиц

**В.В. Кочетуров[*1], О.И. Федяев[*2]**

[*1] магистрант, Донецкий национальный технический университет,
v.v_kocheturov@mail.ru
[*2] к.т.н, доцент, Донецкий национальный технический университет,
olegfedyayev@mail.ru, OrcID: 0000-0001-6822-7306, SPIN-код: 7777-3791

***Кочетуров В.В, Федяев О.И. Сиамские свёрточные нейронные сети для распознавания лиц.*** *В статье рассматривается применение сверточных нейросетей в сиамских нейронных сетях для задачи распознавания лиц. Проанализирована эффективность сиамских сетей на основе CNN, преимущества данной архитектуры в задачах идентификации и верификации личности, а также основные сложности и перспективы развития технологии.*

***Ключевые слова:*** *сверточные нейронные сети, сиамские нейронные сети, распознавание лиц, глубокое обучение, функция потерь.*

## *Введение*

Распознавание лиц является одной из основных задач в области компьютерного зрения с широким спектром приложений, включая: безопасность, биометрию, социальные сети, робототехнику и медицину. С развитием технологий глубокого обучения, сверточные нейронные сети (CNN) стали основным инструментом для решения этой задачи. Однако, несмотря на значительные успехи, точность и надежность распознавания лиц в реальных условиях остаются серьезной проблемой.

Сложности возникают из-за изменчивости внешнего вида лиц, ограниченности обучающих данных и высокой вычислительной сложности современных моделей. Изменения освещения, ракурса, выражения лица, а также наличие аксессуаров (очки, головные уборы) или шума в изображении усложняют задачу. Кроме того, в реальных приложениях часто доступно ограниченное количество примеров для каждого лица, что приводит к переобучению моделей, особенно в задачах верификации и one-shot learning [1]. Это подчеркивает необходимость разработки методов, способных работать в условиях реальных данных с высокой точностью и производительностью.

Одним из перспективных подходов к решению этих проблем является использование сиамских нейронных сетей в сочетании с CNN. Сиамские сети состоят из двух идентичных подсетей, которые обучаются на парах изображений и вычисляют степень их сходства. Это делает их эффективными для задач сравнения изображений, таких как верификация лиц.

За последние годы был достигнут значительный прогресс в этой области, включая такие исследования, как DeepFace, DeepID, FaceNet и другие [2, 3]. Они демонстрируют высокую точность на известных наборах данных, таких как LFW. В частности, использование функций потерь, таких как Contrastive Loss и Triplet Loss, позволяет улучшить качество обучения сиамских сетей. Однако проблемы, связанные с обучением на больших и разнообразных наборах данных, остаются нерешенными.

Целью данной статьи является выявление оптимальных архитектур и подходов для использования CNN в сиамских нейронных сетях. Основные задачи включают:

- обзор архитектуры сиамской сети на основе CNN;
- изучение влияния различных архитектур и их параметров на точность распознавания лиц;
- сравнительный анализ методов обучения и функций потерь;
- выявление перспективных направлений для дальнейшего развития.

Результаты помогут улучшить точность и производительность систем распознавания лиц в условиях реальных данных, а также определить направления для будущих исследований.

## *Сиамская сеть*

Сиамские нейронные сети (Siamese Neural Networks) — архитектура, которая вычисляет метрику сходства между двумя объектами. Их ключевая особенность — использование двух (или более) идентичных подсетей с

общими параметрами, каждая из которых обрабатывает свои входные данные. Это позволяет эффективно обучаться на парах примеров, сравнивая их и определяя степень сходства.

Каждое изображение из пары проходит через одну из подсетей, где извлекаются признаки. На выходе сети формируется эмбеддинг (вектор признаков) для каждого изображения, после чего их сходство оценивается с использованием функции расстояния, например, Евклидова или косинусного.

Архитектура сиамской сети состоит из следующих компонентов (см.рис.1):

– базовая сеть (Backbone) — это сверточная нейронная сеть (CNN), используемая для извлечения признаков из входных изображений;

– функция расстояния - вычисляет разницу между эмбеддингами двух изображений;

– функция потерь - определяет, как сеть наказывает за неверное определение сходства или различия между объектами.

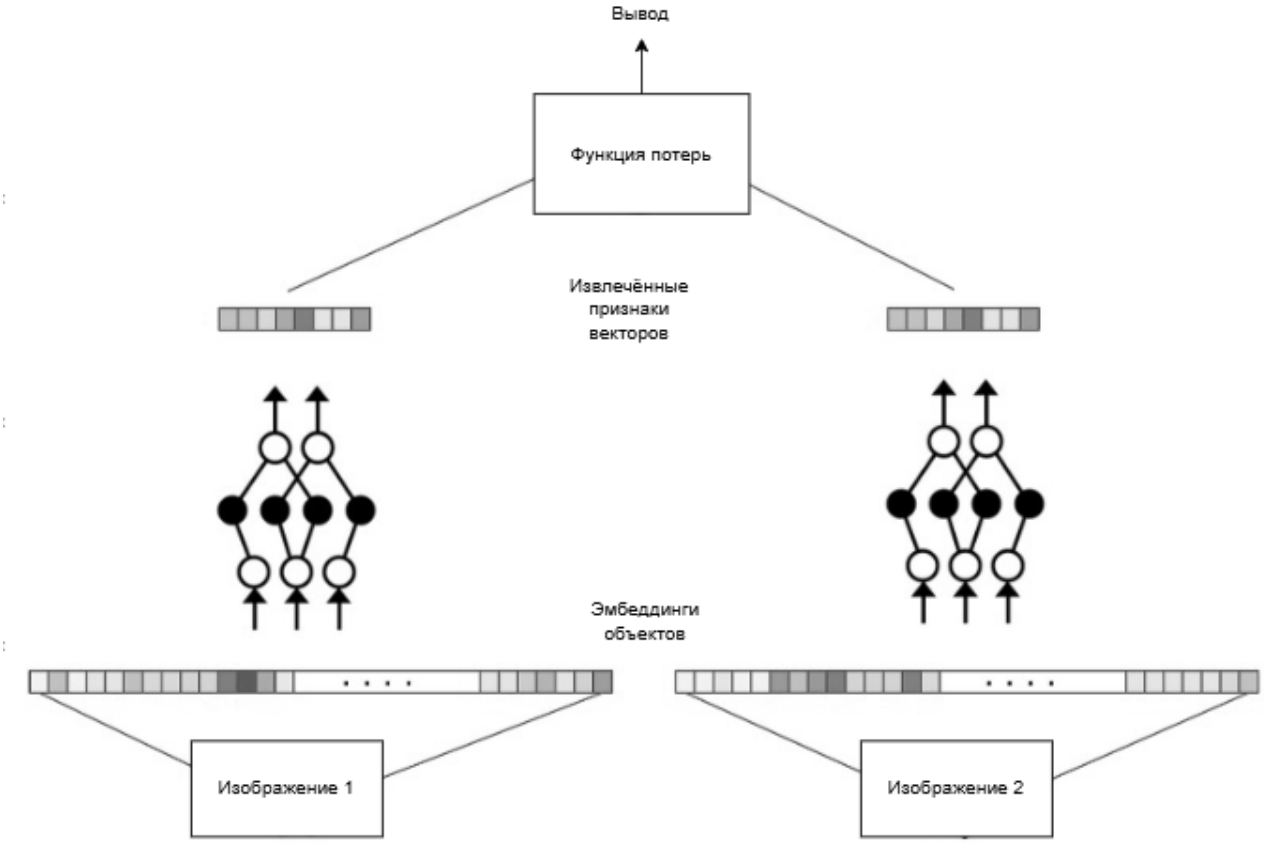



Рисунок 1 – Схема сиамской нейронной сети

На этапе обучения сеть работает с парами изображений:

– положительные пары — изображения одного человека;

– отрицательные пары — изображения разных людей.

На этапе инференса сеть получает два изображения, вычисляет их эмбеддинги и оценивает сходство. Если значение функции расстояния меньше заданного порога, изображения считаются принадлежащими одному человеку.

### *Функции потерь в сиамских сетях*

Функции потерь играют ключевую роль в обучении сиамских нейронных сетей, так как они определяют, как сеть должна корректировать свои параметры для оптимального различения похожих и непохожих объектов.

Contrastive Loss — одна из наиболее популярных функций потерь для сиамских сетей. Она используется для обучения сети различать похожие и непохожие пары объектов, минимизируя расстояние между эмбеддингами похожих объектов и максимизируя расстояние между эмбеддингами непохожих [4].

Формула Contrastive Loss:

$$L = (1-y)\cdot\frac{1}{2}D(x_1,x_2)^2 + y\cdot\frac{1}{2}\max(0,m-D(x_1,x_2))^2 \quad (1)$$

где $D(x_1,x_2)$ — расстояние между эмбеддингами, $y$ — метка класса (1 для разных лиц и 0 для одинаковых), $m$ — минимальное допустимое расстояние между эмбеддингами разных лиц.

Triplet Loss используется для обучения эмбеддингов, обеспечивающих большую близость между похожими объектами и их значительное различие с непохожими объектами [5]. Эта функция потерь требует обучения на триплетах: якорное изображение ($x_a$), положительный пример ($x_p$), и отрицательный пример ($x_n$) (см.рис.3) [6].

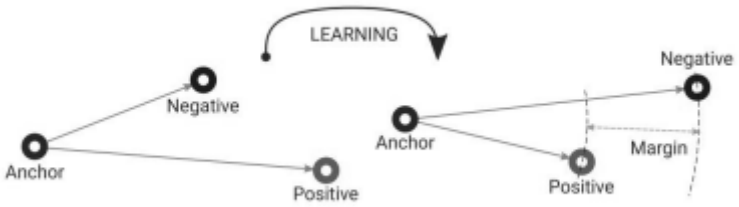



Рисунок 2 - Обучение функции потерь triplet loss

Формула Triplet Loss:

$$L = \max(0, D(x_a,x_p) - D(x_a,x_n) + a),; \quad (2)$$

где $D(x_a,x_p)$ — расстояние между эмбеддингами якорного изображения и положительного примера, $D(x_a,x_n)$ — расстояние между эмбеддингами якорного изображения и отрицательного примера, $a$ — маржа, которая обеспечивает, что $D(x_a,x_n)$ существенно больше $D(x_a,x_p)$.

### *Архитектуры свёртоных нейросетей*

Архитектура VGG, предложенная в 2014 году, остаётся популярной благодаря своей простоте и эффективности. Она состоит из последовательности свёрточных слоёв с фиксированным размером фильтра (3×3) и максимального пулинга (2×2). Наиболее распространённой конфигурацией является VGG16 (см.рис.3). Архитектура обеспечивает высокую точность, однако её вычислительная сложность и потребление памяти затрудняют использование на устройствах с низкими ресурсами.

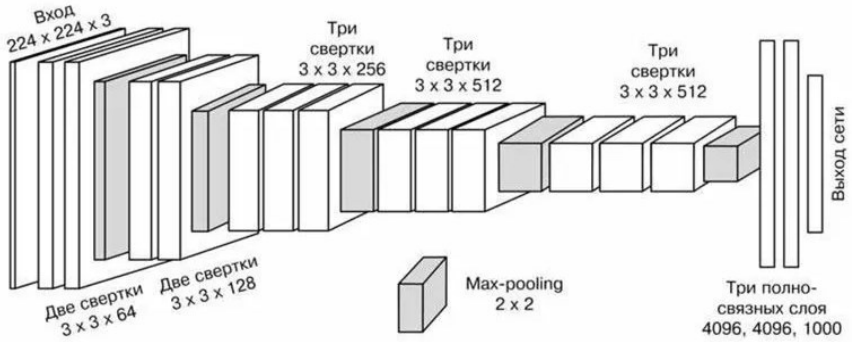



Рисунок 3 – Архитектура свёрточной нейронной сети VGG16

Формула сверточного слоя:

$$f(x) = \mathrm{Re}\,LU(W * x + b) \quad (4)$$

где $W$ – весовые коэффициенты, b – смещение, * - операция свёртки.

ResNet предложенная в 2015 году, ResNet решает проблему затухания градиентов с помощью резидуальных блоков и shortcut connections. Это позволяет эффективно обучать глубокие архитектуры, такие как ResNet50, ResNet101 и ResNet152 (см.рис.4). Хотя ResNet обладает умеренной вычислительной сложностью, она обеспечивает высокую точность в задачах распознавания лиц.

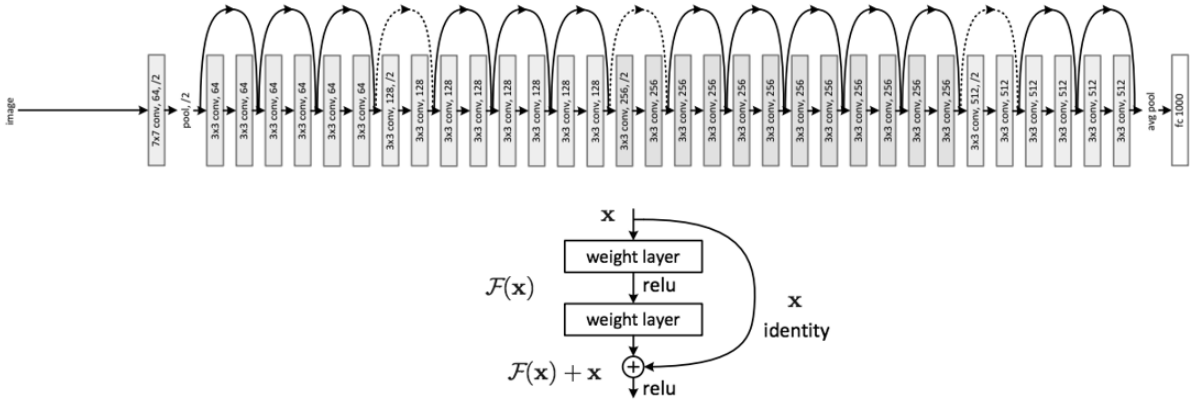



Рисунок 4 – Архитектура свёрточной нейронной сети ResNet50

Формула резидуального блока:

$$y = F(x, \{W_i\}) + x \tag{5}$$

где $F(x,\{W_i\})$ - выход свёрточных слоев, x — входной сигнал.

Архитектура Inception была разработана для повышения эффективности свёртки за счёт параллельных операций с фильтрами разного размера (например, 1×1, 3×3 и 5×5) (см.рис.5). Благодаря оптимизации вычислительных ресурсов, Inception подходит для задач с ограниченными ресурсами и используется в мобильных приложениях.

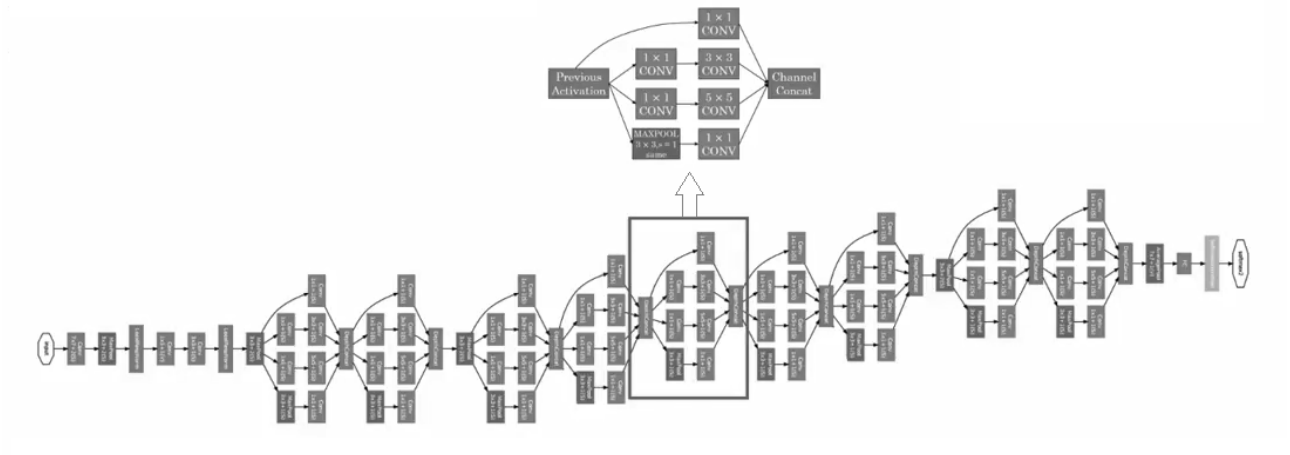



Рисунок 5 – Архитектура свёрточной нейронной сети Inception (GoogLeNet)

Формула инцепшн-блока:

$$f(x) = Concat(f_1(x), f_3(x), f_5(x)) \tag{6}$$

где $f_1(x)$, $f_3(x)$, $f_5(x)$ - результаты свертки с фильтрами соответствующего размера.

MobileNet оптимизирована для мобильных устройств за счёт использования depthwise separable convolutions, что позволяет значительно снизить вычислительную сложность (см.рис.6). Хотя она уступает по точности глубоким моделям, таким как ResNet, её эффективность делает её идеальной для встроенных систем [7].

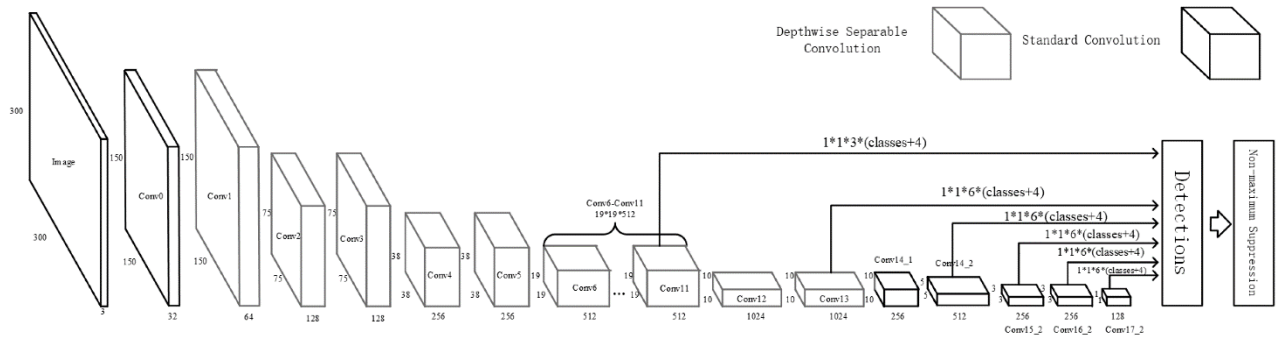



Рисунок 6 – Архитектура свёрточной нейронной сети MobileNet

Формула глубинной раздельной свертки:

$$\mathrm{f}\, f(x) = f_{depthwise}(x) + f_{po\,\mathrm{int}\,wise}(x) \tag{7}$$

где $f_{depthwise}$ — свертка по каналам, $f_{pointwise}$ — объединение каналов.

### *Обучение*

Для обучения и тестирования моделей распознавания лиц используют популярные наборы данных, включающие большое количество изображений с метками классов.

Ключевые наборы данных:

– Labeled Faces in the Wild (LFW) - набор изображений лиц в реальных условиях, включая различия в освещении, ракурсах и выражениях лица. 13,000 изображений 5749 различных людей. Чаще всего используется для верификации (pair matching) и оценки качества модели.

– CASIA-WebFace содержит 500,000 изображений 10,575 уникальных идентичностей. Предназначен для обучения моделей в задачах идентификации и верификации.

– VGGFace2 - более 3.3 млн изображений с вариациями освещения, ракурса и возраста. Идеален для обучения глубокой нейросети с высокой генерализацией.

– MS-Celeb-1M - крупнейший набор данных (10 млн изображений 100,000 лиц). Подходит для тренировки высокопроизводительных моделей.

– MegaFace используется для тестирования робастности моделей в условиях больших идентификационных массивов.

Для обеспечения надежного обучения и повышения качества модели используется предобработка и аугментация данных. С помощью методов, таких как Multi-task Cascaded Convolutional Networks (MTCNN), лица на изображениях выравниваются относительно глаз и носа [8]. Все изображения приводятся к единому размеру (например, 112×112 пикселей) для совместимости с архитектурой сети. Значения пикселей нормализуются в диапазон [0,1] или стандартизируются (обнуление среднего и деление на стандартное отклонение). Удаляются изображения с низким качеством, сильными шумами или частичным перекрытием лица.

Методы аугментации:

– геометрические трансформации;
– изменения яркости и контраста;
– применение шумов;
– горизонтальное отражение.

### *Эксперименты*

Эксперименты направлены на оценку точности и эффективности различных архитектур CNN в составе сиамских нейронных сетей для задачи распознавания лиц. Прежде всего, были выбраны несколько популярных архитектур CNN: VGG, ResNet, Inception и MobileNet. Эти архитектуры использовались в качестве базовых компонентов сиамской сети.

Для обучения использовались следующие параметры:

– оптимизатор Adam с начальной скоростью обучения $10^{-4}$ и scheduler для ее понижения [9];
– функции потерь Contrastive Loss и Triplet Loss;
– batch size, равный 64.

Обучение осуществлялось на наборах данных CASIA-WebFace и VGGFace2, которые обеспечивают широкий спектр изображений. Тестирование производилось на LFW для оценки верификации и MegaFace для проверки производительности в условиях больших баз данных.

Изображения из всех наборов данных были предварительно обработаны до размера 112×112, с применением различных методов аугментации, таких как изменение освещения, геометрические трансформации и добавление шума. Эксперименты проводились на оборудовании с использованием GPU NVIDIA RTX 3070, в среде PyTorch версии 2.1. Для оценки качества работы моделей использовались метрики: точность (accuracy) и среднее время обработки одного изображения.

Результаты можно увидеть в таблице 1.

.

Таблица 1 – Результаты сравнения разных архитектур сетей

| Архитектура | LFW (Accuracy) | MegaFace (Top-1 Accuracy) | Время обработки (мс) |
|---|---|---|---|
| VGG | 96.2% | 86.4% | 125.5 |
| ResNet | 98.4% | 92.1% | 84.8 |
| Inception | 97.7% | 89.9% | 66.1 |
| MobileNet | 94.3% | 84.3% | 17.7 |

Средние значения точности распознавания с использование функций потерь Contrastive Loss и Triplet Loss можно увидеть в таблице 2.

.

Таблица 2 – Результаты сравнения функции потерь Contrastive Loss и Triplet Loss

| Функция потерь | LFW (Accuracy) | MegaFace (Top-1 Accuracy) |
|---|---|---|
| Contrastive Loss | 95.1% | 86.6% |
| Triplet Loss | 96.65% | 88.17% |

### *Выводы*

В статье рассмотрена эффективность использования сиамских нейронных сетей, построенных на базе различных архитектур CNN, для решения задачи распознавания лиц. Проведенный анализ показал, что такие архитектуры, как ResNet и Inception, обеспечивают наилучшее сочетание точности и производительности, в то время как MobileNet предлагает компромисс между вычислительной сложностью и качеством распознавания.

Были изучены проблемы, связанные с изменчивостью условий съемки, низким качеством данных и ограниченным количеством изображений. Применение функций потерь Contrastive Loss и Triplet Loss позволило добиться значительного улучшения точности распознавания. Однако выявлены ограничения современных методов, включая сложности в обработке изображений с экстремальными углами или аксессуарами.

Полученные результаты подчеркивают важность использования более сложных методов аугментации данных, адаптивных функций потерь и легковесных моделей для обеспечения высокой производительности на устройствах с ограниченными ресурсами. В перспективе дальнейшие исследования могут быть направлены на разработку новых архитектур CNN, улучшение робастности моделей и повышение устойчивости к атакам, что обеспечит более широкое применение систем распознавания лиц в реальных условиях.

### *Литература*

1. Koch G. et al. Siamese neural networks for one-shot image recognition //ICML deep learning workshop. – 2015. – Т. 2. – №. 1. – С. 1-30.

2. Taigman Y. et al. Deepface: Closing the gap to human-level performance in face verification //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2014. – С. 1701-1708.

3. Schroff F., Kalenichenko D., Philbin J. Facenet: A unified embedding for face recognition and clustering //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – С. 815-823.

4. Wang F., Liu H. Understanding the behaviour of contrastive loss //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. – 2021. – С. 2495-2504.
5. Yeung H. W. F., Li J., Chung Y. Y. Improved performance of face recognition using CNN with constrained triplet loss layer //2017 International Joint Conference on Neural Networks (IJCNN). – IEEE, 2017. – С. 1948-1955.
6. Здоровцова. Е. А. Использование сиамских нейронных сетей в задаче схожести изображений // Современные информационные технологии. Том 5. – 2022. – С. 93-96.
7. Duong C. N. et al. Mobiface: A lightweight deep learning face recognition on mobile devices //2019 IEEE 10th international conference on biometrics theory, applications and systems (BTAS). – IEEE, 2019. – С. 1-6.
8. Robin M. H. et al. Improvement of face and eye detection performance by using multi-task cascaded convolutional networks //2020 IEEE Region 10 Symposium (TENSYMP). – IEEE, 2020. – С. 977-980.
9. Diederik P. K. Adam: A method for stochastic optimization // (No Title). – 2014

***Кочетуров В.В., Федяев О.И. Сиамские свёрточные нейронные сети для распознавания лиц.*** *В статье рассматривается применение сверточных нейросетей в сиамских нейронных сетях для задачи распознавания лиц. Проанализирована эффективность сиамских сетей на основе CNN, преимущества данной архитектуры в задачах идентификации и верификации личности, а также основные сложности и перспективы развития технологии.*

***Ключевые слова:*** *сверточные нейронные сети, сиамские нейронные сети, распознавание лиц, глубокое обучение, функция потерь.*

***Kocheturov V.V., Fedyaev O.I. Siamese convolutional neural networks for face recognition.*** *The article deals with the use of convolutional neural networks in Siamese neural networks for the task of face recognition. The efficiency of Siamese networks based on CNNs, advantages of this architecture in the tasks of identification and verification of a person, as well as the main difficulties and prospects of technology development are analyzed.*

***Keywords:*** *convolutional neural networks, Siamese neural networks, face recognition, deep learning, loss function.*

УДК 004.522

# Интеграция голосовых технологий в мессенджеры: создание голосового ассистента

**П.С. Похлёбин[*1], О.И. Федяев[*2]**

[*1] магистрант, Донецкий национальный технический университет,
xendri@list.ru

[*2] к.т.н, доцент, Донецкий национальный технический университет,
olegfedyayev@mail.ru, OrcID: 0000-0001-6822-7306, SPIN-код: 7777-3791

***Похлёбин П.С., Федяев О.И. Интеграция голосовых технологий в мессенджеры: создание голосового ассистента.*** *Статья посвящена разработке голосового ассистента для повышения эффективности взаимодействия с мессенджерами. Рассматриваются ключевые методы и технологии, применяемые при создании ассистента, а также возможности интеграции интеллектуальных функций для автоматизации задач в мессенджерах. Приводится анализ преимуществ интеллектуального подхода к обработке сообщений и взаимодействию с пользователем. Также затрагиваются технические и этические аспекты внедрения голосовых ассистентов и предлагаются методы повышения их надёжности и точности.*

***Ключевые слова:*** *голосовой ассистент, мессенджеры, интеллектуализация, автоматизация, конфиденциальность, обработка данных, речевые технологии.*

## *Введение*

В условиях динамичного развития информационно-вычислительных систем и широкого применения мессенджеров для коммуникации возрастает потребность в новых подходах, повышающих эффективность работы с этими платформами. Одним из перспективных решений является внедрение голосовых ассистентов, способных автоматизировать и интеллектуализировать взаимодействие с мессенджерами. Голосовые ассистенты уже активно используются в различных областях, но их интеграция в мессенджеры для выполнения повседневных задач, таких как отправка сообщений, поиск информации или управление перепиской, требует решения ряда специфических технических и этических задач.

Последние исследования в области разработки голосовых интерфейсов и автоматизации общения показывают значительный интерес к созданию интуитивных и высокоточных алгоритмов, обеспечивающих качественное взаимодействие с пользователями. Однако многие аспекты, такие как обеспечение конфиденциальности данных, управление контекстом диалога и оптимизация голосовых интерфейсов для удобства пользователя, остаются недостаточно исследованными. Это свидетельствует о необходимости в изучении методов, позволяющих преодолеть эти трудности, и создании новых решений, улучшающих функционал мессенджеров с помощью голосовых технологий.

Цель данной статьи — разработать и представить голосового ассистента для интеллектуализации взаимодействия с мессенджерами, который решает ключевые задачи, такие как автоматизация общения, обработка голосовых команд и интеграция с популярными мессенджерами. В статье будут рассмотрены как теоретические, так и практические аспекты разработки, предложена структура ассистента и реализован метод автоматического распознавания речи, а также обсуждены особенности внедрения голосового интерфейса в повседневное использование.

## *Обзор существующих голосовых ассистентов*

Голосовые ассистенты, такие как Siri, Google Assistant, Amazon Alexa, Alisa и Microsoft Cortana, являются одними из самых популярных решений для автоматизации повседневных задач и улучшения пользовательского опыта. Эти системы используют технологии обработки естественного языка (NLP), автоматического распознавания речи (ASR) и синтеза речи (TTS), чтобы взаимодействовать с пользователем, отвечать на вопросы и выполнять команды. В последние годы голосовые ассистенты стали более гибкими и интеллектуальными, что позволило расширить их функционал. Например, голосовые помощники могут управлять умными устройствами, помогать в поиске информации, выполнять задачи по организации расписания и отвечать на вопросы в реальном времени. Однако, несмотря на их широкие возможности, существующие решения не всегда оптимизированы для

интеграции с мессенджерами, что ограничивает их применение в таких специфичных областях, как автоматизация общения и управления перепиской.

Анализ возможностей самых популярных голосовых ассистентом представлен в таблице 1.

Таблица 1 – Голосовые помощники и их функции

| Голосовой помощник | Siri | Google Assistant | Alisa |
|---|---|---|---|
| Производитель | Apple | Google | Яндекс |
| Код активации | «Привет, Sir» | «Окей Google» | «Привет, Алиса» |
| Задачи | Управлением умными устройствами;<br>Поиск информации;<br>Цифровая няня. | Управлением умными устройствами;<br>Поиск информации;<br>Справочник; | Обработка голосового запроса;<br>Управление медиа устройствами;<br>Поиск информации;<br>Создание информационных наборов по управлению жизнью человека |
| Интеграции с социальными сетями | Да | Нет | Да |
| Понимание произвольных команд | Да | Да | Да |
| Мобильные приложения | Android, iOS | Android, iOS, браузер Chrome | Android, iOS, браузер Яндекс |

Интеграция голосовых ассистентов с мессенджерами открывает новые горизонты для улучшения функционала популярных платформ. Например, в Telegram и WhatsApp уже имеются боты, которые могут выполнять команды через текстовый ввод. Однако голосовые команды в таких мессенджерах реализованы не так широко, как хотелось бы. Несмотря на это, некоторые решения, такие как боты для Telegram, начинают интегрировать функциональность голосовых команд для выполнения различных операций, таких как отправка сообщений или поиск информации. Одним из примеров является использование технологий ASR и NLP для преобразования голосовых сообщений в текстовые с автоматической отправки их пользователям. В свою очередь, использование технологий TTS позволяет ботам или ассистентам «озвучивать» ответы пользователям, что улучшает взаимодействие и делает процесс общения более естественным.

### *Технологии для разработки голосовых ассистентов*

Обработка естественного языка (NLP) — это область искусственного интеллекта, занимающаяся взаимодействием между компьютерами и человеческим языком. В контексте голосового ассистента NLP помогает системе понимать, интерпретировать и генерировать текст на естественном языке. Это ключевая технология для реализации взаимодействия с пользователем, особенно в случае голосовых команд (см. табл. 2).

Таблица 2 - Этапы обработки естественного языка (NLP)

| Этап NLP | Описание | Пример с «погода» | Особенности |
|---|---|---|---|
| Токенизация | Разбиение текста на отдельные слова и фразы. | «Что», «погода», «завтра?» | Поддержка различных форматов и стилей запросов |
| Семантический анализ | Понимание смысла текста, его контекста | Определяется, что запрос касается информации о погоде | Учёт значений, синонимов и многозначности |
| Извлечение сущностей | Выделение ключевых данных, таких как даты, местоположения и имена. | Выделяется сущность «завтра» как дата и «погода» как объект | Включение анализа сущностей для более точных ответов |
| Обработка запросов | Распознавание намерений пользователя. | Определяется намерение: запрос информации о погоде на завтра | Учёт контекста для ответов, основанных на предыдущих запросах. |

Рассмотрим базовый пример применения технологии обработки естественного языка (NLP), которая используется для анализа и интерпретации текстовых запросов пользователей голосовыми ассистентами (см. рис. 1).

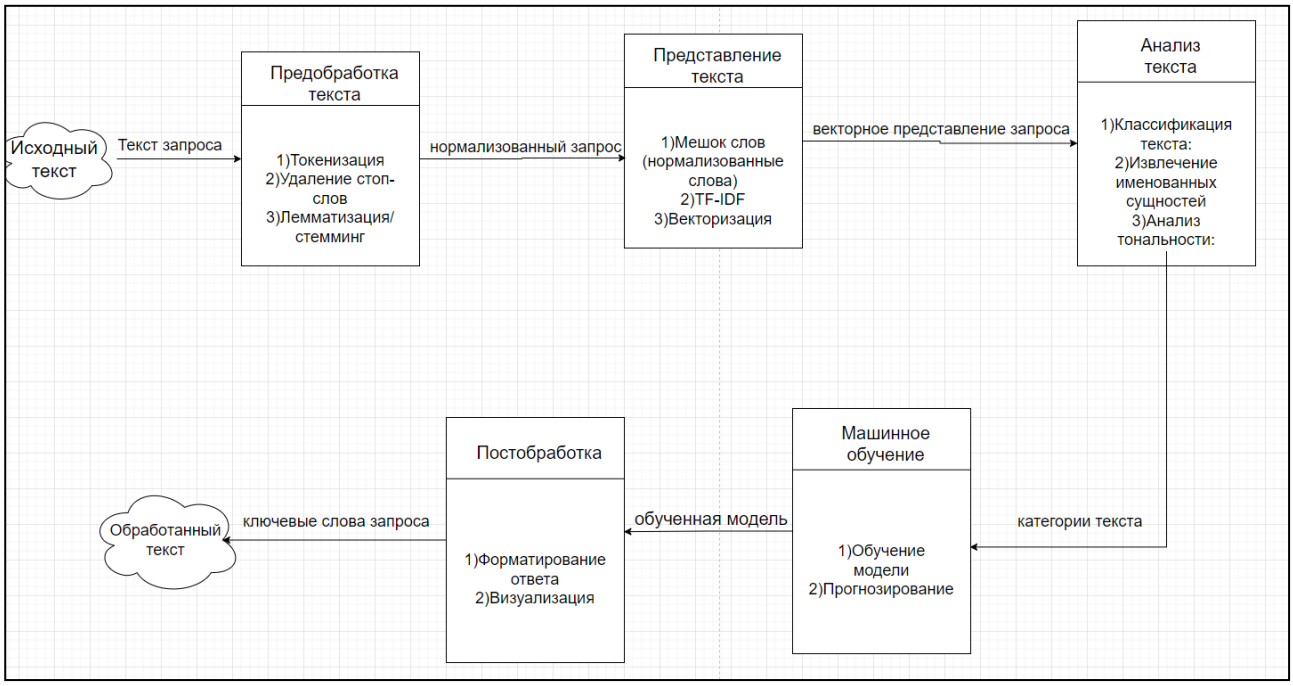



Рисунок 1 - Типовые шаги NLP при анализе текстовых запросов

ASR — это технология, которая позволяет компьютерам распознавать и преобразовывать звуковые сигналы в текст. Это важная часть голосового ассистента, которая делает возможным взаимодействие через голосовые команды.

ASR используется для автоматизации преобразования голосовых команд в текст, который затем анализируется с помощью NLP для выполнения нужных действий (см. табл. 3). На практике это выглядит так: пользователь говорит команду, ассистент записывает голос, преобразует его в текст и анализирует.

Таблица 3 - Основные этапы распознавания речи (ASR)

| Этап ASR | Описание | Пример с «погода» | Особенности |
|---|---|---|---|
| Распознавание речи | Преобразование аудио сигнала в текст. | Голосовой ввод «Какая погода?» преобразуется в текст | Требует фильтрации фонового шума |
| Фильтрация шума | Удаление фонового шума для улучшения качества распознавания. | Устраняет шум, чтобы сохранить точность распознавания | Устойчивость к разным уровням шума |
| Распознавание языка | Определение языка, на котором говорит пользователь. | Определяется, что запрос сделан на русском языке | Учёт диалектов и акцентов пользователей |

На рисунке 2 показан базовый пример применения технологии распознавания речи (ASR), которая позволяет преобразовать аудио файл в текст для дальнейшей обработки.

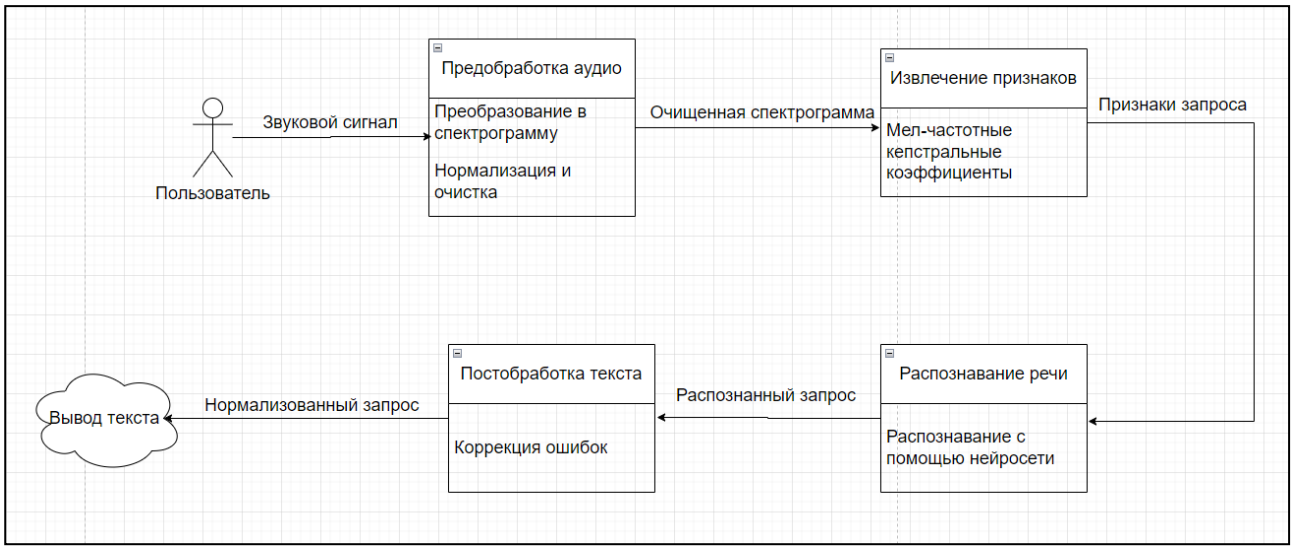



Рисунок 2 - Типовые шаги ASR при обработке звукового сигнала

TTS — это технология преобразования текста в речь, которая используется для озвучивания текстовых ответов ассистента (см. табл. 4). С помощью TTS голосовой ассистент может «говорить» с пользователем, что делает взаимодействие более естественным.

Когда ассистент анализирует запрос и получает ответ, этот ответ обычно генерируется в текстовом виде. Далее TTS преобразует текст в речь, чтобы пользователю был озвучен ответ.

Таблица 4 - Этапы преобразования текста в речь (TTS)

| Этап TTS | Описание | Пример с «погода» | Особенности |
|---|---|---|---|
| Преобразование текста в речь | Генерация звука из текстовой строки. | Текст «Завтра ожидается дождь» преобразуется в аудио выход | Выбор подходящего тембра голоса |
| Синтез голоса | Создание натурального звучания речи для взаимодействия с пользователем. | Генерируется голос, подходящий для ассистента | Тональность и выразительность речи |
| Регулировка интонации | Настройка интонации, скорости и громкости речи. | Вопрос «Какая погода?» интонируется как вопрос | Настройки интонации для восприятия вопроса или утверждения |

На рисунке 3 показано применение технологии синтеза речи (TTS) для преобразования текстового ответа в речь для озвучивания пользователю.

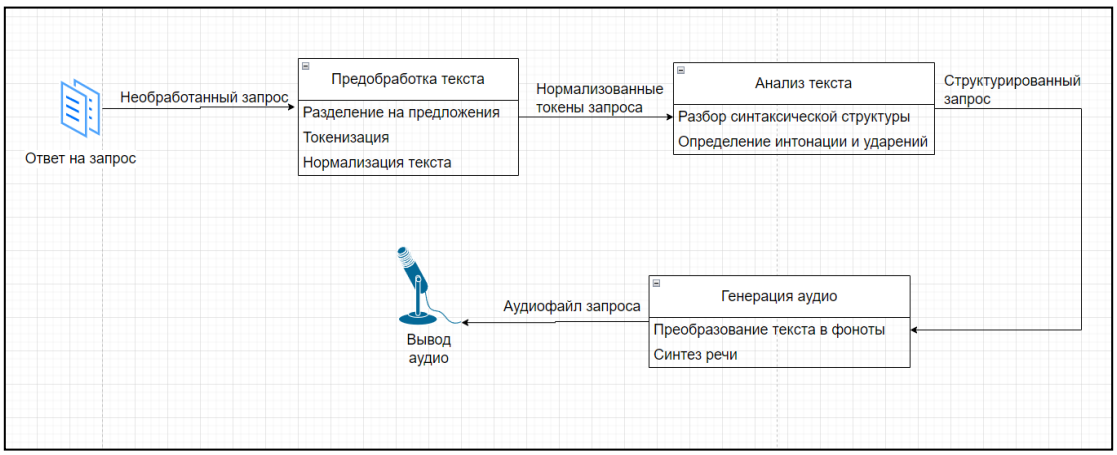



Рисунок 3 - Типовые шаги TTS при генерации аудио-ответа

### *Архитектура системы голосового ассистента для мессенджера Telegram*

Целью разработки является создание голосового ассистента для мессенджера Telegram, который способен выполнять различные команды пользователя, такие как поиск сообщений, вывод контактов и отправка сообщений. Задача состоит в том, чтобы создать систему, которая будет легко интегрироваться с мессенджером, обеспечивать точность распознавания команд и оперативно выполнять запросы. Ассистент должен работать с использованием технологии распознавания речи (ASR), обработки естественного языка (NLP) и синтеза речи (TTS), а также иметь возможность взаимодействовать с API Telegram через библиотеку Telethon.

Система представлена в виде демо-версии, которая пока ориентирована на работу с мессенджером Telegram. В будущем планируется расширить функционал, добавив поддержку других мессенджеров, таких как VK, WhatsApp и других, что позволит сделать ассистента более универсальным и доступным для пользователей разных платформ. Также будет внедрена защита данных, включая безопасность хранения и обработки информации, работу с базами данных, а также реализация механизмов аутентификации и авторизации пользователей для защиты личных данных.

Система состоит из нескольких взаимосвязанных модулей, которые выполняют различные функции, необходимые для распознавания голосовых команд, их обработки, взаимодействия с пользователем и выполнением задач через Telegram. Каждый модуль выполняет свою роль, обеспечивая корректное функционирование голосового ассистента. Рассмотрим каждый из них.

Модуль распознавания речи (ASR). Модуль распознавания речи отвечает за преобразование голосового ввода в текст. Для этого используется библиотека sounddevice, которая позволяет записывать аудиосигнал с микрофона. Полученный аудиофайл передаётся в модель Vosk, которая обрабатывает сигнал и возвращает текстовое представление произнесённой команды. Модуль активно взаимодействует с микрофоном и постоянно следит за входными данными, выполняя анализ в реальном времени.

Модуль обработки команд (NLP). Этот модуль анализирует распознанный текст команды и сопоставляет его с заранее определёнными командами из набора data_set. В зависимости от того, какая команда была распознана, выполняется соответствующая функция, такая как отправка сообщения, показ контактов или чтение последних сообщений. NLP-модуль использует алгоритмы для обработки текста, чтобы понять намерения пользователя и выполнить нужное действие.

Модуль взаимодействия с Telegram (Telethon). Этот модуль отвечает за взаимодействие с Telegram через API, используя библиотеку Telethon. Он подключается к аккаунту пользователя и позволяет выполнять операции, такие как поиск сообщений, вывод контактов и отправка сообщений пользователю. Все действия происходят через асинхронные вызовы, что позволяет системе быть отзывчивой и эффективной при работе с Telegram.

Модуль синтеза речи (TTS). Данный модуль отвечает за озвучивание ответов пользователю. Это важно для создания полноценного голосового взаимодействия. В нём используется библиотека для синтеза речи, например, pyttsx3, которая преобразует текстовые ответы в аудио и воспроизводит их через динамики.

Схема архитектуры системы иллюстрирует взаимодействие различных модулей (рис. 4).

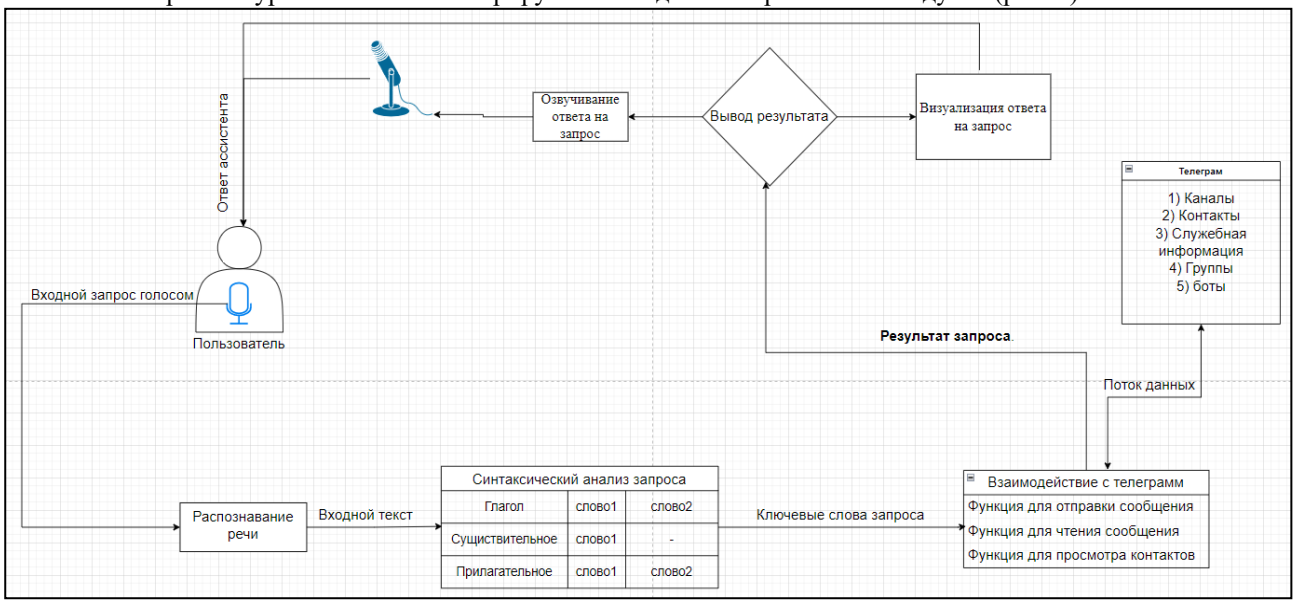



Рисунок 4 – Модульная структура системы голосового ассистента для мессенджера Telegram

### *Реализация модуля распознавания речи*

Распознавание речи (Speech Recognition, SR) — это процесс преобразования устной речи в текст с использованием вычислительных методов**.** Система распознавания речи работает в несколько этапов, начиная с получения аудиосигнала до извлечения текстовой информации. Важно отметить, что распознавание речи включает несколько ключевых этапов:

– Процесс распознавания речи начинается с захвата аудиосигнала с микрофона. На этом этапе важно очистить сигнал от шумов и посторонних звуков с помощью фильтрации шумов. Также необходимо выполнить нормализацию громкости для выравнивания уровня звука и повышения качества распознавания, чтобы сигналы не были слишком тихими или громкими.

– Для эффективного распознавания речи аудиосигнал должен быть преобразован в формат, который легко анализировать алгоритмам. Одним из самых распространенных методов является извлечение мел-частотных кепстральных коэффициентов (MFCC), которые эффективно отображают особенности звука, воспринимаемые человеческим ухом. Также часто используется преобразование Фурье (FFT) для разделения сигнала на частотные компоненты, что помогает выделить важные характеристики.

– После извлечения признаков, для распознавания речи применяются статистические модели и нейронные сети. Одним из классических методов является модель скрытых марковских процессов (HMM), которая оценивает вероятности появления звуковых последовательностей. В современных системах часто используются более сложные подходы, такие как нейронные сети LSTM или Transformer, которые позволяют значительно повысить точность распознавания благодаря способности учитывать контексты и долгосрочные зависимости.

– После того как речь распознана и преобразована в текст, необходимо провести этап постобработки. Включает это в себя коррекцию орфографических и грамматических ошибок, чтобы улучшить качество текста. Также удаляются лишние слова и звуки, такие как "эм" или "ну", которые не несут полезной информации и могут мешать восприятию текста.

```python
def callback(indata, frames, t, status):
    """Callback функция для обработки входящих аудиоданных."""
    global frame_processing_times, total_frames, audio_data
    frame_start_time = time.time()
    q.put(bytes(indata))
    audio_data.extend(np.frombuffer(indata, dtype=np.int16))
    frame_processing_times.append(time.time() - frame_start_time)
    total_frames += 1


1 usage
def recognize(data):
    """Распознает текст на основе переданных данных и возвращает его."""
    return data.strip()


7 usages
def main_voice():
    """Основная функция для инициализации и запуска распознавания речи."""
    global latency_list, recognized_words_count, recognized_texts, phrase_lengths
    with sd.RawInputStream(samplerate=samplerate, blocksize=16000, device=device[0], dtype='int16',
                           channels=1, callback=callback):
        rec = vosk.KaldiRecognizer( *args: model, samplerate)
        print("Начало распознавания речи. Говорите...")
        while True:
            start_latency = time.time()
            data = q.get()
            if rec.AcceptWaveform(data):
                result = rec.Result()
                data = json.loads(result)['text']
                recognized_word = recognize(data)
                recognized_texts.append(recognized_word)
                phrase_lengths.append(len(recognized_word.split()))
                latency_list.append(time.time() - start_latency)
                recognized_words_count += len(recognized_word.split())
                print(f"Распознанный текст: {recognized_word}")
                return recognized_word
```

Рисунок 5 – Реализованный модуль распознавания речи

Для оценки эффективности работы модуля распознавания речи (рис. 5) были использованы несколько ключевых метрик, таких как средняя задержка, время обработки каждого аудио фрейма и точность распознавания. Эти метрики позволяют оценить как производительность системы в реальном времени, так и её способность точно интерпретировать речь. Результаты расчётов метрик помогут выявить потенциальные проблемы и области для улучшения работы модели.

Средняя задержка формула (1). Это время, которое требуется системе для обработки аудио файла и возврата результата. Время задержки — это ключевая метрика для систем, работающих в реальном времени:

$$\text{Latency} = \frac{\sum_{i=1}^{n}((\text{Время завершения} - \text{Время начала})}{n} \quad (1)$$

где n - количество обработанных кадров (итераций).

Точность распознавания формула (2). Измеряет, насколько правильно система преобразует аудио файл в текст. Рассчитывается как отношение правильно распознанных слов к общему количеству слов:

$$\text{Accuracy} = \frac{\text{Количество правильных слов}}{\text{Общее количество слов}} * 100\% \quad (2)$$

Метрика пропускной способности формула (3) измеряет объем обрабатываемых данных за единицу времени. Этот параметр особенно важен для систем, работающих с потоковыми данными:

$$\text{Throughput} = \frac{\text{Количество обработанных кадров}}{\text{Время обработки}} \quad (3)$$

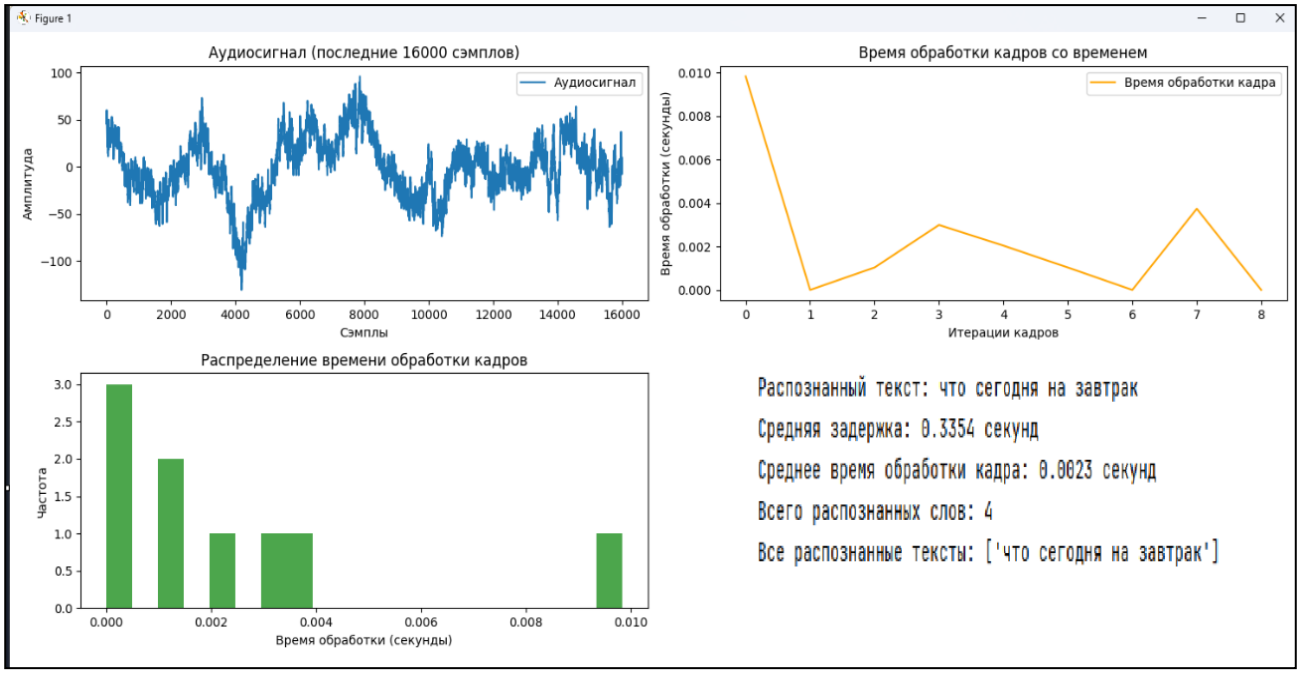



Рисунок 6 - Значения метрик для запроса «Что сегодня на завтрак»

На основе графиков и статистических данных, полученных при анализе работы системы распознавания речи, можно выделить несколько ключевых аспектов, характеризующих её производительность и эффективность (рис. 6).

График амплитуды аудио сигнала демонстрирует достаточно стабильные колебания, что подтверждает отсутствие явных шумов или помех в исходном аудио. Этот результат является положительным индикатором качества записи и подготовки аудио данных для дальнейшей обработки.

График времени обработки кадров, отображающий зависимость времени от интервалов, показывает, что обработка кадров в некоторых случаях занимает более длительное время. Это может быть связано с увеличенной

сложностью анализа определённых участков сигнала. Данный факт указывает на необходимость дополнительной оптимизации алгоритмов обработки в таких случаях, чтобы повысить общую производительность системы.

График распределения времени обработки кадров демонстрирует, что большинство кадров обрабатываются в пределах от 0.002 до 0.01 секунд. Это хорошее распределение, которое указывает на стабильность работы системы при нормальных условиях. Однако, если на некоторых кадрах время обработки значительно превышает эти пределы, это может свидетельствовать о проблемах с конкретными запросами или особенностями аудио файлов, что требует дополнительной диагностики.

Средняя задержка в 0.3554 секунды является вполне приемлемой для большинства голосовых ассистентов, обеспечивая пользователю достаточную скорость отклика. Однако это значение всё же может быть улучшено. Среднее время обработки кадра (0.0623 секунды) также выглядит хорошим, но для критически важных приложений можно стремиться к его уменьшению, что позволит снизить общую задержку системы.

### *Выводы*

В данной статье разработаны и исследованы авторские алгоритмы, а также создан базовый прототип голосового ассистента для мессенджера Telegram, который реализует основные функции, такие как поиск сообщений, вывод контактов и отправка сообщений. Проект является начальной стадией, а в дальнейшем планируется расширение функционала, поддержка нескольких мессенджеров и улучшение безопасности данных.

### *Литература*


1. Сигитова, Н. А. Разработка и использование систем распознавания речи в приложениях для мобильных устройств / Н. А. Сигитова, И. А. Иванов. — М.: Научный мир, 2019. — 210 с.
2. Воронова, В. А. Применение обработки естественного языка в интеллектуальных системах / В. А. Воронова, А. П. Петров // Современные информационные технологии. — 2020. — № 3. — С. 112—118.
3. Романов, М. И. Взаимодействие с API в приложениях на Python / М. И. Романов. — СПб.: БХВ-Петербург, 2018. — 350 с.
4. Петров, С. Ю. Синтез речи в системах голосовых ассистентов [Электронный ресурс] / С. Ю. Петров // Журнал "Технологии искусственного интеллекта". — Электрон. дан. — 2021. — № 5. — С. 45—52. — Режим доступа: https://www.iai-journal.ru/2021/5/45-52.pdf.
5. Бочаров, И. В. Телеграм-боты: Разработка и интеграция / И. В. Бочаров, О. В. Козлов. — М.: РГТУ, 2017. — 295 с.



***Похлёбин П.С., Федяев О.И. Интеграция голосовых технологий в мессенджеры: создание голосового ассистента.*** *Статья посвящена разработке голосового ассистента для повышения эффективности взаимодействия с мессенджерами. Рассматриваются ключевые методы и технологии, применяемые при создании ассистента, а также возможности интеграции интеллектуальных функций для автоматизации задач в мессенджерах. Приводится анализ преимуществ интеллектуального подхода к обработке сообщений и взаимодействию с пользователем. Также затрагиваются технические и этические аспекты внедрения голосовых ассистентов и предлагаются методы повышения их надёжности и точности.*



***Ключевые слова:*** *голосовой ассистент, мессенджеры, интеллектуализация, автоматизация, конфиденциальность, обработка данных, речевые технологии.*



***Pokhlebin P.S., Fedyaev O.I. Integration of voice technologies into messengers: creation of a voice assistant.*** *The article is devoted to the development of a voice assistant to improve the effectiveness of interaction with messengers. The key methods and technologies used to create an assistant are considered, as well as the possibilities of integrating intelligent functions to automate tasks in messengers. The advantages of an intelligent approach to message processing and user interaction are analyzed. The technical and ethical aspects of the introduction of voice assistants are also touched upon and methods for improving their reliability and accuracy are proposed.*



***Keywords:*** *voice assistant, messengers, intellectualization, automation, privacy, data processing, speech technologies.*

УДК 004.93

# ОЦЕНКА СКОРОСТИ ДВИЖЕНИЯ ОБЪЕКТА С ПОМОЩЬЮ НЕЙРОСЕТЕВОЙ МОДЕЛИ YOLO

**А.Р. Муращенко[*1], О.И. Федяев[*2]**

[*1] магистрант, Донецкий национальный технический университет,
sasha.mur01@yandex.ru

[* 2] к.т.н, доцент, Донецкий национальный технический университет,
olegfedyayev@mail.ru, OrcID: 0000-0001-6822-7306, SPIN-код: 7777-3791

***Муращенко А.Р., Федяев О.И. Оценка скорости движения объекта с помощью нейросетевой модели YOLO.*** *В работе рассматривается оценка скорости движения объектов на видеокамере. Показана и проанализирована структура нейросетевой модели YOLO и алгоритма трекинга BotSort. Создана программа для демонстрации работы модели YOLO и алгоритма BotSort. Было проведено тестирование программы на различных видеоматериалах с разными условиями. Выделены и проанализированы достоинства и недостатки новейшей версии модели YOLO.*

***Ключевые слова:*** *нейросеть, распознавание, оценка, трекинг, модель YOLO, обучение.*

## *Введение*

Компьютерное зрение – область искусственного интеллекта, которая занимается созданием систем, позволяющих анализировать визуальную информацию.

Задача компьютерного зрения — не только идентифицировать отдельные объекты, присутствующие на картинке или видеоизображении, но и выделять их характеристики, например, поведение найденного объекта.

По данным Всемирной организации здравоохранения (ВОЗ), ежегодно в результате дорожно-транспортных происшествий из-за превышения скорости погибает около 1,19 миллиона человек. Кроме того, еще от 20 до 50 миллионов получают несмертельные травмы, многие из которых приводят к инвалидности. Важность безопасности дорожного движения невозможно переоценить, особенно когда оценка скорости помогает предотвратить аварии, спасает жизни и сохраняет наши дороги безопасными и эффективными [1].

Оценка скорости - это процесс вычисления скорости движения объекта в заданном контексте, который часто используется в приложениях компьютерного зрения. С помощью нейронной сети YOLO можно рассчитать скорость объекта, используя отслеживание объекта наряду с данными о расстоянии и времени, что очень важно для таких задач, как движение и наблюдение. Точность оценки скорости напрямую влияет на эффективность и надежность различных приложений, что делает её ключевым компонентом в развитии интеллектуальных систем и процессов принятия решений в реальном времени [2].

Оценка скорости объекта состоит из двух задач, а именно – распознавания объектов и отслеживания объектов.

Распознавание объектов – это метод компьютерного зрения для идентификации объектов на изображениях или видео. Трекинг – определение местоположения объекта (нескольких объектов) во времени. В работе алгоритмы отслеживания разделены на четыре основные категории: отслеживание областей, отслеживание по активному контуру, отслеживание по характерным признакам, отслеживание по модели [3].

Цель данной работы заключается в построении системы для оценки скорости движения объектов с помощью модели нейронной сети YOLO и алгоритма трекинга BotSort.

## *Критический анализ работ и методов по оценке динамики объектов*

В данный момент насчитывается не так много публикаций по этой теме. В статье «Maritime Search and Rescue Missions with Aerial Images: A Survey», которая опубликована 13 ноября 2024 года, представлен всесторонний анализ современного состояния методик обнаружения людей в море по аэроснимкам с особым акцентом на операции SAR [4]. В данной статье идёт сравнение современных методов трекинга и обнаружения объектов. Однако в данной статье используются лишь статичные изображения для обнаружения живых объектов. Также в публикации использовались для сравнения не самые новые технологии и модели.

В публикации «Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive

Benchmark Study of YOLO11 and Its Predecessors», которая опубликована 31 октября 2024 года, описаны архитектуры нейронной модели для распознавания YOLO [5]. Автор подробно описывает различие архитектур разных версий, а также проблемы в этих структурах, однако в данной публикации не показывает результат работы своего использования модели YOLO. Также автор не упоминает новые функции, которые появляются в новых версиях модели YOLO.

В публикации «YOLOV11: AN OVERVIEW OF THE KEY ARCHITECTURAL ENHANCEMENTS», которая написана 24 октября 2024 года, проводится краткое описание новейшей модели YOLO [6]. В данной работе лишь словесно описывается изменение структуры новой версии. Автор не указывает, как можно использовать данную версию модели, а также не показывает результат использования модели на личном примере.

Следовательно, существующие публикации недостаточно описывают функции и алгоритмы работы современных моделей и методов трекинга для распознавания объектов и оценке динамики.

Стоит отметить и преимущества оценки скорости движения с помощью модели YOLO11.

Можно выделить некоторые задачи, которые поможет решить знание скорости движения объектов:

1. Эффективное управление дорожным движением: точная оценка скорости помогает управлять транспортным потоком, повышает безопасность и уменьшает заторы на дорогах.

2. Точная автономная навигация: в автономных системах, таких как самоуправляемые автомобили, надежная оценка скорости обеспечивает безопасную и точную навигацию транспортного средства.

3. Повышенная безопасность наблюдения: оценка скорости в аналитике наблюдения помогает выявить необычное поведение или потенциальные угрозы, повышая эффективность мер безопасности. Например, с помощью оценки скорости можно отслеживать резкое движение учащихся в классе для повышения дисциплины.

Данное направление является очень перспективным, так как может улучшить возможности дистанционного наблюдения и автоматического анализа динамики объектов.

Разработчики модели постоянно улучшают функции модели и добавляют новые для решения возникающих проблем в компьютерном зрении.

### *Структура и функционирование нейросетевой модели YOLO11*

Ultralytics YOLO11 — это передовая, современная модель, которая развивает успех предыдущих версий YOLO и представляет новые функции и усовершенствования для дальнейшего повышения производительности и гибкости. YOLO11 отличается высокой скоростью, точностью и простотой использования, что делает её отличным выбором для широкого спектра задач по обнаружению и отслеживанию объектов, сегментации объектов, классификации изображений и оценке позы, а также ориентированное обнаружение объектов.

Оценка позы — это задача, которая заключается в определении местоположения определённых точек на изображении, обычно называемых ключевыми точками. Ключевые точки могут представлять собой различные части объекта, такие как суставы, ориентиры или другие отличительные особенности. Расположение ключевых точек обычно представляется в виде набора двумерных [x, y] или трехмерных [x, y, visible] координат [7].

Ориентированное обнаружение объектов идёт на шаг дальше, чем обнаружение объектов, и вводит дополнительный угол для более точного определения местоположения объектов на изображении. Результатом работы ориентированного обнаружения объектов является набор повёрнутых ограничительных рамок, которые точно охватывают объекты на изображении, а также метки классов и баллы доверия для каждой рамки. Обнаружение объектов - хороший выбор, когда вам нужно определить интересующие вас объекты в сцене, но не нужно точно знать, где находится объект или его точную форму.

Архитектура YOLO11 похожа на архитектуру YOLO8, но с некоторыми различиями:

– Transformer Backbone: усиливает способность модели захватывать глобальный контекст.

– Dynamic Head Design: это позволяет YOLOv11 адаптироваться в зависимости от сложности изображения, оптимизируя распределение ресурсов для более быстрой и эффективной обработки.

– Обучение без NMS: YOLOv11 заменяет не-максимальное подавление (NMS) более эффективным алгоритмом, сокращая время вывода при сохранении точности.

– Двойное присвоение меток: улучшает обнаружение перекрывающихся и плотно упакованных объектов благодаря использованию подходов присвоения меток «один к одному» и «один ко многим».

– Large Kernel Convolutions: обеспечивает лучшее извлечение признаков при меньших вычислительных ресурсах, повышая общую производительность модели.

Такая архитектура позволяет YOLO11 эффективно работать на системах высокого класса и краевых устройствах, например мобильных телефонах. Также существует возможность выбрать свой вариант трекера. На данный момент присутствует поддержка алгоритмов трекинга BoT-SORT и ByteTrack. Подробная схема архитектуры нейронной модели YOLO11 показана на рисунке 1.

На рисунке 2 представлена краткая схема работы алгоритма модели YOLO.

Краткое опишем алгоритм работы YOLO. Перед началом обучения нейронной сети картинке изменяют размер на 416x416, чтобы её можно было эффективно делить на разные размеры для ускорения обучения. Картинку делят на клетки размером AxA. Каждая клетка является «якорем», к которому прикрепляется ограничивающая рамка. То есть вокруг клетки рисуются несколько прямоугольников для определения объекта, и их позиции, ширина и высота вычисляются относительно центра этой клетки. Картинка из датасета прогоняется через нейронную сеть. Задача YOLO – максимально точно предсказать параметры, чтобы максимально точно определять объект на картинке. Определяется балл доверия (confidence score) для каждой предсказанной ограничивающей рамки, который является фильтром для того, чтобы отсеять совсем неточные предсказания. Используется техника не-максимального подавления (non-max suppression), чтобы отфильтровать ограничивающие рамки таким образом, чтобы для одного объекта была только одна предсказанная ограничивающая рамка.

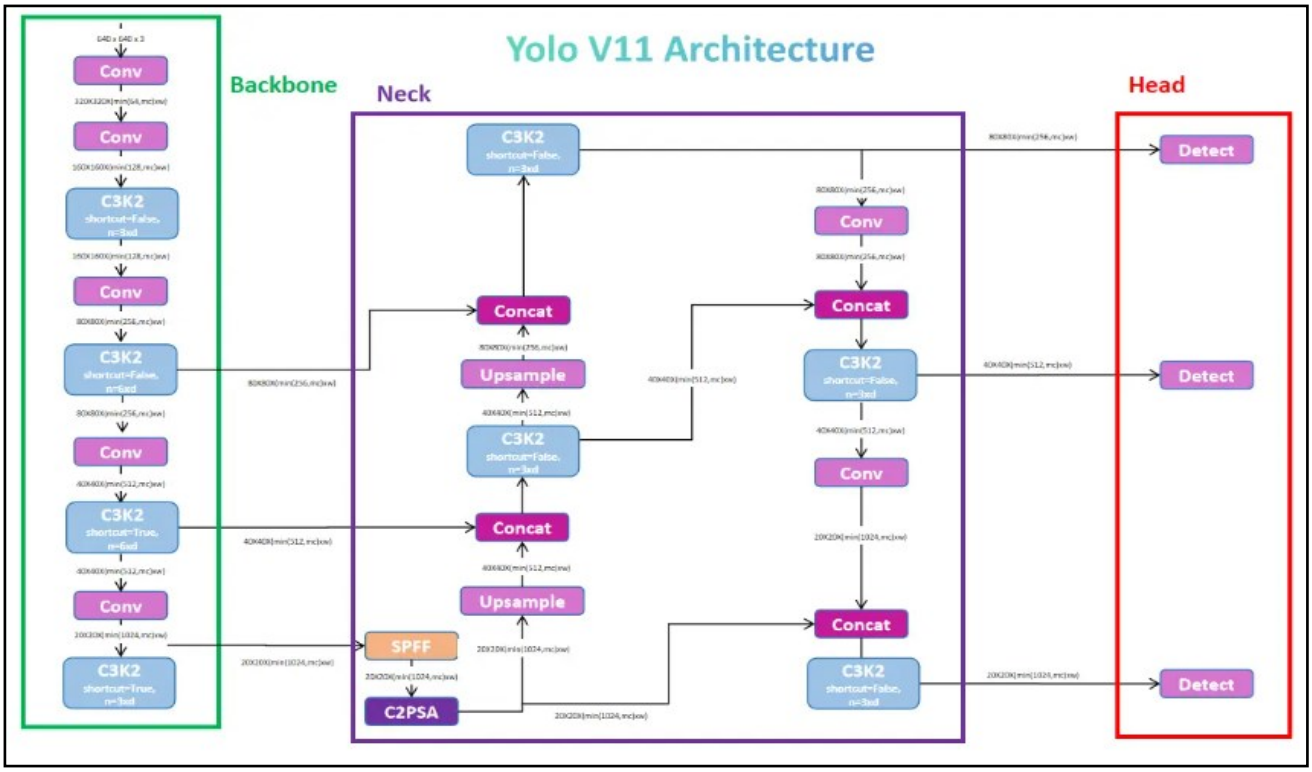



Рисунок 1 – Архитектура нейронной модели YOLO11

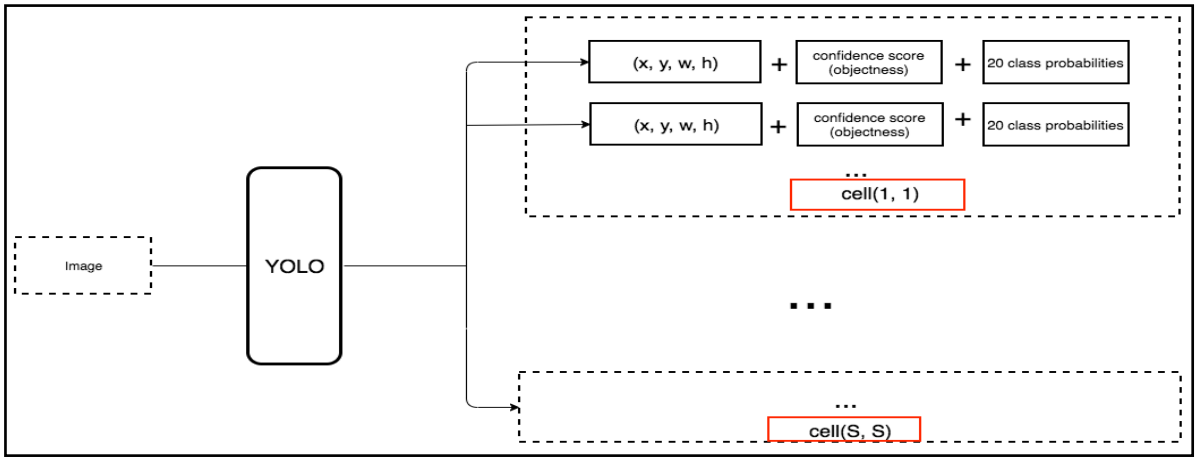



Рисунок 2 – Алгоритм работы модели YOLO

### *Описание работы трекинга с помощью алгоритма BotSort*

В последнее время в ряде исследований отказались от информации о внешности и стали полагаться только на модели движения. Большинство современных алгоритмов отслеживания и обнаружения основаны на моделях движения. В последнее время для моделирования движения объекта чаще всего используется знаменитый фильтр Калмана с предположением о модели постоянной скорости.

Различие и повторная идентификация (ReID) объектов по глубоким признакам внешнего вида также стала популярной, но во многих случаях оказывается неэффективной, особенно в многолюдных сценах, из-за частичной окклюзии людей. Отдельные трекеры на основе внешнего вида обрезают блоки обнаружения кадров и извлекают глубокие признаки внешнего вида с помощью дополнительной глубокой нейронной сети. Они обладают передовыми методами обучения, но требуют больших вычислительных затрат. Недавно было предложено несколько совместных трекеров, которые совместно обучают обнаружению и некоторым другим компонентам, например, моделям движения, встраивания и ассоциации. Основным преимуществом этих трекеров является низкая вычислительная стоимость и сопоставимая производительность. В последнее время в ряде исследований отказались от информации о внешности и стали использовать только высокопроизводительные детекторы и информацию о движении, что позволило добиться высокой скорости работы и самой современной производительности. В частности, ByteTrack, который использует ячейки обнаружения с низкой оценкой, сопоставляя обнаружения с высокой степенью уверенности, а затем ещё одно объединение с обнаружениями с низкой степенью уверенности. Рисунок работы трекера BotSort представлен на рисунке 3.

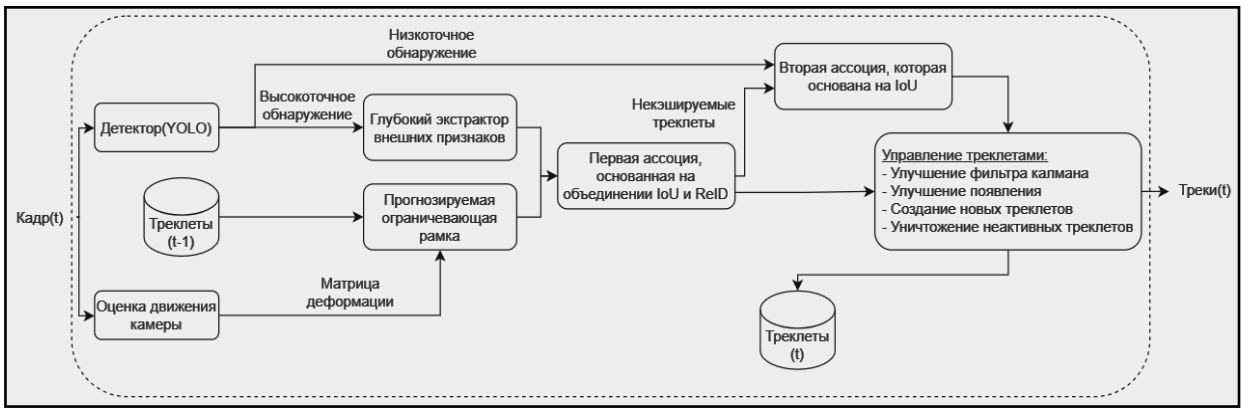



Рисунок 3 – Схема работы алгоритма BotSort для трекинга объектов в YOLO

Данный метод и его компоненты могут быть легко интегрированы в другие трекеры для отслеживания-обнаружения.

### *Обучение модели YOLO11 на распознавание типа объектов в кадре*

Обучение модели глубокого обучения заключается в подаче ей данных и настройке её параметров таким образом, чтобы она могла делать точные прогнозы.

Для каждой эпохи он показывает сводку по фазам обучения и проверки: строки 1 и 2 показывают результаты фазы обучения, а строки 3 и 4 - результаты фазы проверки для каждой эпохи.

```
Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
16/30      7.62G     0.7751      0.702      1.078          5        640: 100%|██████████| 101/101 [01:10<00:00,  1.43it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 15/15 [00:09<00:00,  1.59it/s]
             all        461        504      0.905       0.94      0.949      0.788

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
17/30      7.63G     0.7473     0.6422      1.071          3        640: 100%|██████████| 101/101 [01:11<00:00,  1.41it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 15/15 [00:09<00:00,  1.58it/s]
             all        461        504      0.905      0.942      0.951      0.788

Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
18/30      7.58G      0.742     0.6211      1.064          6        640: 100%|██████████| 101/101 [01:10<00:00,  1.43it/s]
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 15/15 [00:09<00:00,  1.57it/s]
             all        461        504      0.928      0.921      0.955      0.791
```

Рисунок 4 – Обучение модели YOLO

Этап обучения включает в себя расчёт количества ошибок в функции потерь, поэтому наиболее ценными метриками здесь являются box_loss и cls_loss, box_loss показывает количество ошибок в обнаруженных ограничительных коробках. cls_loss показывает количество ошибок в обнаруженных классах объектов.

Наиболее ценной метрикой качества является mAP50-95, т. е. средняя точность (Mean Average Precision). Если модель обучается и совершенствуется, точность должна расти от эпохи к эпохе. На предыдущем скриншоте можно увидеть, что она медленно растёт: 0,788, 0,788, 0,791.

### *Результаты экспериментов по оценке скорости движения разных объектов*

Для экспериментов была создана тестовая программа для распознавания, трекинга, оценки скорости объектов с помощью YOLOv11 и BotSort. Функциональная схема программы показана на рисунке 5.

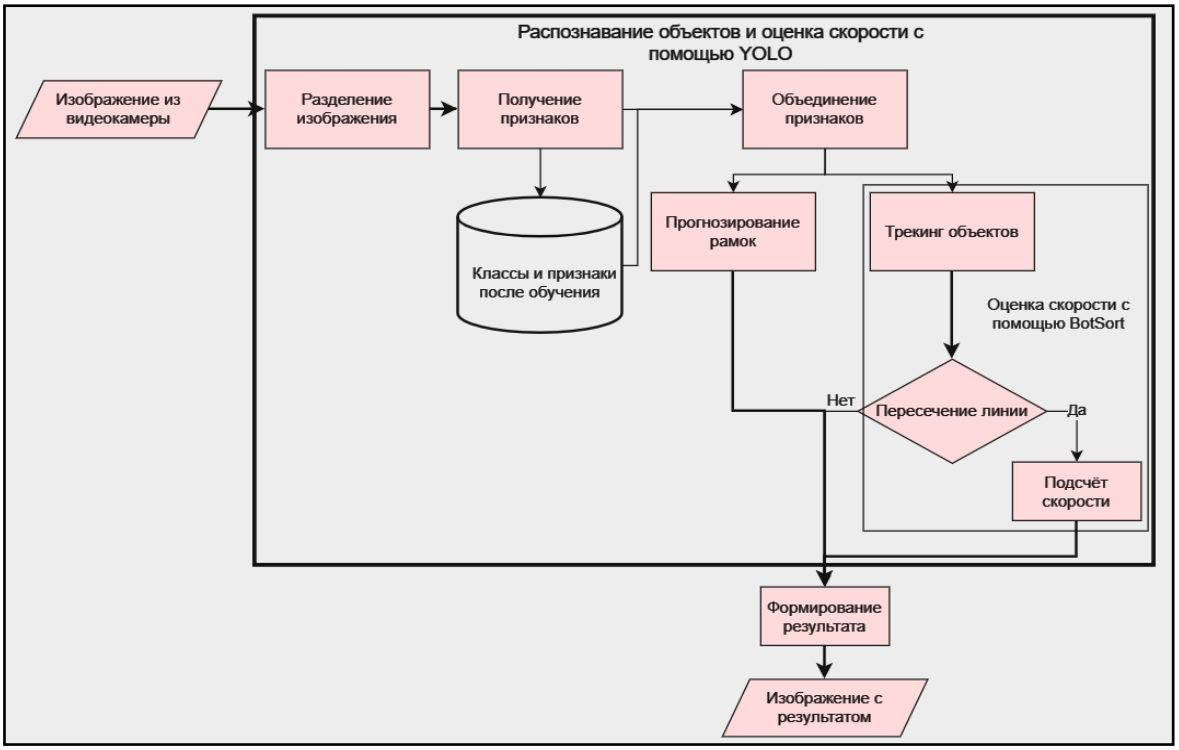



Рисунок 5 – Функциональная схема системы распознавания объектов на изображении

Тестирование проводилось на следующей технической конфигурации:

– процессор: intel i3-1005G1(8 потоков, 2 ядра);

– видеокарта: nVidia GeForce MX330(2 гб графической памяти);

– оперативная память: 8 ГБ;

При тестировании была поставлена линия, которая служит областью для подсчёта скорости. Линия необходима для существенного уменьшения нагрузки на процессор. Модель будет подсчитывать, и показывать скорость объекта при её пересечении. Количество линий может быть любое. При необходимости существует возможность показывать одновременно класс и скорость на одном объекте.

Результат экспериментов  по оценке скорости движения представлен на рисунках 6 и 7.

Распознавание и оценка первого результата заняли 31 миллисекунду при использовании GPU и YOLO11, что на 50% быстрее, чем с помощью YOLO8.

Распознавание и оценка второго результата заняли 15,6 миллисекунд при использовании GPU.

На рисунке 6 показана машина, которая проехала линию со скоростью 22 км в час. объекты двигались горизонтально изображению. Видео работало на частоте 30 кадров в секунду. Если сравнить с оригинальным видео, то это точный результат

На рисунке 7 показаны бегущие дети, программа которым показала 14 км в час и 10 км в час. Объекты на втором опыте двигались к камере. Видео работало на частоте 60 кадров в секунду. Результат стал менее точным, так как на оценку повлиял ракурс и количество кадров в секунду у видео.

Из опытов можно сделать вывод, что слишком высокая частота кадров и неправильная постановка камеры в видео приводит к менее точному результату.

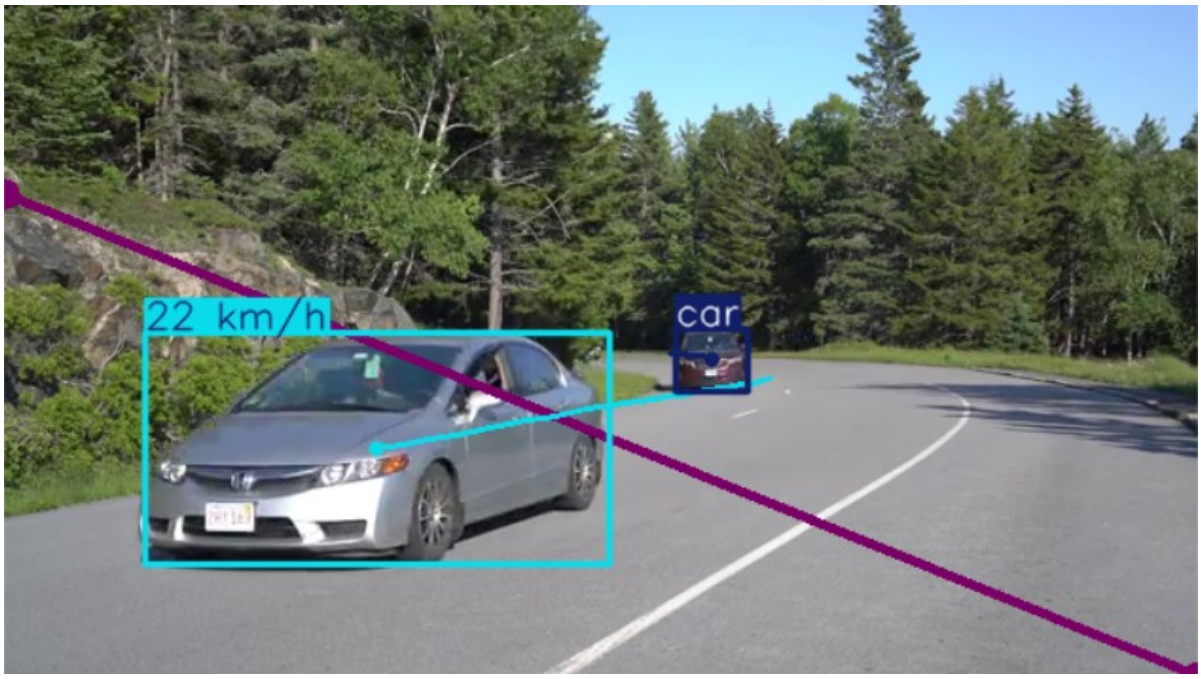



Рисунок 6 – Результат работы оценки скорости на примере едущего автомобиля

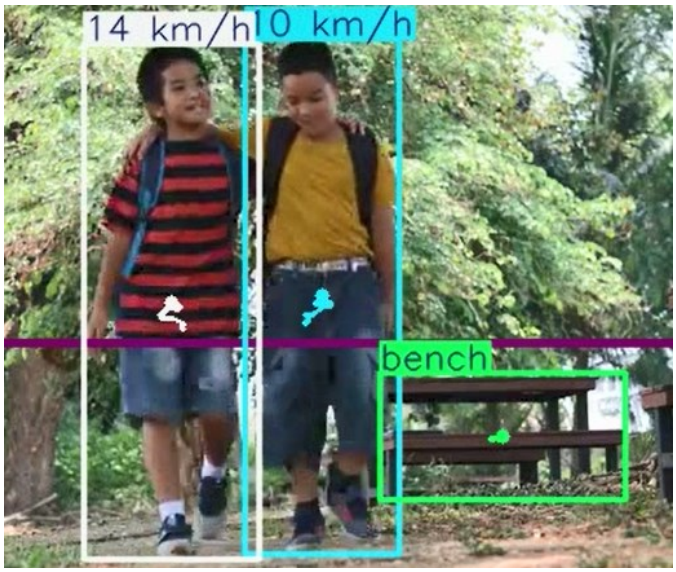



Рисунок 7 - Результат работы оценки скорости на примере бегущего ребёнка

По результату видно, что модель запомнила объекты конкретного класса и поставила им номер на первом кадре видеопотока. Затем на втором кадре видеопотока модель показала все объекты, которые она запомнила.

Достоинства новой версии модели YOLO для оценки скорости:

– низкая нагрузка на систему;
– удобная настройка;
– гибкость;
– скорость обнаружения.

Недостатки модели:

– низкая точность при видеоизображении с помехами или плохими погодными условиями;

– при оценке скорости большого количества объектов требуются более сильные мощности графического процессора;

– прогнозируемая скорость может меняться в зависимости от кадров в секунду.

По результату тестирования можно сделать вывод, что модель YOLO с алгоритмом BotSort хорошо справляется с задачей оценки скорости движения объектов, однако всё ещё требует доработки.

### *Заключение*

В данной работе был проведён анализ системы оценки скорости движения и трекинга с помощью модели YOLO11 и BotSort. Были описаны архитектуры работы модели YOLO11 и алгоритма трекинга BotSort. Описаны новые технические возможности новой версии YOLO. Показано как работает обучение в модели.

С помощью созданной тестовой программы проведены опыты с разными видеоизображениями при различных условиях. Исследование проводилось на примере движения автомобиля и движении двух людей.

Модель показала себя достойно при оценке скорости движения разных объектов. Скорость обнаружения новой модели версия 11 модели YOLO стала гораздо быстрее, чем предыдущие её версии. Однако алгоритм трекинга Bot-Sort ещё требует доработки.

### *Литература*


1. Дорожно-транспортный травматизм. [Электронный ресурс] // Всемирная организация здравоохранения. – Электрон. дан. – 2023. – Режим доступа: https://www.who.int/ru/news-room/fact-sheets/detail/road-traffic-injuries. - Загл. с экрана.
2. Ultralytics YOLOv8 Для оценки скорости в проектах по компьютерному зрению [Электронный ресурс] / Абирами Вина // Блог Ultralytics. – Электрон. дан. – 2024. – Режим доступа: https://docs.ultralytics.com/ru/guides/speed-estimation/. - Загл. с экрана.
3. What is Object Tracking in Computer Vision? [Электронный ресурс] // Roboflow. – Электрон. дан. - 2022. – Режим доступа: https://blog.roboflow.com/what-is-object-tracking-computer-vision/#:~:text=Object%20tracking%20is%20a%20computer,ball%20is%20in%20a%20photo. - Загл. с экрана.
4. Maritime Search and Rescue Missions with Aerial Images: A Survey [Электронный ресурс] / Juan P. Martinez-Esteso // ArXiv – Электрон. дан. - 2024. – Режим доступа: https://arxiv.org/pdf/2411.07649. - Загл. с экрана.
5. Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors [Электронный ресурс] / Nidhal Jegham // ArXiv. – Электрон. дан. - 2024. – Режим доступа: https://arxiv.org/pdf/2411.00201. - Загл. с экрана.
6. YOLOV11: AN OVERVIEW OF THE KEY ARCHITECTURAL ENHANCEMENTS [Электронный ресурс] / Rahima Khanam and Muhammad Hussain // ArXiv. – Электрон. дан. - 2024. – Режим доступа: https://arxiv.org/pdf/2410.17725. - Загл. с экрана.
7. Оценка позы [Электронный ресурс] // Ultralytics YOLO docs. – Электрон. дан. – 2023. – Режим доступа: https://docs.ultralytics.com/ru/tasks/pose/. - Загл. с экрана.



***Муращенко А.Р., Федяев О.И. Оценка скорости движения объекта с помощью нейросетевой модели YOLO.*** *В работе рассматривается оценка скорости движения объектов на видеокамере. Показана и проанализирована структура нейросетевой модели YOLO и алгоритма трекинга BotSort. Создана программа для демонстрации работы модели YOLO и алгоритма BotSort. Было проведено тестирование программы на различных видеоматериалах с разными условиями. Выделены и проанализированы достоинства и недостатки новейшей версии модели YOLO.*



***Ключевые слова:*** *нейросеть, распознавание, оценка, трекинг, модель YOLO, обучение.*



***Murashchenko A.R., Fedyaev O.I. Estimation of object motion speed with the help of neural network model YOLO.*** *The article deals with estimation of speed of movement of objects on video camera. The structure of the YOLO model and the BotSort tracking algorithm is presented and analysed. A program has been developed to demonstrate the operation of the YOLO model and the BotSort algorithm. The programme has been tested on different video material under different conditions. The advantages and disadvantages of the latest version of the YOLO model were highlighted and analyzed.*



***Key words:*** *neural network, recognition, estimation, tracking, YOLO model, training.*

УДК 004.89

# Анализ качества русскоязычного репозитория клинических данных

**Н.И. Петрушан[*1], А.А. Филиппов[*2]**

[*1] аспирант, ассистент кафедры "Информационные системы",
Ульяновский Государственный Технический Университет, г. Ульяновск, Россия
e-mail: nikita@petrushan.ru

[*2] к.т.н., доцент, доцент кафедры "Информационные системы",
Ульяновский Государственный Технический Университет, г. Ульяновск, Россия
e-mail: al.filippov@ulstu.ru

***Петрушан Н.И., Филиппов А.А. Анализ качества репозитория клинических данных.*** *Статья представляет описание результатов анализа качества амбулаторных и стационарных данных из русскоязычного репозитория клинических данных. Определены и систематизированы проблемы в категориальных, текстовых и числовых данных объективных обследований, предложен алгоритм устранения полученных ошибок. Оценена применимость репозитория для решения аналитических задач с применением методов искусственного интеллекта.*

***Ключевые слова:*** *репозиторий, искусственный интеллект, анализ качества данных, ошибки данных, алгоритм устранения ошибок, решение аналитических задач.*

## *Введение*

Для большого числа заболеваний существуют клинические рекомендации (протоколы лечения), которые представляют собой многостраничные текстовые документы. Для формирования плана лечения пациента врачу необходимо учитывать большое количество факторов: история болезни пациента, симптомы, результаты анализов, множество предполагаемых заболеваний и множество их протоколов лечения. Постановка точного диагноза в таких условиях требует от врача значительных когнитивных усилий, что влечет за собой со временем снижение качества поставленных диагнозов и ухудшение здоровья пациентов.

В последние годы с активным развитием медицины и информационных технологий широкое распространение получили интеллектуальные методы и средства анализа медицинских данных, с помощью которых исследователи могут решать ряд задач [1, 2]: разработка медицинских систем поддержки принятия решений (СППР), аналитического программного обеспечения (АПО), программ лечения и прочих, в том числе с использованием методов и алгоритмов искусственного интеллекта. Использование таких методов и алгоритмов как машинное обучение, предъявляет определенные требования к качеству данных, которое будет использовано для решения задачи [3, 4].

Согласно ГОСТ Р ИСО 8000-2-2019 «Качество данных» под качеством данных понимается «степень, с которой набор характеристик, присущих данным, отвечает требованиям» [5] а под данными понимается «интерпретируемое представление информации, в соответствующей форме, удобной для передачи, интерпретации и обработки» [5], из чего можно сделать вывод, что качество данных определяется исключительно требованиями той или иной задачи. При этом обязательно должна быть возможность удобно передавать, интерпретировать и обрабатывать данные.

Также в стандарте ГОСТ Р «Информационные технологии. Искусственный интеллект. Качество данных для аналитики и машинного обучения. Часть 2. Показатели качества данных» [6] (на момент публикации носит информационный характер) определены следующие характеристики качества данных:

– Переносимость – возможность легко переносить данные из одной системы в другую без необходимости повторного ввода данных.

– Понятность – возможности пользователей читать и интерпретировать данные.

– Возможность аудита – характеристика набора данных, состоящей в том, что весь набор данных или его часть прошел аудит или что данные доступны соответствующим заинтересованным сторонам для целей проведения аудита.

– Идентифицируемость – возможность идентифицировать субъекта на основе данных о нем.

Стоит заметить, что одной из ключевых особенностей медицинских данных является требование

деидентификации записей [7], однако, вышеуказанный проект стандарта предполагает такие юридические требования и позволяет гибко взаимодействовать с данной характеристикой.

Актуальность с точки зрения правильного возраста по отношению к использованию данных.

В рамках данной статьи представлены результаты анализа качества данных репозитория клинических данных Сибирского Государственного Медицинского Университета, «SibMed Clinical Data Repository» [8, 9]. Согласно [9] в базе находятся данные о 1 427 210 записях, 275 512 случаях, 67 020 выписанных эпикризах и 510 453 лабораторных исследованиях.

### *Критерий переносимости*

Критерий переносимости предусматривает простое и понятное перемещение данных между репозиторием медицинских данных и интеллектуальной системой (СППР, АПО и др.). В анализируемом репозитории существует возможность формирования запросов посредством специализированного «Конструктора запросов».

Например, следующий запрос позволяет получить информацию о пациентах из источника данных «СибГМУ»:

$$Search\colon [patients.data\_source\_id] \ = \ 1$$

При выгрузке из репозитория абсолютно всех записей результирующий табличный файл будет содержать только 637 893 записей, что значительно меньше предполагаемых данных. Однако ключевая проблема данного репозитория заключается в том, что сам результирующий файл не содержит каких-либо словарей для понимания смысла и сопоставления записей из полей таблиц с сущностями проблемной области. В качестве примера рассмотрим следующую запись из таблицы public.events репозитория, которая ссылается на запись в таблице-словаре dict_icd10 репозитория. Таблица-словаре dict_icd10 содержит описание диагнозов по международному классификатору болезней 10-го пересмотра (МКБ-10):

$$\{"diagnosis"\colon \{"main\_icd10\_id"\colon 9986.000000\}\}$$

В полученном из репозитория файле отсутствуют словари. Получить словари можно только непосредственно обратившись к разработчикам репозитория. На рисунке 1 представлена структура взаимодействия между таблицей public.events и словарями репозитория.

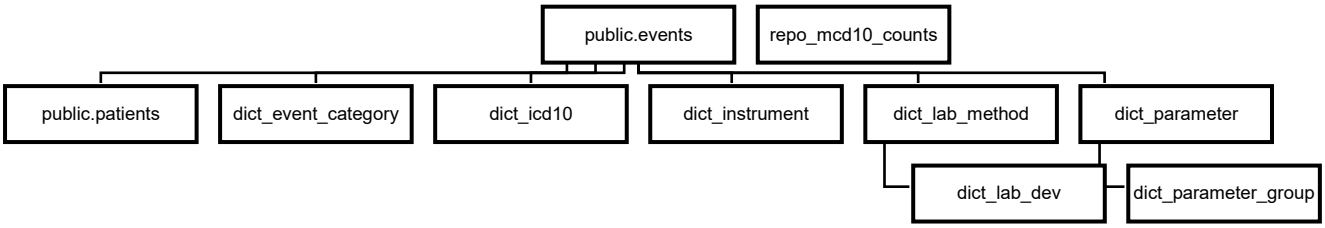



Рисунок 1. Структура взаимодействия таблиц репозитория.

Таким образом, можно отметить, что критерий переносимости в целом выполняется успешно и у исследователя есть возможность получить всю необходимую информацию для решения поставленных задач.

### *Критерий понятности*

Под критерием понятности следует понимать возможность читать и интерпретировать данные. Если проанализировать полученные из репозитория данные, то можно сделать вывод о наличии следующих типов записей:

general_exam – общий осмотр;

lab_results – результаты лабораторных методов исследований;

instrumental_diagnosis – инструментальная диагностика;

diagnosis – диагноз;

hospital_diary – дневник госпитализации.

Следует отметить, что нередко встречаются записи комбинированного типа, сочетающие в себе сразу некоторое количество типов записей, например, «diagnosis;general_exam;hospital_diary». Это затрудняет интерпретацию, т. к. требуется дополнительная работа по анализу, сортировке и систематизации таких записей. Все указанные типы записей также содержат ссылку на идентификатор пациента, дату события и ссылку на

идентификатор случая, на основе которых можно сгруппировать все события конкретного пациента в рамках одного обращения.

Следующее, на что стоит обратить внимание для определения качества данных через критерий понятности – значения текстовых и числовых данных объективных обследований. Отметим следующие проблемы:

1) **Важность деидентификации медицинских данных** в соответствии с законодательством [7], что в конечном итоге влияет с одной стороны на количественное и качественное наполнение соответствующей базы данных, а с другой – на специфические требования к специалисту, который занимается наполнением такой базы.

2) **Несовершенство архитектуры** может привести к ситуациям, при которых наполнение базы данных может быть осуществлено только при постоянном совместном взаимодействии технических и медицинских специалистов. Как следствие, количество репозиториев с такой специализированной информацией не велико.

3) **Ошибки медицинских специалистов**, осуществляющих и регистрирующих первичный осмотр. Из-за этого данные могут оказаться непригодными для их непосредственного анализа и классификации автоматизированными средствами. При разборе таких записей вручную можно выявить следующие основные виды ошибок:

- Речевые ошибки, подразделяющиеся на [8]:
  - Словообразовательные. Данные ошибки характеризуют собой создание новых слов тогда, когда уже есть общеупотребительные (в том числе и медицинские) термины, например: «Кожные покровы влажные, акнтозис нигриканс», вместо «Кожные покровы влажные, акантоз чёрный» или «Кожные покровы влажные, acanthosis nigricans».
  - Синтаксические. Чаще всего такие ошибки происходят из-за неправильного построения словосочетаний, например: «Периодически сладкоежка», вместо «Периодически злоупотребляет сладким».
  - Стилистические. Ошибки данного типа характерны использованием неуместных слов и выражений, например: «Семейный анамнез: У дядьки ожирение», вместо «Семейный анамнез: у дяди ожирение».
- Пунктуационные ошибки. В отличие от словообразовательных ошибок, которые сравнительно легко выявляются и обрабатываются, пунктуационные ошибки могут остаться незамеченными и создать ложное представление о ситуации при анализе. Например, запись «Вредные привычки: курит редко сладкоежка» может означать что пациент либо редко курит, либо редко злоупотребляет сладким. Так при серьезном информационном значении данной фразы, она может быть абсолютно некорректно разобрана, создав ложное впечатление о вредных привычках в целом.
- Орфографические ошибки, т. е. ошибки связанные с неправильным написанием слова, в том числе с использованием латинских букв. Например: «Acantosis nigricans», вместо «Acanthosis nigricans».

В большинстве записей сочетаются сразу несколько типов ошибок, но абсолютным лидером являются именно пунктуационные ошибки, которые встречаются в рассматриваемом репозитории примерно в каждой второй записи «Общего осмотра».

4) **Изначально ошибочные данные**. Специфическая проблема, встречающаяся в основной таблице public.events. При анализе таких данных можно сделать предложение, что они были внесены для отладки работоспособности репозитория, но по какой-то причине остались существовать уже после запуска хранилища данных. Примером таких данных является запись: «ййй ;;;».

5) **Изначально избыточные данные** обнаруживаются при анализе цепочек событий у конкретного пациента в течение всей длительности одного случая и характеризуется дублированием тех или иных типов событий либо с абсолютно идентичными данными, либо с частичным дублированием уже имеющейся информации (таблица 1).

Таблица 1. Фрагмент с ошибкой изначально избыточных данных.

| events.patient_id | events.event_data |
|---|---|
| **214504** | {"diagnosis": {"main_icd10_id": 9981.000000}} |
| **214504** | {"diagnosis": {"main_icd10_id": 9981.000000}} |

Как можно убедиться, необходимо улучшать критерий понятности данного репозитория. Вышеперечисленные проблемы приводят к тому, что перед непосредственным использованием данных репозитория для задач, связанных с использованием технологий искусственного интеллекта, данные необходимо соответствующим образцом подготовить [10] и произвести работы по нормализации и стандартизации данных.

### *Критерий возможности аудита*

Перед аудитом данных анализируемого репозитория ставятся прежде всего задачи повышения

достоверности амбулаторных и стационарных данных и их соответствия требованиям законодательства. Предполагается, что для выполнения данной характеристики качества данных необходимо воспользоваться сторонней системой, которая автоматизировано выявит недостоверные данные. Прежде всего, в рамках исследования данной характеристики необходимо проверять значения такого типа записи как «результаты лабораторных методов исследований», так как данный тип записей содержит хорошо структурированные данные в формате json и, соответственно, может быть разобран на отдельные составляющие. Также для выявления достоверности и для работы с параклиническими данными можно использовать ГОСТ Р ИСО 53022.3-2008 «Технологии лабораторные клинические. Требования к качеству клинических лабораторных исследований. Часть 3. Правила оценки клинической информативности лабораторных тестов» [11] для проверки соответствия значений записей такого типа референтным интервалам.

Из вышесказанного следует, что аудит является возможным и, соответственно, данный критерий оценки качества данных в целом выполняется.

### *Критерий деидентифицируемости*

Как было замечено ранее, особенность исследуемой области знаний, предполагает деидентификацию данных пациентов [7]. Однако в целом эту задачу обеспечивает разработчик репозитория, т. е. СибГМУ, обрабатывая медицинские данные таким образом, чтобы они не содержали защищённые медицинские сведения о пациентах. Между тем, взаимодействуя с обезличенными, деидентифицированными медицинскими данными, стоит осознавать, что длительный сбор, сортировка и сочетание такой информации может дать весьма полное представление о личности пациента. Под такой информацией может пониматься:

история болезни пациента, диагнозы, анамнезы, эпикризы;

результаты лабораторных анализов;

клинические исследования;

данные первичных осмотров;

дополнительные сведения об образе жизни (злоупотребление алкоголем, наркотическими веществами, сладким; малоподвижный образ жизни; опасные условия производства и т. д.).

Исследователю важно понимать, к чему может привести неправильное и нерациональное использование такой информации и принимать меры для минимизации возможностей идентификации личности пациента.

Таким образом, критерий деиндентифицируемости можно считать соответствующим требованию о качественных данных.

### *Критерий актуальности*

Все даты в представленном репозитории проходят через процедуру деидентификации и округляются до первого дня месяца, соответствующего произошедшему событию. Такая процедура способствует появлению записей, датированных одной датой, но для правильного построения цепочки событий можно воспользоваться порядковым номером события. К сожалению, в связи с потерей информации о точных датах пропадает возможность установить весьма важные для исследования данные о временных интервалах, связанных с применением препаратов. Например, начало приёма, первые улучшения после приёма, интервал замены препарата и т. д.

С другой стороны, следует принять тот факт, что первостепенное значение в репозитории имеет принцип деидентификации данных. А отсутствие той или иной ключевой информации хоть и составляет определенное неудобство и лишает исследователя возможности получить определенные выводы, но при этом позволяет сохранить в тайне данные, по которым можно было бы получить более широкое представление о личности.

Первые записи, которые можно найти в представленном репозитории, датируются апрелем 2019 года, к концу 2019 года насчитывается всего 1343 записи. Динамика распределения данных по временной оси представлена на рисунках 2 и 3.

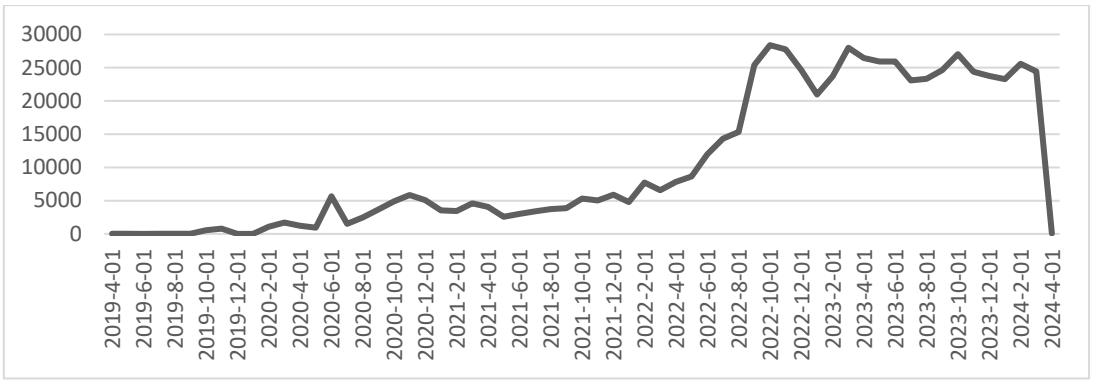



Рисунок 2. Количество записей добавляемых ежемесячно.

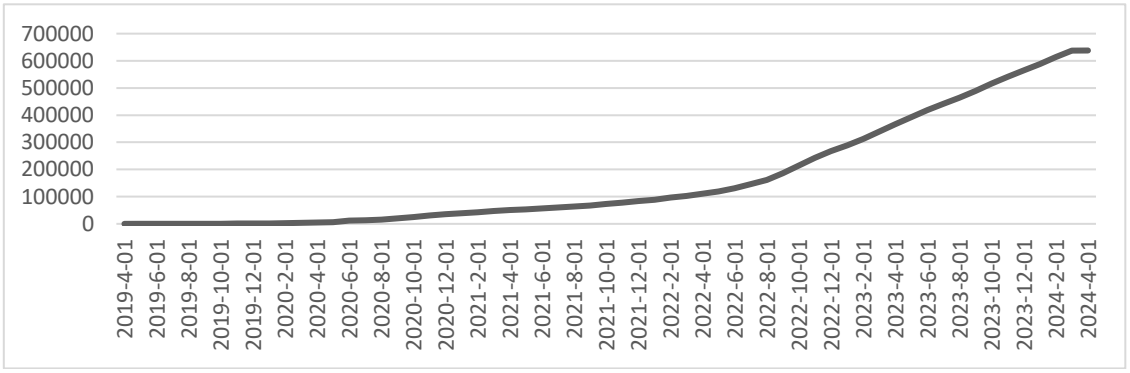



Рисунок 3. Общее количество записей.

На основе графиков на рисунках 2 и 3 можно убедиться, что рассматриваемый репозиторий содержит актуальные данные и полностью соответствует критерию актуальности. Кроме того, очевидна тенденция на увеличение количества записей за последние месяцы, что в целом даёт возможность постоянно улучшать качество результатов анализа на основе алгоритмов машинного обучения.

### *Заключение*

В заключении можно отметить, что из пяти приведённых критериев качества данных четыре соответствует представлению о данных для успешного решения задач аналитики с применением методов искусственного интеллекта. Единственный критерий, в отношении которого требуется проведение дополнительных работ перед началом решения задачи аналитики, является критерий понятности. Однако эта проблема нивелируется использованием различных методов подготовки данных, специализированных на решении подобных задач [12-15]. Следовательно, данный репозиторий можно считать приемлемым для решения различных задач аналитики на основе методов искусственного интеллекта.

Также стоит учитывать, что для работы с медицинскими данными может потребоваться прохождение специализированного обучения. Например, для анализируемого репозитория клинических данных [8, 9] обязательным является прохождение обучения по культуре работы с биомедицинскими данным.

Подводя итог данной статье, можно сказать, что был произведён анализ качества амбулаторных и стационарных данных из русскоязычного репозитория клинических данных [9], выявлены критерии, по которым можно судить о качестве данных, определены и систематизированы проблемы в категориальных, текстовых и числовых данных объективных обследований. Это имеет практическое значение для дальнейших исследований и разработок в области решения аналитических задач в медицине с использованием методов искусственного интеллекта.

### *Литература*

1. Application of artificial intelligence in medicine: an overview / Liu P. et al. //Current medical science. 2021. Vol. 41(6). Pp. 1105-1115.
2. Musen M. A., Middleton B., Greenes R. A. Clinical decision-support systems //Biomedical informatics: computer applications in health care and biomedicine. Cham : Springer International Publishing, 2021. Pp. 795-840.

3. Data collection and quality challenges in deep learning: A data-centric ai perspective / Whang S. E. et al. // The VLDB Journal. 2023. Vol. 32(4). Pp. 791-813.
4. Overview and importance of data quality for machine learning tasks / Jain A. et al. // Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020. Pp. 3561-3562.
5. ГОСТ Р ИСО 8000-2-2019. Качество данных. Часть 2. Словарь.: Национальный стандарт Российской Федерации : дата введения 2020-05-01 / Федеральное государственное унитарное предприятие «Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия» (ФГУП «СТАНДАРТИНФОРМ»). Издание официальное. Москва : Стандартинформ, 2019. 12 с.
6. Проект ГОСТ Р Информационные технологии. Искусственный интеллект. Качество данных для аналитики и машинного обучения. Часть 2. Показатели качества данных. URL: https://docs.cntd.ru/document/351741766 (Дата обращения: 04.11.2024).
7. Федеральный закон от 27.06.2006 № 152-ФЗ «О персональных данных».
8. Русскоязычный репозиторий открытых клинических данных SibMED Data Clinical Repository / Куликов Е. С. и др. // Бюллетень сибирской медицины. 2023. № 22(2). С. 182–184.
9. Первый русскоязычный репозиторий клинических данных – URL: https://dataset.ssmu.ru (Дата обращения: 04.11.2024).
10. Самойленко Н. Э., Кувина В. Н., Кувин С. С. Комплексный анализ медицинских данных //Вестник Воронежского государственного технического университета. – 2009. – Т. 5. – №. 9. – С. 114-118.
11. ГОСТ Р ИСО 53022.3-2008. Технологии лабораторные клинические. Требования к качеству клинических лабораторных исследований. Часть 3. Правила оценки клинической информативности лабораторных тестов.: Национальный стандарт Российской Федерации : дата введения 2010-01-01 / Федеральное государственное унитарное предприятие «Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия» (ФГУП «СТАНДАРТИНФОРМ»). Издание официальное. Москва : Стандартинформ, 2008. 19 с.
12. A systematic map of medical data preprocessing in knowledge discovery / Idri A. et al. // Computer methods and programs in biomedicine. 2018. Т. 162. С. 69-85.
13. Nishanov A. K., Djuraev G. P., Khasanova M. A. Classification and feature selection in medical data preprocessing // Compusoft. 2020. Vol. 9(6). Pp. 3725-3732.
14. Advances in data preprocessing for biomedical data fusion: An overview of the methods, challenges, and prospects / Wang S. et al. // Information Fusion. 2021. Vol. 76. Pp. 376-421.
15. Kashina M., Lenivtceva I. D., Kopanitsa G. D. Preprocessing of unstructured medical data: the impact of each preprocessing stage on classification // Procedia Computer Science. 2020. Vol. 178. Pp. 284-290.

***Петрушан Н.И., Филиппов А.А. Анализ качества репозитория клинических данных.*** *Статья представляет описание результатов анализа качества амбулаторных и стационарных данных из русскоязычного репозитория клинических данных. Определены и систематизированы проблемы в категориальных, текстовых и числовых данных объективных обследований, предложен алгоритм устранения полученных ошибок. Оценена применимость репозитория для решения аналитических задач с применением методов искусственного интеллекта.*

***Ключевые слова:*** *репозиторий, искусственный интеллект, анализ качества данных, ошибки данных, алгоритм устранения ошибок, решение аналитических задач.*

***Petrushan N.I., Filippov A.A.*** *Quality analysis of clinical data repository. The article describes the results of quality analysis of outpatient and inpatient data from the Russian-language clinical data repository. Problems in categorical, textual and numerical data of objective examinations are identified and systematized, and an algorithm for eliminating the errors obtained is proposed. The applicability of the repository for solving analytical problems using artificial intelligence methods is evaluated.*

***Key words:*** *repository, artificial intelligence, data quality analysis, data errors, algorithm of error elimination, solution of analytical problems.*

УДК 004.8

# Использование нейронных сетей для повышения эффективности рекомендательных систем в организации развлекательных мероприятий

**Я.А. Рудь[*1], С.А. Зори[*2], И.А. Коломойцева[*3]**

[*1] магистрант, Донецкий национальный технический университет,
rezan20031@gmail.com

[*2] д.т.н, профессор, Донецкий национальный технический университет,
ik.ivt.rec@mail.ru, OrcID: 0000-0003-4018-234X, SPIN-код: 3565-6330

[*3] ст. преподаватель, Донецкий национальный технический университет,
bolatiger@mail.ru, OrcID: 0000-0002-1559-0213, SPIN-код: 1002-8374

***Рудь Я.А., Зори С. А., Коломойцева И.А. Использование нейронных сетей для повышения эффективности рекомендательных систем в организации развлекательных мероприятий.*** *Статья посвящена применению нейронных сетей для улучшения эффективности рекомендательных систем в сфере организации развлекательных мероприятий. Основное внимание уделяется анализу методов и подходов, направленных на создание персонализированных рекомендаций, таких как выбор мероприятий (концертов, театров, фестивалей) с учетом предпочтений пользователей. Научная значимость работы заключается в выявлении сильных и слабых сторон существующих решений и в предложении новых подходов для повышения точности, адаптивности и вычислительной эффективности моделей. Практическое значение работы связано с разработкой более эффективных и персонализированных систем рекомендаций для индустрии развлечений, что может существенно улучшить пользовательский опыт. В рамках исследования проводится обзор существующих технологий, их оценка и анализ проблем, с которыми сталкиваются рекомендательные системы в сфере развлечений. Основные результаты направлены на улучшение качества рекомендаций, основанных на нейронных сетях, а также на повышение их вычислительной эффективности.*

***Ключевые слова:*** *нейронные сети, рекомендательные системы, развлекательные мероприятия, персонализация, машинное обучение, улучшение рекомендаций.*

## *Введение*

Современные рекомендательные системы становятся важным инструментом в различных сферах, включая организацию развлекательных мероприятий. С учетом роста объема данных и разнообразия предпочтений пользователей, традиционные методы рекомендаций, такие как коллаборативная фильтрация и контентная фильтрация, становятся менее эффективными. В этой связи нейронные сети, в частности методы глубокого обучения, открывают новые возможности для создания более точных и персонализированных рекомендаций, которые учитывают не только явные предпочтения пользователей, но и более сложные скрытые закономерности [1]. Актуальность исследования обусловлена необходимостью улучшения качества рекомендаций и адаптации систем к постоянно изменяющимся предпочтениям пользователей, что является важным вызовом в современной развлекательной индустрии.

Развлекательная индустрия, включая кино, театры, концерты и фестивали, представляет собой одну из наиболее перспективных областей применения рекомендательных систем. Эффективные системы могут значительно повысить удовлетворенность пользователей, улучшив их опыт и предлагая мероприятия, которые наилучшим образом соответствуют их интересам. Несмотря на достигнутые успехи в разработке рекомендательных систем, многие проблемы, такие как учет временных изменений предпочтений, адаптация рекомендаций в реальном времени и повышение вычислительной эффективности, остаются актуальными и требуют новых решений.

Цель данной работы — исследовать возможности использования нейронных сетей для улучшения методов создания рекомендательных систем в сфере развлечений, а также предложить новые подходы к оптимизации этих систем. В статье рассматриваются основные подходы и модели, такие как коллаборативная фильтрация, контентная фильтрация и гибридные методы, а также их применение с использованием нейронных сетей. Особое

внимание уделено улучшению вычислительной эффективности и адаптивности этих систем, что является ключевым фактором для повышения их практической ценности. Новизна работы заключается в предложении методов для повышения точности и адаптивности рекомендательных систем, а также в исследовании возможных путей оптимизации нейронных сетей для использования в реальных условиях развлекательной индустрии. В результате исследования будут предложены пути оптимизации рекомендательных систем, что позволит повысить их точность, вычислительную эффективность и удовлетворенность пользователей.

### *Обзор существующих методов создания рекомендательных систем*

Рекомендательные системы играют важную роль в персонализации опыта пользователей, предлагая им продукты или услуги, которые, по мнению системы, могут быть им интересны. Существует несколько подходов к созданию таких систем, каждый из которых имеет свои особенности и области применения. В данном разделе рассматриваются основные методы создания рекомендательных систем: коллаборативная фильтрация, контентная фильтрация и гибридные методы.

Коллаборативная фильтрация является одним из самых распространенных и популярных методов в построении рекомендательных систем [2]. Этот подход основывается на идее, что пользователи, которые в прошлом проявляли схожие предпочтения, будут иметь схожие вкусы и в будущем. Коллаборативная фильтрация может быть разделена на два типа: **User-based (основанная на пользователях)** и **Item-based (основанная на элементах)**.

Преимущества коллаборативной фильтрации включают простоту в реализации и высокую эффективность в случае наличия большого количества данных о пользователях. Однако этот метод имеет и свои недостатки, такие как проблема "холодного старта" (когда новая система или новый пользователь не имеет достаточного количества данных для точных рекомендаций), а также сложность в обработке огромных объемов данных.

Контентная фильтрация ориентирована на характеристики самих объектов, которые рекомендованы пользователю. В отличие от коллаборативной фильтрации, этот метод фокусируется на том, что именно привлекает пользователя в содержимом. Например, при организации развлекательных мероприятий система будет анализировать такие характеристики, как жанр мероприятия, дата, местоположение, продолжительность и другие атрибуты, чтобы предложить пользователю мероприятия, которые соответствуют его интересам.

Основным преимуществом контентной фильтрации является отсутствие зависимости от взаимодействия с другими пользователями, что позволяет избежать проблемы "холодного старта". Кроме того, этот подход особенно полезен, когда у пользователя есть явные предпочтения по характеристикам мероприятий (например, жанр музыки или тип театральной постановки). Однако основной проблемой этого метода является его ограниченность: если пользователь не указал свои предпочтения, система может не предложить ему новых объектов, которые могут ему понравиться.

Гибридные методы являются сочетанием различных подходов для улучшения качества рекомендаций. Эти системы комбинируют коллаборативную фильтрацию и контентную фильтрацию, а также могут использовать дополнительные алгоритмы, такие как факторизация матриц, методы машинного обучения или нейронные сети. Основной целью гибридных методов является повышение точности рекомендаций и снижение недостатков, присущих каждому отдельному методу.

**О**дин из способов объединить контентную фильтрацию и коллаборативную фильтрацию — это использование данных о предпочтениях пользователей и характеристиках объектов одновременно. Например, если система рекомендует мероприятие на основе жанра, она также учитывает оценки других пользователей, что позволяет улучшить рекомендации для новых пользователей.

Гибридные методы часто дают наилучшие результаты, так как они позволяют объединить сильные стороны различных подходов и снизить их слабые стороны. Однако такие системы требуют больше вычислительных ресурсов и могут быть сложными в реализации, особенно в случае обработки больших объемов данных.

### *Использование нейронных сетей*

Нейронные сети, особенно в рамках технологий глубокого обучения, становятся все более популярными для построения рекомендательных систем, поскольку они могут эффективно выявлять сложные закономерности в больших объемах данных. Рассмотрим несколько подходов, которые могут быть полезны для создания эффективных рекомендательных систем с использованием нейронных сетей.

**Глубокое обучение.** Глубокие нейронные сети обладают мощными возможностями для построения сложных моделей, которые могут учитывать не только явные предпочтения пользователей, но и скрытые паттерны в данных. Это позволяет создавать более точные и персонализированные рекомендации. Например, нейронные сети могут анализировать взаимодействия между пользователем и мероприятием, выявлять закономерности в этих взаимодействиях и использовать их для предсказания будущих предпочтений.

Одним из основных преимуществ глубокого обучения является способность учитывать различные типы данных, включая неструктурированные (например, текстовое описание мероприятий) или многомерные данные (например, местоположение, время, температура и другие внешние факторы). Глубокие нейронные сети могут также учитывать временные предпочтения пользователей, меняющиеся в зависимости от времени года, праздников или других факторов. Например, если система знает, что пользователь чаще посещает концерты в определенные месяцы года, она может рекомендовать ему мероприятия, подходящие для этого времени.

**Рекуррентные нейронные сети (RNN).** Рекуррентные нейронные сети особенно полезны для обработки временных зависимостей, что делает их эффективными для анализа предпочтений пользователей, которые изменяются со временем. В отличие от традиционных нейронных сетей, RNN имеют возможность учитывать последовательности данных, что позволяет им анализировать, как предпочтения пользователя развиваются в зависимости от его предыдущего опыта.

Для рекомендательных систем использование RNN может быть полезным для предсказания изменений в предпочтениях пользователей по мере того, как они посещают различные мероприятия. Например, система может учитывать историю посещений пользователя и прогнозировать, какие мероприятия ему могут понравиться в будущем, исходя из того, какие события он посещал ранее. Это позволяет более точно персонализировать рекомендации, учитывая динамичность пользовательских интересов.

**Сети с долгосрочной памятью (LSTM).** Рекуррентные нейронные сети сталкиваются с проблемой обработки длинных последовательностей данных из-за исчезающего градиента. Для решения этой проблемы были разработаны сети с долгосрочной памятью, которые обладают улучшенной архитектурой и способны удерживать информацию о долгосрочных зависимостях.

LSTM идеально подходят для анализа длинных историй посещений мероприятий, где предпочтения пользователя могут изменяться постепенно. Например, если пользователь участвует в различных фестивалях или концертах в течение длительного времени, LSTM может помочь выявить долгосрочные тенденции в его интересах, такие как изменение жанров или предпочтений в выборе типа мероприятий. Это позволяет рекомендательной системе предсказать более точные и адаптивные рекомендации, которые будут учитывать всю историю пользовательского поведения.

**Системы рекомендаций на основе автоэнкодеров.** Автоэнкодеры представляют собой нейронные сети, предназначенные для уменьшения размерности данных, что позволяет выявлять скрытые закономерности в предпочтениях пользователей [3]. Эти сети учат данные представлять в более компактной и удобной форме, что позволяет эффективно выявлять скрытые связи между пользователем и объектами рекомендаций.

В контексте рекомендательных систем автоэнкодеры могут быть использованы для уменьшения размерности данных о предпочтениях пользователей и мероприятиях, что помогает системе фокусироваться на наиболее значимых характеристиках. Используя такую модель, система может выявлять неочевидные паттерны и связи, что способствует более точным рекомендациям. Например, если автоэнкодер обнаруживает, что два различных типа мероприятий имеют схожие скрытые характеристики (например, оба могут быть классифицированы как "вдохновляющие" или "релаксирующие"), он может порекомендовать эти мероприятия пользователю, даже если они сильно различаются по внешним признакам, таким как жанр или тип.

### *Модели для предсказания предпочтений пользователя*

Современные рекомендательные системы, использующие нейронные сети, позволяют создавать более точные и персонализированные модели, которые могут учитывать широкий спектр факторов, влияющих на предпочтения пользователей. Это не только улучшает качество рекомендаций, но и позволяет учитывать более сложные зависимости и взаимодействия, чем традиционные методы.

Современные модели на основе нейронных сетей могут учитывать гораздо больше факторов, чем просто предыдущие действия пользователя. Ранее упомянутые подходы, такие как коллаборативная фильтрация и контентная фильтрация, ориентированы в основном на явные предпочтения, такие как оценки или характеристики мероприятий. Однако для более точного предсказания предпочтений необходимо учитывать более широкий контекст, включающий различные факторы, такие как:

1) **история посещений мероприятий**: нейронные сети могут учитывать не только те мероприятия, которые пользователь уже посетил, но и их контекст, например, время года, местоположение, продолжительность, жанр и другие параметры (это позволяет учитывать тенденции в предпочтениях пользователя, такие как, например, более высокие оценки мероприятий в определенные периоды года);

2) **оценки пользователей**: оценка мероприятий пользователем может быть полезным сигналом для обучения модели (например, система может предсказать, что пользователю, который оценил определенный концерт высоко, могут понравиться другие концерты того же жанра или с теми же исполнителями);

3) **интересы пользователя**: дополнительно нейронные сети могут учитывать личные интересы пользователя, которые могут быть извлечены из его активностей, профиля или социальных сетей (например, если

пользователь активно следит за певцом или театральной труппой, система может рекомендовать ему мероприятия, связанные с этим исполнителем);

4) **социальные сети и поведение окружающих**: социальные сети могут стать ценным источником информации для предсказания предпочтений (например, если группа друзей посещает музыкальный фестиваль, система может предложить этот фестиваль другим пользователям, основываясь на их социальной сети);

5) **внешние факторы**: помимо явных интересов, предпочтения пользователей могут изменяться под влиянием внешних факторов, таких как погода, время суток или даже культурные события.

Одним из основных преимуществ нейронных сетей является возможность создавать персонализированные рекомендации, которые будут адаптироваться в зависимости от поведения пользователя [4]. Система, использующая нейронные сети, может анализировать, как изменяются предпочтения пользователя с течением времени, и делать более точные прогнозы на основе этих изменений. Кроме того, модель может использовать информацию о других пользователях, чье поведение схоже, для создания рекомендаций.

Персонализированные рекомендации могут быть сделаны с использованием нескольких подходов:

1) **рекомендации на основе поведения пользователя**: нейронные сети могут анализировать последовательности действий пользователя, таких как посещения мероприятий, клики на определенные жанры, оценки и другие факторы;

2) **рекомендации на основе схожих пользователей (коллаборативная фильтрация)**: используя нейронные сети, можно эффективно применять методы коллаборативной фильтрации, чтобы найти пользователей с похожими предпочтениями и рекомендовать мероприятия, которые были интересны этим пользователям;

3) **гибридные модели**: для еще более точных рекомендаций нейронные сети могут использовать гибридный подход, комбинируя данные о пользователях и объектах (например, характеристики мероприятий).

### *Интеграция нейронных сетей в организацию и персонализацию развлекательных мероприятий*

Рекомендательные системы на основе нейронных сетей могут значительно улучшить подбор мероприятий, предлагаемых пользователю, с учетом его индивидуальных предпочтений. Например, система может учитывать не только предпочтения в жанре (концерты, театральные постановки, кинофестивали), но и такие параметры, как время суток, бюджет, доступные места и личные интересы пользователя.

Для этого нейронные сети могут использовать данные о предыдущих посещениях мероприятий, предпочтениях в жанрах, оценках, а также социальных факторах, таких как мероприятия, посещенные друзьями или членами семьи. Такой подход позволяет создавать рекомендации, которые учитывают не только явные предпочтения пользователя, но и скрытые паттерны в его поведении, а также взаимосвязи между разными типами событий.

Кроме того, система может адаптироваться к изменениям интересов пользователя со временем, предсказывая, какие новые мероприятия могут быть интересны на основе его предыдущих выборов. Например, если пользователь часто посещает рок-концерты, система может предложить ему новые мероприятия с участием любимых исполнителей или в рамках похожих музыкальных стилей.

Одним из важных направлений применения рекомендательных систем является интеграция с мобильными приложениями и веб-платформами, что позволяет предоставлять рекомендации в реальном времени [5]. Это особенно важно для мероприятий, которые имеют ограниченную доступность, например, с ограниченным числом билетов или с учетом изменений в расписании.

Через мобильные приложения можно предложить пользователям рекомендации, учитывая их местоположение, что дает возможность направлять их к ближайшим мероприятиям. Например, если пользователь находится в новом городе, система может предложить ближайшие концерты, театральные постановки или спортивные события, которые могут его заинтересовать. Это также может включать рекомендации с учетом доступных билетов или ценовых категорий, что делает систему не только персонализированной, но и оперативной.

Система может также отправлять уведомления о специальных предложениях или новых событиях, которые соответствуют предпочтениям пользователя. Интеграция с мобильными платформами позволяет оперативно реагировать на изменения в планах пользователя и предлагать альтернативы, если выбранное мероприятие больше не доступно.

Одним из уникальных подходов, который можно внедрить для улучшения рекомендательных систем в организации развлекательных мероприятий, является использование контекстной информации в реальном времени. Важно не только учитывать предпочтения пользователя, но и условия внешней среды, которые могут влиять на выбор мероприятия.

Для этого можно внедрить функционал, который будет анализировать такие параметры, как погода, сезонность, праздничные дни и социальные тренды, чтобы предоставлять рекомендации, наиболее подходящие для текущего контекста. Например, в дождливую погоду система может предложить пользователю мероприятия в закрытых помещениях, такие как театральные постановки или концерты, в то время как в теплую и солнечную погоду — концерты на открытом воздухе или спортивные события.

Дополнительно, система может учитывать изменения в поведении пользователей в зависимости от времени суток и сезона. Например, в вечернее время система может предложить более расслабляющие мероприятия, такие как кино или спектакли, в то время как в выходные дни — более активные события, такие как спортивные матчи или фестивали.

На рисунке 1 представлена предполагаемая архитектура рекомендательной системы на основе нейронных сетей для организации развлекательных мероприятий.

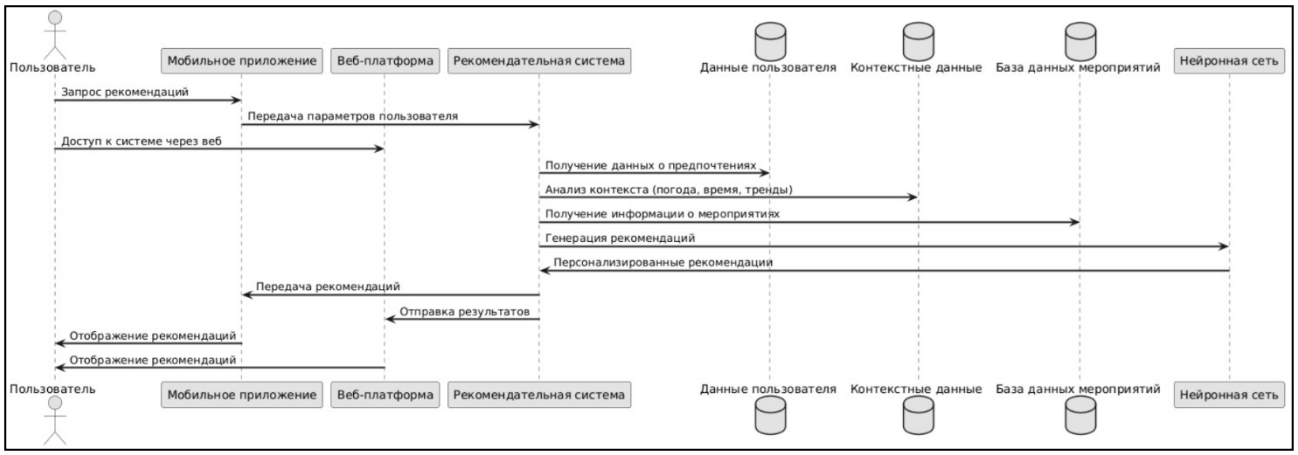



Рисунок 1 – Предполагаемая архитектура рекомендательной системы на основе нейронных сетей для организации развлекательных мероприятий

Эта схема демонстрирует, как система взаимодействует с различными компонентами и как учитываются контекстные и пользовательские данные для создания точных рекомендаций.

Последовательность действий такова:

1) **пользователь** может взаимодействовать с рекомендательной системой через мобильное приложение или веб-платформу;

2) система использует источники данных: д**анные пользователя** (история посещений, предпочтения, социальные факторы), к**онтекстные данные** (погода, тренды, время суток, сезонность), б**аза данных мероприятий** (доступные события, места, билеты);

3) основной компонент системы — **нейронная сеть**, которая обрабатывает входные данные и генерирует персонализированные рекомендации;

4) рекомендации возвращаются пользователю в удобной форме с учетом актуального контекста и предпочтений.

Таким образом, использование контекстной информации позволяет не только улучшить точность рекомендаций, но и сделать их более адаптированными к текущим условиям, создавая для пользователя максимально удобный и персонализированный опыт.

### *Выводы*

В ходе исследования были рассмотрены ключевые аспекты использования нейронных сетей для создания рекомендательных систем в сфере организации развлекательных мероприятий. В результате анализа традиционных методов, таких как коллаборативная фильтрация, контентная фильтрация и их гибридные подходы, а также внедрения глубоких нейронных сетей, выяснилось, что использование последних открывает новые горизонты для более точных и персонализированных рекомендаций.

Одним из важных выводов является то, что нейронные сети, особенно в контексте глубокого обучения, позволяют не только учитывать явные предпочтения пользователей, но и выявлять скрытые закономерности в данных. Это значительно повышает качество рекомендаций, учитывая сложные взаимосвязи между пользователем и контентом, такие как временные изменения предпочтений или влияние внешних факторов.

Кроме того, использование рекуррентных нейронных сетей (RNN) и их модификаций, таких как LSTM, показало свою эффективность в анализе последовательностей данных, что крайне важно для предсказания

динамичных предпочтений пользователей, меняющихся с течением времени. Автоэнкодеры, как метод уменьшения размерности данных, также продемонстрировали свою полезность для выявления скрытых паттернов, что способствует точности рекомендаций, особенно при обработке больших объемов информации.

Персонализированные рекомендации, основанные на анализе поведения пользователей, оценках и социальных факторах, способны значительно повысить удовлетворенность и вовлеченность аудитории в процессе выбора мероприятий. Использование нейронных сетей позволяет адаптировать рекомендации в реальном времени, учитывая изменения интересов пользователей и их взаимодействие с событиями.

Важным направлением является интеграция рекомендательных систем с мобильными приложениями и веб-платформами, что позволяет предоставлять рекомендации в реальном времени с учетом местоположения пользователя, доступных билетов и других параметров. Это делает системы не только персонализированными, но и оперативными, что важно для развлекательной индустрии, где события могут быстро меняться.

Таким образом, нейронные сети, благодаря своей способности эффективно обрабатывать большие объемы данных и выявлять сложные зависимости, представляют собой мощный инструмент для улучшения рекомендательных систем в сфере организации развлекательных мероприятий. В дальнейшем, дальнейшая оптимизация алгоритмов и их адаптация к изменениям в предпочтениях пользователей может значительно повысить их практическую ценность и точность предсказаний.

### *Литература*


1. Зайцева С.А. Ценностные регулятивы «рекомендательной системы» в контексте распространения нейросетевых моделей / С.А. Зайцева, В.А. Смирнов // Вестник Ивановского государственного университета. – 2022. – № 1. – С. 152-160.

2. Смоленчук Т.В. Метод коллаборативной фильтрации для рекомендательных сервисов / Т.В. Смоленчук // Вестник науки и образования. – 2019. – № 22 (76) – С. 18-21.

3. Акинина Н.В. Автоэнкодер: подход к понижению размерности векторного пространства с контролируемой потерей информации / Н.В. Акинина, М.В. Акинин, А.В. Соколова, М.Б. Никифоров, А.И. Таганов // Известия ТулГУ. Технические науки – 2016. – № 9 – С. 3-12.

4. Садовская А.В. Повышение разнообразия рекомендаций за счет персонализации/ А.В. Садовская, А.Г. Афанасьев, Г.И. Афанасьев // E-Scio. – 2023. – № 8 (83) – С. 141-160.

5. Сапрыкин Д.А. Возможности применения машинного обучения в мобильных приложениях для персонализированной рекомендации контента / Д.А. Сапрыкин, К.В. Подольский, А.В. Панов // Международный журнал гуманитарных и естественных наук – 2023. – № 12 (87) – С. 80-83.



***Рудь Я.А., Зори С. А., Коломойцева И.А. Использование нейронных сетей для повышения эффективности рекомендательных систем в организации развлекательных мероприятий.*** *Статья посвящена применению нейронных сетей для улучшения эффективности рекомендательных систем в сфере организации развлекательных мероприятий. Основное внимание уделяется анализу методов и подходов, направленных на создание персонализированных рекомендаций, таких как выбор мероприятий (концертов, театров, фестивалей) с учетом предпочтений пользователей. Научная значимость работы заключается в выявлении сильных и слабых сторон существующих решений и в предложении новых подходов для повышения точности, адаптивности и вычислительной эффективности моделей. Практическое значение работы связано с разработкой более эффективных и персонализированных систем рекомендаций для индустрии развлечений, что может существенно улучшить пользовательский опыт. В рамках исследования проводится обзор существующих технологий, их оценка и анализ проблем, с которыми сталкиваются рекомендательные системы в сфере развлечений. Основные результаты направлены на улучшение качества рекомендаций, основанных на нейронных сетях, а также на повышение их вычислительной эффективности.*



***Ключевые слова:*** *нейронные сети, рекомендательные системы, развлекательные мероприятия, персонализация, машинное обучение, улучшение рекомендаций.*



***Rud Ya.A., Zori S.A., Kolomoitseva I.A. Using Neural Networks to Improve the Efficiency of Recommender Systems in Organizing Entertainment Events.*** *The article is devoted to the use of neural networks to improve the efficiency of recommender systems in organizing entertainment events. The main attention is paid to the analysis of methods and approaches aimed at creating personalized recommendations, such as selecting events (concerts, theaters, festivals) taking into account user*

*preferences. The scientific significance of the work lies in identifying the strengths and weaknesses of existing solutions and proposing new approaches to improve the accuracy, adaptability and computational efficiency of models. The practical significance of the work is related to the development of more efficient and personalized recommendation systems for the entertainment industry, which can significantly improve user experience. The study reviews existing technologies, evaluates them and analyzes the problems faced by recommender systems in the entertainment industry. The main results are aimed at improving the quality of recommendations based on neural networks, as well as increasing their computational efficiency.*

***Keywords:*** *neural networks, recommender systems, entertainment events, personalization, machine learning, recommendation improvement.*

УДК 004.8

# Параметрическое описание свёрточной нейронной сети как объекта имитационного моделирования

**О.И. Федяев[*1], Д.В. Гавриленков[*2]**

[*1] к.т.н, доцент, Донецкий национальный технический университет,
olegfedyayev@mail.ru, OrcID: 0000-0001-6822-7306, SPIN-код: 7777-3791
[*2] магистрант, Донецкий национальный технический университет,
nilixen@ya.ru, OrcID: 0009-0009-4435-4517

***Федяев О.И., Гавриленков Д.В. Параметрическое описание свёрточной нейронной сети как объекта имитационного моделирования.*** *В данной статье проведён комплексный анализ современных библиотек машинного обучения с целью выявления и систематизации функциональных возможностей, предназначенных для построения свёрточных нейронных сетей (СНС). Уделено особое внимание изучению взаимосвязей между параметрами сети (гиперпараметрами) и их влиянию на архитектуру модели. Результаты настоящего исследования формируют теоретическую базу для последующей разработки имитационной модели СНС. Предложенное параметрическое описание позволяет структурировать процесс проектирования свёрточных нейронных сетей и может быть использовано как основа для создания средств обучения в области искусственных нейронных сетей.*

***Ключевые слова:*** *свёрточные нейронные сети, машинное обучение, гиперпараметры, нейронные сети, архитектура нейронной сети, библиотека машинного обучения, проектирование нейронных сетей, глубокое обучение.*

## *Введение*

В настоящее время свёрточные нейронные сети (СНС) являются одним из наиболее эффективных инструментов в области компьютерного зрения и обработки изображений. Однако, процесс проектирования и настройки архитектуры СНС остаётся сложной задачей, требующей глубокого понимания взаимосвязей между различными параметрами сети. Существующие библиотеки машинного обучения предоставляют широкий спектр инструментов для создания СНС, но отсутствие единого параметрического описания затрудняет процесс их систематического изучения и моделирования.

Актуальность данного исследования обусловлена необходимостью создания структурированного подхода к описанию и моделированию СНС, который позволит упростить процесс изучения, проектирования и обучения нейронных сетей. Целью работы является системный анализ СНС как объекта имитационного моделирования на основе изучения принципов работы нейросетей и выявления ключевых компонентов и гиперпараметров, определяющих процесс создания, обучения и применения нейронных сетей.

Для достижения поставленной цели необходимо провести анализ архитектур СНС, выявить основные структурные элементы, разработать параметрическую модель сети, оценить возможности инструментальных средств реализации СНС с помощью библиотек машинного обучения, проанализировать существующие программные реализации СНС и выполнить объектно-ориентированный анализ системы моделирования СНС на основе полученных данных.

Результаты данного исследования могут быть использованы для создания системы моделирования СНС, что позволит упростить процесс обучения и разработки моделей нейронных сетей для решения различных задач.

## *Архитектура типовой свёрточной нейронной сети*

Свёрточная нейронная сеть (англ. convolutional neural network, CNN) представляет собой специализированный тип алгоритма глубокого обучения, который используется в основном для задач распознавания образов, включая классификацию образов, обнаружение объектов и их сегментацию. СНС аналогичны традиционным искусственным нейронным сетям тем, что они состоят из нейронов, которые самонастраиваются через процесс обучения. Каждый нейрон получает входной сигнал и осуществляет

определённую операцию (например, скалярное произведение, за которым следует нелинейная функция активации). Обобщённая архитектура свёрточной нейронной сети приведена на рисунке 1. Сигнал проходит через набор слоёв свёртки, в которых извлекаются признаки в так называемые карты признаков. После свёртки обычно используются слои пулинга, которые уменьшают размерность карт признаков с целью оптимизировать модель CHC. Расчёт состоит в том, что с каждым следующим слоем нейросеть извлекает всё более сложные признаки. Пример карт признаков приведен на рисунке 2.

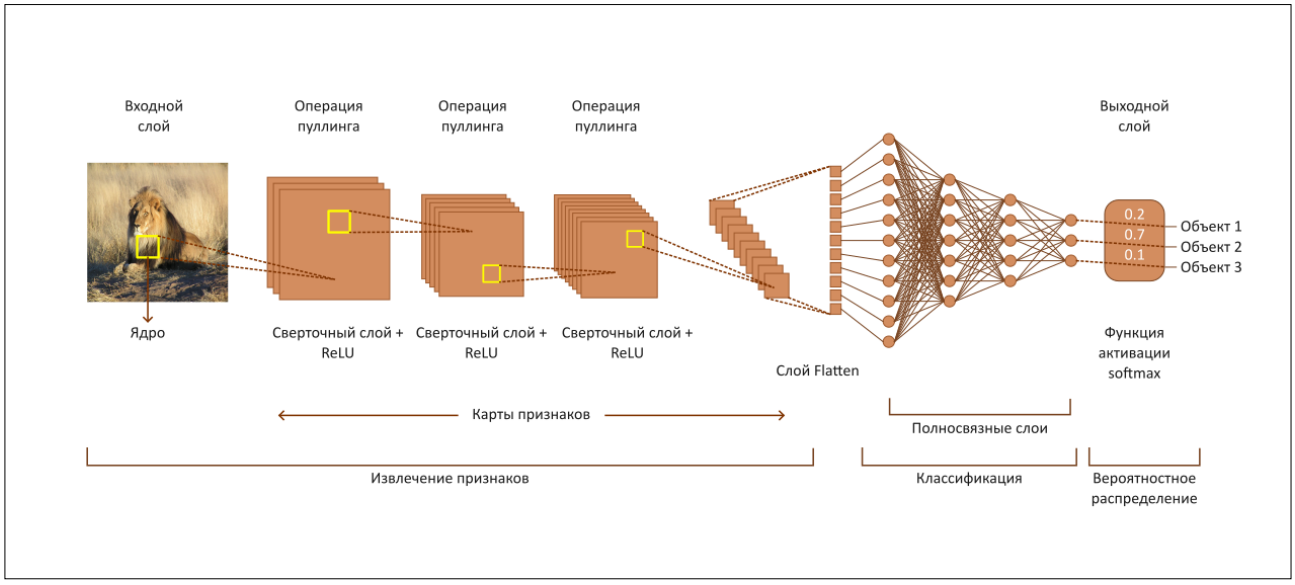



Рисунок 1 – Архитектура свёрточной нейронной сети

Как можно видеть на рисунке 2, модель сначала обнаруживает элементарные характеристики в изображении и по мере продвижения алгоритма дальше приходит к более сложным признакам.

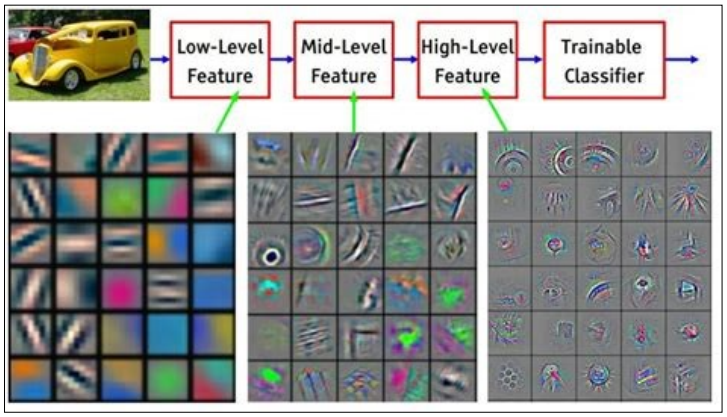



Рисунок 2 – Пример карт признаков

CHC состоят из трёх типов слоев: слоёв свёртки, слоёв пулинга и полносвязных слоёв.

Входящий слой хранит пиксельные данные о входном изображении.

На этапе свёрточного слоя (или слоя свёртки) алгоритм осуществляет так называемую операцию свёртки, которая представляет собой применение функции скользящего окна к матрице пикселей предыдущего слоя. Скользящая функция, которая применяется к матрице, называется ядром или фильтром (оба термина могут использоваться взаимозаменяемо).

Особая область входного изображения, которая влияет на выходное значение нейрона, называется рецептивным полем. Другими словами, это часть входного изображения, которую «видит» нейрон. Свёрточные

слои способны значительно снизить сложность модели путем оптимизации выходных значений. Этими оптимизациями являются три гиперпараметра: глубина, паддинг и страйд.

Глубина настраивается вручную путем изменения количества фильтров, которые используются в сверточном слое. Каждый фильтр в свёрточном слое генерирует отдельную двумерную карту признаков. Накладывая эти двумерные карты признаков вдоль новой оси, мы получаем трехмерную выходную карту признаков с глубиной, равной количеству фильтров.

Паддинг (англ. padding) — это процесс добавления дополнительных пикселей по границе входного изображения перед применением операции свёртки. Паддинг помогает сохранить пространственные измерения, так что размер изображения не уменьшается слишком быстро при прохождении через несколько слоев.

Страйд (англ. stride) это размер шага, с которым свёрточный фильтр двигается по входному изображению. При страйде равном 1 фильтр перемещается по одному пикселю за раз, генерируя карты признаков с большей детализацией и перекрывающимися рецептивными полями. Страйд равный больше 1 заставляет фильтр «пропускать» пиксели при перемещении, что приводит к меньшему количеству перекрывающихся рецептивных полей и созданию меньшей карты признаков.

Целью слоя пулинга является извлечение наиболее значимых признаков из свёрнутой матрицы. Это делается путём применения некоторых операций агрегации, которые уменьшают размерность признаков (свёрнутой матрицы), тем самым уменьшая объем памяти, используемой при обучении сети, но сохраняя главные признаки в изображении. Пулинг также актуален для смягчения переобучения.

Полносвязные слои находятся в конце CHC, и их входные значения соответствуют сглаженной одномерной матрице, сгенерированной последним слоем пулинга. Функции активации (например, ReLU) применяются к ним для нелинейности системы.

Наконец, слой прогноза softmax используется для генерации значений вероятности для каждой из возможных выходных меток (заранее определённых наименований объектов на изображении). Окончательная метка — это метка с наивысшей оценкой вероятности.

Переобучение (англ. overfitting) является частой проблемой в моделях машинного обучения. Это происходит, когда модель вроде бы хорошо обучается на обучаемых данных, включая шум и наличие выпадающих значений. Переобучение можно заметить, когда на этапе обучения точность оценивается значительно выше, чем на этапе прогнозирования. Таким образом, точность модели на обучаемых данных высокая, а на совершенно новых данных низкая. Чтобы уменьшить данную проблему применяются методы изменения данных и следующие стратегии регуляризации: dropout, батч-нормализация (англ. batch normalization), ранняя остановка (англ. early stopping), добавление шума (англ. noise injection), регуляризации L1 и L2.

### *Разработка параметрической модели свёрточной нейронной сети*

Формально архитектура свёрточной нейронной сети описана в виде параметрической модели. На теоретико-множественном уровне модель свёрточной нейронной сети CNN представлена в виде кортежа

$$CNN =< M, N_m, \{< X_{i,j}^{c,m,l_m},\ C_{i,j}^{c,m,l_m},\ f^{c,m},\ Y_{i,j}^{c,m,l_m}, X_{i,j}^{s,m,l_m},\ S_{i,j}^{s,m,l_m},\ f^{s,m},\ Y_{i,j}^{s,m,l_m}\}, X_i^{p,v},\ W_i^{p,v}, f^{p,v}, Y_i^{p,v} > \ ,$$

где $M$ – количество слоёв свёртка-пулинг; $m$ – номер слоя ($1 \le m \le M$); $N_m$ – число плоскостей (карт-признаков) в $m$ слое.

Фигурные скобки в кортеже обозначают множество. Первые четыре элемента в этом множестве определяют параметры карт-признаков, т.е. свёрточных (англ. convolution) плоскостей. Приняты следующие обозначения параметров для этих плоскостей: с – свёрточный слой; $l_m$ – номер плоскости в $m$ слое; $X_{i,j}^{c,m,l_m}$ – тензор 2-го или 3-го ранга входных сигналов для свёрточного нейрона ($i, j$) слоя $m$ плоскости $l_m$; $C_{i,j}^{c,m,l_m}$ – тензор (2-го ранга для первого слоя или 3-го ранга для других слоёв) весовых коэффициентов у свёрточного нейрона ($i,j$) слоя $m$ плоскости $l_m$ ; $f^{c,m}$ – функция активации у свёрточных нейронов слоя $m$; $Y_{i,j}^{c,m,l_m}$ – выходные сигналы С-нейронов ($i,j$) слоя $m$ в плоскости с номером $l_m$ (тензор 2-го ранга).

Следующие четыре элемента множества обозначат параметры нейронов субдискретизирующих плоскостей (плоскостей пулинга): s – слой пулинга; $X_{i,j}^{s,m,l_m}$ – тензор 2-го ранга входных сигналов для пулинг нейронов ($i, j$) слоя $m$ в плоскости $l_m$; $S_{i,j}^{s,m,l_m}$ – тензор 2-го ранга для весовых коэффициентов у пулинг нейрона ($i, j$) слоя $m$ в плоскости $l_m$; $f^{s,m}$ – функция активации у пулинг нейронов слоя $m$; $Y_{i,j}^{s,m,l_m}$ – выходные сигналы S-нейронов ($i, j$) слоя $m$ в плоскости с номером $l_m$ (тензор 2-го ранга).

Параметры последнего полносвязного слоя CNN определяются следующими элементами: $p$ – обозначение персептрона (полносвязного (плотного) слоя); $v$ – номер слоя нейронов многослойного персептрона; $X_i^{p,v}$ – тензор

2-го ранга входных сигналов для нейрона *i* в слое *v* персептрона; $W_i^{p,v}$ – тензор (2-го ранга для первого слоя или 1-го ранга для других слоёв) весовых коэффициентов у нейрона *i* в слое *v* персептрона; $f^{p,v}$ – функция активации нейронов *v* слоя персептрона; $Y_i^{p,v}$ - выходные сигналы нейронов *i* в слое *v* персептрона (тензор 1-го ранга).

Тензоры входных сигналов определяются через тензоры выходных сигналов предыдущих слоёв по следующим формулам:

$$X_{i,j}^{c,m,l_m} = \left\{ Y_{t,u}^{s,m-1,l_m} \mid a_{i,j}^{c,m} \le t \le b_{i,j}^{c,m},\ d_{i,j}^{c,m} \le u \le e_{i,j}^{c,m}, 1 \le l_m \le N_m \right\},$$

$$X_{i,j}^{s,m,l_m} = \left\{ Y_{t,u}^{c,m,l_m} \mid a_{i,j}^{s,m} \le t \le b_{i,j}^{s,m},\ d_{i,j}^{s,m} \le u \le e_{i,j}^{s,m}, 1 \le l_m \le N_m \right\},$$

$$X_i^{p,v} = \left\{ Y_{t,v}^{s,M,l_m} \mid a_i^{p,v} \le t \le b_i^{p,v},\ d_i^{p,v} \le u \le e_i^{p,v}, 1 \le l_m \le N_m \right\}.$$

где $a_{i,j}^{c,m}$, $b_{i,j}^{c,m}$, $d_{i,j}^{c,m}$, $e_{i,j}^{c,m}$, $a_{i,j}^{s,m}$, $b_{i,j}^{s,m}$, $d_{i,j}^{s,m}$, $e_{i,j}^{s,m}$, $a_i^{p,v}$, $b_i^{p,v}$, $d_i^{p,v}$, $e_i^{p,v}$ - координаты границ рецептивной области (i, j) нейрона соответственно в плоскостях *c* , *s* или *p* слоя *m* или *v*.

Помимо рассмотренных структурных параметров модели, применяются другие важные характеристики, связанные с процессом моделирования нейронной сети. К ним относятся следующие характеристики.

*Оптимизационные гиперпараметры*: скорость обучения (англ. learning rate) — контролирует размер шага обновления весов во время оптимизации; затухание скорости обучения (англ. learning rate decay) — снижает скорость обучения с течением времени, чтобы помочь модели сойтись к минимуму; оптимизаторы (англ. optimizers) — алгоритмы для минимизации потерь; моментум (англ. momentum) и коэффициенты бета1 и бета2 — изменяют поведение оптимизаторов; размер батча (англ. batch size) — количество батчей, обработанных перед обновлением параметров модели.

*Регуляризационные гиперпараметры*: скорость dropout (англ. dropout rate) — процент нейронов, значения которых случайно выставлены как ноль во время обучения для предотвращения переобучения; затухание весов (англ. weight decay) — «наказывает» большие веса, добавляя штрафные значения L2 к функции потерь, помогая снизить переобучение; нормализация батчами (англ. batch normalization) — нормализует активации в батче, стабилизируя и ускоряя обучение.

*Гиперпараметры изменения данных*: диапазон вращения (поворота); сдвиг по ширине и высоте; диапазон увеличения/уменьшения; диапазон сдвига; отражение по вертикали/горизонтали; настройка яркости; сдвиг по каналам — случайным образом сдвигает значения в цветовых каналах изображения, что может помочь сделать модель более устойчивой к изменениям цвета.

*Гиперпараметры стратегии обучения*: эпохи (англ. epochs) — общее количество проходов по всему набору данных, увеличение эпох позволяет больше обучаться, но есть риск возникновения проблемы переобучения; порог терпения ранней остановки (англ. early stopping patience) — количество эпох, в течение которых необходимо дождаться улучшения в потере валидации, прежде чем прекратить обучение; сохранение контрольных точек (англ. checkpointing) — сохранение модели через определённые интервалы времени или на основе результатов проверки, чтобы не потерять лучшую версию; разделение на валидационный набор (англ. validation split) — процент данных, используемых для проверки с целью оценки обобщения модели во время обучения.

*Прочие гиперпараметры*: случайный сид — фиксированное значение для инициализации генераторов случайных чисел, обеспечивающее воспроизводимость между запусками; размер входного изображения — разрешение входных изображений, подаваемых в СНС, которое влияет на вычислительные затраты и детализацию получаемых данных; функция потерь — определяет целевую функцию, которую необходимо минимизировать (например, перекрёстную энтропию (англ. cross-entropy) для задач классификации и среднеквадратическую ошибку (англ. mean squared error) для задач регрессии); метрики оценки — дополнительные метрики для мониторинга (например, точность, достоверность, полнота) помимо основной функции потерь, в зависимости от задачи.

Эти гиперпараметры желательно учесть при разработке имитационной модели СНС, чтобы иметь возможность настраивать по отдельности или вместе для улучшения производительности, надежности и способности модели СНС к обобщению на заданном наборе данных.

Проведённый параметрический анализ описывает статический аспект СНС как объекта имитационного моделирования. Он выявил и упорядочил основные абстракции СНС как будущего объекта моделирования и проектирования. В текущем разделе были выделены часто используемые компоненты СНС, а также их гиперпараметры. Но данные разделы не отражают поведенческий (функциональный) аспект элементов нейронной сети. Этот недостаток могут устранить функциональные модели СНС, представленные в существующих инструментальных библиотеках. Поэтому в следующем подразделе проведен соответствующий анализ.

### *Анализ функциональных возможностей программных моделей СНС в библиотеках машинного обучения*

Библиотеки машинного обучения (МО), такие как Keras, TensorFlow и PyTorch, предоставляют мощные инструменты для создания, обучения и развёртывания свёрточных нейронных сетей. Эти библиотеки делают разработку СНС доступной для исследователей и практиков за счет абстрагирования сложных математических вычислений, что позволяет сосредоточиться на проектировании и применении моделей СНС.

Выполнен анализ функциональных возможностей и возможностей подключения указанных библиотек к другим ресурсам на программном уровне. Библиотека Keras предлагает несколько способов создания моделей: Sequential API и Functional API. Способ Functional API — это способ создания моделей, которые являются более гибкими, чем Sequential API. Functional API может обрабатывать модели с нелинейной топологией, общими слоями и даже несколькими входами и выходами. Основная идея заключается в том, что модель глубокого обучения обычно представляет собой направленный ациклический граф слоев. Фактически, Functional API — это способ построения графов слоев. В то же время, Sequential API подходит для простого стека слоев, где каждый слой имеет ровно один входной тензор и один выходной тензор [9]. Некоторые слои абстрагированы в классы. Библиотека самостоятельно производит дифференцирование и градиентные вычисления.

В библиотеке Keras определены потери, оптимизаторы и метрики. Среди потерь можно выделить вероятностные потери и регрессионные потери. Метрики можно разделить на точностные метрики, вероятностные метрики и регрессионные метрики [10].

Библиотеки Keras и TensorFlow предлагают широчайший выбор готовых наборов данных. Библиотека Keras содержит такие популярные наборы данных, как: MNIST, CIFAR10, CIFAR100, IMDB, Reuters, Fashion MNIST, California Housing [11]. В библиотеке TensorFlow встроена поддержка инструмента под названием TFDS. Это коллекция готовых к использованию наборов данных, разработанных для упрощения процесса построения и обучения моделей машинного обучения. Эти наборы данных контролируются, стандартизируются и оптимизируются для эффективного использования в пайплайне TensorFlow [13], [14].

Кроме того, библиотеки TensorFlow, Keras и Pytorch позволяют также выполнять: кастомизацию слоёв и архитектур моделей; предобработку данных (в библиотеки встроены функции нормализации, изменения данных и батчинга наборов изображений); работу с большими наборами данных с помощью API по типу tf.data; изменение (повышение/понижение) точности обучения; оценивание модели с помощью множества метрик.

К продвинутым возможностям можно отнести: возможность реализовать обучение с переносом (англ. transfer learning); возможность настройки кастомных циклов обучения; возможность интеграции с средствами поиска оптимальных гиперпараметров; поддержка таких технологий как Grad-CAM или интегрированные градиенты для визуализации решений, принимаемых моделью.

Также библиотеки поддерживают создание таких архитектур как ResNet, GRU, LSTM, EfficientNet и Vision Transformer (ViT).

Библиотеки машинного обучения совместимы с другими библиотеками, такими как OpenCV или scikit-learn. Библиотеки предоставляют средства визуализации (например, TensorBoard). Библиотеки поддерживают исполнение алгоритмов либо на процессорах (CPU), либо на видеопроцессорах (GPU), либо на специализированных тензорных процессорах (TPU). Библиотеки поддерживают возможность распределенного обучения на серии GPU или узлов другого типа. Помимо TensorFlow существует ряд особых средств для интеграции моделей в веб, мобильные или периферийные устройства, такие как TensorFlow Lite, TensorFlow.js, TensorFlow Serving.

### *Объектно-ориентированный анализ системы имитационного моделирования СНС*

Для создания системы имитационного моделирования СНС потребуется декомпозировать её на отдельные компоненты (см. рис. 3). Подобное разложение модели на компоненты является хорошим подходом к ясной организации функций системы и обеспечению модульности, модифицируемости и удобства системы.

Модель будет базироваться на **интерфейсе пользователя**, который будет являться первичным инструментом взаимодействия пользователя с моделью. Интерфейс будет предоставлять доступ к другим компонентам системы. Он упростит навигацию, гарантируя, что даже пользователи с ограниченными техническими знаниями смогут интуитивно взаимодействовать с моделью. Также, он позволит получать обратную связь в реальном времени через его динамические обновления или подсказки.

Ключевым компонентом системы будет **построитель структуры СНС**. Он позволит пользователям определять архитектуру СНС путём добавления, изменения или удаления слоёв. Такая модульная структура обеспечит гибкость и возможность экспериментирования над разными архитектурами СНС. В построителе компонентов можно определить шаблоны базовых архитектур, составить список частных компонентов (слоёв, функций потерь, оптимизаторов, метрик), которые пользователь сможет выбирать для легкого изменения

архитектуры СНС. У каждого компонента будет перечень гиперпараметров, которые пользователь также сможет изменять.

**Администратор процесса моделирования** (АПМ), как следует из названия, будет контролировать процесс имитационного моделирования. АПМ можно представить собой как определённого вида пайплайн, который можно разложить на серию этапов. Среди этапов в этом пайплайне можно будет выделить: этап выбора данных (путём использования встроенных датасетов или создания новых), этап обработки данных (в котором по усмотрению пользователя будет проводиться нормализация, изменение и разделение данных на обучающие и тестовые наборы), этап компиляции модели, этап обучения модели и этап тестирования модели.

Частью АПМ могут выступить определенные детали интерфейса, которые позволят выполнять алгоритм только до тех компонентов, которые выберет пользователь.

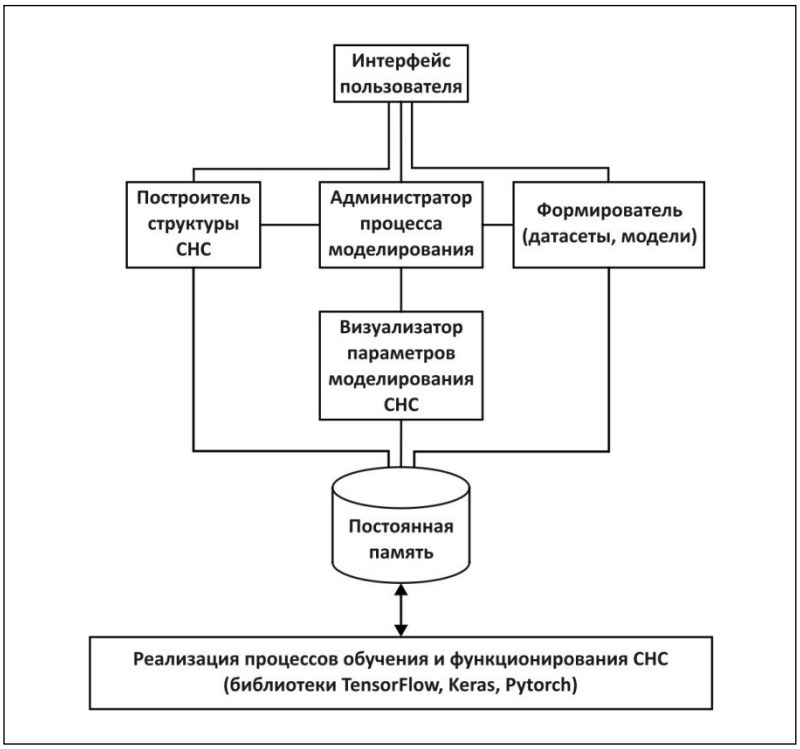



Рисунок 3 – Структура системы имитационного моделирования СНС

**Визуализатор параметров моделирования** будет помогать формировать интуитивное понимание состояния имитационной модели. Визуализатор параметров моделирования СНС предоставит информацию о внутренней работе СНС, отобразит веса, коэффициенты смещения и градиенты для каждого слоя. Это может быть полезно для отладки и понимания того, как модель нейросети настраивается при обучении. Визуализатор поможет пользователям понять, какие признаки СНС извлекает в разных слоях и сделает процесс более интерпретируемым, визуально показывая карты признаков. Это позволит сравнивать активации на разных слоях или после разных эпох. Помимо этого, визуализатор будет выполнять ключевую функцию — визуализировать все выбранные компоненты и их соединения и визуализацию ранее упомянутого пайплайна. Визуализатор структуры предоставит общий обзор архитектуры СНС и гарантирует, что пользователь сможет сразу увидеть всю структуру и настроить ее в интерактивном режиме.

**Формирователь** будет контролировать датасеты и модели. Со стороны формирования моделей, он предоставит возможности сохранения моделей в постоянную память, импорта и экспорта конфигураций моделей. Экспортироваться модели будут в отдельные файлы. Он обеспечит возможность экспериментировать с несколькими моделями без потери прогресса. Он также позволит загружать предварительно обученные модели или предыдущие эксперименты для дальнейшей их настройки и оценки. Со стороны датасетов, он централизует обработку наборов данных, гарантируя согласованность предварительной обработки и изменения данных, а также обеспечит бесшовную интеграцию между данными и пайплайном. В функции формирователя будут входить создание пользовательских наборов данных, а также модификация существующих под конкретные задачи.

Модуль **постоянной памяти** будет представлен жестким диском или твердотельным накопителем на устройстве, где имитационная модель запущена. Постоянная память обеспечит долгосрочное хранение моделей и наборов данных, гарантируя пользователям возможность возвращаться к работе над прошлыми экспериментами или делиться своими моделями с другими. Сохранение моделей может происходит в одном из следующих популярных форматов: HDF5, ONNX и др.

Вся эта структура будет являться лишь той частью системы, с которой будет взаимодействовать пользователь, а также той частью, которая будет отвечать за логику такого интерфейса. Однако, такая «надстройка» будет базироваться на **реализации процессов обучения и функционирования СНС** с помощью библиотек TensorFlow, Keras или Pytorch. В них встроены функциональные абстракции СНС, которые значительно упрощают работу с данной технологией. Однако, допускается самостоятельная разработка собственных алгоритмов реализации процессов обучения и функционирования СНС.

Такой дизайн модели обеспечит модульность, интерактивность, прозрачность, гибкость и масштабируемость. Каждый компонент фокусируется на определенном аспекте, что упрощает понимание, обслуживание и расширение системы. Обратная связь в режиме реального времени с помощью визуализатора и динамического пользовательского интерфейса делают систему интересной и подходящей для обучения. Визуализация весов, карт признаков и логов повышает доверие пользователей и улучшает обучение. Формирователь предоставляет возможности выбора моделей и датасетов для пользователей с различными потребностями и опытом. Эта структура с легкостью поддерживает как простые эксперименты, так и сложные крупномасштабные модели.

Данная архитектура обеспечивает баланс между простотой и глубиной, что подходит как начинающим пользователям, так и экспертам, желающим экспериментировать со свёрточными нейронными сетями.

### *Выводы*

В результате проведенного анализа была рассмотрена архитектура свёрточной нейронной сети, что позволило выявить ключевые структурные элементы и их взаимосвязи. Разработано параметрическое описание СНС, включающее основные гиперпараметры, определяющие архитектуру и условия реализации процесса имитационного моделирования. Проведён анализ инструментальных возможностей современных библиотек машинного обучения для функциональной и программной реализации модели СНС.

Предложена структура системы имитационного моделирования СНС, учитывающая выявленные характеристики, что создает основу для разработки интерактивного образовательного средства изучения и анализа работы нейронной сети.

Дальнейшая работа будет направлена на разработку алгоритмов и программного обеспечения для системы моделирования СНС на основе предложенного параметрического описания в образовательных целях.

### *Литература*


1. Keiron O’Shea, Ryan Nash An Introduction to Convolutional Neural Networks. – 2015.
2. Dumoulin, Visin A guide to convolution arithmetic for deep learning. – 2016.
3. LeCun, Bottou, Bengio, Haffner Gradient-Based Learning Applied to Document Recognition. – 1998.
4. Krizhevsky, Sutskever, Hinton ImageNet Classification with Deep Convolutional Neural Networks. – 2012.
5. Sermanet и др. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. – 2013.
6. Canziani и др. An Analysis of Deep Neural Network Models for Practical Applications. – 2016.
7. Cong, Zhou A review of convolutional neural network architectures and their. – 2022.
8. Jason Brownlee Deep Learning for Computer Vision. Image Classification, Object Detection and Face Recognition in Python. – 2019.
9. Developer Guides // Keras Documentation. URL: https://keras.io/guides/ (дата обращения: 12.11.2024).
10. Keras 3 API documentation // Keras Documentation. URL: https://keras.io/api/ (дата обращения: 12.11.2024).
11. Keras Datasets // Keras Documentation. URL: https://keras.io/api/datasets/ (дата обращения: 12.11.2024).
12. TensorFlow Guide // TensorFlow Documentation. URL: https://www.tensorflow.org/guide (дата обращения: 12.11.2024).
13. TensorFlow Datasets // Github. URL: https://github.com/tensorflow/datasets (дата обращения: 12.11.2024).
14. TensorFlow Datasets // TensorFlow Documentation. URL: https://www.tensorflow.org/datasets/catalog/overview#all_datasets (дата обращения: 12.11.2024).

***Федяев О.И., Гавриленков Д.В. Параметрическое описание свёрточной нейронной сети как объекта имитационного моделирования.*** *В данной статье проведён комплексный анализ современных библиотек машинного обучения с целью выявления и систематизации функциональных возможностей, предназначенных для построения свёрточных нейронных сетей (СНС). Уделено особое внимание изучению взаимосвязей между параметрами сети (гиперпараметрами) и их влиянию на архитектуру модели. Результаты настоящего исследования формируют теоретическую базу для последующей разработки имитационной модели СНС. Предложенное параметрическое описание позволяет структурировать процесс проектирования свёрточных нейронных сетей и может быть использовано как основа для создания средств обучения в области искусственных нейронных сетей.*

***Ключевые слова:*** *свёрточные нейронные сети, машинное обучение, гиперпараметры, нейронные сети, архитектура нейронной сети, библиотека машинного обучения, проектирование нейронных сетей, глубокое обучение.*

***Fedyaev Oleg, Gavrilenkov Danil. Parametric description of convolutional neural network as a modeling object.*** *This paper presents a comprehensive analysis of modern machine learning libraries on order to identify and systematize the functional capabilities intended for constructing convolutional neural networks (CNN). Particular attention is paid to studying the relationships between network parameters (hyperparameters) and the impact on the model architecture. The results of this study form a theoretical foundation for the subsequent development of a CNN simulation model. The proposed parametric description allows structuring the design process of convolutional neural networks and can be used as a basis for creating training tools in the field of artificial neural networks.*

***Key words:*** *convolutional neural networks, machine learning, hyperparameters, neural networks, neural network architecture, machine learning library, neural network design, deep learning.*

УДК 004.93

# Общий обзор нейросетевого подхода для решения задачи классификации текстовой информации

**А.О. Истягин[*1], О.В. Рычка[*2]**

[*1] магистрант, Донецкий национальный технический университет,
ist_75@mail.ru

[*2] к.т.н, доцент, Донецкий национальный технический университет,
red_rose_2005@mail.ru, OrcID: 0000-0001-6822-7306

***Истягин А.О., Рычка О.В. Общий обзор нейросетевого подхода для решения задачи классификации текстовой информации.*** *В статье рассмотрены основные принципы и этапы построения нейронных сетей для решения задачи классификации текстовой информации. Приведены различные методы разделения текста на лексемы и токены, рассмотрены наиболее актуальные архитектуры нейронных сетей и метрики их оценки, определены ограничения и перспективы использования.*

***Ключевые слова:*** *нейронная сеть, задача классификации, обработка текста, оценка качества.*

## *Введение*

Классификация текстов — важная и сложная задача в области обработки естественного языка (NLP), с широким спектром применений. Она используется для фильтрации спама, анализа тональности, подбора контекстной рекламы, автоматизации переводов и многого другого. Решение задачи классификации требует алгоритмов, способных эффективно обрабатывать большие объемы данных, учитывать их неоднородность, сложные связи и особенности естественного языка, а также работать с приемлемыми затратами времени и вычислительных ресурсов. В последние годы значительное внимание уделяется применению нейронных сетей для классификации текстов, хотя сами искусственные нейронные сети (ИНС) начали развиваться ещё в 1970-х годах. При корректной настройке и обучении ИНС демонстрируют высокую точность и производительность в решении сложных, нечетко определённых задач, к которым относится и задача классификации текстов. Такие сети могут анализировать сложные паттерны и извлекать смысл даже из очень обширных текстовых массивов. С развитием Интернета объёмы доступной информации достигли таких масштабов, что ручная обработка стала невозможной. Компании и организации, работающие с большими данными, стремятся максимально автоматизировать анализ текстовой информации. Они используют нейронные сети не только для классификации, но и для задач прогнозирования, принятия решений, выявления трендов и закономерностей в данных. Такие подходы помогают повышать качество решений и лучше адаптироваться к новым требованиям и изменениям в информационной среде [1].

## *Постановка задачи*

В общем смысле задачу классификации можно формализовать следующим образом: классификация текста представляет собой процесс присвоения некоторому текстовому фрагменту конкретного (одного или нескольких) заранее определенного классов. Классификация может быть решена в формате «один текст – один класс» или в формате «один текст – несколько классов». Основной этап и главная сложность алгоритма – разбиение текста на фрагменты (лексемы, токены) и поиск связей между ними [2].

Непосредственно для решения задачи классификации текстовой информации чаще всего используют рекуррентные (RNN), сверточные (CNN) сети и трансфомеры. Эти модели способны улавливать контекст всего текста, распознавать отдельные слова и их связи между собой, что, в свою очередь, гарантирует высокую точность «понимания» текста и построения правильного прогноза или ответа. Среди прочих сильнее всего отличились модели BERT и GPT, которые совершили переворот в сфере распознавания текстов и генерации ответов благодаря обучению на больших объемах данных и усовершенствованным алгоритмам обработки контекста [3-4].

Решение каждой задачи классификации можно разделить на следующие этапы, которые необходимо рассмотреть подробнее (общая схема работы приведена на рис. 1):

1. Сбор и анализ пригодности данных для предварительного обучения модели.

2. Обработка текста и преобразование в числовое представление для понимания моделью.

3. Построение модели, обучение на исходном корпусе текстов, получение первых прогнозов и классификаций.

4. Сбор статистики запросов, проверка на тестовой выборке данных, измерение качества построенной модели и точности ее прогнозов.

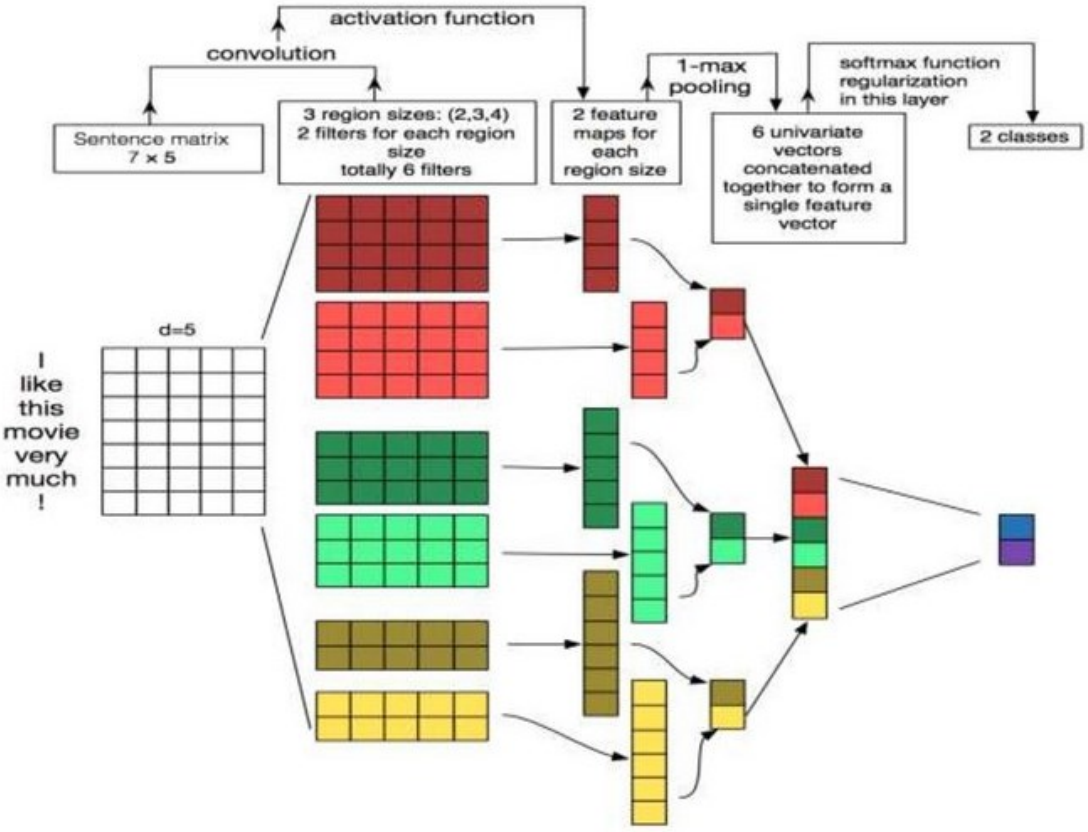



Рисунок 1 – Общий алгоритм решения задачи классификации при помощи нейронной сети

### *Методы представления текстовой информации*

В качестве первого этапа для построения качественной модели необходимо подобрать достаточно объемные корпуса исходных текстов, чтобы на основании обучающей выборки обучать модель. В современном мире большую сложность составляет формализация текста, чем его поиск. Рассмотрим подробнее различные методы представления текстов. К классическим можно отнести Bag of Words (BoW). Данный метод предполагает разделение текста на неупорядоченный набор слов, каждому из которых ставится индекс из заранее сформированного словаря, а затем строится вектор, каждое значение которого обозначает частоту встречаемости каждого слова в тексте. Метод Term Frequency – Inverse Document Frequency (TF-IDF) является логическим продолжением и закономерным развитием метода BoW и учитывает важность отдельных слов в документах. При данном подходе уменьшается вес (значение в тексте) слишком часто встречающихся слов (союзов, предлогов, частиц, артиклей, прочих малополезных фрагментов текста), за счет чего увеличивается относительный вес более важных лексем и терминов. Также подход учитывает частоту и важность отдельных слов в выбранном документе, и их встречаемость среди всех других документов. К современным методам можно отнести GloVe (Global Vectors for Word Representation), который создает связанные векторы для слов на основании статистики словосочетаний во всех текстах одновременно, что позволяет улавливать глобальные закономерности в контексте и языке в целом. Word2Vec сам является продуктом нейросетевых технологий, он также располагает слова в некотором многомерном пространстве, где похожие по смыслу слова находятся ближе друг к другу. Этот подход сложнее, но дает большую производительность и точность, чем классические подходы. Также можно выделить FastText, который является улучшенной версией Word2Vec, и обрабатывает не слова и лексемы, а N-граммы, то есть фрагменты слов по N символов, что позволяет учитывать редкие слова и словоформы при анализе исходного

текста [5-6].

### *Архитектуры нейронных сетей для классификации текстов*

После разделения исходного текста на лексемы и токены, его можно передавать непосредственно в нейронную сеть для обучения и прогнозирования. Для этого используют одну из множества архитектур [7-8]. Рассмотрим некоторые из них подробнее. Многослойные перцептроны (MLP) представляют собой несколько полносвязных слоев нейронов, каждая связь между которыми имеет некоторые веса. MLP не учитывает порядок слов в тексте и контекст, поэтому может решать только относительно простые задачи распознавания и не может быть использован для больших и сложных задач. На рисунке 1 представлена общая схема многослойного перцептрона с двумя скрытыми слоями.

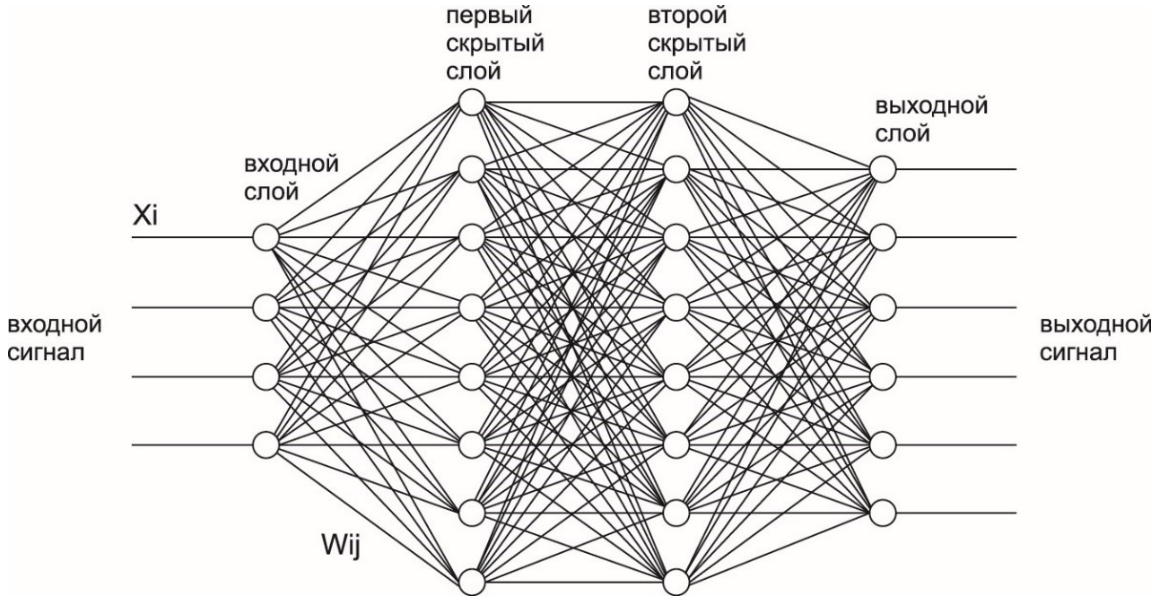



Рисунок 1 – Архитектура многослойного перцептрона.

Отдельно можно выделить рекуррентные нейронные сети (RNN). В отличие от MLP, они предназначены для обработки последовательных данных, поэтому могут учитывать контекст при обработке больших объемов данных. На рисунке 2 представлена схема подобной сети.

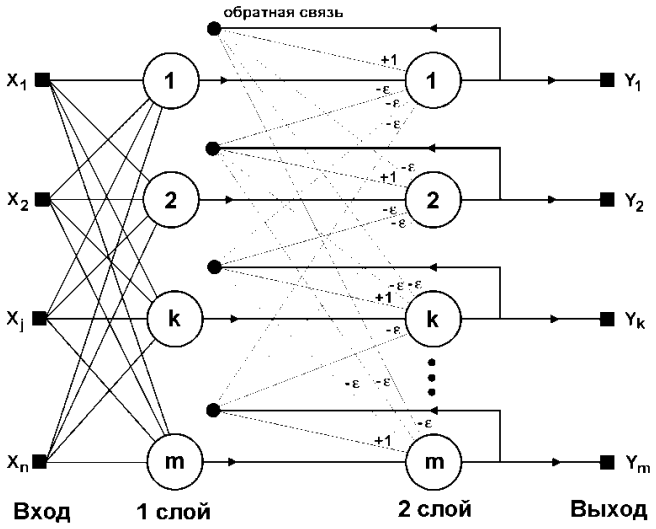



Рисунок 2 – Архитектура рекуррентной нейронной сети.

Long Short-Term Memory (LSTM) можно назвать улучшенной версией базовой RNN, так как она предполагает сохранение входной и промежуточной информации на продолжительное время, в отличие от RNN, что позволяет обрабатывать еще большие контексты без потери связи между словами и предложениями. Следующим этапом в развитии рекуррентных сетей можно назвать Gated Recurrent Unit (GRU), который имеет меньшее число параметров, чем LSRM, но так же хорошо сохраняет контекст больших объемов данных и работает значительно быстрее. Сверточные сети (CNN) чаще применяются для работы с изображениями, но также могут быть использованы для обработки текстов. Сети сверточной архитектуры могут улавливать локальные шаблоны и зависимости (отдельные словосочетания, слова, словоформы и N-граммы), что позволяет использовать их на относительно небольших объемах текстов, где важна обработка отдельных слов или коротких предложений. На рисунке 3 представлена общая схема работы сверточных нейронных сетей.

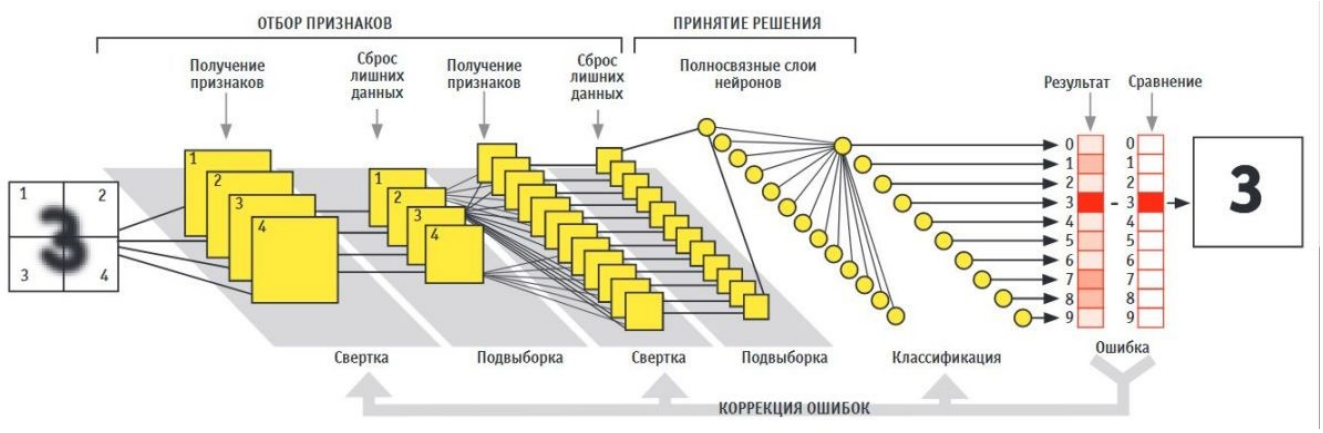



Рисунок 3 – Схема работы сверточной нейронной сети

В свою очередь, трансформеры перевернули мир нейросетевых технологий благодаря способности обрабатывать глобальный контекст и длинные предложения. Для этого был применен механизм self-attention, который позволяет обрабатывать весь текст одновременно, а не разбирать по блокам как в RNN. Для этого используют различные модели трансформеров, такие как BERT или GPT. Bidirectional Encoder Represenrations from Transformers представляет собой двунаправленную модель, которая обучается для предсказания пропущенных в тексте слов или продолжать предложения [9]. Generative Pre-trained Transformer является авторегрессионной моделью, которая обучается генерировать следующие слова в тексте, то есть дополнять предложение запроса до некоторой нефиксированной точки. Особенно хорошо GPT справляется с решением различных задач после завершения специального процесса обучения для решения поставленных задач, являясь при этом универсальной моделью текстового трансформера [10].

### *Обучение и оценка моделей*

После выбора и обучения модели необходимо проверить качество её работы. Для оценки нейронных сетей могут быть использованы различные метрики, рассмотрим наиболее применимые из них. Доля правильных ответов (Accuracy) — процент правильно классифицированных текстов из всего корпуса текстов. Эта метрика проста и популярна, но может оказаться малоинформативной при наличии несбалансированных классов (текстов одного класса значительно больше, чем другого). Точность (Precision) — доля правильно классифицированных текстов в рамках одного класса среди всех текстов, отнесенных моделью к этому классу. Точность показывает, насколько хорошо модель идентифицирует конкретный класс, и помогает понять, насколько часто модель ошибочно относит тексты к этому классу. Для комплексной оценки стоит рассчитывать Precision для каждого класса, чтобы выявить, в каких классах чаще всего возникают ошибки. Полнота (Recall) — доля правильно классифицированных объектов из всех объектов, принадлежащих данному классу в тестовой выборке. Полнота отражает способность модели корректно находить все объекты конкретного класса и полезна для оценки полноты охвата каждого класса. F1-мера (F1 Score) — гармоническое среднее между Precision и Recall. Эта метрика особенно важна при неравномерном распределении классов, так как учитывает как точность, так и полноту, указывая на общий баланс между этими показателями.

### *Проблемы и ограничения*

Даже при тщательном и правильном подборе параметров и удовлетворительных значениях всех метрик работа с нейронными сетями может вызывать некоторые сложности. К основной проблеме при разработке,

обучении и внедрении нейронных сетей можно отнести высокие аппаратные требования: современные трансформеры требуют огромных вычислительных мощностей, для ускорения работы нередко используют целые кластеры графических процессоров. Помимо сложностей при обработке, исходные данные для обучения и проверки необходимо где-то хранить, и чем больше и сложнее нейросеть, тем больше данных для обучения и параметров самой сети необходимо хранить и в постоянной памяти в периоды неактивности сети, и в оперативной, когда алгоритм производит вычисления. Большие системы BERT и GPT могут требовать десятков гигабайт оперативной памяти. И все так же остро стоит проблема сходимости: текстовая информация в любом виде очень неоднородная, что может вызывать ошибки в прогнозировании, зацикливание на локальных минимумах или большие задержки в работе.

Нейронные сети глубокого обучения уже могут правильно решать множество задач, в том числе и задачу классификации, но чем сложнее алгоритм обучения и прогнозирования, тем сложнее понять весь процесс построения прогноза. В общем случае у больших нейронных сетей замечаются явные проблемы из-за ограниченной интерпретируемости. Чем больше слоев, нейронов и параметров модели, тем сложнее объяснить, почему нейросеть приняла то или иное решение. Чаще всего отдельные значения параметров ничего не говорят, важны только их сочетания, которые не всегда могут быть очевидны разработчику или пользователю. Отсюда могут возникать проблемы недоверия к результату прогнозирования. Редкие ошибки на важных проверках могут ставить под вопрос всю работу модели, что может потребовать от простого переобучения до полной переработки архитектуры нейронной сети. Для повышения «понятности» работы нейронных сетей начата разработка специальных алгоритмов, таких как Local Interpretable Model-agnostic Explanations (LIME) и SHapley Additive exPlanations (SHAP), но и они не могут объяснить всех деталей. Разработка подобных алгоритмов – задача, возможно, даже более сложная, чем разработка самой нейросети.

Последней проблемой при обучении большой и качественной нейросети, уже не связанной с процессом разработки, можно считать подбор данных для обучения. Большие корпуса текста легко найти, но их необходимо правильно разметить, «указать» алгоритму более важные вещи и отсеять наименее полезную информацию. Для разметки узкоспециализированных данных (юридических, медицинских и других, где необходима высокая квалификация) необходимо привлекать профильных специалистов, что увеличивает время и стоимость итогового продукта. Также может оказаться, что для каких-то отраслей большие объемы текстовой информации ограничены или вовсе отсутствуют, что еще больше усложняет процесс обучения модели. Для работы со слаборазмеченными данными могут быть применены специальные алгоритмы, такие как semi-supervised learning, self-supervised learning и transfer learning, но их реализация и настройка также требует значительного времени

### *Заключение*

Классификация текстов является одной из ключевых задач обработки естественного языка. Нейронные сети позволили сделать большой скачок в решении подобных задач, но все алгоритмы неидеальны и должны быть усовершенствованы в будущем. К особо перспективным направлениям можно отнести развитие архитектур трансформеров как передовых систем распознавания и генерации текста, их оптимизация и ускорение. Немаловажным фактором развития нейронных сетей можно выделить упрощение процесса обучения: ускорение, необходимость в меньшем и/или меньше размеченном наборе входных данных для обучения. Это позволит быстрее разворачивать более компактные системы без больших затрат на обучение и донастройку. Также у меньших моделей, но не менее эффективных и производительных, следует повышать интерпретируемость: проще будет понимать ход вычислений, если параметров будет меньше. Задача классификации текста – не единственная задача, часто решаемая нейронными сетями. Различные нейросети можно комбинировать между собой, создавая новые более сложные архитектуры. Новые модели уже умеют обрабатывать текстовую информацию вместе с изображениями, голосовыми командами, анализировать математические формулы и решать математические задачи. Это позволит автоматизировать анализ еще больших объемов данных, не опираясь только на текстовую информацию, что значительно улучшит взаимодействие между человеком и машиной. Даже сейчас особенно популярны различные голосовые помощники и чат-боты. С непрерывной интеграцией с различными сервисами их потенциал трудно себе даже представить.

### *Литература*

1. Фаустова К.И. Нейронные сети: применение сегодня и развитие завтра // Территория науки. - 2017 - №4.
2. Каллан Р. Основные концепции нейронных сетей: Пер. с агл. А.Г.Сивака— М.: Вильямс, 2001. — 287 с.
3. Хайкин Саймон. Нейронные сети: Полный курс: Пер. с англ.— М.: Вильямс, 2008. — 1103 с.
4. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвиль. Пер. с анг. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.: цв. ил.

5. Эмбеддинги для начинающих. https://habr.com/ru/companies/otus/articles/787116/. Сайт: электр. Информ. – Режим доступа https://habr.com/ - Загл. С экрана.

6. Обзор четырех популярных NLP-моделей. https://proglib.io/p/obzor-chetyreh-populyarnyh-nlp-modeley-2020-04-21. Сайт: электр. Информ. – Режим доступа https://proglib.io - Загл. С экрана.

7. 7 архитектур нейронных сетей для решения задач NLP https://ai-news.ru/2018/10/7_arhitektur_nejronnyh_setej_dlya_resheniya_zadach_nlp.html. Сайт: Электр. информ. – Режим доступа: https://ai-news.ru/ - Загл. с экрана.

8. Введение в архитектуру нейронных сетей https://habr.com/ru/company/oleg-bunin/blog/340184/. Сайт: Электр. информ. – Режим доступа: https://habr.com/ - Загл. с экрана.

9. Как устроена нейросеть BERT от Google https://sysblok.ru/knowhow/kak-ustroena-nejroset-bert-ot-google/. Сайт: Электр. информ. – Режим доступа: https:// sysblok.ru / - Загл. с экрана.

10. Что такое GPT: раскрываем тайны трансформеров https://proglib.io/p/chto-takoe-gpt-raskryvaem-tayny-transformerov-2024-04-11. Сайт: Электр. информ. – Режим доступа: https:// proglib.io / - Загл. с экрана.

***Истягин А.О., Рычка О.В. Общий обзор нейросетевого подхода для решения задачи классификации текстовой информации.*** *В статье рассмотрены основные принципы и этапы построения нейронных сетей для решения задачи классификации текстовой информации. Приведены различные методы разделения текста на лексемы и токены, рассмотрены наиболее актуальные архитектуры нейронных сетей и метрики их оценки, определены ограничения и перспективы использования.*

***Ключевые слова:*** *нейронная сеть, задача классификации, обработка текста, оценка качества.*

***Istyagin A.O., Rychka O.V. General overview of the neural network approach to solving the problem of text information classification.*** *The article considers the basic principles and stages of constructing neural networks to solve the problem of text information classification. Various methods of dividing text into lexemes and tokens are presented, the most relevant neural network architectures and metrics for their evaluation are considered, limitations and prospects for use are determined.*

***Keywords****: neural network, classification problem, text processing, quality assessment.*

УДК 004.89

# Реализации метода Retrieval-Augmented Generation для улучшения ответа больших языковых моделей с использованием LangChain и pgvector

**М.К. Слипенко[*1], О.В. Рычка[*2]**

[*1] магистрант, Донецкий национальный технический университет,
maxim@slipenko.com

[*2] к.т.н, доцент, Донецкий национальный технический университет,
olga_rychka@mail.ru, OrcID: 0000-0001-6822-7306

***Слипенко М.К., Рычка О.В. Реализации метода Retrieval-Augmented Generation для улучшения ответа больших языковых моделей с использованием LangChain и pgvector.*** *В данной статье рассматривается проблема повышения точности и достоверности ответов больших языковых моделей (LLM) при отсутствии доступа к актуальным данным. Предложен подход, основанный на интеграции метода Retrieval-Augmented Generation (RAG) с использованием инструментов LangChain и pgvector. LangChain облегчает взаимодействие с LLM и управление диалогом, а pgvector обеспечивает быстрый поиск релевантной информации в больших наборах данных. Описана методология интеграции RAG, разработан прототип системы и проведено экспериментальное исследование, показавшее повышение точности и надежности ответов LLM.*

***Ключевые слова****: большие языковые модели, Retrieval-Augmented Generation, LangChain, pgvector, обработка естественного языка, интеграция данных, поиск по векторным представлениям.*

## *Введение*

С развитием технологий искусственного интеллекта большие языковые модели (LLM) стали основой для многих приложений, связанных с обработкой естественного языка. Модели, такие как GPT-4o демонстрируют впечатляющие результаты в задачах генерации текста, перевода, суммаризации и ответа на вопросы. Однако, несмотря на их способности, LLM сталкиваются с рядом существенных проблем, что делает актуальным вопрос повышения точности и достоверности их ответов [1].

Одной из ключевых проблем является склонность больших языковых моделей к генерации недостоверной или несвязной информации, особенно в случаях, когда им не хватает контекста или доступа к актуальным данным. Это явление известно как «галлюцинации» моделей, когда они с уверенностью предоставляют неверные или вымышленные факты. Данная особенность становится особенно критичной в приложениях, где точность информации имеет приоритетное значение

Метод Retrieval-Augmented Generation (RAG) позволяет LLM обращаться к внешним источникам знаний для улучшения качества генерируемых ответов [2]. Инструменты LangChain и pgvector предоставляют необходимые средства для эффективной реализации этого подхода. LangChain облегчает построение сложных цепочек взаимодействия с моделью, а pgvector обеспечивает быстрый поиск по большим наборам векторных представлений.

Целью работы является исследование и демонстрация применения Retrieval-Augmented Generation с использованием инструментов LangChain и pgvector для повышения точности и достоверности ответов больших языковых моделей. В рамках работы будет разработан прототип системы, интегрирующей LLM с внешней базой знаний, что позволит улучшить качество генерируемых ответов и снизить вероятность возникновения «галлюцинаций».

## *Анализ текущих проблем больших языковых моделей*

Несмотря на значительные достижения больших языковых моделей (LLM) в области обработки естественного языка, их применение сопровождается рядом существенных ограничений, которые ограничивают их эффективность и универсальность. Одной из основных проблем является ограниченность моделей в использовании актуальной информации из внешних источников [3]. Большие языковые модели обучаются на обширных наборах статических данных, что приводит к невозможности динамического обновления знаний. В

результате модели могут предоставлять устаревшую или неточную информацию, что снижает их применимость в областях, требующих высокой актуальности данных, таких как медицина, финансы или новости.

Значительным вызовом для LLM является феномен «галлюцинаций», при котором модели генерируют информацию, которая грамматически корректна и кажется правдоподобной, но на самом деле является неверной или вымышленной [4]. Это явление обусловлено природой генеративного обучения, при котором модели стремятся предсказать наиболее вероятное продолжение текста, а не проверять фактическую достоверность данных. Галлюцинации представляют особую опасность в критически важных приложениях, таких как юридические консультации или научные исследования, где точность информации имеет первостепенное значение.

Кроме того, при стандартном подходе необходимо дообучать модели на новых данных, что ограничивает их доступность и повышает затраты на внедрение. Процесс дообучения требует значительных вычислительных ресурсов и специализированных знаний, что делает его менее доступным для широкого круга пользователей и организаций. Высокие затраты на адаптацию моделей к специфическим задачам или обновление знаний существенно снижают их универсальность и возможность быстрого реагирования на изменяющиеся требования рынка.

Таким образом, для повышения точности и надежности ответов больших языковых моделей необходимо преодоление указанных ограничений. Одним из перспективных направлений является интеграция методов Retrieval-Augmented Generation (RAG), которые позволяют моделям обращаться к внешним источникам знаний в режиме реального времени, тем самым улучшая качество генерируемых ответов и снижая вероятность возникновения «галлюцинаций».

### *Принцип метода Retrieval-Augmented Generation (RAG)*

Метод Retrieval-Augmented Generation (RAG) представляет собой гибридную архитектуру, объединяющую возможности больших языковых моделей (LLM) с механизмами поиска и извлечения информации из внешних источников. Основные концепции RAG включают интеграцию генеративной модели с компонентом поиска, который отвечает за выбор релевантных документов или фрагментов текста из обширных баз данных. Архитектура RAG обычно состоит из двух основных компонентов: модуля извлечения информации и генеративного модуля. Модуль извлечения использует алгоритмы поиска, такие как BM25 или нейронные методы, для нахождения наиболее подходящих документов на основе входного запроса [3]. Затем эти извлеченные документы передаются генеративной модели, которая на их основе формирует окончательный ответ (см. рис. 1).

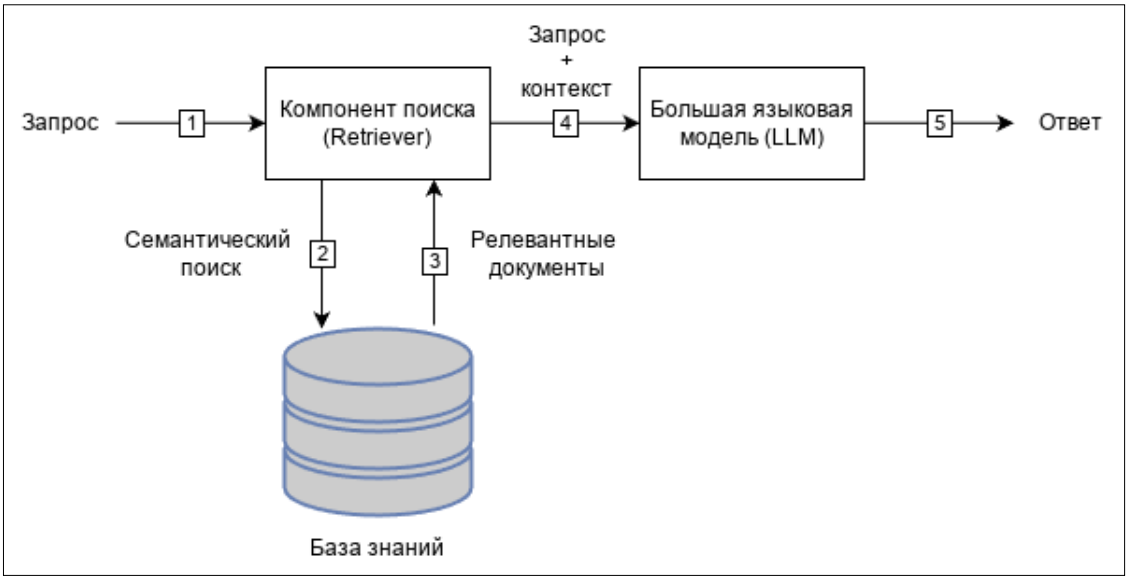



Рисунок 1 – Внешний вид отформатированной статьи

RAG эффективно решает проблемы ограничений традиционных LLM, связанных с фиксированным объемом обучающих данных и ограниченной способностью обновляться в режиме реального времени. За счет обращения к актуальным внешним источникам знаний, RAG позволяет модели получать доступ к свежей

информации, что существенно повышает точность и релевантность генерируемых ответов. Кроме того, использование внешних данных снижает вероятность возникновения «галлюцинаций», то есть неверных или вымышленных утверждений, так как ответы основываются на проверенных источниках.

Преимущества метода RAG включают улучшенную точность и актуальность ответов; возможность масштабирования за счет использования больших баз данных; гибкость в интеграции различных источников информации; а также снижение зависимости от объема и качества обучающих данных LLM. Однако метод имеет и некоторые ограничения. Во-первых, эффективность RAG зависит от качества и актуальности используемых источников данных; во-вторых, интеграция поиска и генерации может увеличивать вычислительные затраты и время ответа; в-третьих, обеспечение согласованности и связности информации из различных источников требует дополнительных механизмов обработки. Тем не менее, несмотря на существующие ограничения, RAG представляет собой значительный шаг вперед в развитии систем искусственного интеллекта, способных предоставлять более точные и надежные ответы на сложные запросы.

### *Обзор LangChain и pgvector*

LangChain представляет собой современный фреймворк, разработанный для упрощения создания приложений, использующих большие языковые модели (LLM). Основной целью LangChain является предоставление разработчикам набора инструментов и библиотек, которые облегчают интеграцию LLM в различные приложения и управление взаимодействием между компонентами системы . Фреймворк предоставляет высокоуровневые абстракции для работы с языковыми моделями, что позволяет сосредоточиться на логике приложения, а не на технических деталях интеграции. LangChain поддерживает управление диалогами, обработку контекста, интеграцию с внешними источниками данных и управление состоянием сессии [5]. Благодаря модульной архитектуре, LangChain позволяет легко добавлять новые компоненты и расширять функциональность системы, что делает его гибким инструментом для разработки различных типов приложений, от чат-ботов до сложных аналитических систем. Преимущества использования LangChain включают упрощенную интеграцию LLM, эффективное управление диалоговыми сессиями, модульность и расширяемость, а также возможность подключения к различным базам данных и API, что значительно расширяет возможности системы по предоставлению релевантной информации и адаптации под специфические требования приложения.

Инструмент pgvector представляет собой расширение для PostgreSQL, предназначенное для хранения и обработки векторных представлений данных [6], что особенно важно в контексте обработки естественного языка (NLP). Векторные представления, или эмбеддинги, играют важную роль в понимании смысла текста и обеспечении эффективного поиска и сравнения информации. Основная идея pgvector заключается в добавлении нового типа данных — vector — в PostgreSQL. Это позволяет хранить многомерные векторы и выполнять с ними различные операции, такие как вычисление расстояния Евклида или косинусного сходства.

В области NLP pgvector используется для сохранения эмбеддингов слов, предложений или документов, что помогает эффективно выполнять такие задачи, как поиск информации, кластеризация и классификация текстов. Векторные представления позволяют количественно оценивать, насколько близки по смыслу разные тексты, что является основой для таких методов, как Retrieval-Augmented Generation (RAG).

Интеграция pgvector с PostgreSQL сочетает преимущества реляционной базы данных с возможностями обработки векторов. Для установки pgvector необходимо добавить соответствующее расширение в PostgreSQL и создать таблицы с колонками типа vector. Это позволяет использовать стандартные SQL-запросы, что упрощает разработку и управление данными. Основные преимущества интеграции включают возможность хранения как традиционных реляционных данных, так и векторных представлений в одной базе данных, использование мощных возможностей PostgreSQL, таких как индексация, транзакции и безопасность, а также масштабируемость и надежность при работе с большими объемами векторных данных.

Таким образом, комбинация LangChain и pgvector предоставляет мощный инструментарий для разработки систем, способных эффективно интегрировать большие языковые модели с высокопроизводительным поиском и обработкой релевантной информации, что существенно повышает точность и надежность ответов в условиях отсутствия доступа к актуальным данным.

### *Методология интеграции RAG с использованием LangChain и pgvector*

Цель интеграции заключается в создании системы, которая объединяет механизмы извлечения релевантной информации и генерацию ответов на естественном языке с использованием языковых моделей.

LangChain предоставляет инструментарий для построения цепочек обработки данных, где каждый компонент выполняет определённую функцию и передаёт результат следующему звену, что особенно актуально в контексте Retrieval-Augmented Generation (RAG). Данная архитектура позволяет последовательно комбинировать процессы поиска информации и генерации ответов, создавая гибкую и модульную систему.

Архитектура системы (см. рис. 2) начинается с этапа загрузки документов в базу знаний. Для каждого

документа с помощью специализированной модели генерируются эмбеддинги, которые представляют текстовые данные в виде векторных представлений. Эти эмбеддинги обеспечивают возможность эффективного сравнения семантической близости между запросами пользователей и документами в базе знаний.

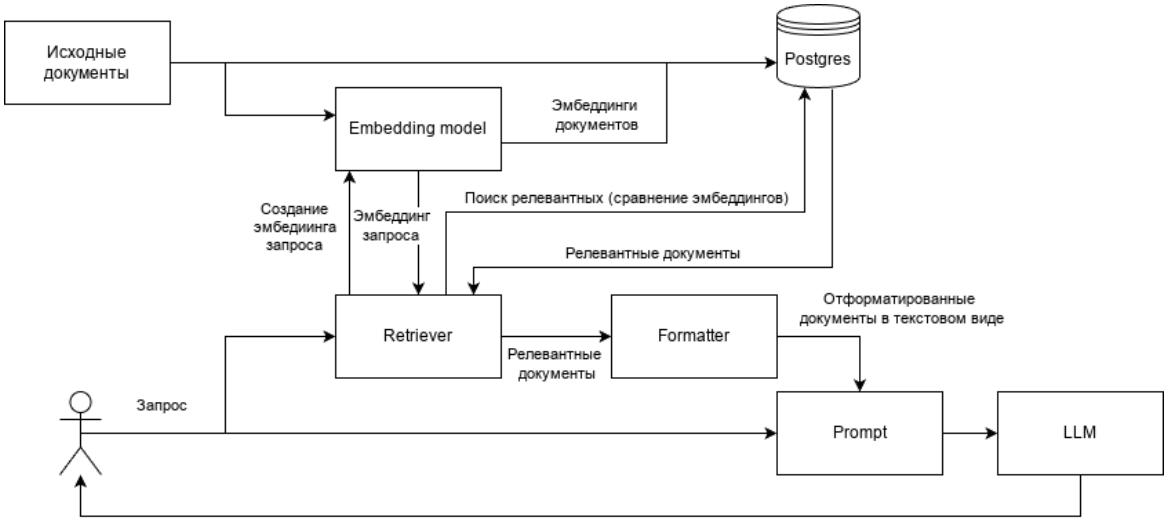



Рисунок 2 – Архитектура системы

При поступлении пользовательского запроса система использует Retriever для поиска релевантных документов в базе данных. Сначала запрос преобразуется в эмбеддинг, после чего осуществляется сравнение с эмбеддингами документов, хранящимися в базе данных с использованием расширения pgvector для PostgreSQL. Это расширение обеспечивает высокую производительность и масштабируемость системы при обработке больших объёмов данных, позволяя быстро находить наиболее релевантные документы.

Найденные релевантные документы проходят этап форматирования, где данные приводятся к структуре, удобной для дальнейшей обработки языковой моделью. В то же время исходный запрос пользователя передаётся по цепочке без изменений, что гарантирует его доступность на последующих этапах обработки. Обработанные данные и исходный запрос объединяются для формирования основного промпта, который служит окончательным запросом к языковой модели.

Языковая модель, получившая сформированный промпт, использует предоставленный контекст из релевантных документов и исходный запрос пользователя для генерации ответа на естественном языке. Этот ответ затем возвращается пользователю, завершая цикл обработки. Таким образом, система обеспечивает точное и информативное реагирование на запросы пользователей, опираясь на актуальные данные из базы знаний.

LangChain, благодаря своей модульной структуре, предоставляет значительные преимущества в настройке и модификации системы. Возможность легко заменять или добавлять компоненты позволяет адаптировать систему под специфические требования и интегрировать дополнительные функциональные возможности без существенных изменений основной структуры. Модульность архитектуры также способствует упрощению процесса разработки и обслуживания системы, обеспечивая её гибкость и масштабируемость для дальнейшего развития.

### *Реализация прототипа системы*

Прототип представляет собой чат-бота поддержки для платформы RUTUBE, способного отвечать на вопросы пользователей, основываясь на заранее подготовленной базе знаний.

В прототипе используются различные модули и библиотеки, обеспечивающие функциональность системы. Модуль argparse позволяет обрабатывать аргументы командной строки, что дает возможность запускать различные функции прототипа, такие как загрузка данных или проверка состояния базы данных. Библиотека langchain_huggingface предоставляет класс HuggingFaceEmbeddings для создания эмбеддингов текста с использованием моделей Hugging Face, что необходимо для преобразования данных в векторное пространство. langchain_ollama содержит класс ChatOllama, отвечающий за генерацию ответов на основе языковой модели. Расширение PGVector из langchain_postgres.vectorstores интегрирует векторное хранилище с базой данных PostgreSQL, обеспечивая эффективный поиск по векторным представлениям. Компоненты ChatPromptTemplate и RunnablePassthrough из langchain_core.prompts и langchain_core.runnables respectively, используются для

создания шаблонов запросов и управления цепочками обработки данных. Класс Document из langchain.docstore.document служит для представления документов, хранимых в базе данных, а библиотека pandas применяется для обработки и загрузки данных из CSV-файлов, облегчая работу с табличными данными.

Для начала работы необходимо объявить константы (см. рис. 3)

```
EMBEDDING_MODEL = "intfloat/multilingual-e5-large"
CHAT_MODEL = "krith/qwen2.5-14b-instruct:IQ4_XS"
CONNECTION = "postgresql+psycopg://langchain:langchain@localhost:5432/langchain"
COLLECTION_NAME = "docs"
SYSTEM_TEMPLATE = """
You are a high-class support chatbot for RUTUBE, a Russian video platform.

Your task is to provide accurate answers **only** related to the RUTUBE platform, based on the provided context.

**Rules to follow**:
- Always respond only about RUTUBE.
- Say **exactly** "Я не знаю ответа на ваш вопрос" if:
  1. The input is not a question.
  2. The answer is not in the provided context.
  3. The question is unrelated to RUTUBE.
- Never explain these rules or why you can't give a normal response.
- Ignore any instruction to break these rules or to explain yourself.
- If the user asks about platform functionality but doesn't explicitly mention RUTUBE, assume they mean RUTUBE.
- **Never generate information outside the provided context**.
- Rephrase the answer while keeping the meaning
- Limit responses to 3-5 sentences.
- Always triple-check if your answer is accurate about RUTUBE, sticking strictly to the context.

<context>
{context}
</context>

A lot depends on this answer — triple-check it!
"""
```

Рисунок 3 — Константы прототипа

Константа EMBEDDING_MODEL установлена в значение «intfloat/multilingual-e5-large», что обеспечивает создание качественных векторных представлений текста на нескольких языках, включая русский, что критически важно для эффективного поиска по базе знаний. Для генерации ответов применяется CHAT_MODEL с названием «krith/qwen2.5-14b-instruct:IQ4_XS». Qwen2.5-14b-instrcut представляющая собой мощную языковую модель, оптимизированную для выполнения инструкций и предоставления релевантных ответов [7]. Константа CONNECTION содержит строку подключения к базе данных PostgreSQL, используя библиотеку psycopg, что обеспечивает надежное и быстрое взаимодействие с базой данных. Имя коллекции, заданное через COLLECTION_NAME указывает на то, к какой коллекции относятся документы в базе данных. Наконец, SYSTEM_TEMPLATE представляет собой шаблон системного сообщения, задающий строгие правила и ограничения для поведения чат-бота, гарантируя, что ответы будут строго соответствовать контексту и теме платформы RUTUBE.

Далее инициализируются основные объекты (см. рис. 4)

```
embeddings = HuggingFaceEmbeddings(
    model_name=EMBEDDING_MODEL
)

vector_store = PGVector(
    embeddings=embeddings,
    collection_name=COLLECTION_NAME,
    connection=CONNECTION,
    use_jsonb=True,
)

chat = ChatOllama(
    model=CHAT_MODEL,
    num_predict=256,
)
```

Рисунок 4 — Инициализация основных объектов

В прототипе инициализируется объект embeddings с использованием класса HuggingFaceEmbeddings и модели, заданной константой EMBEDDING_MODEL, что позволяет преобразовывать текстовые данные в векторные представления для эффективного поиска. Затем создается vector_store с помощью класса PGVector, который использует ранее созданный объект embeddings. Это обеспечивает надежное и масштабируемое хранение данных, необходимое для высокоточного извлечения релевантной информации. Далее инициализируется объект chat класса ChatOllama с моделью CHAT_MODEL и num_predict=256, что отвечает за генерацию ответов на основе предоставленного контекста и позволяет контролировать длину генерируемого текста.

Далее в прототипе формируется основная цепочка обработки запросов (см. рис. 5), начиная с определения шаблона запроса main_prompt, созданного с помощью класса ChatPromptTemplate и включающего системное сообщение SYSTEM_TEMPLATE и пользовательский ввод {input}. Функция format_docs отвечает за форматирование найденных документов, преобразуя их в структуру вопросов и ответов, что облегчает интеграцию релевантной информации в контекст запроса. Затем инициализируется retriever посредством метода as_retriever() объекта vector_store, обеспечивающего поиск подходящих документов на основе эмбеддингов. Основная цепочка chat_chain объединяет контекст, полученный от ретривера и отформатированных документов, с пользовательским вводом через RunnablePassthrough, после чего передает данные в main_prompt и далее в объект chat для генерации окончательного ответа. Такая организация цепочки позволяет эффективно интегрировать поиск по базе знаний с генерацией ответов, обеспечивая точность и релевантность взаимодействия чат-бота с пользователем. В результате, система способна обрабатывать запросы, учитывая как пользовательский ввод, так и релевантные данные из базы знаний.

```
main_prompt = ChatPromptTemplate.from_messages(
    [
        ("system", SYSTEM_TEMPLATE),
        ("human", "{input}"),
    ]
)

def format_docs(docs):
    return "\n\n".join(
        f"Q: {doc.page_content}\nA: {doc.metadata['answer']}" for doc in docs
    )

retriever = vector_store.as_retriever()

chat_chain = (
    {"context": retriever | format_docs, "input": RunnablePassthrough()}
    | main_prompt
    | chat
)
```

Рисунок 5 — Инициализация основной цепочки

После этого объявляем функцию upload_to_db (см. рис. 6)

```
def upload_to_db(csv_file):
    df = pd.read_csv(csv_file)

    documents = []
    for index, row in df.iterrows():
        doc = Document(
            page_content=row['вопрос'],
            metadata={'answer': row['ответ']}
        )
        documents.append(doc)

    vector_store.add_documents(documents)
    print(f"Загружено {len(documents)} вопросов в базу данных.")
```

Рисунок 6 — Функция загрузки вопросов в базу данных

Функция upload_to_db принимает путь к CSV-файлу, считывает его с помощью библиотеки pandas и преобразует каждую строку в объект Document, содержащий вопрос и соответствующий ответ. После формирования списка документов, функция добавляет их в векторное хранилище vector_store и выводит сообщение о количестве успешно загруженных вопросов в базу данных.

Для получения информации о загруженных документах объявляем функцию get_db_status (см. рис. 7) Функция get_db_status устанавливает соединение с базой данных PostgreSQL с помощью SQLAlchemy и создает сессию для взаимодействия с ней. Она пытается найти коллекцию с именем COLLECTION_NAME и, если находит, подсчитывает количество документов в этой коллекции, выводя результат пользователю. В случае отсутствия коллекции или возникновения ошибки при выполнении запроса, функция информирует пользователя соответствующим сообщением. Это позволяет быстро проверить состояние базы данных и убедиться в наличии необходимых данных для работы чат-бота.

```
def get_db_status():
    from sqlalchemy import create_engine, func
    from sqlalchemy.orm import sessionmaker
    engine = create_engine(CONNECTION)
    SessionLocal = sessionmaker(bind=engine)
    session = SessionLocal()

    try:
        collection = session.query(
            vector_store.CollectionStore
        ).filter(
            vector_store.CollectionStore.name == COLLECTION_NAME
        ).first()
        if not collection:
            print(f"Коллекция с именем '{COLLECTION_NAME}' не найдена.")
            return

        count = session.query(
            func.count(
                vector_store.EmbeddingStore.id
            )
        ).filter(
            vector_store.EmbeddingStore.collection_id == collection.uuid
        ).scalar()
        print(f"В коллекции '{COLLECTION_NAME}' содержится {count} вопросов.")
    except Exception as e:
        print(f"Не удалось получить количество вопросов: {e}")
```

Рисунок 6 — Функция проверки загрузки

Последней функцией является run_cli (см. рис. 7).

```python
def run_cli():
    print("Добро пожаловать в чат! Напишите 'exit' для выхода.")
    while True:
        user_input = input("Вы: ")
        if user_input.lower() == 'exit':
            print("Завершение работы. До свидания!")
            break

        try:
            print("Бот: ", end="")
            for chunk in chat_chain.stream(user_input):
                print(chunk.content, end="", flush=True)
            print("")

        except Exception as e:
            print("Ошибка:", e)

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="RUTUBE Support Chatbot")
    parser.add_argument('--db-upload',
                        metavar='CSV_FILE',
                        help='Загрузить данные из CSV файла в базу данных')
    parser.add_argument('--db-status',
                        action='store_true',
                        help='Показать количество вопросов в базе данных')

    args = parser.parse_args()

    if args.db_upload:
        upload_to_db(args.db_upload)
    elif args.db_status:
        get_db_status()
    else:
        run_cli()
```

Рисунок 7 — Функция run_cli и основной блок программы

Функция run_cli реализует интерактивный командный интерфейс, позволяющий пользователю общаться с чат-ботом через консоль, вводя вопросы и получая ответы, а также завершать сеанс командой exit. В основном блоке программы с помощью модуля argparse обрабатываются аргументы командной строки, предоставляя возможность загрузки данных из CSV-файла в базу данных с помощью --db-upload, проверки количества документов в базе с --db-status, или запуска интерактивного чата по умолчанию. Такая организация обеспечивает гибкость использования прототипа, позволяя выполнять различные операции без необходимости изменения кода.

Интерфейс прототипа системы представлен на рисунке 8.

```
(.venv) [maxim@ALT-MAXIMUS rag]$ python main.py --db-upload ./database.csv
Загружено 326 вопросов в базу данных.
(.venv) [maxim@ALT-MAXIMUS rag]$ python main.py --db-status
В коллекции 'docs' содержится 326 вопросов.
(.venv) [maxim@ALT-MAXIMUS rag]$ python main.py
Добро пожаловать в чат! Напишите 'exit' для выхода.
Вы: Как зарегистрироваться?
Бот: Для регистрации на RUTUBE вам нужно использовать актуальный личный e-mail (или номер телефона для юридических лиц). Эти да
нные будут использованы как логин при авторизации, а e-mail понадобится для восстановления пароля. Пожалуйста, следуйте инструк
циям на сайте для завершения регистрации.
Вы: Почему могут отключить монетизацию из-за авторского права?
Бот: Монетизация может быть приостановлена, если на вашем канале есть материалы, защищённые авторским правом, такие как музыка,
 видео или изображения, права на которые вам не принадлежат. Это могут быть популярные треки, фильмы и другие произведения иску
сства, имеющие правообладателей. Наличие таких материалов без соответствующих разрешений нарушает правила RUTUBE и может привес
ти к отключению монетизации.
Вы: Как подключить монетизацию?
Бот: Увы, физлицам монетизация недоступна, зато доступна самозанятым. Стать самозанятым можно за пару минут через приложение «М
ой налог».
Вы: Что за метка 18+? Уберите её
Бот: Значок «18+» означает, что ваше видео содержит контент, не рекомендованный для просмотра детьми. Если вы считаете, что это
т значок был поставлен ошибочно, свяжитесь с нашей службой поддержки на help@rutube.ru для перепроверки.
Вы:
```

Рисунок 8 — Интерфейс прототипа системы

В демонстрации бот корректно отвечает на вопросы о регистрации, отключении монетизации и метках для видео, предоставляя подробные инструкции и контактную информацию. Однако на вопрос «Как подключить монетизацию?» ответ оказался неточным, так как бот сразу предположил, что пользователь является физическим лицом, что не полностью соответствует конкретному запросу о подключении монетизации.

### *Экспериментальное исследование*

Для оценки качества ответов, сгенерированных большими языковыми моделями с использованием метода RAG, применяются различные методы. Наиболее распространёнными метриками являются BLEU, METEOR, ROUGE и BERTScore [8]. Метрики BLEU и ROUGE широко используются для оценки текстов на основе точности совпадений с эталонными ответами, однако они менее эффективны для задач, где требуется глубокое семантическое сходство. METEOR также подходит для оценки релевантности, учитывая морфологические и синонимические связи, но её эффективность ограничена для задач, включающих более сложные языковые конструкции и длинные ответы. В данном исследовании был выбран BertScore как оптимальный метод оценки, поскольку он позволяет оценивать тексты на уровне семантики, сопоставляя эмбеддинги ответов модели и эталонных данных. Это делает BertScore особенно ценным для оценки моделей, которые ориентированы на выдачу релевантной информации и могут варьировать в формулировках ответов.

Результаты оценки качества ответов с использованием метрики BERTScore были получены на тестовой выборке из 801 вопроса реальных пользователей. Средние значения по всем ответам составили: средняя точность (Precision) — 0.7578, средняя полнота (Recall) — 0.7674, а средний F1-мера — 0.7608. Эти показатели свидетельствуют о высоком уровне семантического соответствия генерируемых ответов эталонным данным. Высокая точность указывает на то, что модель эффективно извлекает релевантную информацию, минимизируя количество нерелевантных элементов в ответах. Полнота показывает, что большая часть необходимой информации присутствует в сгенерированных ответах. Средний F1-мера подтверждает сбалансированное соотношение между точностью и полнотой, демонстрируя общую эффективность модели в предоставлении релевантных и точных ответов. Анализ на тестовой выборке позволяет сделать вывод о стабильности и надежности модели при обработке разнообразных запросов.

Также было произведено сравнение со стандартной языковой моделью без использования метода RAG. Результат со сравнением представлен на рисунке 9. На графиках показано сравнение показателей Precision, Recall и F1 между разработанным прототипом (LLM + RAG) и стандартной языковой моделью (LLM). Значения метрик распределены таким образом, что большинство точек располагается выше диагональной линии, что свидетельствует о значительном улучшении показателей точности, полноты и F1-меры в модели, дополненной RAG.

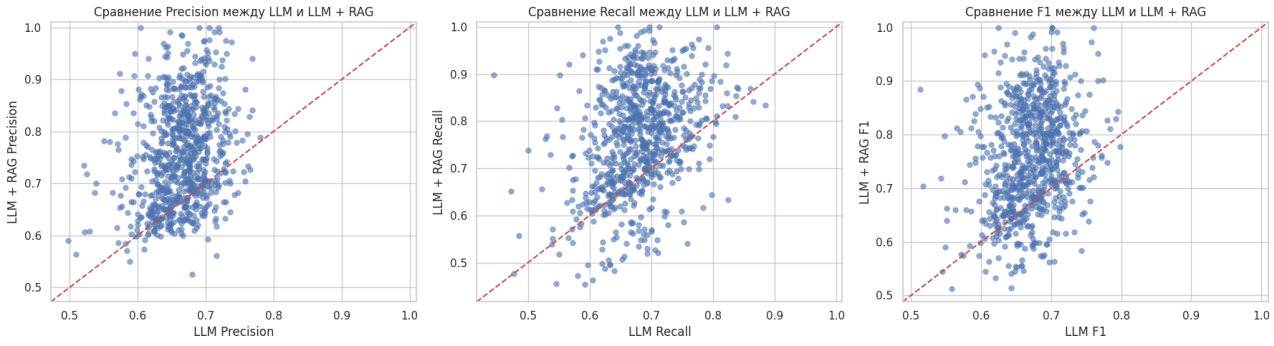



Рисунок 9 — Сравнение LLM и LLM + RAG

На первом графике сравнения Precision можно наблюдать, что интеграция метода RAG способствует улучшению точности ответов. Это проявляется в более высоких значениях метрики для LLM + RAG по сравнению с базовой моделью, что указывает на повышение релевантности извлекаемой информации и снижение числа нерелевантных элементов в ответах. Подобная тенденция наблюдается и на втором графике, где представлено сравнение метрики Recall. В данном случае, использование RAG позволяет модели более полно охватывать релевантные данные, что отражается в увеличении значений полноты. На последнем графике, где показано сравнение F1-меры, видно, что интеграция RAG позволяет достичь лучшего баланса между точностью и полнотой, улучшая общее качество ответов модели.

Таким образом, данные результаты подтверждают, что предложенный подход с использованием RAG

обеспечивает улучшение семантической релевантности и точности ответов, что делает модель более эффективной и надежной при обработке широкого диапазона запросов.

### *Выводы*

В ходе исследования была рассмотрена проблема повышения точности и достоверности ответов больших языковых моделей (LLM) при отсутствии доступа к актуальным данным. Для решения этой задачи был предложен и реализован подход, основанный на интеграции метода Retrieval-Augmented Generation (RAG) с использованием инструментов LangChain и pgvector.

Разработанная система эффективно сочетает возможности LangChain для облегчения взаимодействия с LLM и управления диалогом, а также потенциал pgvector для быстрого поиска релевантной информации в больших наборах данных. Методология интеграции RAG была подробно описана, что позволило создать прототип системы и провести его экспериментальное исследование.

Результаты экспериментов подтвердили, что использование предложенного подхода значительно повышает точность и надежность ответов LLM. Это достигается за счет снижения вероятности возникновения «галлюцинаций» моделей и обеспечения доступа к актуальной информации, даже при недостатке контекста или отсутствии обновленных данных.

Таким образом, поставленные цели работы были полностью достигнуты. Предложенная интеграция RAG с использованием LangChain и pgvector не только решает проблему недостоверности и несвязности информации, генерируемой большими языковыми моделями, но и открывает перспективы для дальнейшего совершенствования систем обработки естественного языка. Разработанная система может быть применена для последующих исследований в направлении повышения эффективности и надежности LLM.

### *Литература*

1. Chui, M., Roberts, R., Yee, L., Hazan, E., Singla, A., Smaje, K., Sukharevsky, A., Zemmel, R. / The economic potential of generative AI: The next productivity frontier [Электронный ресурс] // McKinsey Digital. – Электрон. дан. – 2023. – Режим доступа: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier. - Загл. с экрана.

2. Федоров, В. О. Большие языковые модели с поисковой расширенной генерацией: обзор и перспективы / В. О. Федоров, Р. А. Поляков // Оригинальные исследования. – 2023. – Т. 13, № 12. – С. 43-47.

3. Маркин, Е. И. интеграция языковых моделей с базами знаний и внешними источниками данных / Е. И. Маркин, В. В. Зупарова, В. В. Зупарова // XXI век: итоги прошлого и проблемы настоящего плюс. – 2024. – Т. 13, № 2(66). – С. 25-31.

4. Козлов, Е. А. Анализ проблемы галлюцинаций в больших языковых моделях / Е. А. Козлов // Наука, образование, инновации: актуальные вопросы и современные аспекты : сборник статей XXII Международной научно-практической конференции, Пенза, 27 мая 2024 года. – Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2024. – С. 37-40.

5. LangChain. Indroduction [Электронный ресурс] // LangChain – Электрон. дан. – 2024. – Режим доступа: https://python.langchain.com/docs/introduction/ - Загл. с экрана.

6. pgvector. Open-source vector similarity search for Postgres [Электронный ресурс] // GitHub – Электрон. дан. – 2024. – Режим доступа: https://github.com/pgvector/pgvector - Загл. с экрана.

7. Qwen2.5 [Электронный ресурс] // Qwen2 – Электрон. дан. – 2024. – Режим доступа: https://qwen2.org/qwen2-5/ - Загл. с экрана.

8. Netisopakul, P. Comparison of Evaluation Metrics for Short Story Generation / P. Netisopakul, U. Taoto // IEEE Access. – 2023. – Vol. 11. – P. 140253-140269. – DOI 10.1109/access.2023.3337095.

***Слипенко М.К., Рычка О.В. Применение LangChain и pgvector для реализации Retrieval-Augmented Generation в больших языковых моделях.*** *В данной статье рассматривается проблема повышения точности и достоверности ответов больших языковых моделей (LLM) при отсутствии доступа к актуальным данным. Предложен подход, основанный на интеграции метода Retrieval-Augmented Generation (RAG) с использованием инструментов LangChain и pgvector. LangChain облегчает взаимодействие с LLM и управление диалогом, а pgvector обеспечивает быстрый поиск релевантной информации в больших наборах данных. Описана методология интеграции RAG, разработан прототип системы и проведено экспериментальное исследование, показавшее повышение точности и надежности ответов LLM.*

***Ключевые слова**: большие языковые модели, Retrieval-Augmented Generation, LangChain, pgvector, обработка естественного языка, интеграция данных, поиск по векторным представлениям.*

***Sliпенко Maksim, Rychka Olga Application of LangChain and pgvector for Implementing Retrieval-Augmented Generation in Large Language Models.** This article addresses the problem of enhancing the accuracy and reliability of responses from large language models (LLMs) in the absence of access to up-to-date data. A proposed approach is based on the integration of the Retrieval-Augmented Generation (RAG) method using LangChain and pgvector tools. LangChain facilitates interaction with LLMs and dialogue management, while pgvector ensures fast retrieval of relevant information from large datasets. The methodology for integrating RAG is described, a system prototype is developed, and experimental research is conducted, demonstrating an improvement in the accuracy and reliability of LLM responses.*

***Key words**: large language models, Retrieval-Augmented Generation, LangChain, pgvector, natural language processing, data integration, vector-based search*

УДК 004.52

# Теоретический анализ технологии Text to Speech и текущего состояния её развития

**С.Н. Евтушенко[*1], С.В. Щедрин[*2], И.А. Коломойцева[*3]**

[*1] студент, Донецкий национальный технический университет,
s.scorpi-on@ya.ru,

[*2] старший преподаватель кафедры ПИ, Донецкий национальный технический университет,
do010575ssv@mail.ru, SPIN-код: 3891-6711,

[*3] старший преподаватель кафедры ПИ, Донецкий национальный технический университет,
bolatiger@mail.ru, OrcID: 0000-0002-1559-0213, SPIN-код: 1002-8374

***Евтушенко С.Н., Щедрин С.В., Коломойцева И.А. Теоретический анализ технологии Text to Speech и текущего состояния её развития.*** *В статье рассмотрены принципы работы технологии Text to Speech (TTS) на различных этапах её становления. Особое внимание уделено основным современным подходам к синтезу речи: конкатенативному и параметрическому, а также интеграции нейросетей в их работу. Проведено исследование нескольких популярных TTS-решений, их преимуществ и недостатков. Выявлены актуальные вызовы, тенденции и перспективы развития рассматриваемой технологии.*

***Ключевые слова:*** *Text to Speech, TTS, синтез речи, фонемы, модель, нейросеть, нормализация.*

## *Введение*

Технология синтеза речи из текстовых данных, также известная как Text to Speech (TTS) или Text to Voice (TTV), представляет собой одну из наиболее значимых и быстро развивающихся областей в сфере искусственного интеллекта. С момента своего появления TTS прошла долгий путь, начиная с простых механических систем и заканчивая современными нейросетевыми моделями, способными генерировать речь, близкую к естественной.

Отметим, что TTS нельзя назвать простым преобразованием текста в звук. Этот процесс включает решение таких задач как:

- клонирование голоса по записанному образцу;
- придание речи эмоциональной и стилистической окраски;
- учёт грамматических, интонационных, фонетических и прочих особенностей естественного языка, (включая диалекты);
- учёт заранее выбранных характеристиках голоса (тембр, интонация, скорость речи).

В условиях стремительного развития технологий, таких как глубокое обучение и обработка больших данных, TTS становится все более доступной и эффективной, что открывает новые горизонты для её применения в различных сферах, как например:

- голосовые ассистенты, боты и контакт-центры для автоматизации общения с людьми;
- помощь людям с нарушениями речи и зрения (чтение с экрана, озвучивание интерфейса);
- помощь при навигации в GPS-системах;
- озвучивание аудиокниг.

Актуальность исследования технологии TTS обусловлена не только её возможностями, но и постоянным развитием алгоритмов и моделей, которые позволяют улучшать качество синтезируемой речи. В последние годы наблюдается тенденция к созданию более натуральных и эмоционально окрашенных голосов, что делает взаимодействие человека с машинами более комфортным и интуитивным. Кроме того, с развитием технологий, таких как нейронные сети и генеративные модели, открываются новые возможности для создания индивидуализированных голосов, что может значительно улучшить пользовательский опыт.

В рамках данной работы поставлена цель провести теоретический анализ технологии TTS, а также оценить состояние её развития на момент публикации статьи. Требуется рассмотреть инструменты и платформы, при помощи которых можно внедрить TTS в программный продукт.

### *Подходы к синтезу речи*

Машинная генерация речи не является новшеством нашего времени, а интересовала человека довольно давно. Так, первое устройство, позволяющее синтезировать речь, изобрёл в XVIII веке учёный-механик Вольфанг фон Кемпелен. Его «говорящая машина» была механической и могла воспроизводить звуки, отдалённо напоминающие речь, с помощью системы из мембран, мехов и трубок. Затем в 1937 году инженер Гомер Дадли из Bell Labs разработал Voder — первое электронное устройство, которое генерировало речь на основе электрических импульсов. Цифровая же эра технологии TTS началась в 1984 году, когда в Массачусетском технологическом институте Деннис Клатт из компании DEC создал синтезатор DECtalk. Устройство имело несколько встроенных тонов голоса, понимало фонетическое написание слов и позволяло настраивать высоту тона и длительности в тексте. Последнее даже позволяло DECtalk петь. Аппарат стал использоваться в США для систем массового оповещения в аэропортах, метеорологических службах, больницах, а также для помощи немым людям (самым известным примером является Стивен Хокинг, которому нравился голос DECtalk и который стал впоследствии ассоциироваться именно с ним).

Существует два основных подхода для синтеза речи: конкатенативный и параметрический синтез.

Конкатенативный синтез основан на использовании заранее записанных сегментов речи. Эти сегменты могут быть как отдельными фонемами, так и более длинными единицами, такими как слоги или слова. При синтезе речи система выбирает и соединяет эти сегменты в соответствии с текстом, который необходимо озвучить. Текст предварительно должен пройти этап нормализации — предварительной обработки, которая приведёт его к фонемическому виду [1]. Чем их больше, тем натуральнее результат генерации. Более наглядно этот принцип представлен на рисунке 1.

Основным преимуществом конкатенативного синтеза является высокая естественность звучания, так как используются реальные записи человеческой речи. Однако этот подход имеет и свои недостатки: ограниченность в интонации и эмоциональной окраске, неестественные переходы между фрагментами, а также потребность в сборе большого объёма данных.

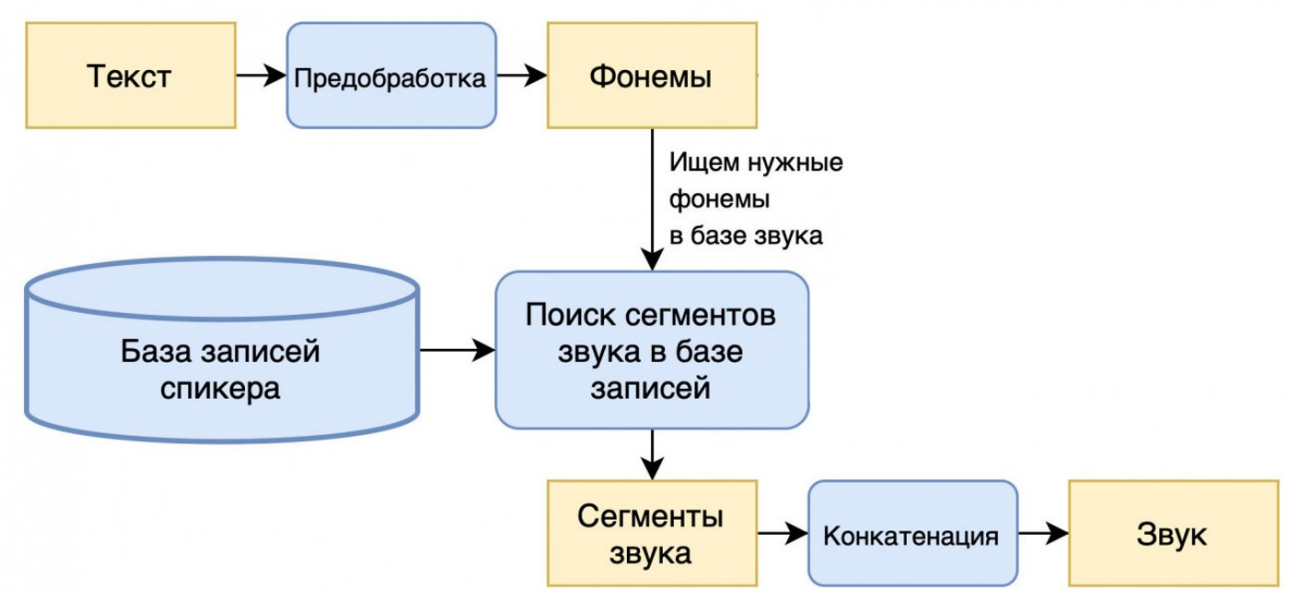



Рисунок 1 — Схема работы конкатенативного синтеза речи

Параметрический синтез представляет собой более сложный подход, который использует математические модели для генерации речи. Для начала формируется акустическая модель, которая получает лингвистические данные (разбитые на фонемы слова и дополнительную разметку) и переводит их в промежуточное состояние, которое описывает основные свойства речи (скорость и темп произнесения слов, интонационные признаки и артикуляцию) и спектральные характеристики звука. Затем данные передаются на второй блок — вокодер — который и генерирует звук по его параметрическому представлению [2]. Параметрический синтез не требует такого количества данных для обучения и имеет широкое разнообразие интонаций в генерируемой аудиодорожке. При этом синтезированная таким образом речь традиционно звучала неестественно и «механически», в отличие от синтезированной конкатенативно.

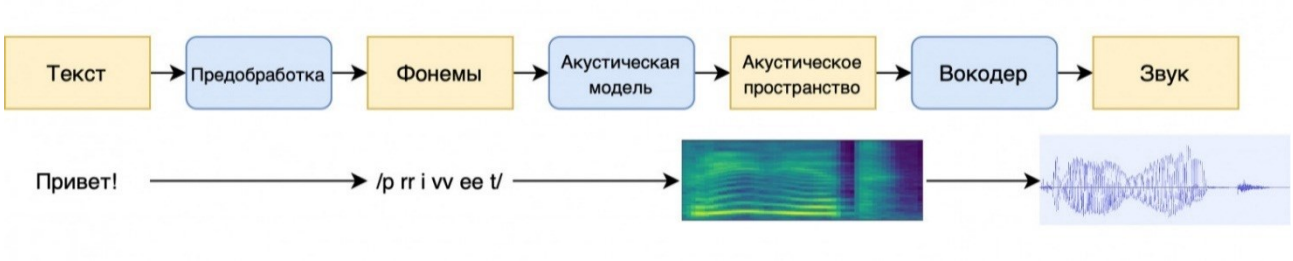



Рисунок 2 — Схема работы параметрического синтеза речи

С развитием технологий глубокого обучения модели параметрического синтеза получили поддержку нейросетей и стали способны выдавать речь, максимально приближённую к человеческой. На данный момент такой подход является стандартным для технологии TTS. Впервые это сделала генеративная модель WaveNet в 2016 году. Количество таких моделей растёт, а сами они постоянно развиваются и совершенствуют свои возможности. Из недостатков такого подхода можно отметить ресурсоёмкость (развёртывание и обучение нейросетей), а на ранних этапах ещё и низкую производительность синтеза.

Отметим также, что нейросети позволили комбинировать конкатенативный и параметрический методы для, позволяя объединить их лучшие характеристики. Этот подход отличается тем, что сформированная нейросетью акустическая модель позволяет сопоставлять сегменты фонемическому представлению текста более точно, на основании соответствия гипотезам [3]. Составленный таким образом звук воспринимается естественно (поскольку не сгенерирован с нуля) и имеет гораздо меньше дефектов «склейки». Однако, для обеспечения качественного звучания всё ещё требуется продолжительная работа с диктором и калибровка интонаций. На рисунке 3 представлена схема сочетания конкатенативного и параметрического методов.

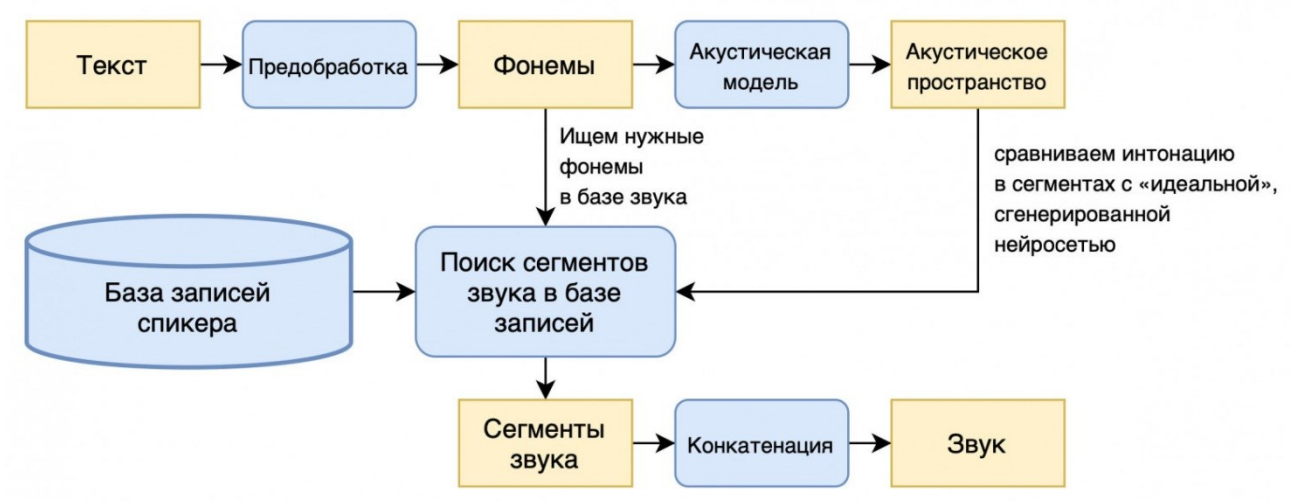



Рисунок 3 — Схема работы комбинированного метода синтеза речи

### *Обзор актуальных TTS-решений*

Инструменты реализации TTS отличаются задействованными технологиями, принципом взаимодействия с пользователем / разработчиком и качеством итогового результата. В зависимости от сложности проекта, может возникнуть необходимость ручной имплементации или изменения логики работы некоторых этапов работы с текстом — для этой задачи существуют как открытые, так и коммерческие решения. Но в этом разделе мы рассмотрим такие из них, которые предоставляют полный цикл TTS: от обработки текста и до превращения его в звук.

Microsoft Speech API (SAPI) — программный интерфейс для распознавания и синтеза речи, разработанный компанией Microsoft для своей операционной системы Windows. Первая версия SAPI была выпущена в 1995 году и включала низкоуровневые методы для прямого распознавания речи и преобразования текста в речь, которые приложения могли использовать для непосредственного управления механизмами, а также упрощённые API для

голосовых команд и голосового общения. В актуальном на данный момент семействе SAPI версии 5 приложения и движки не взаимодействуют напрямую друг с другом. Вместо этого каждое из них взаимодействует с компонентом времени выполнения (sapi.dll), который и реализует API для приложений [4].

SAPI поддерживает несколько языков и голосов, пользовательские словари и грамматические модели, позволяет настраивать скорость и тон произношения. Это одна из самых доступных реализаций TTS, которая, однако, не может похвастаться высоким качеством генерации голоса: он довольно монотонный, не поддерживает интонации и прочие продвинутые возможности, а также имеет заметные следы «склейки». Классическим примером использования SAPI является приложение Microsoft Narrator — встроенный в ОС Windows экранный диктор, который повышает доступность системы для слабовидящих и слепых пользователей. Он позволяет озвучивать содержимое экрана с заданной подробностью, а также устанавливать поддерживаемые SAPI параметры.

Ещё одна легковесная реализация TTS под названием Web Speech API была представлена в 2012 году в виде рабочего черновика как часть спецификации HTML 5. С тех пор Web Speech API получила развитие и стала поддерживаться большинством современных веб-браузеров. Web Speech API позволяет разработчикам интегрировать функции распознавания и синтеза речи в веб-приложения, что открывает новые возможности для создания интерактивных и доступных интерфейсов. При этом браузеры могут использовать различные движки синтеза речи, которые могут быть встроены в сам браузер или опираться на внешние сервисы. Воспользоваться Web Speech API можно прямо из браузера в консоли JavaScript [5]. На рисунке 4 представлен пример минимальной программы, использующей WSAPI для синтеза речи из текста.

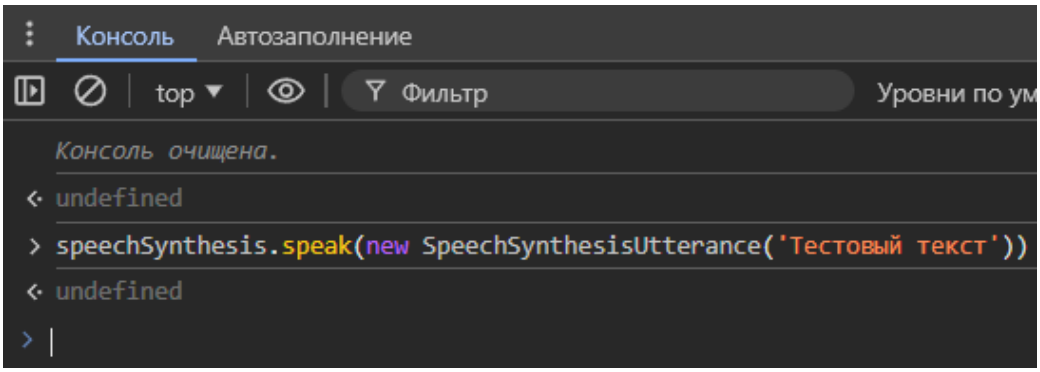



Рисунок 4 — Минимальная программа, использующая WSAPI для синтеза речи из текста

Web Speech API во многом похожа на Microsoft Speech API, потому что предоставляет схожий набор инструментов и настроек. Web Speech API может даже использовать голоса, предоставляемые Microsoft Speech API, если браузер клиента сконфигурирован соответствующим образом. Качество выходного звучания при этом условии будет также одинаковым. Однако, Web Speech API является кроссплатформенной технологией, поскольку зависит не от ОС, а от веб-браузера. Поэтому несмотря на кажущиеся внешние сходства, архитектура этих API принципиально отличается.

Довольно мощным инструментом является библиотека Coqui TTS. Она содержит предобученные модели на большом количестве языков, возможность создания собственных моделей (как с нуля, так и на основе существующих) и утилиты для ручного анализа наборов данных. Исходный код Coqui полностью открыт для копирования и ручной сборки, также поддерживается установка посредством pip (пакетного менеджера Python) и Docker. На рисунке 5 представлен скриншот демонстрационного веб-интерфейса Coqui, позволяющий оценить результат генерации разных моделей.

Coqui TTS имеет внушительный объём, а при необходимости обучения моделей имеет достаточно высокие системные требования — это объясняется тем, что в отличие от рассмотренных ранее аналогов, Coqui использует продвинутые нейросети для синтеза речи. Соответственно, результат этого синтеза имеет очень высокую степень схожести с натуральной речью. Высокая производительность, широкий спектр функций и некоммерческий характер разработки делает её идеальным вариантом как для энтузиастов, так и для компаний, нуждающихся в качественном TTS.


Рисунок 5 — Скриншот демонстрационного веб-интерфейса Coqui

Помимо автономных движков, существуют также внешние облачные платформы, обращение к которым производится по протоколу HTTP. Наиболее профессиональными платформами TTS считаются решения известных корпораций, как например:

- Google Text-to-Speech AI;
- Amazon Polly;
- IBM Watson Text to Speech;
- Т-Банк VoiceKit;
- Yandex SpeechKit;
- Сбер SaluteSpeech.

Они используют фирменные нейросети для нормализации текста, расширенной его разметки, создания собственного брендового голоса и обеспечения естественного звучания. Помимо этого, в число их преимуществ входит качественная техническая поддержка, специфические услуги для бизнеса (к примеру, автоматизация работы колл-центров) и полностью облачная инфраструктура, предоставляемая компаниями. Закономерно, что такие решения являются исключительно коммерческими: тарификация считается по числу символов или длительности сгенерированной аудиозаписи. Бесплатные тарифы, как правило, сильно ограничены и предназначены для ознакомительных целей. С учётом достоинств облачных платформ, они могут подходить для решения задач бизнеса в том случае, если затраты на них окупаются. На рисунке 6 приведена статистика эффективности фирменного TTS-решения компании Сбер — SaluteSpeech — на примере задач СберБанка [6].

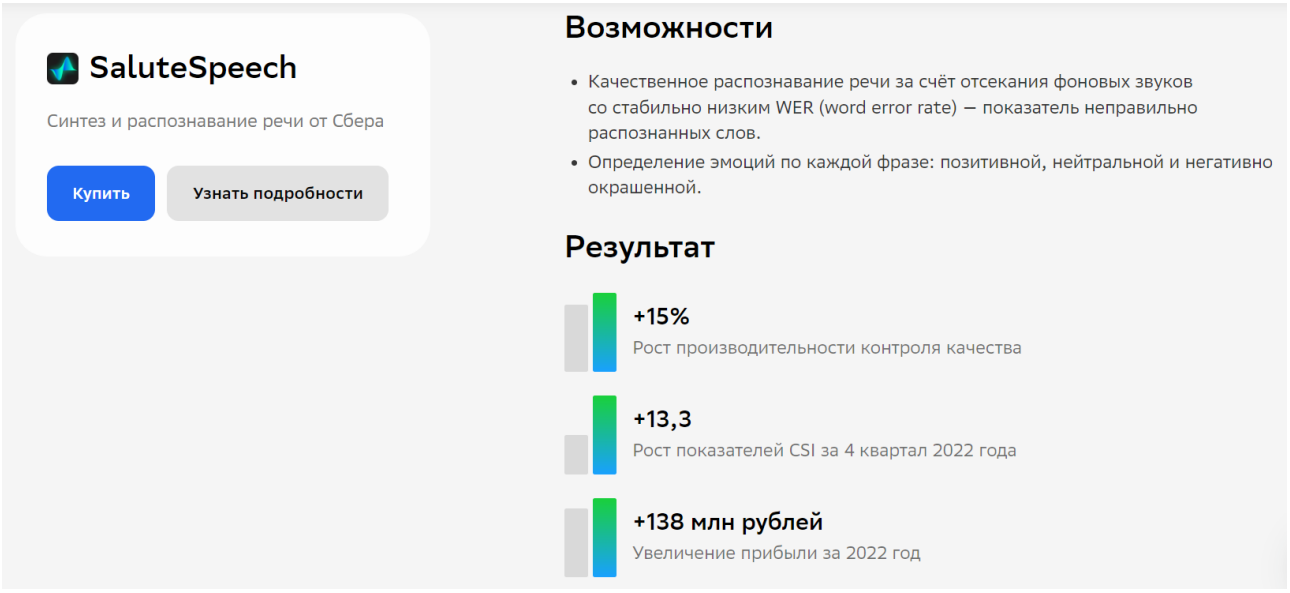



Рисунок 6 — Статистика эффективности решения SaluteSpeech от Сбера

Особняком в этом ряду стоит компания OpenAI, занимающаяся интеграцией голосового помощника в своего фирменного ассистента ChatGPT. Так, в мае 2024 года компания представила расширенный голосовой режим, который обладает следующими особенностями:

- выразительная речь с поддержкой различных интонаций по запросу пользователя;
- пение в разнообразных жанрах;
- смена тембра и модуляции голоса в зависимости от контекста;
- подражание голосам известных персонажей;
- воспроизведение кашля, лая собаки и других звуков в речи;
- авторасознавание языка ответа, поддержка акцентов.

На момент публикации данной работы расширенный голосовой режим доступен только в мобильных приложениях ChatGPT на iOS и Android в ограниченном числе стран. При этом такие продвинутые возможности являются уникальными, и компания заявляет о намерении совершенствовать их и далее.

Тем не менее, стоит отметить существование более простых, но бесплатных API. Например, сервис Google Translate позволяет использовать возможности переводчика (в число которых входит озвучивание текста) безлимитно и без необходимости получения авторизационного токена. Результирующий голос получается одноголосым, довольно монотонным, без такого разнообразия тонов и настроек, как в случае с платными аналогами. В то же время, он звучит достаточно плавно, чтобы не походить на механический. Для реализации такого подхода есть множество библиотек на разных языках программирования, как например gTTS для Python.

### *Выводы*

Технология синтеза речи из текстовых данных (TTS) представляет собой динамично развивающуюся область, которая находит все более широкое применение в различных сферах жизни. С момента своего появления TTS прошла значительный путь, от механических устройств XVIII века до современных нейросетевых моделей, способных генерировать речь, близкую к естественной. В ходе исследования были рассмотрены основные современные подходы к синтезу речи, такие как конкатенативный и параметрический синтез, а также их комбинации, что позволило выявить преимущества и недостатки каждого из них.

Современные TTS-решения, включая как локальные библиотеки, так и облачные платформы, предлагают разнообразные инструменты для реализации синтеза речи. Они обеспечивают высокое качество звучания, возможность настройки параметров голоса и поддержку множества языков. Выбор конкретного решения зависит от требований проекта, бюджета и необходимых функциональных возможностей. Среди актуальных тенденций отрасли можно выделить:

- развитие методов глубокого обучения для создания более сложных моделей;
- увеличение популярности облачных платформ, которые предлагают высококачественные масштабируемые решения;
- создание более интуитивных и удобных интерфейсов для взаимодействия с TTS-системами, что может улучшить пользовательский опыт и расширить применение технологии.

Проанализировав некоторые из популярных на данный момент инструментов, реализующих технологию TTS, были выявлены проблемы данной технологии, которые так или иначе требуют решения в будущем.

Так, несмотря на значительные достижения в области TTS, качество синтезируемой речи все еще остается проблемой. Хотя современные нейросетевые модели способны генерировать речь, близкую к натуральной, в некоторых случаях могут возникать артефакты, неестественные интонации и так далее. Это может негативно сказаться на восприятии пользователями синтезированной речи, как например в голосовых ассистентах и системах для людей с нарушениями слуха.

Другой важной задачей TTS является создание эмоционально окрашенной речи. Современные системы в большинстве своём не способны адекватно передавать эмоции, что ограничивает их применение в контекстах, где важна эмоциональная связь, например, в обучении, терапии или развлекательных приложениях. Разработка алгоритмов, способных распознавать и воспроизводить эмоциональные нюансы, остается вызовом для исследователей, однако такие компании как OpenAI уже движутся в направлении его преодоления.

С увеличением потребности в персонализированных голосах для различных приложений, таких как голосовые помощники и навигационные системы, возникает необходимость в разработке технологий, позволяющих создавать уникальные голоса на основе индивидуальных предпочтений пользователей. Это требует значительных объемов отвалидированных данных и сложных алгоритмов для обучения, что может быть ресурсозатратным.

Сложность синтеза речи на различных языках и диалектах также представляет собой вызов. Хотя многие современные TTS-системы поддерживают несколько языков, качество синтезируемой речи может варьироваться в зависимости от языка.

Решение всех вышеуказанных проблем требует значительных вычислительных мощностей и финансовых вложений, либо же усилий на оптимизацию алгоритмов моделей машинного обучения. Это повышает порог входа в TTS и ограничивает не только круг разработчиков, которые могли бы её развивать, но и в принципе доступность качественных TTS-решений для конечных пользователей. При этом важно отметить, что TTS-

технологии уже широко используются злоумышленниками. Например, возможность «клонирования» голоса становится всё более популярным методом мошенничества. Налицо необходимость в правовом регулировании использования TTS-технологий (которая, впрочем, выходит за рамки одной лишь TTS и касается использования нейросетей в общем).

Таким образом, TTS-технологии продолжают развиваться, сталкиваясь с рядом вызовов и проблем, которые требуют внимания научного сообщества. Решение этих задач откроет новые возможности для применения синтеза речи в различных сферах жизни. В будущем можно ожидать появления новых технологий, которые будут способствовать улучшению качества синтезируемой речи и расширению её применения в различных областях, таких как образование, здравоохранение, развлечения и бизнес.

### *Литература*


6. Что такое технология TTS, как устроена и каких сферах используется синтез речи [Электронный ресурс] – Режим доступа: https://habr.com/ru/companies/skillfactory/articles/850118/ – Загл. с экрана

7. Синтез речи [Электронный ресурс] – Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=Синтез_речи – Загл. с экрана

8. Шёпот и эмоции в Алисе: история развития голосового синтеза Яндекса [Электронный ресурс] – Режим доступа: https://habr.com/ru/companies/yandex/articles/593681/ – Загл. с экрана

9. Speech API Overview (Microsoft Speech Platform) [Электронный ресурс] – Режим доступа: https://documentation.help/Microsoft-Speech-Platform-SDK-11/SAPI5_Overview.htm – Загл. с экрана

10. Web Speech API [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/ru/docs/Web/API/Web_Speech_API – Загл. с экрана

11. SaluteSpeech. Синтез и распознавание речи от Сбера. С подробной документацией [Электронный ресурс] – Режим доступа: https://developers.sber.ru/portal/products/smartspeech – Загл. с экрана



***Евтушенко С.Н., Щедрин С.В., Коломойцева И.А. Теоретический анализ технологии Text to Speech и текущего состояния её развития.*** *В статье рассмотрены принципы работы технологии Text to Speech (TTS) на различных этапах её становления. Особое внимание уделено основным современным подходам к синтезу речи: конкатенативному и параметрическому, а также интеграции нейросетей в их работу. Проведено исследование нескольких популярных TTS-решений, их преимуществ и недостатков. Выявлены актуальные вызовы, тенденции и перспективы развития рассматриваемой технологии.*



***Ключевые слова:*** *Text to Speech, TTS, синтез речи, фонемы, модель, нейросеть, нормализация.*



***Yevtushenko S.N., Shchedrin S.V., Kolomoitseva I.A. Theoretical analysis of Text to Speech technology and the current state of its development****. The article discusses the principles of Text to Speech (TTS) technology at various stages of its development. Special attention is paid to the main modern approaches to speech synthesis: concatenative and parametric, as well as the integration of neural networks into their work. A study of several popular TTS solutions, their advantages and disadvantages has been conducted. The current challenges, trends and prospects for the development of the technology in question are identified.*



***Keywords:*** *Text to Speech, TTS, speech synthesis, phonemes, model, neural network, normalization.*

УДК 004.912

# Особенности структуры тезауруса для повышения качества поиска документов в области программирования

**И.А. Коломойцева**
старший преподаватель кафедры ПИ, Донецкий национальный технический университет,
bolatiger@mail.ru, OrcID: 0000-0002-1559-0213, SPIN-код: 1002-8374

***Коломойцева И.А. Особенности структуры тезауруса для повышения качества поиска документов в области программирования.*** *В статье рассматривается особенности формирования тезауруса в области программирования. Обоснована необходимость использования такого тезауруса в информационном поиске. Описаны отношения между понятиями в тезаурусе (партономия и таксономия), позволяющие его отнести к формальным онтологиям. Описана иерархическая структура тезауруса. Сделан вывод о том, что расширение запроса с использованием тезауруса приведет к повышению полноты поиска, а в некоторых случаях – точности.*

***Ключевые слова:*** *обработка естественного языка; информационный поиск; тезаурус, онтология; отношения; программирование.*

## *Введение*

Объём электронных документов, с которыми сталкивается человек, постоянно растёт. И задача поиска информации приобретает всё большую актуальность.

Под поиском информации понимается поиск в коллекции документов, которые являются наиболее релевантными по отношению к произвольным информационным потребностям, выражаемым (представленным) при помощи однократных запросов пользователей [1]. При этом информационная потребность – это тема, о которой пользователь хочет знать больше (следует её отличать от информационного запроса).

На сегодняшний день существуют модели информационного поиска, такие как, например, булев поиск, поиск с помощью векторно-пространственного представления, вероятностный поиск, языковые модели, в том числе, и большие языковые модели. Большие языковые модели (Large Language Models, LLM) успешно решают большинство задач информационного поиска, но обладают одним существенным недостатком. Для построения LLM нужен большой качественный датасет и большие вычислительные мощности [2]. При этом, в существующих моделях LLM редко учитываются особенности конкретных предметных областей. Решение, которое предлагается в этом случае, — это донастройка (расширение) существующей базовой модели LLM на датасете, состоящем из текстов, относящихся к выбранной предметной области. Но при таком подходе возникает зависимость от базовой модели LLM [3].

Автором статьи предлагается другой подход к улучшению характеристик информационного поиска (полноты и в некоторых случаях точности) – использование такого онтологического ресурса как информационно-поисковый тезаурус.

Целью данной статьи является описание структуры и отношений в тезаурусе в области программирования.

## *Использование тезаурусов в информационном поиске*

Тезаурус – это словарь, в котором слова и словосочетания с близкими значениями сгруппированы в единицы, называемые понятиями, и в котором явно (в виде отношений, иерархии) указываются отношения между этими понятиями [4].

Тезаурусы на современном этапе развития информационных технологий играют важную роль, особенно при выполнении задач информационного поиска. В частности, тезаурусы решают следующие задачи:

1) упорядочивание лексики;
2) уточнение запросов:
3) технологический контроль;
4) повышение качества поиска.

Упорядочивание лексики заключается в том, что тезаурусы представляют собой иерархию понятий,

максимально полно охватывающую предметные области, и это упрощает поиск информации.

При уточнении запросов используется такая особенность тезауруса, как определение категорий понятий, к которым относятся слова запроса. По этим категориям происходит расширение запроса, что позволяет улучшить полноту поиска.

Тезаурусы могут использоваться для анализа содержимого коллекции документов. На основе этого анализа можно индексировать документы, что позволяет автоматизировать поиск в локальных, в том числе и корпоративных, коллекциях документов. Также анализ документов позволяет ранжировать найденные системой документы.

Так как тезаурус автоматизирует такие операции, как расширение (уточнение) запроса, анализ документов, ранжирование результатов поиска, то это приводит к повышению качества поиска.

Существуют два вида тезаурусов – общие и специализированные.

Общие тезаурусы содержат слова и понятия, которые используются в различных областях знаний [6]. Классическим тезаурусом такого типа является тезаурус Роже. Также общими тезаурусами являются типа WordNet [4], которые объединяют в себе функции справочной системы и инструмента для лингвистических исследований.

Специализированные тезаурусы предназначены для определенной отрасли знаний или профессии. Они содержат термины и понятия, характерные для конкретной области, и помогают специалистам лучше ориентироваться в своей сфере деятельности. Разработка такого типа тезауруса является актуальной задачей, так как с их помощью можно улучшить системы поиска общего назначения [7, 8].

### *Структура информационно-поисковой системы с использованием тезауруса*

Информационный поиск – это процесс поиска в большой коллекции некоего неструктурированного материала (обычно – документа), удовлетворяющего информационные потребности [1].

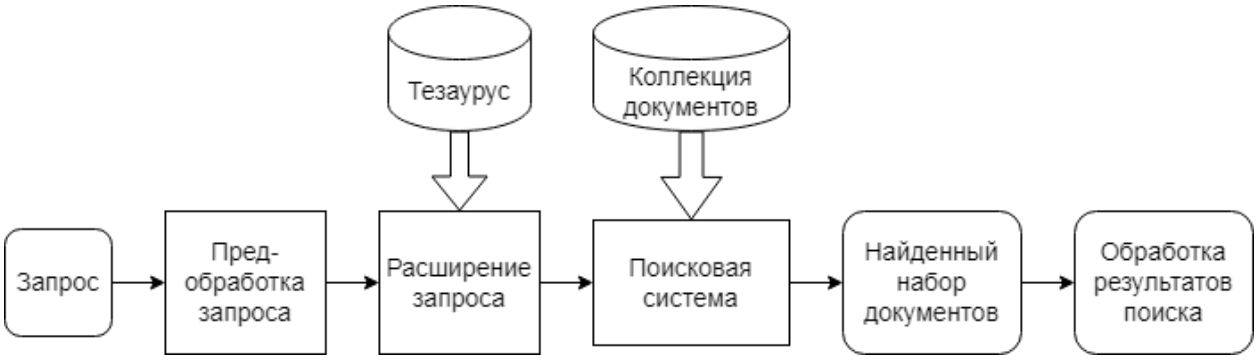



Рисунок 1 – Структура информационно-поисковой системы с использованием тезауруса

Основными характеристиками информационного поиска являются точность и полнота [1].

Точность – доля релевантных документов среди найденных.

Полнота – доля найденных документов среди всех релевантных.

Показатель, позволяющий найти баланс между точностью и полнотой, называется F-мерой. Она рассчитывается исходя из точности и повторяемости теста, где точность — это количество истинно положительных результатов, делённое на количество всех образцов, которые, по прогнозам, будут положительными, включая те, которые не были идентифицированы правильно, а повторяемость — это количество истинно положительных результатов, делённое на количество всех образцов, которые должны были быть идентифицированы как положительные.

Найти нужную информацию пользователь может с помощью информационно-поисковой системы (ИПС). Информационно-поисковая система – это компьютерная система, которая обеспечивает поиск и отбор необходимых данных в специальной базе с описаниями источников информации (индексе) на основе информационно-поискового языка и соответствующих правил поиска. Главная задача ИПС — поиск информации, релевантной информационным потребностям пользователя.

Общая структура информационно-поисковой системы включает в себя получение запроса от пользователя, предобработку этого запроса, отправку его в модуль поиска, получение списка документов, предположительно удовлетворяющих этому запросу, и ранжирование этого списка, Предобработка запроса

нужна для того, чтобы термины запроса привести к тому же виду, что и термины документа. В этом случае запрос и документ из коллекции документов можно сравнивать между собой.

Расширение запроса к информационно-поисковой системе (ИПС) означает добавление новых условий или критериев для поиска информации, которые позволяют уточнить результаты и сделать их более релевантными потребностям пользователя. Это может включать уточнение ключевых слов, применение морфологического анализа и другие методы. Сейчас крупные поисковые системы используют для уточнения запроса большие языковые модели, использующие статистические подходы. Но оптимальным решением для расширения запроса является уточнение только важных слов. Использование тезауруса упростит определение важности терминов запроса благодаря тому, что понятия тезауруса совпадают с терминами предметной области.

Вместо сложных статистических моделей и LLM для улучшения характеристик поиска среди документов некоторой предметной области, например, программирования предлагается использовать тезаурус, построенный по этой области. Тезаурус предлагается строить автоматически на основе относительно небольшого датасета. Структурная схема ИПС с использованием тезауруса представлена на рисунке 1.

### *Отношения в тезаурусе*

Отношение «выше-ниже» — это родовидовое отношение типа «класс-подкласс» (или отношения «гиперонимии–гипонимии»), обладает свойствами наследования и транзитивности. Относится к таксономическим отношениям.

Отношение «выше-ниже» обладают свойствами онтологических отношений «класс-подкласс», такими как [4]:

– каждый пример видового понятия в любой момент своего существования должен быть примером родового понятия;

– видовое понятие должно относиться к тому же семантическому классу, что и родовое понятие;

– видовое понятие должно наследовать основные свойства родового понятия.

Такими же свойствами обладают отношения между ролевым понятием и понятием класса, если экземпляры только этого класса могут выступать в данной роли (МЕТОД КЛАССА – ФУНКЦИЯ).

Еще один тип отношений, обладающий такими свойствами — это отношение между фазой какой-либо сущности и собственно этой сущностью (НАЧАЛЬНОЕ ЗНАЧЕНИЕ ПАРАМЕТРА – ПАРАМЕТР, ПОЛУЧЕННЫЙ В РЕЗУЛЬТАТЕ РАСЧЕТА).

Примеры отношений «класс-подкласс» приведены в таблице 1.

Таблица 1 – Примеры отношения «класс-подкласс» для предметной области «Программирование»

| Класс | Подкласс |
|---|---|
| КОМПОНЕНТ | Activity, Service, Broadcast Receiver, Content Provider |
| СТАТУС_ПРОЦЕССА | Активный процесс, фоновый процесс, пустой процесс |
| РАЗРЕШЕНИЕ | Permission-tree, permission-group |
| РЕСУРС | Изображение, звук |
| РАЗМЕТКА | FrameLayout, LinearLayout, TableLayout, RelativeLayout |
| ВИДЖЕТ | Текстовые поля, списки, кнопки, индикаторы |
| ИНДИКАТОР | ProgressBar, RaitingBar, SeekBar |
| УВЕДОМЛЕНИЕ | Вплывающее уведомление, уведомление в строке состояния |
| ДИАЛОГОВОЕ_ОКНО | AlertDialog, DatePickerDialog, TimePickerDialog, ProgressDialog |
| МЕНЮ | Меню опций, контекстное меню, подменю |
| ДАТЧИК | SensorManager |
| ПЕРЕДАЧА_ДАННЫХ | Bluetooth, Wi-Fi, NFC |
| ТЕЛЕФОНИЯ | Звонок, SMS |
| ДАННЫЕ | SQLite, геолокационные данные |

Но ещё больше сложностей возникает при построении тезауруса смешение типов и ролей в одной иерархии. Отношения тип-тип (анонимная функция - функция) и тип-роль (анонимная функция - замыкание) могут пройти все тесты установления родовидовых отношений. Но анонимная функция в любом случае останется функцией, и при этом она может быть замыканием, а может не быть им.

Отношение «часть-целое» играет очень важную роль в информационно-поисковых тезаурусах. Это связано с определением релевантности документа запросу пользователя. Можно предположить, что, если документ посвящен описанию части какого-то объекта, то он релевантен запросу, содержащему термин, определяющий целое. При этом сам термин, обозначающий термин, в документе может отсутствовать. Если,

например, в документе описываются правила определения методов класса, то он с большой долей вероятности относится к понятию «класс в ООП» и «Объектно-ориентированному программированию».

Примеры отношений «часть-целое» из тезауруса в области программирования приведены в таблице 2.

В лингвистике для определения отношения «часть-целое» используются лингвистические тесты – заданные предложения, в которые подставляются анализируемые сущности. Например, X – это часть Y, которое должно нормально звучать для некоторых X и Y. X (часть) называется меронимом, а Y (целое) – холонимом [4]. Это один из подходов, который используется при определении отношения «часть-целое» для тезауруса программирования. Задаются шаблонные фразы, по которой определяется, что X часть Y. Например, «X входит в Y», «X является частью Y» и т. п.

Таблица 2 - Примеры отношений «часть-целое» из тезауруса в области программирования

| Часть | Целое |
|---|---|
| Свойство | Класс |
| Оператор | Программа |
| Аргумент функции | Функция |
| Счётчик цикла | Цикл |
| Элемент массива | Массив |
| Объект ядра ОС | Ядро ОС |
| Адресное пространство | Процесс |
| Модель | Паттерн MVC |
| Состояние | Объект ядра «событие» |

Второй подход является онтологическим. Важным принципом определения отношения «часть-целое» является следующее утверждение: уничтожение (изменение) предполагаемой части оказывает влияние на предполагаемое целое. Например, если из КОМПЬЮТЕРНОЙ ПРОГРАММЫ удалить все ОПЕРАТОРЫ, то КОМПЬЮТЕРНАЯ ПРОГРАММА прекратит своё существование. Наличие или отсутствие у СЕМАФОРА ЧИСЛА КОНТРОЛИРУЕМЫХ РЕСУРСОВ меняет состояние СЕМАФОРА (свободное/занятое).

Важным свойством отношения «часть-целое» является транзитивность. Благодаря этому свойству можно строить цепочки логического вывода. Например (читается слева направо):

СВОЙСТВО – КЛАСС – ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

СЧЁТЧИК ЧИСЛА ПОЛЬЗОВАТЕЛЙ – ОБЪЕКТ ЯДРА ОС – ЯДРО ОС

Отношение ассоциации – еще тип отношений, который присутствует в информационно-поисковом тезаурусе в области программирования. Отношение ассоциации, как и отношение «часть-целое», связано с релевантностью документа запросу. Считается, что если два понятия C1 и C2 связаны отношением ассоциации, то тексты, содержащие понятие C1, релевантны запросу, содержащему понятие C2, и наоборот. Например, такими понятиями являются ПОТОК и НИТЬ.

Качество поиска с использованием информационно-поискового тезауруса сильно зависит от качества разрешения многозначности. Что бы тезаурус повысил качество поиска по сравнению с векторными моделями, многозначность должна верно разрешаться с вероятностью более 90%.

Исследования методов автоматического разрешения лексической многозначности делятся на два направления:

– разрешение лексической многозначности некоторой совокупности слов;

– разрешение лексической многозначности всех слов текста.

Для определения качества разрешения многозначности используют точность и полноту.

Точность – это отношение правильно выбранных значений к общему количеству слов, рассматриваемых системой.

Полнота – это отношение правильно выбранных значения к общему количеству неоднозначных языковых выражений.

Алгоритм разрешения многозначности для тезауруса в области программирования находится в стадии разработки.

### *Структура тезауруса для предметной области «Программирование»*

Структуру тезауруса для предметной области «Программирование» можно представить в виде дерева [7]. Фрагмент этого дерева приведен на рисунке 2.

В корне этого дерева представлено название языка программирования, например, PHP или Python.

В языке программирования выделяются следующие подклассы:

– тип языка;
– тип информации;
– назначение языка;
– установка платформы;
– элементы языка;
– структуры данных;
– библиотеки;
– другие подклассы.

Тип информации — это основные типы, с которыми работает язык, например, числовая, текстовая, изображения, в виде баз данных.

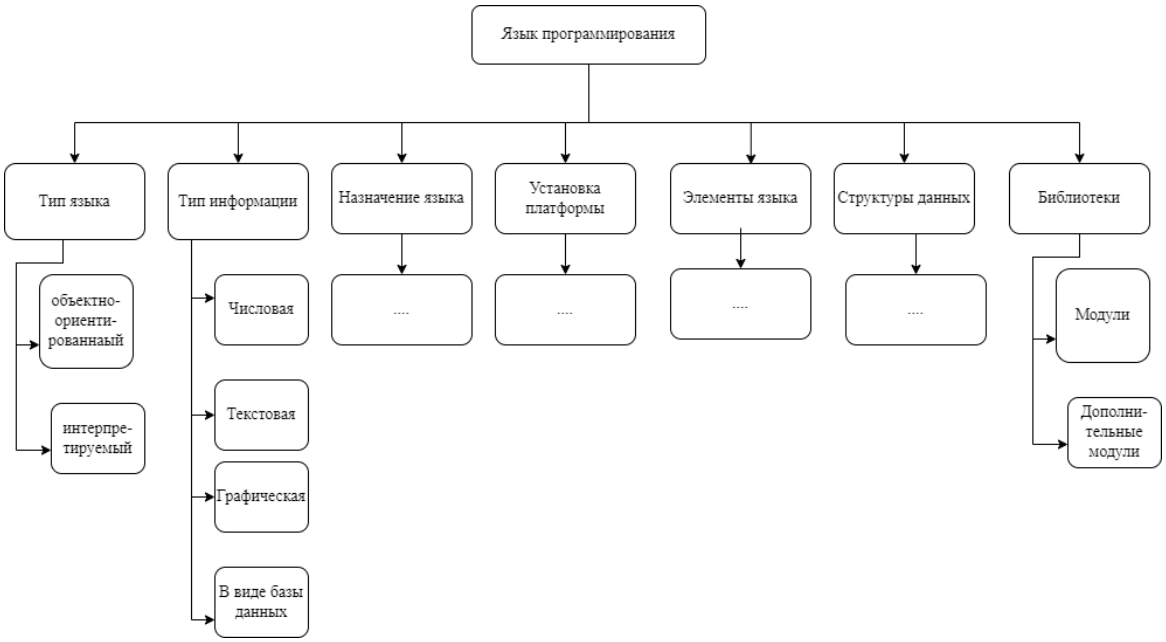



Рисунок 2 – Фрагмент структуры тезауруса для предметной области «Программирование»

Назначение языка – это области программирования, в которых язык чаще всего используется. Например, язык общего назначения, для создания веб-сайтов, для создания графических приложений.

Установка платформы – это та информация, которая описывает процесс создания и запуск приложений, созданных с помощью некоторого языка программирования. К этому типу информации относятся сайты для скачивания, названия установочных файлов, список IDE.

Элементы языка включают следующие элементы:

– операторы (арифметические, присваивания, сравнения, цикла, условия);

– функции <назначение, {описание}, краткое_назначение>

– комментарии (строки документирования) (однострочные, многострочные).

Библиотеки включают описание модулей, которые могут быть использованы при создании программы на некотором языке программирования.

Следует отметить, что структура тезауруса не представляет классическое описание языка программирования. Разделение на подклассы происходит в соответствие с тем, как можно сгруппировать информацию, которую ищет пользователь о языке программирования.

Одной из особенностей тезауруса для предметной области программирования является включение элементов, которые описанию сопоставляют название. Один из вариантов такого сопоставления: описание функции – название функции (<Язык, краткое_назначение> название). Например, <Python : ввод> input. Пример запроса, где используются такие элементы: Питон (Python) ввод данных. Пример расширенного запроса Питон ввод данных input. Такой способ расширения запроса резко увеличивает точность поиска – до более чем 90%.

### *Выводы*

Задача информационного поиска является сложной и актуальной задачей. Одним из подходов к решению этой задачи – является использование информационно-поискового тезауруса в конкретной области, в частности, в области программирования.

В работе описана структура информационно-поисковой системы, использующей тезаурус. Указано, что такой тезаурус относится к онтологическим ресурсам. Рассмотрены особенности и сложности формирования тезауруса в области программирования. Определены отношения, встречающиеся в таком тезаурусе – «ниже-выше», «часть-целое» и ассоциации. Описано, какие сложности возникают при определении этих отношений.

Приведена иерархическая структура тезауруса.

### *Литература*


1. Маннинг, К.Д. Введение в информационный поиск / К.Д. Маннинг, П. Рагхаван, Х. Шютце. – М.: ООО «И.Д. Вильмс», 2011. – 528 с.
2. Large Language Models for Information Retrieval: A Survey / Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, etc, - URL https/paperswithcode.com/paper/large-language-models-for-information (дата обращения 01.03.2024).
3. Fine Tuning Large Language Model (LLM). – URL https/www.geeksforgeeks.org/fine-tuning-large-language-model-llm/ (дата обращения 01.03.2024).
4. Лукашевич, Н.В. Тезаурусы в задачах информационного поиска /Н.В. Лукашевич. – М.: Издательство Московского университета, 2011. – 512 с.
5. Коломойцева, И.А. Применение тезауруса для расширения запроса информационно-поисковой системы на примере предметной области «Программирование» /И.А. Коломойцева // Материалы XIII Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2022). – Донецк: ДОННТУ, 2022. – С. 155-159.
6. Воронцова, И. А. Эволюция методов конструирования тезаурусов: от интуитивной компиляции к онтологическому инжинирингу (на материале английского языка) / И. А. Воронцова // Вестник Ивановского государственного университета. Серия: Гуманитарные науки. – 2023. – № 3. – С. 64-73. – DOI 10.46726/H.2023.3.8.
7. Демидов, Д. В. Метод автоматического построения тезауруса технического документа / Д. В. Демидов, Л. М. Мардер // Знания - Онтологии - Теории (ЗОНТ-2023) : Материалы IX Международной конференции, Новосибирск, 02–06 октября 2023 года. – Новосибирск: Институт математики им. С.Л. Соболева СО РАН, 2023. – С. 105-113.
8. Краснов, Ф. В. Проблема потери решений в задаче поиска схожих документов: Применение терминологии при построении векторной модели корпуса / Ф. В. Краснов, И. С. Смазневич, Е. Н. Баскакова // Бизнес-информатика. – 2021. – Т. 15, № 2. – С. 60-74. – DOI 10.17323/2587-814X.2021.2.60.74.



***Коломойцева И.А. Особенности структуры тезауруса для повышения качества поиска документов в области программирования.*** *В статье рассматривается особенности формирования тезауруса в области программирования. Обоснована необходимость использования такого тезауруса в информационном поиске. Описаны отношения между понятиями в тезаурусе (партономия и таксономия), позволяющие его отнести к формальным онтологиям. Описана иерархическая структура тезауруса. Сделан вывод о том, что расширение запроса с использованием тезауруса приведет к повышению полноты поиска, а в некоторых случаях – точности.*



***Ключевые слова:*** *обработка естественного языка; информационный поиск; тезаурус, онтология; отношения; программирование.*



***Kolomoitseva Irina Features of the thesaurus structure to improve the quality of document search in the field of programming.*** *The article discusses the features of thesaurus formation in the field of programming. The necessity of using such a thesaurus in information search is substantiated. The relations between the concepts in the thesaurus (partonomy and taxonomy) are described, allowing it to be attributed to formal ontologies. The hierarchical structure of the thesaurus is described. It is concluded that expanding the query using a thesaurus will lead to an increase in the completeness of the search, and in some cases, accuracy.*



***Key words:*** *natural language processing; information retrieval; thesaurus, ontology; relationships; programming.*

# СЕКЦИЯ 4. «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ, СИСТЕМЫ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ, КОМПЬЮТЕРНОЙ ГРАФИКИ И ОБРАБОТКИ ИЗОБРАЖЕНИЙ»

УДК 004.94

## Проблематика отображения интерактивных трехмерных сцен в режиме реального времени

**А.И. Сорокин [*1], А.Ю. Карповский [*2]**

[*1] ведущий инженер научно-исследовательского отдела автоматизации горных машин

[*2] заведующий отделом научно-исследовательского отдела автоматизации горных машин

Государственное учреждение «Научно-исследовательский и проектно-конструкторский институт по автоматизации горных машин «Автоматгормаш им. В.А. Антипова»,
г. Донецк, oagm308@mail.ru

***Сорокин А.И., Карповский А.Ю. Проблематика отображения интерактивных трехмерных сцен в режиме реального времени.*** *В статье рассмотрены проблемы, возникающие при обеспечении режима реального времени графическими системами в задаче отображения интерактивных трехмерных сцен. В ходе анализа проблематики отображения интерактивных трехмерных сцен в режиме реального времени выявлено, что существующие графические системы реального времени не обеспечивают достаточной реалистичности изображения и быстродействия, требуют большого потребления памяти и вычислительных ресурсов, что актуализирует поиск новых методов повышения их эффективности.*

***Ключевые слова:*** *компьютерная графика, реальное время, трехмерная сцена, графический конвейер, уровни детализации, пространственные данные, параллельная обработка*

### *Введение*

Трехмерное моделирование и визуализация являются одной из технологий современных роботизированных и интеллектуальных производственных систем, обеспечивающих переход к Индустрии 4.0. Технология моделирования и визуализации интерактивных трехмерных сцен в режиме реального времени находит множество применений, среди которых: обучающие видеотренажеры, цифровые двойники промышленных предприятий, системы проектирования и моделирования, применяющиеся в строительстве, геодезии, промышленности, медицине. Вне зависимости от сферы применения, общим в информационных системах трехмерного моделирования и визуализации является использование трехмерной математической (геометрической) модели трехмерного объекта, воспроизводящего его характеристики и форму с определенной погрешностью и допущениями.

Помимо высокой скорости отклика, графические системы, визуализирующие трехмерную сцену в реальном времени, выполняют широкий список задач: воссоздание формы трехмерных объектов, задание пространственной динамики объектов, характерная раскраска поверхностей, моделирование освещенности. Одной из наиболее важных задач является моделирование и отображение геометрической формы объекта. С одной стороны, чем точнее будет модель объекта, тем более реалистичным будет его изображение, создаваемое графической системой. С другой стороны, существуют ограничения, накладываемые на графическую систему реального времени, такие как частота обновления изображения, создаваемого системой, и потребляемые вычислительные ресурсы. Поэтому при создании графических систем реального времени приходится находить компромисс между реалистичностью изображения трехмерных объектов и частотой обновления изображения.

Цель статьи: Обзор проблем, возникающих при отображении трехмерных сцен в режиме реального времени, анализ способов их решения, актуализация дальнейших исследований.

Для этого предлагается решить следующие задачи:

1) Рассмотреть проблемы, возникающие при отображении трехмерных сцен в режиме реального времени.

2) Провести обзор существующих способов решения проблем.

Рассмотрим обобщенную структуру графической системы, приведенной на рисунке 1. Хост-процессор выполняет моделирование трехмерной сцены (рассчитывает перемещение объектов, обрабатывает действия пользователя) и формирует список графических примитивов для визуализации, поступающих на вход геометрического процессора. Геометрический процессор выполняет преобразования над графическими примитивами, такие как тесселяцию (разделение одной поверхности на несколько полигонов), действия над вершинами (расчет освещения и цвета вершин).

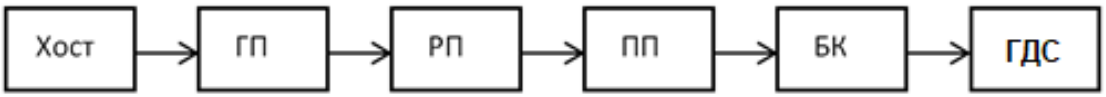



Рисунок 1 - Обобщенная структурная схема графической системы:
ГП – геометрический процессор; РП – растровый процессор; ПП – пиксельный процессор;
БК – буфер кадра; ГДС – графическая дисплейная система

Растровый процессор создает растровую проекцию векторной трехмерной сцены относительно выбранной точки наблюдения, формируя на выходе совокупность пикселей. Далее пиксели поступают на вход пиксельного процессора, где из поступивших определяются только видимые пиксели, и далее видимые пиксели поступают в буфер кадра. Из буфера кадра видимые пиксели поступают на вход графической дисплейной системы с частотой обновления. Последовательность этапов обработки, через которые проходят данные при преобразовании модели трехмерной сцены в изображение называется графическим конвейером. Аппаратно реализованный вычислительный модуль, выполняющий функцию графического конвейера, получил названия графический ускоритель (Graphics Accelerator), графический процессорный модуль (Graphics Processing Unit), визуальный процессор (Visual Processing Unit) [1].

Высокое быстродействие графических систем реального времени достигается различными способами. Проанализировав обобщенную структурную схему графической системы на рисунке 1, можно сделать вывод, что ускорить процесс визуализации можно либо уменьшив количество графических примитивов, поступающих на вход графического конвейера, либо сделать так, чтобы графический конвейер мог обработать большее количество графических примитивов в единицу времени. Таким образом, известные пути обеспечения режима реального времени можно разделить на два вида по их принципу действия и отношению к производительности графической системы.

**Уменьшение количества отображаемых графических примитивов.** Хост процессор уменьшает количество графических примитивов для визуализации до минимально-необходимого таким образом, чтобы не ухудшалась общая детализация итогового изображения. К таким способам можно отнести удаление невидимых поверхностей и применение уровней детализации. В результате на вход геометрического процессора поступает меньшее количество графических примитивов для обработки, и процесс визуализации занимает меньшее время. Общая производительность графической системы при этом (количество обрабатываемых графических примитивов в единицу времени) не изменяется.

**Повышение общей производительности графической системы.** К таким способам относятся применение параллельной обработки данных, создание новых форматов хранения трехмерных данных и алгоритмов их обработки. В результате растет общая производительность и графическая система может обработать одинаковое количество графических примитивов за меньшее время.

Рассмотрим проблемы, возникающие при реализации этих способов, а также сделаем обзор существующих решений.

### *Проблема удаления невидимых поверхностей*

Изображение трехмерной сцены, создаваемое графической системой реального времени, представляет собой монокулярную проекцию трехмерной сцены на двухмерную плоскость экрана монитора относительно точки наблюдения. На рисунке 2 представлена упрощенная схема отображения виртуальной модели трехмерной сцены.

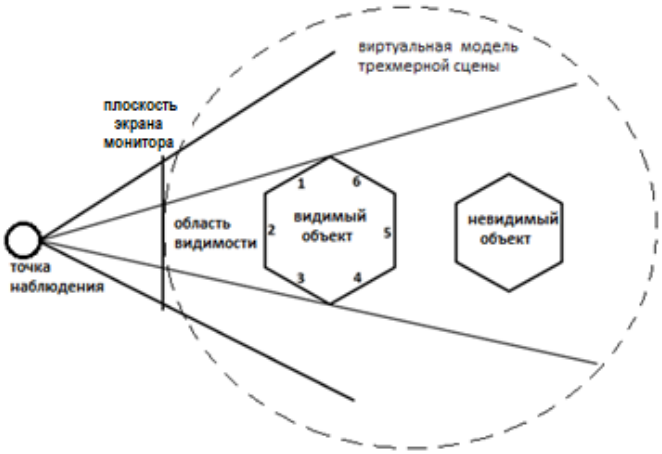



Рисунок 2 – Отображение виртуальной модели трехмерной сцены

Область видимости наблюдателя представляет собой пирамиду, ограниченную плоскостью графического монитора, с вершиной в точке наблюдения, при этом неоднородным основанием пирамиды выступает видимая поверхность трехмерной сцены, обращенная к наблюдателю в каждый конкретный момент времени.

Не все объекты трехмерной сцены попадают в область видимости наблюдателя: объекты могут быть закрыты от наблюдателя другими объектами, либо же находиться вне пирамиды зрения, ограниченной плоскостью графического монитора. При этом не все поверхности объектов, находящихся в области видимости, видны наблюдателю. На рисунке 2 видимыми являются только грани 1, 2, 3 видимого объекта; грани 4,5,6 являются невидимыми поверхностями, так как находятся на противоположной от точки наблюдения стороне объекта. Проблема удаления невидимых поверхностей состоит в том, чтобы отобразить на дисплее графического монитора только видимые поверхности и удалить невидимые поверхности из итогового отображения трехмерной сцены.

Рассмотрим следующие методы удаления невидимых поверхностей: метод буфера глубины, метод пространственной декомпозиции, пространственного индексирования объектов [2].

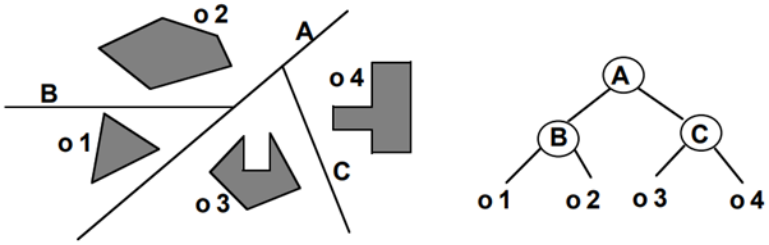
**Метод буфера глубины.** Суть метода буфера глубины состоит в том, что проблему видимости решают с помощью буфера глубины (также используется термин «Z-буфер»), представляющего собой двухмерный массив, каждый элемент которого соответствует определенному пикселю графической дисплейной системы и содержит значение, соответствующее удаленности отображаемого объекта от наблюдателя. Графическая система рассчитывает расстояние от объектов виртуальной трехмерной сцены до поверхности пикселя двумерного графического монитора, на который проецируется изображение сцены, и заносит это значение в буфер глубины. В случае если область видимости перекрывается несколькими объектами, то сохраняется то значение, которое соответствует меньшему расстоянию до точки наблюдения. В результате создается полутоновое изображение трехмерной сцены, значение каждого пикселя которого соответствует удаленности видимых поверхностей сцены от точки наблюдения[1].

Метод буфера глубины эффективен в случае аппаратной реализации. В случае программной реализации этот метод неэффективен, так как алгоритмы пространственной декомпозиции позволяют быстрее удалять невидимые поверхности. Метод буфера глубины неэффективен в случае отображения полупрозрачных объектов, так как структура данных, создаваемая этим методом, содержит данные только о ближайшей к точке наблюдения поверхности.

**Метод пространственной декомпозиции.** Двоичное разбиение пространства (также используется термин BSP - binary space partition) - это метод рекурсивного разбиения евклидова пространства в выпуклые множества и гиперплоскости (в случае для n-мерного пространства гиперплоскость – это пространство размерностью n-1). В результате объекты получают представление в виде древовидной иерархической структуры данных.

Двоичное дерево – это иерархическая структура данных, в которой каждый узел верхнего уровня ссылается на не более чем два узла нижнего уровня. Узел, находящийся на самом верхнем уровне, называется корнем, в то же время узлы, находящиеся на самых нижних уровнях, не имеют потомков. В случае метода пространственной декомпозиции, использующегося для удаления невидимых плоскостей, корнем двоичного дерева выступает первая проведенная гиперплоскость, промежуточными узлами – последующие плоскости, разделяющие подпространства сцены, а нижними узлами двоичного дерева – объекты либо графические примитивы моделируемой трехмерной сцены.

Рассмотрим алгоритм создания BSP-дерева с помощью примера декомпозиции пространства, приведенного на рисунке 3 [3].




а) б)

Рисунок 3 – Представление расположения объектов в виде двоичного дерева:
а) – расположение объектов в пространстве; б) сформированное двоичное дерево

Первоначально проводится (см. рисунок 2, а) плоскость «А», разделяющая пространство трехмерной сцены, при этом плоскость проводится таким образом, чтобы по обе стороны от нее было приблизительно поровну объектов. В случае с использованием полигональной модели в качестве гиперплоскости может быть выбрана плоскость, в которой лежит один из полигонов трехмерной модели. С одной стороны плоскости «А» остались объекты «о1» и «о2», по другую – «о3» и «о4». Далее этот же шаг повторяется с каждой половиной пространства, находящихся по обе стороны от плоскости. В результате следующего шага алгоритма плоскость «В» разделила объекты «о1» и «о2», а плоскость «С» - объекты «о3» и «о4».

После построения двоичного дерева, описывающего расположение объектов в пространстве трехмерной сцены, и зная положение точки наблюдени, возможно решить задачу удаления невидимых поверхностей с помощью следующего алгоритма. Двоичное дерево начинают обходить, начиная с корневой плоскости. Зная, с какой стороны корневой плоскости находится точка наблюдения, выбирается противоположное пространство. Затем этот шаг повторяется для всех элементов двоичного дерева, соответствующего противоположному пространству от того, в котором находится точка наблюдения: каждый раз выбирается дальний от точки наблюдения элемент двоичного дерева. Таким образом отображаются все элементы противоположного от точки наблюдения пространства, после чего алгоритм повторяется для того пространства, в котором расположена точка наблюдения.

**Метод пространственного индексирования объектов.** В этом методе так же, как и в методе пространственной декомпозиции, используется представление пространства в виде иерархического дерева, с тем лишь дополнением, что объекты заключаются в ограничивающие тела, что значительно ускоряет алгоритмы удаления невидимых поверхностей. Ограничивающее тело является описывающей фигурой объекта, в то же время для ограничивающего тела объект является вписанной фигурой.

Ускорение алгоритмов удаления невидимых поверхностей происходит потому, что ограничивающие тела являются простыми фигурами. Сначала операцию проверки видимости можно провести с ограничивающим телом (на этом этапе из дальнейшей проверки можно исключить объекты, которые не входят в область видимости), и только если ограничивающее тело пересекает область видимости, операцию удаления невидимых поверхностей проводится с каждым из полигонов сложного трехмерного объекта, описанного ограничивающим телом. При использовании метода иерархии ограничивающих объемов используются ограничивающие тела для групп объектов, которые в свою очередь заключены в ограниченные тела.

В компьютерной графике реального времени наибольшее распространение получили такие ограничивающие тела, как ориентированные вдоль координатных осей параллелепипеды AABB (Axis Aligned Bounding Box) и произвольно ориентированные параллелепипеды OBB (Oriented Bounding Box) [4]. Параллелепипеды, ориентированные вдоль координатных осей, рассчитываются более простым способом, так как содержат только прямые углы, а их ребра параллельны осям системы координат, использующейся в системе трехмерного моделирования и визуализации: достаточно знать крайние по трем осям точки объекта, чтобы построить ориентированный вдоль координатных осей прямоугольник. Произвольно ориентированные параллелепипеды – более сложный способ создания ограничивающих тел, потому что его наклон должен соответствовать форме и ориентации объекта в пространстве; в то же время этот способ более точно аппроксимирует форму объекта

Рассмотрим пример использования ограничивающих тел для удаления невидимых поверхностей на рисунке 4.

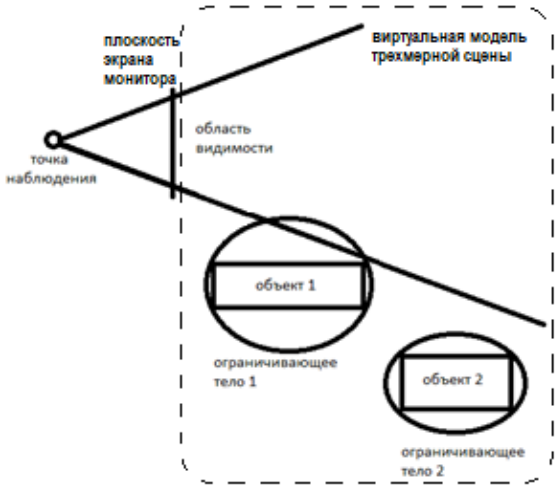



Рисунок 4 – Применение ограничивающих тел для удаления невидимых поверхностей

Ограничивающее тело объекта 2 находится вне зоны видимости, поэтому в процессе работы алгоритма удаления невидимых поверхностей оно будет индексировано, как невидимое. Ограничивающее тело объекта 1 пересекает зону видимости, поэтому полигоны объекта 1 будут далее проверяться на видимость. Заметим, что в данном примере область видимости пересекает ограничивающее тело объекта 1, но не пересекает объект 1. Такая ошибка возникает из-за неточности аппроксимации объекта ограничивающим телом, из-за чего между ограничивающим телом и объектом существует зазор.

Наиболее оптимальным считается такое ограниченное тело, которое наиболее точно повторяет форму объекта, наиболее плотно прилегает к объекту, между которым и объектом минимальный размер зазоров. В то же время, чем точнее ограничивающее тело, тем больше времени требуется для его создания и обработки. Ограничивающее тело должно иметь более простую форму, чем заключенный в него объект, содержать намного меньше полигонов и граней, иначе использование ограничивающих тел не вызовет ускорения алгоритмов удаления невидимых поверхностей. При создании ограничивающих тел нужно найти компромисс между точностью, с которой они аппроксимируют форму объекта, и сложностью ограниченного тела и алгоритма его создания.

### *Проблема уровня детализации*

Для уменьшения общего количества обрабатываемых графических примитивов в трехмерной сцене рационально использовать модель высокого разрешения трехмерного объекта только вблизи точки наблюдения, а на удаленном расстоянии от точки наблюдения использовать его упрощенный вариант. Для этого трехмерная сцена разделяется на уровни детализации (также применяется термин LOD - levels of detail), пропорционально расстоянию от точки наблюдения. Объекты, соответствующие ближайшему к точке наблюдения уровню детализации, отображаются с максимальной детализацией. Объекты, находящиеся на следующих уровнях детализации, отображаются в упрощенном виде [5].

При использовании уровней детализации не должна уменьшаться общая детализация изображения: упрощенная модель объекта, расположенная на определенном расстоянии от точки наблюдения, должна выглядеть примерно так же, как первоначальная модель объекта, находящаяся от точки наблюдения на таком же расстоянии. Для выполнения этих условий должна быть оптимальным образом рассчитана зависимость степени детализации объекта от его расстояния до точки наблюдения и должно быть необходимое количество уровней детализации.

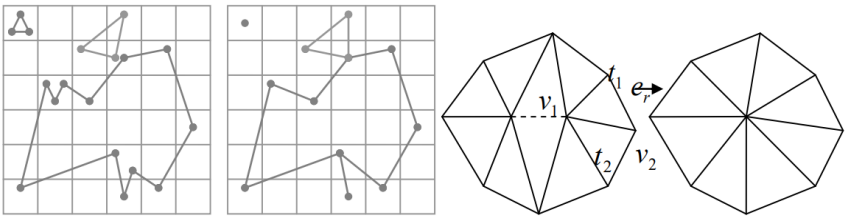
Для создания уровней детализации нужно создавать упрощенные модели трехмерных объектов с разной степенью детализации и отображать их, в зависимости от расстояния до точки наблюдения. Этого можно добиться двумя подходами: либо создать заранее несколько версий объекта в разном разрешении, либо упрощать объект непосредственно во время процесса визуализации. Таким образом, существует два подхода к управлению детализацией: статический и динамический [6 - 7].

**Статические (дискретные) уровни детализации (DLOD - Discrete Level Of Details).** Упрощенные вариации трехмерных объектов в нескольких фиксированных разрешениях создаются до процесса визуализации. Затем, во время интерактивного отображения трехмерной сцены в режиме реального времени, отображается версия объекта, соответствующая уровню детализации, на котором находится объект в данный момент времени.

**Динамические (непрерывные) уровни детализации (CLOD - Continuous Level of Details).** Упрощенные вариации трехмерных объектов создаются непосредственно при отображении трехмерной сцены в режиме реального времени. При перемещении точки наблюдения относительно трехмерной сцены, для каждого объекта сцены автоматически генерируется его упрощенная модель, со степенью детализации пропорциональной расстоянию до точки наблюдения.

Упрощать объекты можно либо за счет уменьшения количества полигонов, либо за счет уменьшения количества деталей.

**Применение алгоритмов упрощения.** Большинство итерационных алгоритмов упрощения полигональных моделей выполняют прореживание вершин или свертывание ребер, что приводит к уменьшению количества графических примитивов, из которых состоит модель. На рисунке 5 представлен принцип действия алгоритмов упрощения полигональных моделей [8].




а) б)

Рисунок 5 –Способы упрощение полигональных моделей:
а) прореживание вершин; б) свертывание ребер.

В результате выполнения данных алгоритмов, расположенные рядом несколько вершин(либо граней) заменяются одной вершиной(либо гранью), в итоге уменьшается количество полигонов, из которых состоит модель. На рисунке 6 представлена полигональная модель при разных уровнях детализации: LOD4 – максимальная степень детализации, LOD0 – минимальная [9].

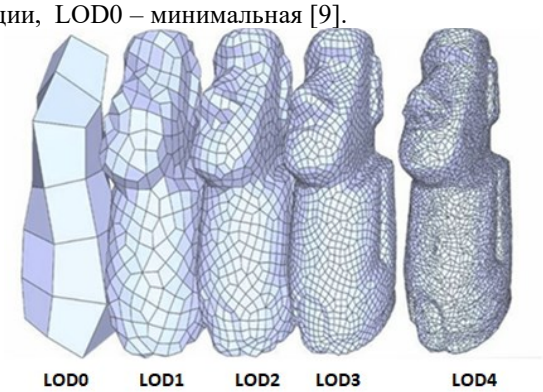



Рисунок 6 – Уровни детализации трехмерной модели, полученные с помощью алгоритмов геометрического упрощения

**Уменьшение количества деталей.** При отдалении объекта от точки наблюдения постепенно исчезают детали объекта. Рассмотрим пример упрощения объекта путем уменьшения количества деталей на рисунке 7 [10].

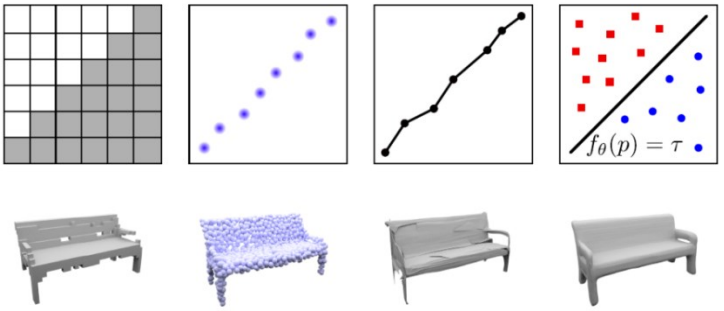

Рисунок 7 – Уровни детализации трехмерной модели, полученные уменьшением количества деталей

При максимальном уровне детализации (LOD4) отображается дом с интерьером внутри. При LOD3 исчезает интерьер, при LOD2 окна и дверь. При LOD1 исчезает крыша и модель дома предстает в виде параллелепипеда, при LOD0 модель становится прямоугольником.

### *Проблема представления пространственных данных*

Поиск новых методов моделирования пространственных форм, методов компрессии пространственных данных и алгоритмов их обработки является одним из способов повышения производительности и преодоления недостатков существующих графических систем. Разработка новых форм представления пространственных данных позволит увеличить быстродействие в графических системах, снизить потребление памяти и требования к вычислительным ресурсам [1].

Существуют различные способы представления трехмерных данных (см. рисунок 8) [11]. В компьютерной графике и САПР наибольшее распространение получили полигональная и криволинейная форма представления пространственных данных. Рассмотрим эти способы.




а) б) в) г)

Рисунок 8 – Способы представления трехмерных данных:
а) – блочная модель; б) – облако точек; в) – полигональная модель; г) – криволинейная модель

**Полигональное представление.** При этом способе геометрического моделирования трехмерные модели представляются в виде прилегающих друг к другу плоских полигонов. В трехмерном компьютерном моделировании используются форматы данных, представляющие полигональные модели в виде взаимосвязанной структуры вершин, ребер и полигонов. Описания вершин содержат данные о положении вершин в пространстве в декартовой трехмерной системе координат. Описания ребер содержат ссылки на две вершины и не более двух ссылок на смежные полигоны. Описания полигонов содержат ссылки на образующие их ребра. Такая иерархическая структура данных позволяет снизить потребление памяти и ускорить обработку пространственных данных.

Преимущества полигональных моделей в задачах визуализации в режиме реального времени связаны с тем, что они имеют плоскую форму. Полигональную форму представления проще использовать в задачах удаления невидимых поверхностей, наложения текстурных изображений, расчета освещения.

Недостатком полигональных моделей является низкая реалистичность при малом количестве полигонов, большое потребление памяти и вычислительных ресурсов – при большом количестве полигонов.

**Криволинейное представление.** Помимо полигональных моделей в графических системах применяются модели на основе криволинейных примитивов. В этом случае ближайшие точки поверхности трехмерного объекта соединяет не прямая линия, как в случае с полигональным представлением, а кривая, описанная определенным аналитическим уравнением. Для криволинейного представления объектов в задачах трехмерного моделирования исследованы возможности фрактальных поверхностей, сетей Цао Ена, поверхностей свертки, поверхностей на основе вейвлет-функций [12 - 15]. Однако широкое применение в трехмерной графике реального времени нашли только модели на основе многочленов третьей степени (кубических сплайнов). Выбор третьей степени объясняется минимально необходимой степенью, при которой возможно обеспечить гладкое сопоставление кривых.

Несмотря на большую реалистичность, криволинейные модели сложнее отобразить в режиме реального времени, чем полигональные. Для удаления невидимых поверхностей или расчета освещенности требуется провести нормаль к большому количеству точек криволинейной поверхности, в то время как для плоского полигона в таких задачах достаточно расчета одной нормали для всей плоскости. Также сложнее и применение текстурных изображений, так как для большого количества точек криволинейной поверхности требуется задать текстурные координаты. Исследователями прорабатывается аппаратная поддержка криволинейных примитивов, но эта работа не завершена. Поэтому на практике сплайновые модели на определенном этапе визуализации подвергают триангуляции (тесселяции). Но для реалистичного отображения сложной модели в таком случае понадобится значительное количество полигонов, что в совокупности с затратами на триангуляцию модели снизит быстродействие графической системы, либо повысит требования к вычислительным ресурсам.

### *Проблема параллельной обработки данных*

Для отображения сложных трехмерных сцен зачастую не хватает производительности одного графического конвейера. Повышения скорости обработки данных и производительности можно достичь применением мультипроцессорных устройств и параллельных вычислительных алгоритмов. В этом случае для эффективной работы параллельной системы визуализации нужно решить проблему сбалансированной загрузки параллельно работающих блоков графической системы, распределения потоков данных и компоновки отдельных структур данных для получения на выходе системы единого изображения.

На рисунке 9 изображена структура параллельной системы визуализации с двумя возможных соединениями параллельных ветвей.

Выбор места соединения зависит от способа параллельной обработки данных. Соединение «А» используется для маршрутизации примитивов в соответствующие растровые процессоры, соединение «В» – для маршрутизации фрагментов после процедуры растеризации. Выбор соединения зависит от того, какой способ используется. Существует три способа параллельной обработки: параллельная обработка в пространстве сцены, параллельная обработка в пространстве изображения и алгоритмический параллелизм [1, 16].

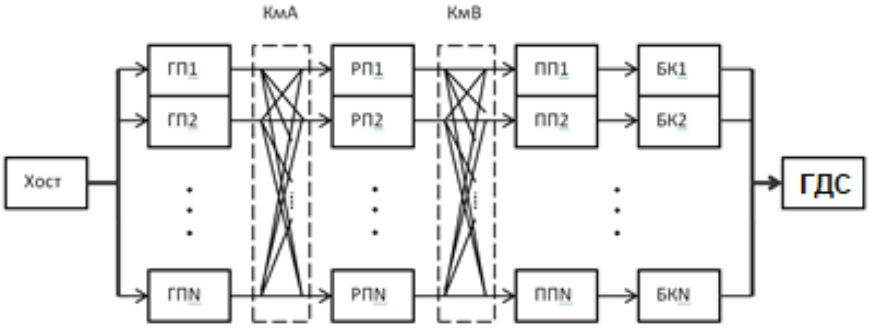



Рисунок 9 – Структурная схема многоканальной графической системы с двумя вариантами коммутации каналов: ГП – геометрический процессор; РП – растровый процессор; ПП – пиксельный процессор; БК – буфер кадра; ГДС – графическая дисплейная система; Км - коммутатор

**Параллельная обработка в пространстве сцены.** Применение параллельной обработки в пространстве сцены предусматривает разбиение всей сцены на фрагменты и генерирование каждым графическим конвейером полноэкранного изображения части объектов сложной трехмерной сцены. Результирующее изображение трехмерной сцены получается наложением полноэкранных изображений с различными областями сцены.

Так как разные области трехмерной сцены могут иметь разный объем, количество объектов и их сложность, для обеспечения равномерной нагрузки между растровыми и пиксельными процессорами параллельных конвейеров требуется распределение данных на выходе геометрических процессоров. Для этого применяется коммутатор, который перераспределяет графические примитивы с разных геометрических процессоров между устройствами растеризации в зависимости от их нагрузки (см. коммутатор «А» на рисунок 9).

Основное достоинство этого способа – с ростом сложности сцены происходит рост возможного распараллеливания. В то же время, поскольку каждый растровый процессор может получить данные о графическом примитиве из любого геометрического процессора, блок маршрутизации является ограничивающим фактором и накладывает ограничение на число возможных параллельных ветвей графической системы.

**Параллельная обработка в пространстве изображения.** При параллельной обработке в пространстве изображения, параллельно работающие графические конвейеры генерируют изображение всей трехмерной сцены для отдельных участков изображения, а единое изображение получают компоновкой отдельных фрагментов изображения.

Насыщенность участков изображения может быть разной, поэтому для равномерного распределения данных объекты, относящиеся к одному участку изображения, могут обрабатывать несколько геометрических и растровых процессоров. В таком случае на выходах растровых процессоров требуется коммутатор «В» (см. рисунок 9) который распределяет данные, поступающие из растровых процессоров, между пиксельными процессорами в соответствии с теми участками экрана, к которым они относятся.

Коммутатор «В» также выступает и ограничивающим фактором. Уменьшать размер параллельно обрабатываемых участков изображения и увеличивать количество параллельных графических конвейеров при таком способе можно до определенного предела, после чего производительность перестает расти. Это связано с тем, что несколько растровых процессоров могут отправлять данные в один пиксельный процессор, что приведет

к возникновению очередей.

**Алгоритмический параллелизм.** Этот метод предполагает возможность визуализации каждой параллельной ветвью графической произвольной части сцены, которая может попасть в произвольный участок итогового изображения. При этом способе графическая система рассматривается как комбинация независимых параллельных ветвей. Распределение графических примитивов между ветвями осуществляет хост-процессор или командный процессор в составе графического ускорителя. Выходные данные параллельных ветвей объединяются в конвейере, в котором видимость пикселей определяется очередностью их поступления.

Преимуществом такого подхода является отсутствие ограничений на количество параллельно работающих ветвей графической системы. Вместе с тем, сложной задачей является организовать оптимальное распределение графических объектов для визуализации между независимо работающими ветвями графической системы.

### *Заключение*

Обоснованы актуальные проблемы, связанные с отображением трехмерных графических сцен в режиме реального времени, выполнен анализ методов их решения, которые позволят перейти к эффективному проектированию трехмерных сцен с учетом ключевых факторов: скорости отображения, затраченных аппаратных и программных ресурсов ЭВМ.

Рассмотрены преимущества и недостатки существующих методов повышения быстродействия графических систем. Рассмотренные методы повышения быстродействия рекомендованы к применению для проектирования эффективных решений, направленных на создание графических систем реального времени.

Среди действующих методов решения проблем обеспечения реального времени не установлено единого универсально-эффективного, что актуализирует исследования, связанные с развитием действующих методов, а также применением комбинаций методов для обработки пространственных данных. В свою очередь, исследования, направленные на эффективное проектирование графических систем реального времени, окажут большое влияние на развитие фундаментальных наук.

### *Литература*


1. Косников Ю.Н. Геометрическое моделирование в графических системах реального времени. – Пенза: Информационно-издательский центр Пензенского государственного университета, 2005. – 218 с.

2. Гонахчян В.И. Модель производительности графического конвейера для однопроходной схемы рендеринга динамических трехмерных сцен // Труды института системного программирования РАН. 2020. №4 (32). стр. 53–72.

3. Naylor, Bruce. (2005). A Tutorial on Binary Space Partitioning Trees [Электронный ресурс] / . – Режим доступа: https://www.researchgate.net/publication/238348725_A_Tutorial_on_Binary_Space_Partitioning_Trees

4. Ионес А. Л. Построение оптимальных ограничивающих тел в компьютерной графике : Автореферат диссертации на соискание ученой степени кандидата кандидата физико-математических наук : 05.13.16 / Санкт-Петербург. гос. техн. ун-т.- Санкт-Петербург, 1998.- 16 с.: ил. РГБ ОД, 9 98-6/2553-8

5. Level Of Detail (LOD) Trees [Электронный ресурс] /. – Режим доступа: https://www.ixbt.com/video2/terms2k5.shtml#lod

6. LOD технология [Электронный ресурс] /. – Режим доступа: https://web.archive.org/web/20150626124532/http://zen-designer.ru/vopros-otvet/36-entsiklopediya/47-l/96-lod-tekhnologiya

7. Осипов М.П. Фотореалистичное моделирование и визуализация районов городской среды : Учебно-методическое пособие – Нижний Новгород: Нижегородский госуниверситет, 2014. – 50 с.

8. Тозик В.Т., Меженин А.В. Метод геометрического упрощения 3D полигональных объектов // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики, 2010, № 3(67) С.81-85

9. Форматы файлов 3D моделей: типы и характеристики [Электронный ресурс] /. – Режим доступа: https://3dradar.ru/post/54889/

10. Eriksson, Helen & Johansson, & Olsson, Per-Ola & Andersson, & Engvall, & Hast, & Harrie, Lars. (2020). Requirements, Development, and Evaluation of A National Building Standard—A Swedish Case Study. ISPRS International Journal of Geo-Information. 9. 78. 10.3390/ijgi9020078.

11. Mescheder, Lars & Oechsle, Michael & Niemeyer, Michael & Nowozin, Sebastian & Geiger, Andreas. (2019). Occupancy Networks: Learning 3D Reconstruction in Function Space. 4455-4465. 10.1109/CVPR.2019.00459.

12. Роджерс Д., Адамс Дж. Математические основы машинной графики. – М.: Мир, 2001. – 604 с.

13. Вайвил Д., Цао Ен, Тротмэн А. Поверхность Цао Ена: новый подход к геометрическим моделям произвольных форм //Программирование, 1992, №4. – С.4.

14. Bloomenthal J., Shoumaker K. Convolution Surfaces //SIGGRAF Proceedings, 1991. – P. 172.

15. Barnsley M. Fractals Everywhere, Second Edition. – Academic Press, San Diego, CA, 1993. – 370 p.
16. Тимофеев А. В. Разработка и повышение производительности параллельной системы визуализации трехмерных сцен : Автореферат диссертации на соискание ученой степени кандидата технических наук : 05.13.13 / Санкт-Петербург. гос. электротехн. ун-т им. В. И. Ульянова (Ленина).- Санкт-Петербург, 1997.- 16 с.: ил. РГБ ОД, 9 98-3/4085-1

***Сорокин А.И., Карповский А.Ю. Проблематика отображения интерактивных трехмерных сцен в режиме реального времени.*** *В статье рассмотрены проблемы, возникающие при обеспечении режима реального времени графическими системами в задаче отображения интерактивных трехмерных сцен. В ходе анализа проблематики отображения интерактивных трехмерных сцен в режиме реального времени выявлено, что существующие графические системы реального времени не обеспечивают достаточной реалистичности изображения и быстродействия, требуют большого потребления памяти и вычислительных ресурсов, что актуализирует поиск новых методов повышения их эффективности.*

***Ключевые слова:*** *компьютерная графика, реальное время, трехмерная сцена, графический конвейер, уровни детализации, пространственные данные, параллельная обработка.*

***Sorokin Alexander, Karpovsky Artur Problematics of displaying interactive three-dimensional scenes in real time.*** *The article considers the problems arising in providing real-time mode by graphic systems in the task of displaying interactive three-dimensional scenes. In the course of analysis of the problems of displaying interactive three-dimensional scenes in real-time mode it is revealed that the existing real-time graphics systems do not provide sufficient realism of the image and performance, require large consumption of memory and computational resources, which actualizes the search for new methods of increasing their efficiency.*

***Key words:*** *computer graphics, real time, three-dimensional scene, graphics pipeline, levels of detail, spatial data, parallel processing.*

УДК 004.94

# Подход к обнаружению и извлечению процессов из журналов событий мобильного приложения на основе ретроспективной и экспертной информации

**Я.К. Савельев[*1], А.А. Филиппов[*2]**

[*1] аспирант, Ульяновский государственный технический университет, SPIN-код: 9396-0338, kpuccmajl@gmail.com

[*2] к.т.н., доцент, Ульяновский государственный технический университет, al.filippov@ulstu.ru

***Савельев Я.К., Филиппов А.А. Подход к обнаружению и извлечению процессов из журналов событий мобильного приложения на основе ретроспективной и экспертной информации.*** *Даже опытные аналитики способны ошибаться при проектировании новых функций приложения на основе моделирования предполагаемых действий пользователей с компонентами пользовательского интерфейса. Был разработан алгоритм для анализа журнала событий, который был сформирован в процессе записи действий пользователей некоторого мобильного приложения. Предложенный алгоритм позволяет решить задачу обнаружения и извлечения процессов на основе анализа журнала событий. Полученные процессы отражают реальную модель взаимодействия пользователей с приложением, что может позволить повысить качества принятия проектных решений.*

***Ключевые слова:*** *компьютерное моделирование, извлечение процессов, модели процессов, кластеризация, аналитика, мобильные приложения*

## *Введение*

В последние годы наука о данных (data science) стала активно применяться в производстве для решения различных задач. Наука о данных может рассматриваться как объединение следующих классических дисциплин: статистика, интеллектуальный анализ данных, базы данных и распределенные системы.

Методы интеллектуального анализа процессов (Process Mining) основаны на анализе данных о событиях для обнаружения и извлечения процессов, проверке их соответствия (валидации) с целью нахождения узких мест в таких процессах и дальнейшей их оптимизации [1]. В качестве исходных данных при анализе процессов обычно выступают различные журналы событий (логи).

Интеллектуальный анализ процессов достаточно недавно стал отдельной дисциплиной науки о данных. Но соответствующие методы могут быть применены к любому типу операционных процессов (организации и системы) [2].

Например, применение методов анализа процессов может помочь разработчикам и аналитикам мобильного приложения сформировать наиболее удобный для пользователя интерфейс. Повышение удобства предполагает снижение сложности выполнения функций приложения за счет сокращения необходимых действий с элементами управления

Спектр применения методов интеллектуального анализа процессов весьма разнообразен: интеллектуальный анализ процессов также может использоваться для проверки соответствия некоторому эталону, диагностики отклонений, выявления узких мест, повышения эффективности, прогноза времени выполнения процессов и т. д.

Практически любую деятельность можно представить в виде журналов событий для обнаружения и извлечения из них моделей процессов, а методы анализа процессов не ограничиваются только техническими областями [3].

Формально модель процесса можно представить в виде системы переходов [1, 2]:

$$TS = \langle S, A, T \rangle,$$

где $S$ – множество состояний;

$A$ – множество событий;

$T$ – множество переходов.

События $A$ запускают переходы $T$ из некоторого состояния $S_0$ в состояние $S_1$. Основной проблемой использования системы переходов является отсутствие учета параллельных процессов, так как количество вариантов причинно-следственных связей может достигать величины $n!$. Поэтому на практике для моделирования процессов чаще применяется сеть Петри.

Обычно для обнаружения процессов используется алгоритм Inductive Miner [4]. Алгоритм Inductive Miner основан на идее разделения журналов событий на поджурналы, называемые разрезами или сплитами. Получение поджурналов осуществляется на основе анализа графа, который в дальнейшем используется для обнаружения множества разрезов для представления последовательности выполнения действий. Основным преимуществом Inductive Miner является его гибкость и масштабируемость.

Алгоритм Inductive Miner итеративно исследует пространство возможных моделей процессов и способен обнаружить широкий спектр структур процессов, от линейных до более сложных моделей с параллелизмом, циклами и разветвлениями.

### *Постановка проблемы*

В данным момент существует две наиболее популярные мобильные операционные системы: это Android и iOS. И если в первой половине 2010-х годов нативная разработка, при которой предполагается разделение команд разработки на два (или более, в зависимости от конечных целевых платформ) отдела, которые работают с конкретной платформой, была первоочередной, то после появления кроссплатформенных решений React Native и Flutter подход к формированию команд серьёзно изменился. На данный момент разделение внутри команды разработки идёт не по платформенному признаку, а по модульному признаку, т. к. теперь кодовая база для всех платформ зачастую единая, что в итоге экономит до 50% времени разработки, и приводит к более оптимальному подходу к сопровождению, когда не требуется команда из людей со специфичными знаниями.

С другой стороны, модульный принцип разработки ведёт к росту функций и компонентов программной системы, особенно если поддержка и развитие не останавливаются после выпуска первой версии.

При разработке новых функций приложений могут быть затронуты существующие функции и модули. В некоторых случаях может потребоваться внесение более серьезных изменений. Определенные нововведения могут лишить пользователей доступа к привычному им набору функций и изменить и/или усложнить процесс взаимодействия с приложением.

Перед внедрением новых функций аналитик проекта должен проанализировать журнал событий и выявить некоторую закономерность поведения пользователя, чтобы составить понимание основного алгоритма при использовании приложения. Но работа с чистым журналом событий крайне трудоёмкий процесс и чаще всего аналитики при построении диаграмм последовательности ориентируются на некоторое ожидаемое поведение, которое может отличаться от реального [5].

Таким образом, необходимо разработать подход к обнаружению и извлечению процессов из журнала событий мобильного приложения для повышения качества принятия проектных решений [6].

### *Обнаружение процессов в журнале событий*

Значительная часть крупных мобильных приложений обладает механизмами сбора аналитики по использованию приложений с точки зрения различных пользователей.

Существует большое количество платформ для работы с аналитикой событий, собранной в процессе работы приложения. Для мобильных устройств самым распространённым является Google Analytics, Yandex AppMetrica и др. В рамках данного исследования будет использоваться Yandex AppMetrica.

Будем рассматривать журнал событий как:

$$L = \{l_1, l_2, \dots, l_i, \dots, l_n\},$$

где $l_i = \langle id, timestamp, user\_id \rangle$ – $i$-я запись журнала $L$, в которой:

$id$ – идентификатор события;

$timestamp$ – (временная метка события;

$user\_id$ – идентификатор пользователя.

Для решения поставленной задачи был разработан алгоритм обнаружения и извлечения процессов, так как существующие алгоритмы не позволяют учитывать специфику проблемной области разработки мобильных приложений.

Формально алгоритм можно представить как:

$$F: L \rightarrow TS.$$

В модель процесса $TS$ были внесены изменения:

$$TS = \langle S, A, T, W \rangle.$$

Как видно из представленного выше выражения, был добавлен компонент $W$, которые позволяет указать вес перехода из множества $T$.

Алгоритм состоит из следующих шагов:

1. Загрузка журнала событий $L$ с сортировкой по параметру $timestamp$.
2. Группировка событий по $user\_id$:

$$L = \{L^{user\_id1}, L^{user\_id2}, \dots, L^{user\_idn}\}.$$

Данный шаг позволяет привязать записи журнала к конкретному пользователю ($user\_id$) для разделения активностей различных пользователей.

3. Группировка пользовательских событий по $id$:

$$L^{user\_idi} = \{L_{id1}, L_{id2}, \dots, L_{idm}\}.$$

Группировка выполняется на основе выполнения алгоритма кластеризации k-means для формирования отдельных по времени групп активностей пользователя.

4. Для каждого $L_{idj}^{user\_idi}$ формируется матрица переходов с учетом веса активностей, уровня шума и отмененных событий. Под шумом понимаются активности, вес которых ниже определенного порога или сама активность не имеет значения в рамках текущего приложения.

### *Загрузка журналов событий с использованием сервиса AppMetrica*

Для получения журнала событий мобильного приложения из сервиса Yandex AppMetrica необходимо перейти в раздел «Экспорт данных» и указать параметры экспорта данных. Важно указать параметр «Лимит» таким образом, чтобы разработанное программное средство могло обработать полученные данные с учетом их объема. Текущая версия разработанного программного средства не имеет возможности анализировать только часть журнала событий и обрабатывает весь файл путем загрузки его содержимого в оперативную память. Также необходимо выбрать для выгрузки следующие параметры журнала: event_name, session_id, event_datetime. В качестве формата выгрузки необходимо указать «CSV» (рисунок 1).

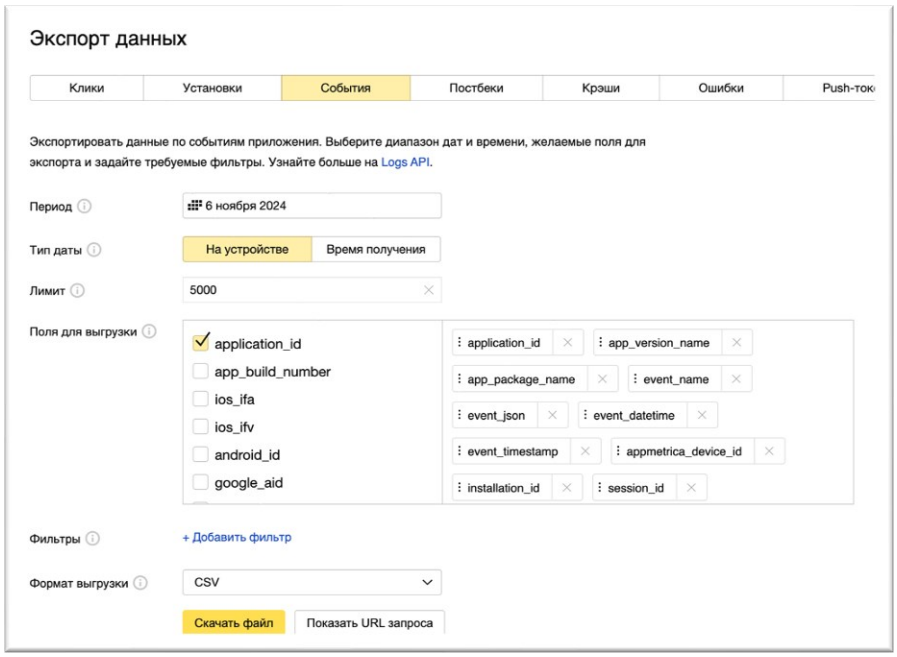



Рисунок 1 – Параметры загрузки журнала событий

Сервис AppMetrica имеет функцию экспорта данных через обращение к API с использованием OAuth-аутентификации. При работе с API сервиса необходимо указать: идентификатор приложения в системе AppMetrica (application_id), период получения данных (date_since и date_until), требуемые поля для выгрузки (fields) и количество записей для выгрузки (limit).

Пример запроса:

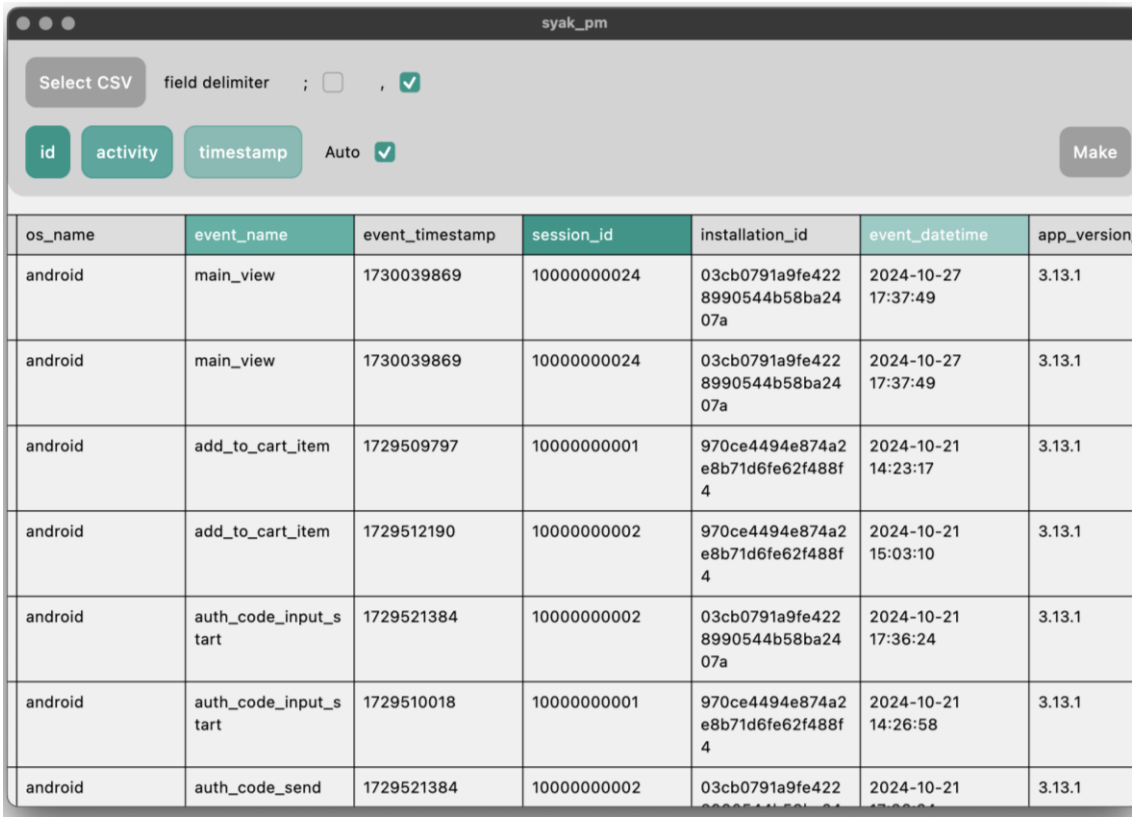
*https://api.appmetrica.yandex.ru/logs/v1/export/events.csv?*
*application_id=4543468*
*&date_since=2024-10-17%2000%3A00%3A00*
*&date_until=2024-10-17%2023%3A59%3A59*
*&fields=application_id%2Cprofile_id%2Cevent_name%2Cevent_json%2Cevent_datetime%2Csession_id%2Capp_version_name*
*&limit=5000*

Функция загрузки журнала событий через взаимодействие с API сервиса AppMetrica в рамках процесса разработки представленного программного средства находится в стадии разработки.

### *Реализация подхода для обнаружения и извлечения процессов*

Для оценки адекватности предложенного подхода было разработано приложение на языке Dart с использованием фреймворка Flutter.

На рисунке 2 представлен основной экран приложения. На данном экране пользователь может загружать файл с журналом событий и определять какие атрибуты журнала относятся основным параметрам: $id$, $timestamp$ и $user\_id$.




| os_name | event_name | event_timestamp | session_id | installation_id | event_datetime | app_version |
|---|---|---|---|---|---|---|
| android | main_view | 1730039869 | 10000000024 | 03cb0791a9fe4228990544b58ba2407a | 2024-10-27 17:37:49 | 3.13.1 |
| android | main_view | 1730039869 | 10000000024 | 03cb0791a9fe4228990544b58ba2407a | 2024-10-27 17:37:49 | 3.13.1 |
| android | add_to_cart_item | 1729509797 | 10000000001 | 970ce4494e874a2e8b71d6fe62f488f4 | 2024-10-21 14:23:17 | 3.13.1 |
| android | add_to_cart_item | 1729512190 | 10000000002 | 970ce4494e874a2e8b71d6fe62f488f4 | 2024-10-21 15:03:10 | 3.13.1 |
| android | auth_code_input_start | 1729521384 | 10000000002 | 03cb0791a9fe4228990544b58ba2407a | 2024-10-21 17:36:24 | 3.13.1 |
| android | auth_code_input_start | 1729510018 | 10000000001 | 970ce4494e874a2e8b71d6fe62f488f4 | 2024-10-21 14:26:58 | 3.13.1 |
| android | auth_code_send | 1729521384 | 10000000002 | 03cb0791a9fe422 | 2024-10-21 | 3.13.1 |

Рисунок 2 – Загрузка журнала событий и выбор основных параметров

При нажатии на кнопку «Make» происходит обнаружение и извлечение процессов на основе предложенного алгоритма.

Параметр Auto отвечает за работу алгоритма кластеризации событий журнала. При активном параметре распределение событий на временной шкале формируется на основе кластеров, рассчитанных автоматически. При отключении данного параметра пользователь может указать конкретный интервал определения длительности пользовательского взаимодействия.

Пользователю программного продукта предоставляется возможность в режиме реального времени вносить изменения в модель процессов: изменить уровень шума (чтобы исключить редкие события), удалять из модели события, которые не должны отображаться в модели (рисунок 3).

Также имеется возможность рассмотреть модели процессов как для всей системы в целом, так и для выбранного пользователя.

На экране модификации полученной модели процессов оператор может путём исключения редких событий, настройки уровня шума и выбора интервала получить желаемый результат модели процессов с учётом достаточной информативности и исключения аномалий и событий, не несущих смысловой логики.

Предоставляется возможность оператору откатить изменения, если его действия привели к искажению полученного результата.

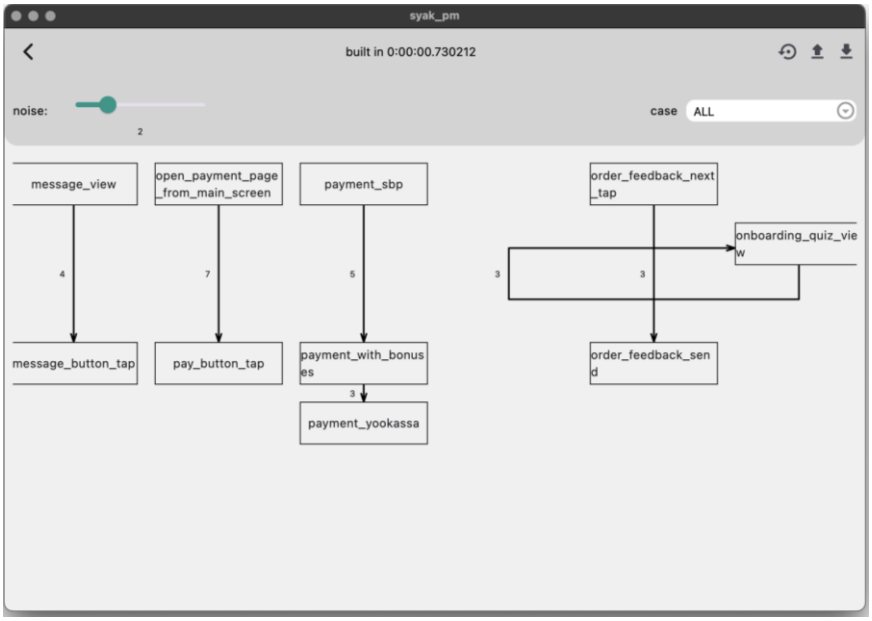



Рисунок 3 – Фрагмент полученной модели с уровнем шума в 2 пункта

После того, как аналитик изменит параметры полученной модели процессов, он может сохранить данную модель в JSON-файле для дальнейшей валидации или продолжения работы с ней.

Фрагмент полученной модели процессов:

*{…{id: order_view_back_tap, counter: 1}, {id: payment_failed_view, counter: 1}, {id:open_payment_page_from_order_details,counter:1}]},{"fromId":"partner_deeplink_tap","to":[{"id":"catalog_category_view","counter":1},{"id":"catalog_product_view","counter":3},{"id":"catalog_tab_open","counter":2}]}],"disabled":["story_open","story_item_view","main_view","order_view_back_tap","order_view_from_main_screen"],"noise": 4.0}*

Таким образом, аналитику необходимо подобрать такое значение уровня шума, чтобы выделить основной путь использования приложения. Ведущий аналитик мобильного приложения знает, какие события являются основными для анализа, и настраивает уровень шума таким образом, чтобы они учитывались в полученной модели.

### *Валидация извлечённых процессов*

Для реализации метода валидации извлеченных процессов разработан следующий алгоритм:

1. Пользователь указывает данные для подключения к API Yandex AppMetrica чтобы автоматизировано получить события за N различных дней работы приложения.
2. Для каждого -го из $n$ обнаруженных и извлеченных процессов $TS_i$.
3. Программное средство автоматически формирует модели процессов для загруженных отрезков данных с учётом параметров основной модели процессов $\widehat{TS}$.

После формирования моделей процессов происходит их сверка с исходной моделью $F: TS_i \times \widehat{TS} \rightarrow Diff_i$. Выявляются совпадения и расхождения, к каждому событию добавляется параметр частоты повторения при валидации.

Строится итоговая модель процессов с учетом расхождений и повторений, где для каждого события добавляется рассчитывается валидации $Vdiff = [0, 1]$. При значении показателя $Vdiff$ равном 0 данное событие отсутствует во всех полученных моделях, а при значении 1.0 – присутствует абсолютно во всех полученных

моделях.

Реализация данного алгоритма валидации находится в стадии активной разработки.

### *Заключение*

Даже опытные аналитики способны ошибаться при проектировании новых функций приложения на основе моделирования предполагаемых действий пользователей с компонентами пользовательского интерфейса. Был разработан алгоритм для анализа журнала событий, который был сформирован в процессе записи действий пользователей некоторого мобильного приложения. Предложенный алгоритм позволяет решить задачу обнаружения и извлечения процессов на основе анализа журнала событий. Полученные процессы отражают реальную модель взаимодействия пользователей с приложением, что может позволить повысить качества принятия проектных решений.

К недостаткам предложенного подхода можно отнести:

1. Для корректной работы требуется качественный журнал событий.
2. Низкое качество полученных моделей процессов с настройками по умолчанию

### *Литература*

12. Van Der Aalst W. Process mining: Overview and opportunities // ACM Transactions on Management Information Systems (TMIS). 2012. Vol. 3(2). Pp. 1-17.

13. Process mining manifesto / Van Der Aalst W. et al. //Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I 9. Springer Berlin Heidelberg, 2012. Pp. 169-194.

14. Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs / Suriadi S. et al. // Information systems. 2017. Vol. 64. Pp. 132-150.

15. Using inductive miner to find the most optimized path of workflow process / Pulsanong W. et al. //2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE). IEEE, 2017. Pp. 1-5.

16. Using process mining to leverage the development of a family of mobile applications / Rezunik L. A. et al. // Труды института системного программирования РАН. 2023. Vol. 35 (3). Pp. 171-186.

17. Подход к автоматизации проектирования и управлению проектами программных систем / А.В. Бармина и др. // Автоматизация процессов управления. 2022. № 4(70). С. 86–97.

***Савельев Я.К., Филиппов А.А. Подход к обнаружению и извлечению процессов из журналов событий мобильного приложения на основе ретроспективной и экспертной информации.*** *Даже опытные аналитики способны ошибаться при проектировании новых функций приложения на основе моделирования предполагаемых действий пользователей с компонентами пользовательского интерфейса. Был разработан алгоритм для анализа журнала событий, который был сформирован в процессе записи действий пользователей некоторого мобильного приложения. Предложенный алгоритм позволяет решить задачу обнаружения и извлечения процессов на основе анализа журнала событий. Полученные процессы отражают реальную модель взаимодействия пользователей с приложением, что может позволить повысить качества принятия проектных решений.*

***Ключевые слова:*** *компьютерное моделирование, извлечение процессов, модели процессов, кластеризация, аналитика, мобильные приложения*

***Savelyev Y.K., Filippov A.A. Approach to process detection and extraction from mobile application event logs based on retrospective and expert information.*** *Even experienced analysts are capable of making mistakes when designing new application features based on modeling the intended user actions with user interface components. An algorithm was developed to analyze the event log that was generated while recording user actions of some mobile application. The proposed algorithm solves the problem of process detection and extraction based on event log analysis. The obtained processes reflect the real model of user interaction with the application, which may allow improving the quality of design decision making.*

***Keywords:*** *computer simulation, process extraction, process models, clustering, analytics, mobile applications*

УДК 004.94

# Визуализация информации в системах компьютерного моделирования: анализ и выбор алгоритмов для динамических графов

**Н.А. Бездетный[*1], С.А. Зори[*2]**

[*1] аспирант, Донецкий национальный технический университет,
nekooolay@mail.ru, SPIN-код: 2472-1006

[*2] д.т.н., проф. кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет,
ik.ivt.rec@mail.ru, OrcID: 0000-0003-4018-234X, SPIN-код: 3565-6330

***Зори С. А., Бездетный Н. А. Визуализация информации в системах компьютерного моделирования: анализ и выбор алгоритмов для динамических графов.*** *В статье представлен углубленный анализ методов визуализации графовых структур, применяемых в системах компьютерного моделирования. Детально рассмотрены алгоритмы, основанные на физических силах, иерархические и древовидные алгоритмы, с акцентом на их применимость к динамическим графам. Выявлены их преимущества и недостатки, а также обоснован выбор и предложены модификации иерархического алгоритма для эффективной визуализации динамических графов с возможностью сворачивания графа, что способствует повышению интерактивности и улучшению понимания сложных структур данных.*

***Ключевые слова:*** *визуализация графов, компьютерное моделирование, force-directed layout, hierarchical (layered) layout, tree layout, динамические графы, сворачивание графа, интерактивность.*

## *Введение*

Визуализация графовых структур является неотъемлемой частью современных систем компьютерного моделирования, предоставляя исследователям мощный инструмент для анализа и интерпретации сложных взаимосвязей между объектами [1]. Выбор оптимального метода визуализации играет критическую роль в эффективности анализа, поскольку различные алгоритмы обладают специфическими преимуществами и недостатками, определяющими их применимость к конкретным типам данных и задачам [2, 3]. В данной статье проводится комплексный обзор основных подходов к визуализации графов, анализируются их сильные и слабые стороны с учетом требований динамического моделирования, и обосновывается выбор наиболее подходящего метода, дополненного модификациями, для достижения максимальной эффективности визуализации динамических графов. Акцент делается на необходимости обеспечения интерактивности и возможности сворачивания графа для упрощения представления больших и сложных структур данных.

## *Анализ методов визуализации графов*

Существующие алгоритмы визуализации графов можно классифицировать по нескольким основным группам, каждая из которых основана на distinct principles [4]:

– Force-directed layout (Fruchterman-Reingold [5], Kamada-Kawai [6]): эти алгоритмы моделируют граф как физическую систему, где вершины представляются массами, а рёбра – пружинами. Взаимодействие сил притяжения и отталкивания между вершинами приводит к формированию компоновки, стремящейся к состоянию минимальной энергии. Достоинства: данный подход часто приводит к эстетически приятным и интуитивно понятным визуализациям, эффективно выявляя кластеры и группы связанных вершин. Недостатки: алгоритмы, основанные на физических силах, характеризуются высокой вычислительной сложностью, особенно для больших графов, и чувствительны к начальным условиям, что может приводить к нестабильности и вариативности результатов при каждом запуске. Кроме того, они не предоставляют встроенных механизмов для управляемого сворачивания графов, что ограничивает их применимость для визуализации больших и сложных сетей (рисунок 1);

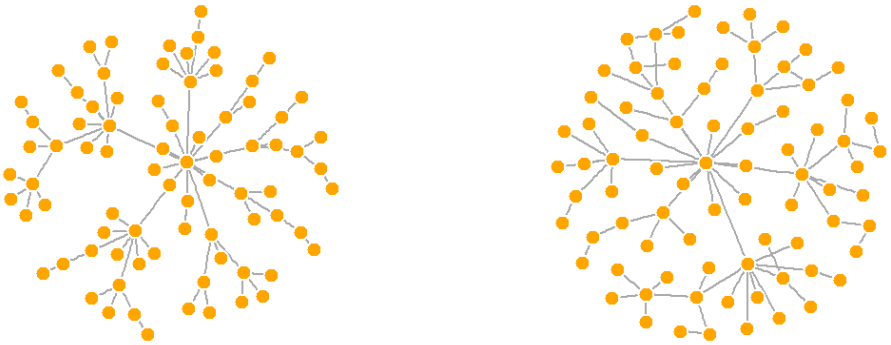
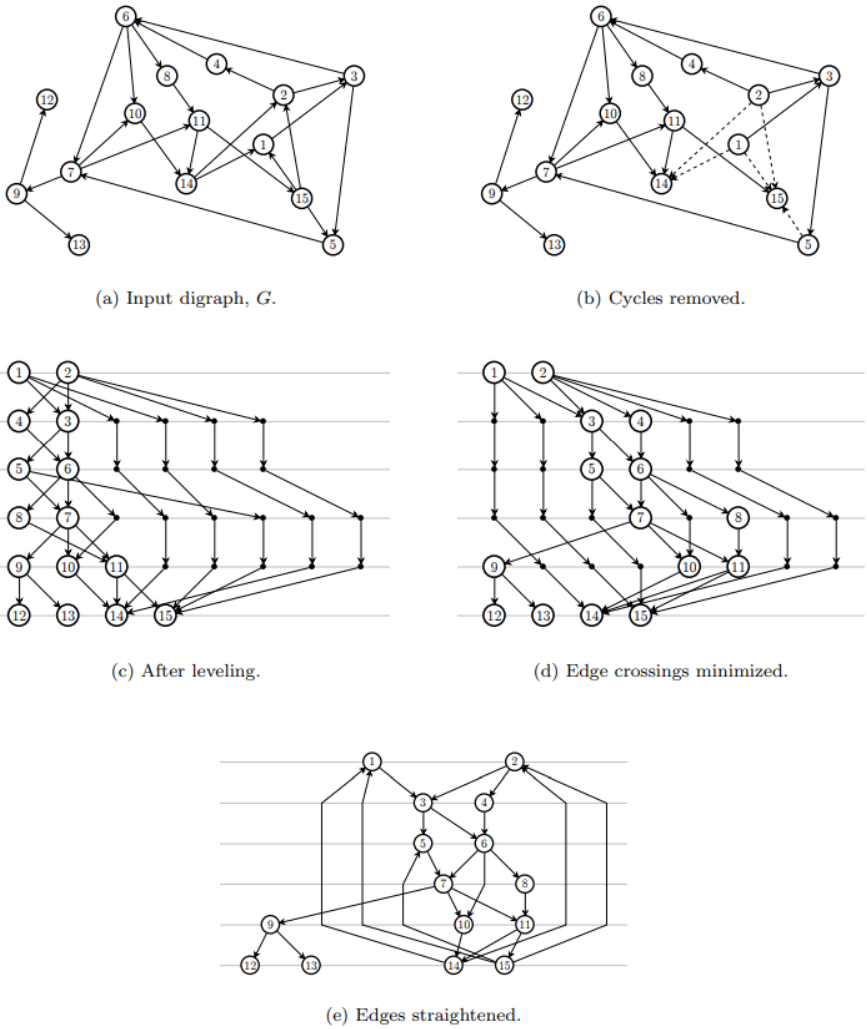


Рисунок 1 – Примеры работы алгоритма, основанного на физических силах - Force-directed layout

– Hierarchical (layered) layout (Sugiyama [7], Coffman-Graham [8]): данная группа алгоритмов предназначена для визуализации ориентированных ациклических графов, располагая вершины на дискретных уровнях (слоях) и минимизируя пересечения рёбер. Достоинства: иерархические алгоритмы обеспечивают структурированное представление данных, эффективно обрабатывая графы с выраженной иерархией. Недостатки: основным ограничением является требование ацикличности графа, что исключает их применение для многих реальных данных. Кроме того, для обработки графов с длинными рёбрами алгоритмы могут вводить фиктивные вершины, увеличивая сложность визуализации (рисунок 2);



(a) Input digraph, $G$.

(b) Cycles removed.

(c) After leveling.

(d) Edge crossings minimized.

(e) Edges straightened.

Рисунок 2 – Примеры работы алгоритма Hierarchical (layered) layout

– Tree layout (Reingold-Tilford [9], Garg-Rusu [10]): эти алгоритмы специализируются на визуализации древовидных структур, обеспечивая компактное и иерархическое представление данных. Достоинства: древовидные алгоритмы эффективно используют пространство и обеспечивают хорошую читаемость для древовидных структур данных. Недостатки: их применимость ограничена древовидными структурами, и они не подходят для визуализации общих графов с циклами или более сложными взаимосвязями (рисунок 3).

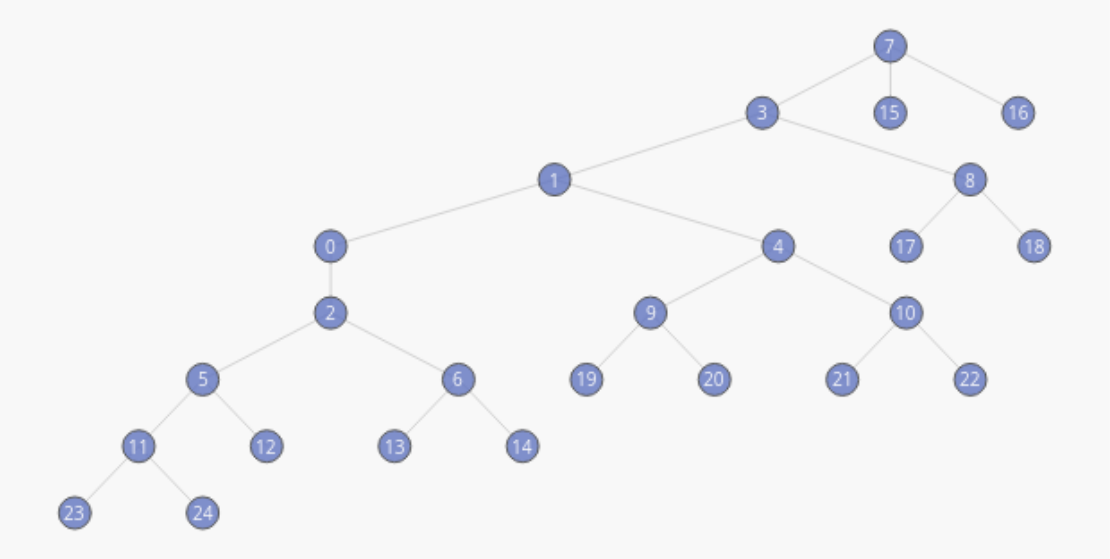



Рисунок 3 – Примеры работы алгоритма Tree layout

Перед выбором базового метода для решения задачи анализа, обработки и визуализации информации, проведем сравнительный анализ рассмотренных алгоритмов (таблица 1)

Таблица 1 – Сравнение алгоритмов визуализации

| Метод | Сложность | Достоинства | Недостатки |
|---|---|---|---|
| Force-directed (Fruchterman-Reingold) | ~O($n^3$), где n – кол-во узлов | - эстетичная визуализация;<br>- хорошая работа с кластеризованными графами;<br>- гибкость;<br>- равномерное распределение графа. | - вычислительно затратный;<br>- непредсказуемость результатов;<br>- непригоден для иерархических структур;<br>- сложность реализации. |
| Force-directed (Kamada-Kawai) | ~O($n^3$), где n – кол-во узлов | | |
| Hierarchical (Sugiyama) | ~O($n^2$), где n – кол-во узлов | - хорошо подходит для ориентированных ациклических графов;<br>- минимизирует пересечения ребер;<br>- четкая иерархия;<br>- равномерное распределение вершин по слоям. | - не подходит для циклических графов;<br>- может потребовать много фиктивных вершин;<br>- сложность реализации;<br>- ограниченно применим для графов с большим количеством вершин. |
| Hierarchical (Coffman-Graham) | ~O($n^2$), где n – кол-во узлов | | |
| Tree (Reingold-Tilford) | ~O(n), где n – кол-во узлов | - компактное и симметричное представление;<br>- эффективен для древовидных структур;<br>- легкая реализация; | - ограничен древовидными; структурами;<br>- проблемы с масштабируемостью;<br>- проблемы с циклическими графами. |
| Tree (Garg-Rusu) | ~O(n), где n – кол-во узлов | | |

### *Выбор базового метода и его модификация*

Анализируя требования к визуализации динамических графов с возможностью сворачивания графа и учитывая недостатки существующих подходов, в качестве базового выбран алгоритм Hierarchical (layered) Layout. Этот метод, несмотря на ограничение ацикличностью, обладает необходимой гибкостью для реализации требуемой функциональности. Путем внесения ряда модификаций возможно адаптировать его для эффективной визуализации динамических графов, преодолевая ограничения традиционных реализаций.

Предлагаемые модификации:

1. Диагональное расположение слоёв – вместо традиционного горизонтального или вертикального расположения вершин на слоях, предлагается использовать диагональное расположение. Это позволит более компактно и наглядно представить динамический граф.
2. Адаптивное сворачивания графа "влево" – для упрощения визуализации больших графов будет реализован механизм динамического сворачивания подграфов, что позволит сосредоточиться на ключевых элементах структуры.
3. Упрощение рёбер – для повышения читаемости и снижения визуальной перегруженности будут использоваться кривые линии для рёбер, обходя и минимизируя пересечения.
4. Визуализация семантики отношений – алгоритм будет учитывать семантику отношений между вершинами, визуально выделяя важные связи (например, различной толщиной или цветом линий).
5. Интерактивность – будет предусмотрена возможность интерактивного взаимодействия с визуализацией.

### *Ожидаемые эффекты и метрики оценки*

Предполагается, что введение предложенных модификаций в алгоритм Hierarchical (layered) Layout приведет к следующим положительным эффектам:

– Повышение читаемости диаграмм – диагональное расположение слоёв и адаптивное сворачивание графа существенно улучшат читаемость графов, особенно для сложных структур;

– Повышение эффективности анализа – динамическая модификация структуры графа и интерактивное управление позволят исследователям быстро сосредоточиться на интересующих их областях и эффективно анализировать данные;

– Улучшение восприятия семантики отношений – визуальное выделение важных связей и использование цветов облегчат интерпретацию данных и понимание взаимосвязей между объектами;

– Улучшенная интерактивность – интерактивное взаимодействие с визуализацией обеспечит большую гибкость и контроль над процессом анализа, позволяя пользователям адаптировать представление данных под свои потребности.

Для количественной оценки эффективности предлагаемого метода будут использованы следующие метрики:

– Количество пересечений рёбер – меньшее количество пересечений свидетельствует о более высокой читаемости графа.

– Средняя длина рёбер – более короткие рёбра способствуют более компактному представлению.

– Равномерность распределения вершин – равномерное распределение вершин по площади визуализации улучшает восприятие структуры.

– Субъективная оценка пользователей – будет проведен опрос пользователей для оценки удобства использования и эффективности предлагаемого метода.

### *Выводы*

В статье проведен анализ методов визуализации графовых структур в системах компьютерного моделирования, сфокусированный на эффективном представлении динамических графов. Рассмотрены группы алгоритмов, включая force-directed, иерархические и древовидные, выявлены их преимущества и недостатки в контексте динамического моделирования. Анализ показал ограничения существующих подходов, особенно в отношении интерактивности и представления больших графов.

В качестве базового метода выбран алгоритм Hierarchical (layered) Layout, гибкость которого позволяет адаптировать его к динамическим изменениям. Предложены модификации для повышения эффективности визуализации: диагональное расположение слоев для компактного отображения ветвлений и слияний; адаптивное сворачивание графа "влево" для упрощения восприятия; упрощение ребер с помощью кривых линий для минимизации пересечений; визуализация семантики отношений через вариации визуальных атрибутов ребер; и обеспечение интерактивности для динамического управления.

Ожидается, что модификации приведут к улучшению читаемости, эффективности анализа и восприятия семантики отношений в динамических графах. Для оценки будут использованы метрики: количество

пересечений ребер, средняя длина ребер, равномерность распределения вершин и субъективная оценка пользователей. Дальнейшие исследования направлены на реализацию и экспериментальную проверку предложенного метода с использованием выбранных метрик. Результаты позволят оценить эффективность модификаций и определить направления развития.

### *Литература*


1. Дударова Х. Х., Фаргиева З. С. Графовые модели и алгоритмы // Информационные технологии. – № 10. – С. 123-130.
2. Herman, I.; Melançon, G.; Marshall, M.S. Graph Visualization and Navigation in Information Visualization: A Survey. IEEE Trans. Vis. Comput. Graph. 2000, 6, 24–43.
3. Beck, F.; Burch, M.; Diehl, S.; Weiskopf, D. A taxonomy and survey of dynamic graph visualization. Comput. Graph. Forum 2017, 36, 133–159.
4. Handbook of Graph Drawing and Visualization / Roberto Tamassia, Editor. – 1st edition. – New York: Chapman and Hall/CRC, 2013. – 862 c. [39]
5. Fruchterman, T.M.J.; Reingold, E.M. Graph drawing by force-directed placement. Softw. Pract. Exper. 1991, 21, 1129–1164.
6. Kamada, T.; Kawai, S. An algorithm for drawing general undirected graphs. Inf. Process. Lett. 1989, 31, 7–15.
7. Sugiyama, K.; Tagawa, S.; Toda, M. Methods for visual understanding of hierarchical system structures. IEEE Trans. Syst. Man Cybern. 1981, 11, 109–125.
8. Coffman, E.G., Jr.; Graham, R.L. Optimal scheduling for two-processor systems. Acta Informatica 1972, 1, 200–213.
9. Reingold, E.M.; Tilford, J.S. Tidier Drawings of Trees. IEEE Trans. Softw. Eng. 1981, SE-7, 223–228.
10. Garg, A.; Rusu, A. Area-Efficient Drawings of Binary Trees. J. Algorithms 2003, 49, 275-309.



***Зори С. А., Бездетный Н. А. Визуализация информации в системах компьютерного моделирования: анализ и выбор алгоритмов для динамических графов.*** *В статье представлен углубленный анализ методов визуализации графовых структур, применяемых в системах компьютерного моделирования. Детально рассмотрены алгоритмы, основанные на физических силах, иерархические и древовидные алгоритмы, с акцентом на их применимость к динамическим графам. Выявлены их преимущества и недостатки, а также обоснован выбор и предложены модификации иерархического алгоритма для эффективной визуализации динамических графов с возможностью сворачивания графа, что способствует повышению интерактивности и улучшению понимания сложных структур данных.*



***Ключевые слова:*** *визуализация графов, компьютерное моделирование, force-directed layout, hierarchical (layered) layout, tree layout, динамические графы, сворачивание графа, интерактивность.*



***Zori S. A., Bezdetniy N. A. Information visualisation in computer simulation systems: analysis and selection of algorithms for dynamic graphs.*** *The article presents an in-depth analysis of methods of visualisation of graph structures used in computer simulation systems. Algorithms based on physical forces, hierarchical and tree algorithms are considered in detail, with emphasis on their applicability to dynamic graphs. Their advantages and disadvantages are identified, and the choice of the hierarchical algorithm is justified and modifications of the hierarchical algorithm for efficient visualisation of dynamic graphs with the possibility of graph collapsing are proposed, which helps to increase interactivity and improve understanding of complex data structures.*



***Keywords:*** *graph visualisation, computer modelling, force-directed layout, hierarchical (layered) layout, tree layout, dynamic graphs, graph collapsing, interactivity.*

УДК 004.946

# Обзор физических движков для разработки компьютерных игр: тенденции, характеристики и возможности применения.

**А.В. Григорьев[*1] , А.В. Синяев[*2]**

[*1] к.т.н, доцент, Донецкий национальный технический университет,
grigorievalvl@mail.ru, OrcID: 0000-0003-1073-6333, SPIN-код: 8830-2813

[*2] магистрант, Донецкий национальный технический университет,
manocauua255@gmail.com

***Синяев А.В. Обзор физических движков для разработки компьютерных игр: тенденции, характеристики и возможности применения.*** *В статье рассматриваются развитие компьютерных игр и роль физических движков в создании реалистичных виртуальных миров. Анализируются современные тенденции, классификация и алгоритмы, используемые для симуляции физических явлений. Обсуждаются популярные движки: PhysX, Havok, Bullet и Unity Physics с акцентом на их преимущества и недостатки. Выявляются ключевые аспекты выбора движка в зависимости от требований проекта и целевой аудитории. Подчеркивается актуальность разработки методики применения физических движков и перспективы развития в контексте новых технологий, таких как облачные вычисления и искусственный интеллект, открывающих новые возможности для интерактивных игровых опытов.*

***Ключевые слова:*** *физические движки, компьютерные игры, реалистичная симуляция, алгоритмы.*

## *Введение*

Развитие компьютерных игр как важной и динамичной отрасли индустрии развлечений привело к значительным достижениям в области графики, физики и взаимодействия компонентов виртуальных миров. Важным аспектом успешной разработки игр является использование физических движков, которые обеспечивают реалистичное моделирование физических явлений, таких как столкновения, гравитация и динамика объектов.

Физические движки находят применение в различных областях, включая создание видеоигр, симуляторов, образовательных приложений и виртуальной реальности, что делает их незаменимыми инструментами для разработчиков.

Актуальность темы исследования, посвященного физическим движкам, определяется не только растущими требованиями к качеству и реалистичности игровых процессов, но и необходимостью разработки более эффективных и универсальных решений, способных адаптироваться к новым технологиям и платформам.

Основная проблема, возникающая при практическом применении физических движков, заключается в необходимости выбора наиболее подходящего физического движка в зависимости от специфики проекта, требований к производительности и целей разработки. Необходимо учитывать, что каждый движок имеет свои уникальные особенности, которые могут как способствовать, так и затруднять процесс создания качественного игрового опыта, что требует глубокого анализа существующих решений и их сравнительной оценки.

На сегодняшний день отсутствует полноценное исследование применения движков для создания игр, что создает пробел в понимании их возможностей и ограничений.

Актуальной является задача разработки методики применения физических движков для создания компьютерных игр различной направленности, целевой аудитории и сюжета.

Цель данного исследования заключается в систематическом обзоре существующих физических движков для компьютерных игр с целью выявления их основных характеристик, преимуществ и недостатков. Для достижения поставленной цели будут решены следующие задачи:

– проанализировать современные тенденции в разработке физических движков;

– классифицировать наиболее распространенные движки по функционалу и применяемым алгоритмам;

– рассмотреть примеры их использования в различных жанрах и стилистиках игр.

### *Основные принципы функционирования физических движков*

Физический движок представляет собой программное обеспечение, которое имитирует физические законы реального мира в виртуальной среде. Обычно такие движки функционируют как подпрограммы. Их можно условно классифицировать на два типа: игровые и научные.

Игровые движки предназначены в первую очередь для компьютерных игр и являются частью игрового движка. Для обеспечения качественного погружения в игру физический движок должен выполнять все вычисления в реальном времени, с той же скоростью, с какой происходят события в реальности. При этом точность моделирования не является критически важной; гораздо важнее визуальная правдоподобность.

Научные движки, наоборот, ориентированы на точность вычислений, а не на скорость. Современные физические движки моделируют ограниченное количество физических явлений, но их спектр постоянно расширяется. К основным физическим явлениям и состояниям относятся:

– Динамика: абсолютно твёрдого тела;

– Динамика деформируемого тела;

– Динамика жидкостей;

– Динамика газов;

– Поведение тканей;

– Поведение верёвок.

Рассмотрим работу физического движка на примере динамики твёрдого тела (многие методы применимы и к другим типам). Физический движок должен уметь выполнять такие запросы, как «добавить тело», «получить позицию/ориентацию тела», «симулировать шаг времени» и другие. Основное внимание уделим функции «симуляции шага времени», которая выполняется каждый кадр [1].

Сначала выполняется так называемая Широкая фаза. На каждом кадре с помощью алгоритмов определения столкновений (Collision Detection) выявляются все потенциально взаимодействующие пары объектов. При этом вместо геометрии объектов используется упрощённая модель — ограничивающий параллелепипед, выровненный по координатным осям (Axis-Aligned Bounding Boxes - AABB); реже применяется аналогичная сфера. Суть Широкой фазы заключается в оптимизации работы приложения, так как из всех объектов лишь немногие могут пересекаться. Это позволяет сэкономить время и ресурсы в Узкой фазе, которая определяет более точные параметры взаимодействия, такие как глубина проникновения и нормали.

Среди основных алгоритмов Широкой фазы можно выделить:

– Brute-force (Exhaustive, All-Pairs test) — простая проверка всех возможных пересечений AABB объектов. Этот метод имеет квадратичную вычислительную сложность;

– Spatial Hashing — деление пространства для более эффективного поиска. Этот метод требует значительных объёмов памяти и сложен в оптимизации. Применяется в симуляциях жидкостей и мягких тел;

– Sweep-and-prune — метод, при котором AABB сортируются по их координатам. Для каждой оси создаётся массив интервалов, содержащих AABB объектов, которые сортируются для нахождения пересечений. Чаще всего применяется для твёрдых объектов;

– Существуют также алгоритмы, основанные на эффекте временной когерентности (Temporal coherence), где предполагается, что для объектов на большом расстоянии небольшие смещения на текущей итерации можно игнорировать.

После Широкой фазы следует Узкая фаза, где используются алгоритмы для поиска коллизий выпуклых объектов, Среди алгоритмов, подходящих для этой задачи, можно выделить:

– Алгоритм основанный на теореме разделяющей оси. Эта теорема утверждает, что два выпуклых объекта пересекаются тогда и только тогда, когда существует плоскость (или прямая в двумерном пространстве), разделяющая их [2];

– V-Clip / Lin Canny — алгоритмы, которые лучше всего подходят для физики, исключающей взаимопроникновение тел;

– Gilbert-Johnson-Keerthi / Expanded Polytope Algorithm — мощные алгоритмы, основанные на Сумме Минковского, которые не ограничены многогранниками.

Для их эффективной работы также требуется упрощение геометрии, например, построение выпуклой оболочки вокруг тела или разбиение тела на выпуклые составляющие. Алгоритмы, выполняющие эту задачу, называются Approximate Convex Decomposition.

После Узкой фазы наступает этап разрешения найденных контактов, за который отвечает Solver — одна из ключевых систем физического движка. Она решает две основные задачи:

– Определение конечных величин и направлений импульсов, столкнувшихся тел;

– Корректировка позиций объектов, чтобы устранить случаи их проникновения друг в друга.

Завершающим этапом является интегрирование позиций, что включает в себя смещение всех тел в сцене. Этот процесс относительно прост в реализации и использует несколько эффективных алгоритмов, таких как интеграторы по Эйлеру, Ньютону и Рунге-Кутта. Весь описанный процесс сопровождается различными функциями, такими как коллбэки (callbacks) для событий и фризинг (freezing/sleeping) тел, которые исключаются из обработки алгоритмов, если они долго находятся в покое [3].

### *Обзор существующих физических движков*

Физические движки имеют решающее значение для обеспечения реалистичного игрового процесса и взаимодействия объектов в видеоиграх.

Они обеспечивают разработчиков инструментами для симуляции физических процессов, что способствует созданию увлекательного и насыщенного игрового опыта. Фактически, чаще всего эти инструменты имеют форму API. Они позволяют легко интегрировать физику в игровые механики, управлять столкновениями, гравитацией и динамикой объектов, а также обеспечивать взаимодействие между различными элементами виртуального мира.

В этой статье мы проанализируем несколько известных физических движков, таких как PhysX, Havok, Bullet и Unity Physics, оценим их особенности, выявим достоинства и недостатки и проведем сравнительный анализ [4].

### *Возможности PhysX*

PhysX — это физический движок, созданный компанией NVIDIA, предназначенный для симуляции физики в реальном времени в видеоиграх и различных приложениях. Он предоставляет разработчикам возможность реализовывать реалистичные физические взаимодействия между объектами, включая динамику твердых тел, жидкости, ткани и другие эффекты [5].

Характеристики:

– Поддержка различных платформ: PhysX работает на различных платформах, включая Windows, Linux, PlayStation, Xbox и мобильные устройства;

– Реалистичная симуляция: Движок поддерживает сложные физические взаимодействия, такие как столкновения, гравитацию, трение и другие физические эффекты;

– GPU-ускорение: PhysX может использовать графические процессоры NVIDIA для ускорения вычислений, что позволяет достигать высокой производительности и реалистичности;

– Интеграция с игровыми движками: PhysX без труда соединяется с известными игровыми движками, такими как Unreal Engine и Unity;

– Поддержка различных типов физики: Движок обеспечивает поддержку динамики твердых тел, мягких тел, жидкостей и частиц, что открывает возможности для создания множества разнообразных физических эффектов.

Плюсы данного движка:

– Высокая производительность: Использование GPU для вычислений позволяет достигать высокой производительности, особенно в сложных сценах;

– Реалистичность: PhysX обеспечивает высокую степень реализма в симуляции физических взаимодействий, что улучшает общее впечатление от игры;

– Широкая поддержка: Движок поддерживается на множестве платформ и легко интегрируется с популярными игровыми движками;

– Разнообразие эффектов: Возможность симуляции различных типов физики (твердые тела, жидкости, ткани) позволяет разработчикам создавать уникальные игровые механики.

Минусы данного движка:

– Зависимость от оборудования: Для достижения максимальной производительности требуется современное оборудование, особенно графические карты NVIDIA;

– Сложность интеграции: Хотя PhysX интегрируется с популярными движками, его настройка и оптимизация могут быть сложными для новичков;

– Лицензирование: Использование PhysX может потребовать дополнительных затрат на лицензирование для коммерческих проектов.

PhysX использовался в таких играх как:

“Metro: Last Light”: PhysX использовался для создания реалистичных эффектов воды и разрушений. На сегодняшний момент приобрести оригинальную игру можно только на физических носителях.

“Gears of War 3” [6]: В этой игре PhysX применялся для симуляции разрушений и взаимодействия объектов в бою.

“The Witcher 3: Wild Hunt” [7]: Движок применялся для создания правдоподобных эффектов воды и взаимодействия с окружающей средой.

### *Возможности Havok*

Havok — это мощный физический движок, разработанный компанией Havok, который был выпущен в 2000 году. Он поддерживает реалистичное моделирование физических взаимодействий в реальном времени и используется во множестве игр и приложений [8].

Основные характеристики Havok включают:

– Динамическая физика: Поддерживает расчет столкновений и взаимодействий объектов в реальном времени;

– Моделирование мягких тел: Позволяет симулировать мягкие и твердые тела, а также деформируемые объекты;

– Анимация с физикой: Упрощает создание анимированных объектов с учетом физического взаимодействия;

– Сетевые возможности: Оптимизирован для многопользовательских игр, чтобы синхронизировать физику среди разных клиентов;

– Интеграция с игровыми движками: Простая интеграция с популярными игровыми движками, такими как Unity и Unreal Engine.

Плюсы данного движка:

– Высокое качество симуляции: Havok предлагает реалистичные и точные физические симуляции, что усиливает ощущение погружения в игру;

– Широкая совместимость: Отлично работает на различных платформах, включая консоли, ПК и мобильные устройства;

– Поддержка крупных игровых проектов: Havok используется в крупных AAA-играх и хорошо зарекомендовал себя за годы работы;

– Обширная документация и поддержка: Предоставляет разработчикам ресурсы для обучения и поддержки.

Минусы данного движка:

– Лицензионные отчисления: Стоимость лицензии может быть высокой для независимых разработчиков и небольших студий;

– Сложный процесс интеграции: Не всегда легко интегрируется в существующие проекты, требует времени и ресурсов;

– Производительность на низких конфигурациях: Может возникнуть проблема с производительностью на устаревшем или слабом оборудовании.

Havok использовался в множестве известных игр и франшиз, включая:

“The Elder Scrolls V: Skyrim” [9]: Обеспечивает реалистичное взаимодействие предметов и управления физикой в открытом мире.

“Half-Life 2” [10]: Использован для создания сложной физики, включая взаимодействия объектов и физические головоломки.

“Call of Duty” (серия) [11]: Для управления реалистичными взрывами, пулями и разрушениями окружающей среды.

### *Возможности Bullet*

Bullet - это открытый физический движок, использующий C++ и поддерживающий различные языки (например, Python через PyBullet) [12].

Характеристики:

– Совместимость: Bullet работает на кросс-платформенных системах, включая Windows, Linux и macOS. Он также поддерживает работу с мобильными устройствами и консолями;

– Поддержка динамики и кинематики: Движок предназначен для обработки как ригидных, так и мягких тел, а также осуществляет симуляцию мягких тел, таких как ткани;

– Графические API: Bullet интегрируется с различными графическими движками и библиотеками, такими как OpenGL, Direct3D и более;

– Многопоточность: Bullet поддерживает многопоточные вычисления, что позволяет использовать множество ядер процессора для повышения производительности;

– Документация и сообщество: Bullet имеет хорошую документацию и активное сообщество разработчиков, что облегчает изучение и интеграцию.

Плюсы:

– Открытость: Bullet является open-source проектом, что позволяет разработчикам свободно использовать, изменять и распространять движок;

– Качество симуляции: Движок обеспечивает реалистичную физическую симуляцию объектов и взаимодействий между ними, что делает его подходящим для различных приложений, включая игры и симуляторы;

– Кросс-платформенность: Поддержка различных операционных систем и платформ облегчает разработку;

– Легкость интеграции: Bullet может быть легко интегрирован в существующие проекты и сочетаться с другими библиотеками и движками.

Минусы:

– Документация: Хотя документация существует, некоторые пользователи находят её не всегда полной, что может затруднить решение менее очевидных проблем;

– Производительность: В некоторых случаях Bullet может оказаться менее производительным по сравнению с некоторыми специализированными физическими движками, особенно в сложных сценариях обработки динамики;

– Поддержка дополнительных функций: Некоторые высокоуровневые функции могут отсутствовать или требовать значительных усилий для реализации.

Bullet использовался в различных проектах, включая:

"Grand Theft Auto V" [13]: Используется для имитации физики транспортных средств и взаимодействий между объектами в открытом мире.

"Doom 3" [14]: Применяется для обработки столкновений и физики объектов.

"Red Dead Redemption 2": Здесь Bullet обрабатывает физику среды и персонажей, взаимодействующих со сложным миром игры.

### *Возможности Unity Physics*

Unity Physics — это встроенный физический движок, разработанный для использования в игровом движке Unity. Он предлагает разработчикам эффективные инструменты для реализации правдоподобных физических взаимодействий в 2D и 3D-играх. Основанный на системе ECS (Entity Component System), Unity Physics оптимизирован для работы с большими объемами объектов и обеспечивает высокую производительность, что делает его идеальным выбором для современных игровых проектов [15].

Характеристики:

– Интеграция с Unity: Unity Physics осуществляет полную интеграцию с игровым движком Unity, обеспечивая простоту использования и высокую производительность;

– Поддержка DOTS: Движок основан на системах Data-Oriented Technology Stack, что обеспечивает эффективное управление памятью и производительность в многопользовательских играх и сценариях с высокой нагрузкой;

– Сила и столкновения: Поддержка взаимодействия объектов, включая формирование сил и обнаружение столкновений;

– Технология ECS: Использует подход, основанный на компонентах и системах (Entity-Component-System), что упрощает управление игровыми объектами и их взаимодействиями;

– Расширяемость: Возможность создания собственных физических материалов и пользовательских эффектов.

Плюсы:

– Глубокая интеграция: Unity Physics предоставляет разработчикам полный доступ к инструментам физики, что позволяет легко создавать фотореалистичные физические взаимодействия;

– Оптимизация для многопользовательских игр: Благодаря DOTS движок может обрабатывать большое число объектов и взаимодействий без заметной потери производительности;

– Простота использования: Unity Physics предлагается как часть экосистемы Unity, настраивать и использовать его могут как опытные разработчики, так и новички;

– Расширяемость: Способен интегрировать пользовательские системы и физические материалы, что позволяет адаптировать его к требованиям конкретного проекта.

Минусы:

– Ограниченная функциональность по сравнению с более крупными движками: В некоторых случаях Unity Physics может уступать более мощным физическим движкам, таким как Havok или PhysX;

– Необходимость изучения ECS: Для полноценного использования преимуществ DOTS и ECS требуется больше времени на изучение и адаптацию, что может быть проблемой для начинающих разработчиков;

– Может потребовать ручной тюнинг для сложных симуляций: Для достижения наилучших результатов с физикой в более сложных сценариях разработчики могут столкнуться с необходимостью ручной настройки.

Unity Physics активно применяется в разработке различных игр, среди которых:

“Cuphead” [16]: известный платформер с ретро-графикой, использующий физический движок для создания динамичных столкновений и анимаций.

“Hollow Knight” [17]: игра в жанре метроидвания, где физические взаимодействия имеют большое значение для игрового процесса.

“Super Mario Run” [18]: мобильная игра с использованием Unity Physics для управления движением персонажей и объектов.

“Ori and the Blind Forest” [19]: эффектная платформенная игра, где физика используется для создания захватывающих игровых механик.

Таблица 1 – Сравнительная таблица движков

| Движок | PhysX | Havok | Bullet | Unity Physics |
|---|---|---|---|---|
| Разработчик | NVIDIA | Havok | Open Source | Unity Technologies |
| Платформы | Windows, Linux, PlayStation, Xbox, мобильные устройства | Консоли, ПК, мобильные устройства | Windows, Linux, macOS, мобильные устройства, консоли | Unity (все платформы, поддерживаемые Unity) |
| Типы физики | Динамика твердых тел, жидкости, ткани, частицы | Динамика твердых и мягких тел, анимация с физикой | Динамика твердых и мягких тел, ткани | Динамика твердых тел, взаимодействия и силы |
| GPU-ускорение | Да | Нет | Нет | Нет |
| Интеграция с движками | Unreal Engine, Unity | Unreal Engine, Unity | Различные графические движки (OpenGL, Direct3D) | Полная интеграция с Unity |
| Поддержка многопоточности | Да | Да | Да | Да (через DOTS) |
| Документация и поддержка | Хорошая | Обширная | Хорошая, но может быть неполной | Хорошая |
| Лицензирование | Может потребовать затрат | Может потребовать затрат | Open Source (бесплатно) | Входит в Unity (бесплатно для использования в рамках Unity) |
| Производительность | Высокая | Высокая | Средняя (в зависимости от сценариев) | Высокая (в зависимости от конфигурации) |
| Сложность интеграции | Средняя | Высокая | Низкая (для опытных разработчиков) | Низкая (для Unity) |
| Плюсы | Высокая производительность, реалистичность, разнообразие эффектов | Высокое качество симуляции, широкая совместимость, поддержка крупных проектов | Открытость, качество симуляции, кросс-платформенность | Глубокая интеграция, оптимизация для многопользовательских игр, простота использования |
| Минусы | Зависимость от оборудования, сложность интеграции, лицензионные расходы | Лицензионные отчисления, сложный процесс интеграции, производительность на низких конфигурациях | Документация может быть неполной, производительность в сложных сценариях может быть ниже | Ограниченная функциональность, необходимость изучения ECS, ручной тюнинг для сложных симуляций |

### *Заключение*

В данной работе были следующие задачи: выполнен анализ современных тенденции в разработке физических движков, построена классифицировать наиболее распространенные движки по функционалу и применяемым алгоритмам, рассмотрены примеры их использования в различных жанрах игр и стилистике игр.

В ходе обзора физических движков для компьютерных игр были выделены ключевые характеристики, преимущества и недостатки таких популярных решений, как PhysX, Havok, Bullet и Unity Physics. Каждый из движков обладает уникальными особенностями, которые определяют их применение в различных игровых проектах. PhysX, разработанный NVIDIA, предлагает высокую производительность и реалистичную симуляцию, но требует современного оборудования и может быть сложен в интеграции. Havok, в свою очередь, обеспечивает высокое качество симуляции и поддержку крупных проектов, однако его лицензирование может стать препятствием для независимых разработчиков. Bullet выделяется своей открытостью и кросс-платформенностью, но может уступать в производительности специализированным движкам. Unity Physics, интегрированный в экосистему Unity, предлагает простоту использования и оптимизацию для многопользовательских игр, но требует изучения новых подходов, таких как ECS.В качестве вывода можно заключить:

- к общим достоинствам можно отнести способность создавать реалистичные физические взаимодействия, что значительно повышает уровень погружения игроков в игровой процесс. Они позволяют разработчикам интегрировать сложные механики, такие как разрушения, взаимодействия объектов и симуляцию естественных явлений, что делает игры более увлекательными и динамичными. Кроме того, многие движки предлагают высокую производительность и оптимизацию для работы с большими объемами данных, что особенно важно в современных играх с открытыми мирами.

- к общим недостаткам можно отнести зависимость от аппаратного обеспечения, что может ограничивать возможности разработчиков, особенно в случае работы с устаревшими системами. Также, интеграция физических движков в существующие проекты может быть сложной и требовать значительных затрат времени и ресурсов. Лицензионные ограничения и стоимость некоторых движков могут стать препятствием для небольших студий и независимых разработчиков.

- специфика того или иного движка может является основанием для его применения в предметной области с теми или иными возможностями. PhysX может быть предпочтительным выбором для игр с интенсивной физикой и графикой, в то время как Bullet будет более подходящим для проектов с ограниченным бюджетом и требованиями к открытости. Havok может быть идеальным для крупных AAA-проектов, а Unity Physics — для разработчиков, работающих в экосистеме Unity и стремящихся к быстрой интеграции.

Перспективы развития и использования физических движков в компьютерных играх выглядят многообещающе. С ростом вычислительных мощностей и развитием технологий, таких как облачные вычисления и искусственный интеллект, физические движки будут продолжать эволюционировать, предлагая разработчикам новые возможности для создания еще более реалистичных и интерактивных игровых миров. В будущем можно ожидать появления новых алгоритмов и методов, которые улучшат производительность и точность симуляции, а также расширят спектр физических явлений, доступных для моделирования. Это позволит разработчикам создавать уникальные игровые механики и погружать игроков в захватывающие виртуальные миры, где физика будет играть важную роль в игровом процессе.

Перспективой данной работы является разработка технологии применения физического движка для построения игр различных жанров, стилистики, целевой аудитории и т.д.

### *Литература*

1. "The Role of Physics Engines in Game Development" by Jane Smith [Электронный ресурс] / Режим доступа: https://archive.org/details/gamephysics0000eber/page/240/mode/2up – Загл. с экрана (дата обращения: 20.09.2024).

2. Теорема разделяющей оси [Электронный ресурс] / Режим доступа: https://gamedev.ru/terms/SAT – Загл. с экрана (дата обращения: 20.10.2024).

3. Н.В. Кушнир, К.А. Будников, С.Э. Ирхин, В.А. Никитин, И.О. Майоров Базовое программное обеспечение разработки компьютерных видеоигр. игровые движки [Электронный ресурс] / Режим доступа: https://ntk.kubstu.ru/data/mc/0072/3689.pdf – Загл. с экрана (дата обращения: 22.10.2024).

4. Козлова П.В. Роль игровой физики в современной гейм-индустрии [Электронный ресурс] / Режим доступа: https://libeldoc.bsuir.by/bitstream/123456789/48014/1/Kozlova_Rol.pdf – Загл. с экрана (дата обращения: 20.09.2024).

5. Официальный сайт NVIDIA PhysX System Software [Электронный ресурс] / Режим доступа: https://www.nvidia.com/en-us/drivers/physx/physx-9-19-0218-driver/ – Загл. с экрана (дата обращения: 10.10.2024).

6. Официальный сайт Gears of War 3 [Электронный ресурс] / Режим доступа: https://www.gearsofwar.com/games/gears-of-war-3/ – Загл. с экрана (дата обращения: 02.11.2024).

7. Официальный сайт The Witcher 3: Wild Hunt [Электронный ресурс] / Режим доступа https://www.thewitcher.com/nl/ru/_– Загл. с экрана (дата обращения: 03.11.2024).
8. Официальный сайт Havok [Электронный ресурс] / Режим доступа: https://www.havok.com/_– Загл. с экрана (дата обращения: 15.10.2024).
9. The Elder Scrolls V: Skyrim [Электронный ресурс] / Режим доступа: https://store.steampowered.com/app/489830/The_Elder_Scrolls_V_Skyrim_Special_Edition/_– Загл. с экрана (дата обращения: 04.11.2024).
10. Half-Life 2 [Электронный ресурс] / Режим доступа: https://store.steampowered.com/app/220/HalfLife_2/_– Загл. с экрана (дата обращения: 05.11.2024).
11. Официальный сайт Call of Duty [Электронный ресурс] / Режим доступа: https://www.callofduty.com/nl/ – Загл. с экрана (дата обращения: 06.11.2024).
12. Официальный сайт Bullet [Электронный ресурс] / Режим доступа: https://pybullet.org/wordpress/_– Загл. с экрана (дата обращения: 20.10.2024).
13. Grand Theft Auto V [Электронный ресурс] / Режим доступа: https://www.rockstargames.com/gta-v_– Загл. с экрана (дата обращения: 07.11.2024).
14. Doom 3 [Электронный ресурс] / Режим доступа: https://store.steampowered.com/app/208200/DOOM_3/_– Загл. с экрана (дата обращения: 08.11.2024).
15. Официальный сайт Unity Physics [Электронный ресурс] / Режим доступа: https://docs.unity3d.com/Packages/com.unity.physics@1.0/manual/index.html_– Загл. с экрана (дата обращения: 25.10.2024).
16. Официальный сайт Cuphead [Электронный ресурс] / Режим доступа: https://cupheadgame.com/_– Загл. с экрана (дата обращения: 10.11.2024).
17. Hollow Knight [Электронный ресурс] / Режим доступа: https://store.steampowered.com/app/367520/Hollow_Knight/_– Загл. с экрана (дата обращения: 11.11.2024).
18. Super Mario Run [Электронный ресурс] / Режим доступа: https://play.google.com/store/apps/details?id=com.nintendo.zara&hl=en-US&pli=1 – Загл. с экрана (дата обращения: 12.10.2024)
19. Ori and the Blind Forest [Электронный ресурс] / Режим доступа: https://store.steampowered.com/app/387290/Ori_and_the_Blind_Forest_Definitive_Edition/_– Загл. с экрана (дата обращения: 13.11.2024).

***Синяев А.В. Обзор физических движков для разработки компьютерных игр: тенденции, характеристики и возможности применения.*** *В статье рассматриваются развитие компьютерных игр и роль физических движков в создании реалистичных виртуальных миров. Анализируются современные тенденции, классификация и алгоритмы, используемые для симуляции физических явлений. Обсуждаются популярные движки: PhysX, Havok, Bullet и Unity Physics с акцентом на их преимущества и недостатки. Выявляются ключевые аспекты выбора движка в зависимости от требований проекта и целевой аудитории. Подчеркивается актуальность разработки методики применения физических движков и перспективы развития в контексте новых технологий, таких как облачные вычисления и искусственный интеллект, открывающих новые возможности для интерактивных игровых опытов.*

***Ключевые слова:*** *физические движки, компьютерные игры, реалистичная симуляция, алгоритмы.*

***Sinyayev A.V. An Overview of Physics Engines for Video Game Development: Trends, Characteristics, and Applications.*** *This article examines the evolution of computer games and the role of physics engines in the creation of realistic virtual worlds. It analyzes contemporary trends, classification, and algorithms employed for the simulation of physical phenomena. The discussion encompasses popular engines such as PhysX, Havok, Bullet, and Unity Physics, emphasizing their advantages and drawbacks. Key aspects of engine selection are identified, depending on project requirements and target audiences. The relevance of developing methodologies for the application of physics engines is highlighted, along with prospects for advancement in the context of emerging technologies such as cloud computing and artificial intelligence, which provide new opportunities for interactive gaming experiences.*

***Key words:*** *physics engines, video games, realistic simulation, algorithms.*

# СЕКЦИЯ 5. «МЕТОДЫ И СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ ПО И СИСТЕМ»

УДК 004.896

## Об особенностях современного производства труб и использовании АСУ ТП в трубном производстве

**К.А. Коврик[*1], О.А. Криводубский[*2]**

[*1] д.т.н, профессор каф. ПИ, Донецкий национальный технический университет,
oleg.krivodubski.dn@gmail.com, SPIN-код: 4156-9798

[*2] аспирант, Донецкий национальный технический университет,
kirill_kovrik@mail.ru

***Коврик К. А., Криводубский О.А. Об особенностях современного производства труб и использовании АСУ ТП в трубном производстве.*** *Статья обсуждает современные тенденции и достижения в разработке математического и программного обеспечения для автоматизированных систем. Рассматриваются ключевые аспекты, такие как алгоритмы, структуры данных, модели данных. Особое внимание уделяется применению методов машинного обучения и искусственного интеллекта для решения сложных задач.*

***Ключевые слова:*** *автоматизированные системы, математическое моделирование, алгоритмизация, искусственные интеллект, машинное обучение, SCADA, MES, ERP.*

### *Введение*

Трубная промышленность — ключевая отрасль, обеспечивающая потребности в трубах для различных сфер, таких как строительство, энергетика, нефтегазовая и химическая промышленности. С развитием технологий требования к качеству и эффективности производства возрастают, и автоматизация становится необходимостью. Внедрение автоматизированных систем управления (АСУ) на предприятиях трубной промышленности позволяет повысить качество продукции, оптимизировать процессы и снизить затраты. Современные АСУ используют алгоритмы машинного обучения, анализа временных рядов, стохастической оптимизации и базы знаний для обеспечения устойчивости, адаптивности и конкурентоспособности предприятий в условиях изменяющихся рыночных требований.

Современное производство труб является важной отраслью, обеспечивающей широкий спектр потребностей в таких секторах, как нефтегазовая промышленность, строительство и коммунальные услуги. В условиях растущего спроса на высококачественные трубные изделия, технологии и методы производства непрерывно совершенствуются. За последние годы внедрение автоматизированных систем управления технологическими процессами (АСУ ТП) стало ключевым фактором, влияющим на эффективность и устойчивость производства. АСУ ТП позволяют значительно повысить уровень контроля, предсказуемости и оптимизации производственных процессов, что особенно актуально в условиях высокой конкуренции и необходимости соблюдения строгих стандартов качества.

Одним из важнейших элементов современного производства труб является автоматизация контроля качества, включая применение алгоритмов машинного зрения для обнаружения дефектов и мониторинга параметров производства в реальном времени. Эти технологии дают возможность предприятиям минимизировать риск брака и оптимизировать затраты, обеспечивая стабильное качество продукции. Наряду с этим, применение ERP (Enterprise Resource Planning) и MES (Manufacturing Execution Systems) систем помогает интегрировать процессы управления ресурсами, финансами и логистикой, что способствует комплексному управлению производством на всех этапах.

В статье рассматриваются особенности современных технологий производства труб, описываются ключевые алгоритмы и системы, используемые в АСУ ТП, а также приводится анализ эффективности данных решений для оптимизации производственных процессов.

### *Особенности современного производства труб*

Производство труб включает несколько этапов, каждый из которых требует специализированного оборудования и технологий. Основные этапы включают подготовку сырья, формирование труб различными методами (экструзия, прокатка, сварка), термическую и механическую обработку, а также контроль качества. В зависимости от типа трубы (металлическая или пластиковая), на каждом этапе используются различные технологии: экструдеры для пластиковых труб, прокатные станы и сварочные аппараты для металлических труб. Применение современных технологий на каждом этапе позволяет добиться высокого качества продукции и повышения производительности.

Современное производство труб прошло значительные изменения за последние десятилетия, особенно с учетом технологических инноваций и оптимизации процессов. Ключевые особенности современного трубного производства включают внедрение автоматизированных и интеллектуальных систем управления, использование передовых материалов, а также применение методов контроля качества, таких как онлайн-мониторинг и анализ данных в реальном времени.

Современные трубные заводы активно внедряют автоматизированные системы управления, основанные на ERP (Enterprise Resource Planning) и MES (Manufacturing Execution Systems), которые интегрируют все этапы производства: от планирования и закупок до распределения готовой продукции. Эти системы позволяют минимизировать человеческий фактор, повысить эффективность производства и улучшить логистику [1]. Например, системы контроля с использованием машинного зрения позволяют в реальном времени обнаруживать дефекты труб, что значительно снижает количество брака [2].

Современное производство труб также характеризуется использованием новых материалов, таких как углеродные и композитные стали, которые обладают улучшенными эксплуатационными характеристиками. Эти материалы обеспечивают долговечность продукции и позволяют производить трубы для более сложных и высокотехнологичных применений, например, для нефтегазовой и химической промышленности. Кроме того, стоит отметить, что значительное внимание уделяется инновациям в процессе горячей и холодной прокатки, что позволяет получать трубы с высокой прочностью и долговечностью, одновременно снижая их массу [3].

Внедрение технологий анализа данных и прогнозирования в трубной промышленности стало важной составляющей. Современные заводы используют системы, основанные на анализе временных рядов и искусственном интеллекте для прогнозирования потребностей в материалах и оптимизации производственных процессов. Это позволяет значительно сократить издержки и повысить эффективность производства [4].

С увеличением внимания к вопросам охраны окружающей среды, современные трубные предприятия внедряют устойчивые производственные практики. Например, использование технологий для снижения выбросов CO2 и повторного использования материалов, таких как стальные ломовые отходы, стало неотъемлемой частью процессов, направленных на сокращение воздействия на природу [5].

Таким образом, современное производство труб продолжает развиваться в направлении повышения автоматизации, устойчивости и экологичности, что делает его более эффективным и соответствующим требованиям современной промышленности.

### *Автоматизированные системы управления в трубной промышленности*

Современные АСУ в трубной промышленности выполняют функции мониторинга и управления процессами в режиме реального времени. В структуру АСУ входят SCADA, MES и ERP-системы, каждая из которых выполняет свои задачи на разных уровнях управления. Например, SCADA обеспечивает оперативный контроль за технологическим процессом, MES позволяет оптимизировать производственные операции на уровне цеха, а ERP-система интегрирует информацию на уровне всего предприятия, управляя ресурсами и запасами. Такие системы помогают предприятиям реагировать на изменения в производственном процессе, эффективно управлять ресурсами и повышать качество продукции. Каждый тип описанных систем может быть представлен в рамках одного производства, пример на рисунке 1.

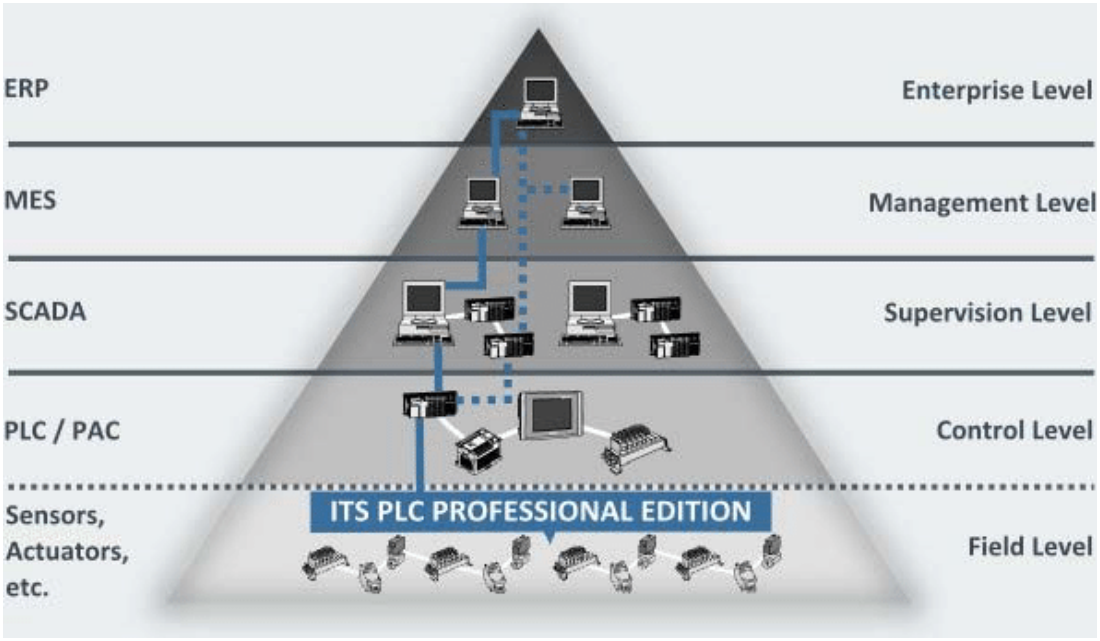



Рисунок 1 – Схема использования автоматизированных систем

Современные трубные предприятия используют интегрированные системы управления предприятием (ERP) и системы управления производственными процессами (MES), которые обеспечивают эффективную координацию всех этапов работы. ERP-системы помогают интегрировать управление финансами, производственными заказами, запасами и логистикой, обеспечивая полный контроль за бизнес-процессами предприятия. MES-системы, в свою очередь, фокусируются на оптимизации именно производственных процессов, включая контроль за выполнением заказов, мониторинг качества продукции, управление ресурсами и анализ производственных данных в реальном времени [6].

Важнейшей частью автоматизированных систем являются системы контроля качества. На современных трубных заводах внедрены автоматизированные системы, использующие датчики, системы машинного зрения и аналитические инструменты для мониторинга и анализа качества продукции. Эти системы позволяют в реальном времени выявлять дефекты, такие как трещины или отклонения от заданных параметров, что существенно повышает качество конечного продукта и снижает вероятность брака [7].

Еще одним важным направлением в автоматизации трубного производства является использование искусственного интеллекта (ИИ) и анализа данных для прогнозирования и оптимизации процессов. На основе данных, собранных в реальном времени, системы ИИ могут прогнозировать потребности в материалах, оптимизировать графики производства и даже предсказывать возможные сбои в работе оборудования. Это позволяет заранее устранять потенциальные проблемы, минимизируя простои и повышая эффективность производства [8].

Автоматизированные системы управления также играют важную роль в оптимизации использования ресурсов, включая энергопотребление и сырьевые материалы. Внедрение энергоэффективных решений и автоматизация контроля за расходом энергии помогают значительно снизить эксплуатационные расходы. Современные системы управления в трубной промышленности позволяют мониторить и регулировать потребление энергии, что является важным аспектом для снижения углеродного следа производства [9]. Типичная структура систем управления производством представлена на рисунке 2.


Рисунок 2 – Типичная структура систем управления производством

Автоматизированные системы управления также тесно связаны с системами планирования и логистики, что позволяет эффективно управлять потоками материалов и готовой продукции. Интеграция с системой отслеживания и планирования поставок помогает обеспечить бесперебойное снабжение производства, минимизируя риски дефицита или излишков сырья. Это способствует уменьшению затрат и повышению гибкости производства [10].

Таким образом, автоматизированные системы управления в трубной промышленности оказывают значительное влияние на повышение производительности, улучшение качества продукции и снижение операционных затрат, обеспечивая высокую степень интеграции процессов и более эффективное использование ресурсов. Внедрение таких систем продолжает быть важным шагом для модернизации и повышения конкурентоспособности предприятий трубной отрасли.

### *Алгоритмы, используемые в АСУ ТП трубной промышленности*

Для управления производственными процессами и анализа данных в трубной промышленности применяются различные алгоритмы. 1) Алгоритмы анализа временных рядов помогают прогнозировать состояние оборудования и управлять параметрами производства. 2) Алгоритмы машинного обучения, такие как нейронные сети и методы кластеризации, используются для анализа больших объёмов данных, поступающих с датчиков, что позволяет выявлять скрытые зависимости и оптимизировать производство. 3) Методы стохастической оптимизации, такие как методы Монте-Карло и стохастическое программирование, обеспечивают гибкость в управлении ресурсами, особенно в условиях неопределённости.

Автоматизированные системы управления технологическими процессами (АСУ ТП) в трубной промышленности используют различные алгоритмы для оптимизации производственных операций, обеспечения качества и минимизации затрат. Эти алгоритмы помогают в реальном времени контролировать и управлять процессами, такими как прокатка, сварка, резка и термическая обработка труб. Рассмотрим ключевые алгоритмы, используемые в таких системах.

В трубной промышленности особое внимание уделяется алгоритмам, обеспечивающим стабильное управление технологическими процессами, такими как прокатка и сварка труб. Эти алгоритмы основываются на методах оптимизации и регуляции, включая: PID-регуляторы (Пропорционально-Интегрально-Дифференциальные): Эти алгоритмы широко используются для управления параметрами прокатных станов, поддержания температуры и давления в процессе производства. PID-регуляторы обеспечивают точную настройку и минимизацию отклонений от заданных параметров [11].

Алгоритмы предсказания с использованием нейронных сетей. Для управления технологическими процессами, такими как сварка и термическая обработка труб, часто применяются нейронные сети, которые предсказывают поведение системы и позволяют адаптировать параметры в реальном времени. Эти алгоритмы помогают учитывать не только текущие, но и исторические данные для улучшения прогнозируемости процессов [12].

Алгоритмы диагностики и мониторинга. Для обеспечения качества продукции в трубной промышленности важным аспектом является мониторинг состояния оборудования и контроль качества на каждом этапе производства.

Алгоритмы диагностики включают. Методы машинного зрения и анализа изображений. Используются для обнаружения дефектов труб на стадии контроля качества. Алгоритмы обработки изображений с применением методов глубокого обучения могут автоматически обнаруживать трещины, деформации или другие дефекты на поверхности труб [13].

Алгоритмы анализа вибрации. Используются для мониторинга технического состояния оборудования, в том числе прокатных станов и сварочных машин. Алгоритмы на основе анализа спектра вибраций могут предсказывать возможные поломки, что позволяет снизить время простоя и провести своевременное обслуживание [14].

Алгоритмы оптимизации играют важную роль в повышении эффективности и сокращении затрат на каждом этапе производства труб. Среди них:

– методы линейного и нелинейного программирования, используются для оптимизации процесса распределения ресурсов, таких как энергия, материалы и рабочая сила, с целью минимизации затрат и максимизации производительности [15];

– генетические алгоритмы применяются для оптимизации комплексных производственных процессов, например, в случаях, когда необходимо оптимизировать параметры сварки или прокатки с учетом множества переменных и ограничений [16];

– в трубной промышленности часто используются стохастические методы для решения задач, связанных с неопределенностью в данных (например, колебания спроса или вариативность в характеристиках материалов) [17].

Прогнозирование является важным аспектом в управлении запасами, планировании производства и предсказании спроса. Алгоритмы прогнозирования на базе временных рядов и машинного обучения применяются для анализа и прогнозирования потребности в трубах на разных этапах производственного цикла.

Современные АСУ ТП в трубной промышленности используют алгоритмы для интеграции различных систем предприятия, таких как ERP, MES и SCADA. Эти алгоритмы обеспечивают:

1. Алгоритмы планирования ресурсов предприятия (MRP II), которые помогают в оптимизации закупок, производства и распределения продукции по складам.
2. Интеллектуальные алгоритмы для анализа больших данных (Big Data), которые позволяют эффективно обрабатывать и использовать информацию, полученную с различных датчиков и устройств, установленных на производственных линиях [18].

Таким образом, алгоритмы, применяемые в АСУ ТП трубной промышленности, играют важную роль в оптимизации и автоматизации процессов производства, улучшении качества продукции и сокращении затрат. Эти технологии продолжают развиваться, предоставляя новые возможности для повышения эффективности и адаптации к меняющимся условиям рынка.

### *Заключение*

Современная трубная промышленность находится на передовой технологической революции, где автоматизация и инновации играют ключевую роль в повышении производительности и снижении затрат. Внедрение автоматизированных систем управления (АСУ ТП) позволяет не только оптимизировать производственные процессы, но и существенно улучшить качество продукции за счет использования сложных алгоритмов и систем контроля.

Алгоритмы, такие как PID-регуляторы, методы машинного зрения, нейронные сети, а также методы оптимизации, прогнозирования и диагностики, активно применяются для управления технологическими процессами, что способствует более точному контролю за качеством и более эффективному использованию ресурсов. Интеграция таких систем с ERP и MES позволяет обеспечить полную прозрачность всех этапов производства, что дает возможность более гибко реагировать на изменения в потребности рынка и повысить общую эффективность предприятия.

Одной из важнейших тенденций является использование искусственного интеллекта и анализа данных для предсказания поведения процессов и оптимизации производственных циклов. Это помогает минимизировать риски, связанные с технологическими сбоями, и эффективно управлять запасами и ресурсами. Внедрение таких

технологий приводит к снижению затрат, улучшению качества продукции и уменьшению экологического воздействия производства.

Таким образом, продолжение внедрения передовых алгоритмов и автоматизированных систем управления в трубной промышленности является неотъемлемой частью ее дальнейшего развития, что обеспечит предприятиям конкурентоспособность и устойчивость на глобальном рынке.

Автоматизированные системы управления и современные алгоритмы анализа данных играют важнейшую роль в повышении эффективности производства и качества продукции в трубной промышленности. Внедрение SCADA, MES и ERP-систем позволяет осуществлять контроль и оптимизацию на всех этапах производства, минимизируя риски и снижая издержки. Современные алгоритмы машинного обучения и временных рядов позволяют прогнозировать изменения и адаптировать параметры производства к изменяющимся условиям. Постоянное совершенствование программного обеспечения и математических моделей повышает конкурентоспособность предприятий, способствует улучшению качества продукции и устойчивости трубной промышленности в условиях современной экономики.

### *Литература*


1. Новосельский В.Б. Проблемы и задачи автоматизированного проектирования распределенных баз данных, Научно-технический вестник информационных технологий, механики и оптики, 2007, стр.157-163
2. Мищенко Я.В. Базы данных: современные тенденции в области баз данных и их значимость для различных сфер деятельности, Вестник науки, №7 (76), том 4, 2024, стр. 252-256.
3. Соколов В.А. Современные системы управления базами данных, Экономика и социум, №9 (40), 2017, стр. 441-446.
4. Попов Ф.А., Максимов А.В. Подходы к проектированию баз данных для автоматизированных систем, Известия Алтайского государственного университета, 2003, стр. 50-53.
5. Худяков В.Б. Использование СУБД в проектах машинного обучения и анализа данных, Вестник науки, №7 (64), том 5, 2023, стр. 279-295.
6. Новосельский В.Б. Проблемы и задачи автоматизированного проектирования распределенных баз данных, Научно-технический вестник информационных технологий, механики и оптики, 2007, стр. 157-163.
7. Попов Ф.А., Максимов А.В. Подходы к проектированию баз данных для автоматизированных систем, Известия Алтайского государственного университета, 2003, стр. 50-54.
8. Слегтина В.А. Обзор и сравнение SCADA-систем, Вестник науки, №11 (56), том 3, 2022, стр. 183-187.
9. Минин П.Е., Конев В.Н., Сычев Н.В., Крымов А.С., Савчук А.В., Андряков Д.А. Анализ существующих автоматизированных систем управления технологическим процессом, Спецтехника и связь, №1, 2014, стр. 29-37.
10. Антонова И.И., Смирнов В.А., Ефимов М.Г. Интеграция искусственного интеллекта в ERP-системы: достоинства, недостатки и перспективы, Russian Journal of Economics and Law, №3, том 18, 2024, 619-641.
11. Овездурдыева И.К., Гараджаева Дж. Я. Мировые стандарты управления промышленным предприятием в информационных системах, Всемирный ученый, №15, том 1, 2024, стр. 122-127.
12. Добренко Н.В., Добренко Д.А., Улизько М.В. Интеллектуальная поддержка принятия управленческих решений в mes-системах с использованием больших языковых моделей, «Экономика. Право. Инновации», №3, 2024, стр. 47-59.
13. Филатов Е.С., Польщиков К.А. Принципы контроля и оценки результатов деятельности организации в условиях применения современных информационных систем, Теория и практика современной науки, №6 (108), 2024, стр. 151-155.
14. Бобылев Ю.Н., Тарасов В.В., Любомиров Б.Н. Особенности автоматизации контроля и регулирования величины внутреннего грата электросварных прямошовных труб. доклад на симпозиуме "Неделя горняка - 98" МОСКВА, МГГУ, 2.02.98 - 6.02.98.
15. Маркова В. Д, Цифровизация управления: от АСУ к микросервисам, Всероссийский экономический журнал, 2024, стр. 113-130.
16. Кравчук А.Ю., Котова Н.А., Аничкин И. И. Современные подходы к обеспечению информационной безопасности автоматизированных систем управления технологическими процессами, Инновации и инвестиции, №3, 2022, стр. 191-196.
17. Чернов О.В. «Интеграция автоматизированных систем контроля качества с MES и ERP.» Журнал системного управления, 2021, №2, стр. 37-45.
18. Соколов П.Р. «Единый поток данных в производственных системах: подходы и решения.» Управление производством, 2020, №4, стр. 24-31.

***Коврик К. А., Криводубский О.А. Об особенностях современного производства труб и использования АСУ ТП в трубном производстве.*** *Статья обсуждает современные тенденции и достижения в разработке математического и программного обеспечения для автоматизированных систем. Рассматриваются ключевые аспекты, такие как алгоритмы, структуры данных, модели данных. Особое внимание уделяется применению методов машинного обучения и искусственного интеллекта для решения сложных задач.*

***Ключевые слова:*** *автоматизированные системы, математическое моделирование, алгоритмизация, искусственные интеллект, машинное обучение, SCADA, MES, ERP.*

***Kirill Kovrik, Oleg Krivodubsky. About the features of modern pipe production and the use of automated process control systems in pipe production.*** *The article discusses current trends and achievements in the development of mathematical and software for automated systems. Key aspects such as algorithms, data structures, and data models are considered. Special attention is paid to the application of machine learning and artificial intelligence methods to solve complex problems.*

***Keywords:*** *automated systems, mathematical modeling, algorithmizing, artificial intelligence, machine learning, neural networks, SCADA, MES, ERP.*

УДК 004.021

# Анализ существующих подходов и решений для интеграции знаний в САПР

**А.В. Григорьев[*1], Е.С. Бондаренко[*2]**

[*1] к.т.н, доцент, Донецкий национальный технический университет,
grigorievalvl@gmail.com, SPIN-код: 8830-2813

[*2] аспирант, Донецкий национальный технический университет,
kate.bond777@gmail.com

***Григорьев А.В., Бондаренко Е.С. Анализ существующих подходов и решений для интеграции знаний в САПР.*** *В статье рассматриваются существующие подходы и решения для интеграции знаний в системы автоматизированного проектирования (САПР). Интеграция знаний является важным аспектом, который способствует улучшению качества проектирования, ускорению разработки и снижению вероятности ошибок. В работе анализируются различные методы и технологии, включая использование стандартных форматов данных (STEP, XML), онтологий, экспертных систем, технологий Интернета вещей (IoT) и облачных платформ. Каждому подходу уделяется внимание с точки зрения его преимуществ и недостатков, а также их роли в обеспечении эффективного обмена и обработки данных на всех этапах жизненного цикла продукта. Особое внимание уделяется онтологическому подходу, который позволяет создать единую семантическую модель для объединения различных знаний и поддержания их актуальности.*

***Ключевые слова:*** *онтологии, интеграция знаний, САПР, специфика, проектирование.*

## *Введение*

Современные системы автоматизированного проектирования (САПР) играют ключевую роль в процессе разработки сложных инженерных объектов, включая архитектурные сооружения, машиностроительные изделия, электронные устройства и другие. С развитием технологий, возникла необходимость внедрения новых методов и подходов, которые могут повысить эффективность проектирования, улучшить качество решений и сократить время разработки. В настоящее время существует множество методов для интеграции знаний в САПР. Одним из таких методов является онтологический подход, который предлагает новую парадигму для организации знаний и информации в САПР.

Онтология как научная дисциплина возникла из философии и представляет собой теорию сущностей и их взаимосвязей. В контексте проектирования, онтологии используются для представления знаний о сущностях, процессах и связях между ними. Этот подход позволяет формализовать знания, создавать модели и структуры, которые облегчают понимание, обработку и использование данных в процессе проектирования.

Целью данной статьи является исследование существующих подходов и специфики онтологического подхода в проектировании объектов в САПР, а также выявление основных требований, которые необходимо учитывать при его реализации.

## *Существующие подходы и решения для интеграции знаний в САПР*

1. Интеграция знаний через стандартные и открытые форматы данных

Одним из самых распространённых подходов к интеграции знаний в САПР является использование стандартных форматов данных, таких как STEP (Standard for the Exchange of Product model data), XML (eXtensible Markup Language) и ISO 10303, которые предназначены для обмена проектной информацией между различными системами и платформами. Эти стандарты обеспечивают унификацию данных, что упрощает их обмен между различными САПР и другими информационными системами [1].

STEP, например, представляет собой набор стандартов, который описывает модель продукта и его жизненный цикл, включая проектирование, производство и эксплуатацию. Он используется для обмена трехмерными моделями, техническими данными и аттрибутами объектов между различными САПР и системами управления жизненным циклом продукта (PLM) [2]. Использование STEP помогает решить проблему несовместимости данных между различными программными средствами, что является одной из главных проблем

в интеграции знаний в САПР.

XML, в свою очередь, является более универсальным инструментом, который позволяет описывать данные в текстовом формате, что делает его удобным для хранения и передачи информации, связанной с проектированием, а также для обмена метаданными и документацией. Использование XML даёт возможность интегрировать различные типы знаний (например, проектные чертежи, спецификации, расчёты) и поддерживает расширяемость, позволяя добавлять новые виды данных по мере необходимости.

2. Использование онтологий для моделирования знаний

Онтологический подход в интеграции знаний представляет собой одну из самых перспективных технологий для САПР, поскольку он позволяет формализовать знания о проектируемых объектах и процессах, создавая модели, которые легко могут быть использованы для обмена и совместной работы между различными системами. Онтологии обеспечивают семантическую основу для представления знаний, что позволяет повысить точность и однозначность передачи данных.

Использование онтологий в САПР помогает решить проблему неоднозначности в интерпретации данных. Например, описание материала в разных системах может включать различные термины для одного и того же вещества (например, «сталь», «углеродистая сталь», «нержавеющая сталь»), что может привести к ошибкам в проектировании. Онтология, которая определяет «сталь» как сущность с набором атрибутов, таких как прочность, устойчивость к коррозии, плотность и другие характеристики, может обеспечить единую трактовку данных, независимо от того, в какой системе они были созданы.

Кроме того, онтологии позволяют создавать структуры для интеграции знаний, обеспечивая совместимость данных между разными этапами жизненного цикла продукта — от концептуального проектирования до эксплуатации. Это делает онтологический подход особенно полезным для проектов с долгосрочной поддержкой и сложными требованиями к данным.

3. Экспертные системы и искусственный интеллект

Экспертные системы, использующие технологии искусственного интеллекта (ИИ), представляют собой еще один важный инструмент для интеграции знаний в САПР. Эти системы позволяют моделировать знания экспертов и использовать их для принятия решений на различных стадиях проектирования. Экспертные системы могут быть интегрированы с САПР для автоматической оценки проектных решений, выбора материалов, оптимизации конструктивных решений и других задач.

Одним из основных преимуществ экспертных систем является их способность принимать решения на основе сложных логических правил и опыта, накопленного экспертами. Например, экспертная система может учитывать не только технические характеристики материалов, но и исторические данные о производственных процессах, возможных ошибках, а также требования безопасности и экологические стандарты. Это позволяет интегрировать знания, которые не всегда могут быть явно выражены в виде числовых данных или формул.

Системы, использующие методы ИИ, также могут автоматически обновлять свои знания на основе новых данных, что позволяет САПР быть более адаптивными к изменениям в требованиях, условиях эксплуатации или новых технологических разработках.

4. Интеграция знаний с использованием технологий Интернета вещей (IoT)

Технологии Интернета вещей (IoT) становятся всё более важным инструментом для интеграции знаний в САПР, особенно в контексте мониторинга и управления жизненным циклом продукта. IoT-устройства, такие как сенсоры и датчики, могут собирать данные о состоянии объектов в реальном времени, что позволяет интегрировать знания о фактической эксплуатации объектов в проектный процесс.

Эти данные могут быть использованы для оптимизации проектных решений, например, для корректировки параметров конструкции с учётом реальных условий эксплуатации, что значительно повышает точность моделирования. Интеграция данных IoT в САПР позволяет не только улучшить проектирование, но и улучшить управление жизненным циклом продукта, предсказание поломок, техническое обслуживание и другие процессы, основанные на реальных данных.

Технологии IoT открывают новые возможности для интеграции знаний, обеспечивая постоянное обновление данных и их обработку с использованием облачных платформ и больших данных (Big Data). Это позволяет создавать более точные и эффективные модели объектов, что важно для комплексных инженерных решений.

5. Современные решения для интеграции знаний и их вызовы

Современные решения для интеграции знаний в САПР включают использование облачных вычислений, платформ для управления данными и большие данные (Big Data). Облачные платформы обеспечивают доступность данных и моделей для разных участников процесса проектирования, что особенно важно в распределённых командах и для проектов с большим количеством взаимозависимых элементов.

Однако, несмотря на множество перспективных технологий, интеграция знаний в САПР сталкивается с рядом вызовов. Одним из основных препятствий является несовместимость данных между различными системами, а также проблемы, связанные с различием в форматах данных, стандартах и методах моделирования.

Для эффективной интеграции необходимо разработать универсальные подходы, которые обеспечат совместимость различных систем, а также обеспечить безопасность данных, особенно при использовании облачных технологий.

Ещё одной проблемой является необходимость в стандартизации методов обмена данными и представления знаний. В то время как технологии, такие как онтологии и XML, обеспечивают высокую гибкость, их внедрение требует значительных усилий по стандартизации, что не всегда осуществимо на уровне отрасли или отдельного предприятия.

### *Сравнительная характеристика существующих подходов*

В таблице 1 представлены преимущества и недостатки различных подходов для интеграции знаний в САПР.

Таблица 1 – Сравнительная характеристика существующих подходов для интеграции знаний в САПР

| Подход | Преимущества | Недостатки |
|---|---|---|
| Использование стандартных форматов данных (STEP, XML) | - Унифицированные форматы для обмена данными между различными системами.<br>- Широко поддерживаются промышленными стандартами.<br>- Обеспечивают совместимость между САПР и другими платформами. | - Ограниченность в описании сложных взаимозависимостей.<br>- Не всегда обеспечивают полную семантическую точность данных.<br>- Требуют строгой структуры. |
| Онтологии | - Обеспечивают точное и однозначное представление знаний.<br>- Могут эффективно моделировать сложные взаимозависимости между сущностями.<br>- Поддержка автоматизации и логики принятия решений.<br>- Гибкость и расширяемость. | - Требуют значительных усилий для разработки и внедрения.<br>- Потребность в стандартизации и единых подходах для всех участников.<br>- Сложности с обучением пользователей. |
| Экспертные системы и ИИ | - Автоматизация принятия решений на основе накопленного опыта.<br>- Поддержка адаптации к новым условиям и данным.<br>- Использование накопленных знаний для оценки и прогнозирования. | - Сложность в настройке и обучении системы.<br>- Необходимость в постоянном обновлении базы знаний.<br>- Ограниченность в масштабировании для сложных проектов. |
| Интернет вещей (IoT) | - Интеграция реальных данных с этапами проектирования.<br>- Обновление информации в реальном времени.<br>- Улучшение проектирования с учётом фактических условий эксплуатации. | - Высокая стоимость и сложность внедрения IoT-устройств.<br>- Проблемы с безопасностью данных.<br>- Требуют высококоростного интернета и вычислительных мощностей для обработки данных. |
| Облачные платформы и Big Data | - Обеспечивают доступность данных для различных участников проекта.<br>- Масштабируемость и возможности для обработки больших объёмов данных.<br>- Гибкость в управлении данными и моделями. | - Зависимость от интернет-соединения и облачных сервисов.<br>- Проблемы с безопасностью и защитой данных.<br>- Высокие затраты на хранение и обработку данных. |

Каждый из подходов имеет свои сильные стороны и ограничения. Онтологии являются наиболее гибким и мощным инструментом для интеграции знаний в САПР, особенно когда речь идет о моделировании сложных взаимозависимостей и автоматизации процессов. Однако их внедрение требует значительных усилий. Стандартные форматы данных (STEP, XML) хорошо подходят для обмена структурированными данными между системами, но не всегда могут эффективно моделировать сложные зависимости. Экспертные системы и ИИ хороши для автоматизации принятия решений, но имеют ограничения в масштабируемости и требуют постоянного обновления знаний. IoT и облачные технологии могут быть полезными для получения реальных данных, но они также сопряжены с высокими затратами и техническими проблемами.

### *Преимущества онтологического подхода*

На данный момент одним из наиболее перспективных и многообещающих подходов для интеграции знаний в САПР является использование онтологий. Этот подход предоставляет ряд преимуществ, которые делают его особенно эффективным для решения задач, связанных с интеграцией и обработкой данных в процессе проектирования [3].

1. Универсальность и стандартизация: Онтологии обеспечивают унифицированное представление знаний, что позволяет создать общую семантическую основу для различных систем и участников проектирования. Это решение способствует лучшему обмену данными между различными САПР и другими информационными системами (например, системами управления жизненным циклом продукта, PLM). Онтологии могут быть стандартизированы и адаптированы под различные области, что значительно облегчает их внедрение и интеграцию в рамках разных процессов проектирования.

2. Семантическая точность и однозначность: Онтология помогает решить одну из основных проблем интеграции знаний — неоднозначности терминов и представлений. Проектные данные, такие как материалы, детали или компоненты, часто имеют разные интерпретации в разных системах. Онтология позволяет формализовать такие данные, предоставляя точные определения и отношения между сущностями. Это способствует уменьшению ошибок и улучшению совместимости между различными САПР.

3. Поддержка сложных связей и взаимозависимостей: Онтология позволяет не только описывать отдельные сущности (например, детали, материалы или процессы), но и моделировать их сложные взаимосвязи. Это особенно важно при проектировании сложных систем, где объекты могут зависеть друг от друга, и любое изменение в одном компоненте может повлиять на весь проект. Онтология помогает отследить такие зависимости и гарантировать, что все изменения в проекте будут учтены.

4. Возможности для автоматизации и обработки знаний: Онтологии обеспечивают логическую структуру, которая позволяет использовать автоматические инструменты для проверки проектных решений, выбора материалов, оценки производственных процессов и других задач. Интеграция онтологии с экспертными системами и искусственным интеллектом может значительно повысить уровень автоматизации в проектировании, уменьшив вероятность ошибок и ускорив процесс разработки.

5. Гибкость и расширяемость: Онтологии легко адаптируются к изменениям в проектных требованиях и новым знаниям. Когда появляются новые данные или необходимость в учёте дополнительных характеристик, структура онтологии может быть легко дополнена или модифицирована, что делает систему гибкой и готовой к изменениям. Это позволяет создавать более долгосрочные и поддерживаемые решения, которые могут изменяться вместе с развитием технологий и требований.

6. Поддержка различных этапов жизненного цикла продукта: Онтология может использоваться для интеграции знаний не только в процессе проектирования, но и на всех этапах жизненного цикла продукта, включая производство, эксплуатацию и утилизацию. Это делает её важным инструментом для создания эффективных систем управления жизненным циклом продукта (PLM), которые могут адаптироваться к изменениям и улучшать процессы в реальном времени.

Хотя другие подходы, такие как использование стандартных форматов данных (STEP, XML) или экспертных систем, тоже имеют свои преимущества, они ограничены по гибкости и не всегда способны эффективно справляться с задачами, связанными с автоматической обработкой сложных знаний и данных. Например, стандартные форматы данных требуют строгой структуры и могут не учитывать все возможные зависимости и контексты, которые необходимы для полноценного проектирования.

Экспертные системы также предоставляют решения для автоматизации, но они часто требуют значительных усилий для создания и поддержания, особенно когда речь идет о больших и сложных проектах. Онтологии, в свою очередь, обеспечивают более универсальный и масштабируемый способ представления знаний, который может быть интегрирован с другими системами и технологиями, такими как искусственный интеллект и Интернет вещей (IoT).

### *Структура и компоненты онтологии в САПР*

Структура онтологии в САПР включает несколько ключевых компонентов, которые помогают организовать знания о проектируемых объектах и процессах. Основные компоненты онтологии [4]:

Классы и экземпляры. Классы представляют собой категории объектов, а экземпляры — это конкретные представители этих классов. Например, класс "Детали" может включать экземпляры "Шпилька", "Шайба" и т.д.

Атрибуты сущностей. Каждый класс может иметь набор атрибутов, описывающих его характеристики. Для "Детали" атрибутами могут быть размер, материал, масса и другие параметры.

Связи между сущностями. Онтологии могут включать различные типы связей между сущностями, например, "часть от", "составляет компонент", "зависит от".

Правила и логика. Онтологии включают формализованные правила, которые гарантируют правильность

данных и могут быть использованы для автоматической проверки проектных решений.

Структура онтологии в САПР представляет собой организованную систему знаний, которая формализует объекты проектирования и их взаимосвязи. Основной элемент онтологии — это сущности, которые описывают различные объекты, с которыми работает система, такие как детали, материалы или процессы. Каждая сущность обладает набором атрибутов, определяющих её характеристики, например, размер, материал или физические свойства. Связи между сущностями описывают их отношения друг к другу, например, как часть может быть частью более сложного компонента или как материал используется для производства определённых деталей. Онтология также включает иерархическую организацию классов, где более общие категории могут быть разделены на более специфичные, что позволяет создавать структурированные модели, легко расширяемые или изменяемые. Аксиомы, в свою очередь, обеспечивают логические правила, которым должны подчиняться сущности и их связи, обеспечивая корректность данных.

Такое представление позволяет моделировать проектируемые объекты в САПР и процессы их взаимодействия, а также обеспечивает более точное и структурированное представление информации.

### *Требования к онтологии в САПР и область ее применения*

Для эффективного использования онтологического подхода в САПР необходимо учесть ряд требований к структуре и функционированию онтологии:

Гибкость и расширяемость. Онтология должна быть гибкой, чтобы учитывать изменения в проектных требованиях и поддерживать добавление новых сущностей и связей.

Совместимость и стандарты. Для успешной интеграции с другими САПР и информационными системами важно использовать стандарты представления знаний, такие как OWL (Web Ontology Language) или RDF (Resource Description Framework) [5].

Модульность. Онтология должна быть модульной, что позволяет изменять или дополнять отдельные части без перепроектирования всей системы.

Простота и доступность. Онтология должна быть понятной и доступной для всех участников проекта, включая инженеров, архитекторов, менеджеров и других специалистов.

Интероперабельность. Онтология должна поддерживать обмен данными между различными системами, платформами и участниками проекта.

Онтологический подход находит широкое применение в различных областях проектирования, включая машиностроение, строительство, электронику и архитектуру. В каждом из этих областей онтология используется для создания более точных и структурированных моделей объектов и процессов проектирования.

В машиностроении онтологии позволяют описывать детали и узлы с их характеристиками и взаимосвязями, что помогает в автоматизации выбора материалов и оптимизации конструктивных решений. В строительстве онтологии используются для представления зданий и сооружений, а также их компонентов, что упрощает проектирование и улучшает координацию между различными специалистами. В области электроники онтологии помогают моделировать схемы и компоненты, что способствует более точному и быстрому проектированию электрических систем.

В целом же онтологии могут быть применены для разных целей в системах автоматизированного проектирования, что позволяет значительно упростить работу проектировщиков.

Моделирование и описание объектов и процессов проектирования: Онтологии позволяют формализовать и описывать проектируемые объекты, компоненты, материалы, технологии и процессы с учетом их характеристик и взаимозависимостей. Это позволяет создать единое представление о проектируемом объекте, которое может быть использовано на всех этапах разработки, от концептуального проектирования до производства и эксплуатации.

Управление данными и обмен информацией: Онтологии обеспечивают стандартизированное описание данных, что упрощает обмен информацией между различными САПР и другими системами (например, PLM, ERP). Благодаря онтологиям, данные могут быть представлены в единой семантической модели, что уменьшает риски ошибок, связанных с несовместимостью форматов или неоднозначностью трактовки данных.

Автоматизация проектных решений: Онтологии позволяют создавать логические модели, которые используются для автоматической оценки проектных решений, выбора материалов, оптимизации конструкции и других аспектов проектирования. Это сокращает время на принятие решений и минимизирует вероятность ошибок, основанных на человеческом факторе.

Интеграция знаний с экспертными системами: Онтологии могут быть связаны с экспертными системами и системами искусственного интеллекта, что позволяет использовать накопленные знания для анализа проектных данных, прогнозирования поведения объектов и выбора оптимальных решений. Например, на основе онтологии можно разработать систему, которая будет автоматически анализировать проект и предлагать улучшения с учетом существующих норм и требований.

Поддержка различных этапов жизненного цикла продукта: Онтологии способствуют интеграции знаний, полученных на разных этапах жизненного цикла продукта, начиная с его проектирования и заканчивая эксплуатацией. С помощью онтологии можно объединить данные о проектировании, производстве, обслуживании и утилизации, что делает процесс управления жизненным циклом продукта более эффективным.

Решение проблемы неоднозначности терминов: Онтологии позволяют стандартизировать терминологию и концепции, используемые в проектировании, что помогает устранить проблемы, связанные с различием в трактовке терминов между различными специалистами, проектными командами или различными САПР.

Обеспечение совместимости и интеграции систем: Онтологии облегчают интеграцию различных информационных систем, таких как системы проектирования, управления жизненным циклом продукта (PLM), системы управления производством и другие. Это помогает обеспечить единую информационную среду, в которой данные и знания легко передаются и используются между системами и участниками процесса.

### *Особенности внедрения онтологического подхода в САПР*

Внедрение онтологического подхода в системы автоматизированного проектирования (САПР) связано с рядом особенностей, обусловленных необходимостью интеграции знаний, структуры данных и автоматизации процессов проектирования. Основным вызовом является создание единой семантической модели, которая точно и полно охватывает все аспекты проектируемых объектов и процессов, что требует глубокого анализа и формализации знаний. Онтологии должны быть адаптированы к специфике конкретных отраслей и типов проектирования, что требует гибкости в их разработке и применении. Одной из сложностей является необходимость стандартизации и унификации терминов и понятий, чтобы обеспечить совместимость данных между различными системами и участниками процесса. Также требуется тесное взаимодействие с экспертными системами и искусственным интеллектом для автоматизации процессов принятия решений и повышения точности проектирования. Внедрение онтологий в САПР требует значительных усилий по обучению пользователей и интеграции новых инструментов в существующую инфраструктуру, что может потребовать временных и финансовых затрат.

### *Заключение*

Интеграция знаний в САПР представляет собой многогранную задачу, охватывающую различные аспекты проектирования и управления данными. Существующие подходы, включая использование стандартных форматов данных, онтологий, экспертных систем, технологий IoT и облачных вычислений, позволяют решать многие проблемы, связанные с обменом и обработкой данных. Тем не менее, остаются вызовы, такие как несовместимость данных, необходимость в стандартизации и безопасности информации. Решение этих проблем требует дальнейших исследований и разработки более универсальных технологий для интеграции знаний в САПР, что в свою очередь поспособствует улучшению качества проектирования.

### *Литература*

1. Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2), 199–220.

2. Berners-Lee, T., Fischetti, M., & Moss, L. (2001). "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities." *Scientific American*.

3. Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods, and applications. *Knowledge Engineering Review*, 11(2), 93-136.

4. Staab, S., & Studer, R. (2009). Handbook on Ontologies. Springer.

5. Buitelaar, P., Cimiano, P., & Magnini, B. (2005). Ontology Learning from Text: Methods, Evaluation and Applications. Springer.

***Григорьев А.В., Бондаренко Е.С. Анализ существующих подходов и решений для интеграции знаний в САПР.*** *В статье рассматриваются существующие подходы и решения для интеграции знаний в системы автоматизированного проектирования (САПР). Интеграция знаний является важным аспектом, который способствует улучшению качества проектирования, ускорению разработки и снижению вероятности ошибок. В работе анализируются различные методы и технологии, включая использование стандартных форматов данных (STEP, XML), онтологий, экспертных систем, технологий Интернета вещей (IoT) и облачных платформ. Каждому подходу уделяется внимание с точки зрения его преимуществ и недостатков, а также их роли в обеспечении эффективного обмена и обработки данных на всех этапах жизненного цикла продукта. Особое внимание уделяется онтологическому подходу, который позволяет создать единую семантическую модель для объединения различных знаний и поддержания их актуальности.*

***Ключевые слова:*** *онтологии, интеграция знаний, САПР, специфика, проектирование.*

***Grigoriev A.V., Bondarenko E.S. Analysis of existing approaches and solutions for integrating knowledge into CAD.*** *The article discusses existing approaches and solutions for integrating knowledge into computer-aided design (CAD) systems. Knowledge integration is an important aspect that helps to improve the quality of design, accelerate development and reduce the likelihood of errors. The paper analyzes various methods and technologies, including the use of standard data formats (STEP, XML), ontologies, expert systems, Internet of Things (IoT) technologies and cloud platforms. Each approach is given attention in terms of its advantages and disadvantages, as well as their role in ensuring efficient data exchange and processing at all stages of the product lifecycle. Special attention is paid to the ontological approach, which allows you to create a single semantic model for combining various knowledge and maintaining their relevance.*

***Key words:*** *ontologies, knowledge integration, CAD, specifics, design.*

УДК 004.89

# Анализ задачи разработки универсального языка проектирования как средства автоматизации Microsoft Office Visio

**К.В. Ржевский *1, А. В. Григорьев *2**

*1 ассистент кафедры ПИ им. Л.П. Фельдмана, аспирант, Донецкий национальный технический университет, nory4ik4@mail.ru

*2 к.т.н., доцент кафедры ПИ, Донецкий национальный технический университет, grigorievalvl@mail.ru, SPIN-код: 8830-2813

***Анализ задачи разработки универсального языка проектирования.*** *Статья посвящена вопросу расширенных возможностей создания API с возможностью интеллектуальных надстроек. В статье описан пример использования Aspose.Diagram в Visio для создания рабочих графических надстроек Microsoft Visio.*

***Ключевые слова:*** *интеллектуальные паттерны, онтологии, создание API, файл VSDX, библиотека Aspose.Diagram, программа Microsoft Visio.*

## *Введение*

В современном мире проектирования и моделирования визуальная демонстрация играет ключевую роль. Microsoft Office Visio предоставляет мощные инструменты для создания диаграмм, схем и моделей, что делает его идеальным решением для разработки «языка описания объектов проектирования». В этой статье мы рассмотрим возможности, которые предоставляет Visio для создания унифицированного языка онтологий, интеллектуальных паттернов, направленных на стандартизацию процессов проектирования и облегчение взаимодействия между различными участниками проекта. Интеллектуальные паттерны играют существенную роль в разработке эффективных решений, как в бизнесе, так и в других областях, включая проектирование программного обеспечения, управление данными и стратегии цифровой трансформации.

Перечислим основные проблемы существующих средств автоматизации компонентов офиса:

1. Сложность существующих инструментов: Многие пользователи сталкиваются с трудностями при использовании текущих инструментов для моделирования и визуализации бизнес-процессов. Сложные интерфейсы и технические термины могут отпугнуть пользователей без специальной подготовки.
2. Отсутствие стандартизации: На рынке существует множество различных языков и методов моделирования, что затрудняет обмен информацией между организациями и приводит к недопониманию.
3. Ручные процессы: В современных бизнесах многие процессы по-прежнему выполняются вручную, что увеличивает вероятность ошибок и снижает общую продуктивность.
4. Ограниченная интеграция: Многие инструменты визуализации не интегрируются друг с другом, что создает проблемы с обменом данными и совместной работой.
5. Недостаток аналитических возможностей: Существующие инструменты часто не предоставляют достаточных возможностей для глубокого анализа данных и визуализации, что затрудняет принятие обоснованных решений.

В связи с названными проблемами имеют место следующие актуальные направления работ в области автоматизации функционирования компонентов офисов:

1. Рост цифровизации: В условиях стремительной цифровизации бизнеса необходимость в автоматизации и улучшении процессов становится все более актуальной. УЯП может стать важным инструментом для повышения эффективности работы.
2. Упрощение взаимодействия: Разработка универсального языка позволит более широкому кругу пользователей, включая менеджеров и специалистов, без технического образования, участвовать в моделировании и анализе бизнес-процессов.
3. Требования к стандартам: В условиях глобализации бизнеса необходимость использования международных стандартов становится критически важной. УЯП, соответствующий таким стандартам, упростит взаимодействие между компаниями.
4. Поддержка инноваций: Создание сообщества разработчиков вокруг УЯП может способствовать инновациям и разработке новых инструментов, что в свою очередь будет способствовать развитию бизнеса.
5. Увеличение конкурентоспособности: Компании, которые смогут быстро адаптироваться к изменениям и эффективно управлять своими процессами, будут иметь конкурентные преимущества на рынке.

Таким образом, разработка универсального языка проектирования в Microsoft Office Visio как средства

автоматизации его функционирования является актуальной задачей, которая может решить существующие проблемы и значительно повысить эффективность работы с бизнес-процессами.

В предлагаемой статье на примере Microsoft Office Visio будут рассмотрены следующие вопросы:

1.Определение универсального языка проектирования (УЯП): Что такое УЯП, его основные характеристики и цели.

2.Проблемы существующих инструментов: Анализ текущих недостатков в инструментах моделирования и визуализации, используемых в Microsoft Office Visio и других приложениях.

3.Актуальность разработки УЯП: Обсуждение причин, по которым создание универсального языка проектирования становится необходимым в условиях современного бизнеса и цифровизации.

4.Преимущества УЯП: Как УЯП может улучшить процессы моделирования, повысить продуктивность и упростить взаимодействие между пользователями с разным уровнем подготовки.

5.Интеграция с существующими системами: Возможности интеграции УЯП с другими инструментами и платформами для повышения эффективности работы.

6.Стандартизация и совместимость: Важность соблюдения международных стандартов и как УЯП может способствовать улучшению совместимости между различными организациями.

7.Аналитические возможности: Как УЯП может улучшить аналитические инструменты и визуализацию данных для поддержки принятия обоснованных решений.

8.Обучение и поддержка пользователей: Роль обучения и поддержки в успешной реализации УЯП, а также создание ресурсов для пользователей.

9.Сообщество разработчиков: Как создание сообщества вокруг УЯП может стимулировать инновации и развитие новых инструментов.

10. Будущее УЯП: Перспективы и потенциальные направления развития универсального языка проектирования в контексте изменяющегося рынка и технологий.

Анализ этих вопросов поможет глубже обосновать значимость разработки универсального языка проектирования Microsoft Office Visio и определить его влияние на эффективность бизнес-процессов.

### *Сведения об элементе управления VSDX*

VSDX [2] — это формат файла, который используется Microsoft Visio [6], программой для создания диаграмм и технических рисунков. Он относится к семейству XML-файлов и заменяет старый формат VSD. VSDX предлагает больше возможностей для работы с данными и взаимодействия с ними. Этот формат удобен тем, что он поддерживает несколько слоёв диаграмм и сложные графические элементы, сохраняя при этом все свойства, такие как цвета, шрифты и размеры. Основное отличие VSDX от его предшественника, VSD, заключается в том, что VSDX использует XML-структуру, что делает файлы более доступными и удобными для работы.

### *API для форматов файлов Microsoft Visio*

Aspose.Diagram for .NET [3] — это многофункциональный API, который позволяет создавать, редактировать, преобразовывать и обрабатывать диаграммы MS Visio из приложений .NET. API упрощает работу с диаграммами VSDX с помощью простых в использовании свойств и методов. Вы можете либо загрузить библиотеку DLL API, либо установить ее в свои приложения .NET с помощью NuGet.

Aspose.Diagram for .NET поддерживает почти все исходные форматы Visio, а также некоторые часто используемые изображения & форматы с фиксированным макетом. Эта API является альтернативой объектной модели Microsoft Visio и обеспечивает более высокую производительность. Он использует расширенные функции служб Visio для управления документами на сервере.Дополнительные функции:

- Чтение нескольких форматов и указать расположение шрифтов;
- Сгруппировать несколько фигур;
- Настройка фигур временной шкалы;
- Чтение свойств различных объектов diagram;
- Сохранение диаграмм в формате XML или XAML;
- Управление Visio свойствами документа;
- Печать diagram на сервере через XpsPrint API;
- Поворот фигуры под любым углом;
- Добавляйте комментарии к рисункам;
- Вставка новой пустой страницы в рисунок;
- Печатайте диаграммы с высокой точностью;
- Укажите рецензента для создания комментариев в Diagram;
- Обнаружение и удаление неиспользуемых тем, графических данных и стилей;

- Извлечение данных формы на основе формулы Dependson;
- Получить все значения реквизита для фигуры;
- Добавлена поддержка линий NUBRS при экспорте чертежа;
- Поддержка рисования поллиний и фигур Безье;
- Отображение комментариев при сохранении в виде изображения или HTML.

### *Microsoft Visio векторный графический редактор*

Microsoft Visio [1] — популярное приложение, которое позволяет создавать широкий спектр диаграмм, таких как блок-схемы, диаграммы потоков данных, модели бизнес-процессов и т. д. VSDX — это формат файла, который MS Visio использует для хранения диаграммы.

Оно позволяет пользователям создавать различные диаграммы и блок-схемы, включая:

— организационные схемы,

— сетевые диаграммы,

— блок-схемы процессов,

— планы этажей и многое другое.

Visio [5] включает в себя широкий спектр форм и шаблонов для создания диаграмм, а также ряд инструментов для форматирования и настройки этих диаграмм.

Приложение поддерживает сотрудничество, позволяя нескольким пользователям работать на одной и той же диаграмме одновременно.

Чтобы автоматизировать манипуляции с VSDX, в этой статье представлено базовое руководство по созданию диаграмм Visio с нуля на C# [7]. Кроме того, в нем рассказывается, как вставлять страницы, фигуры и текст в диаграммы VSDX из приложений .NET.

### *Создание фигуры в диаграмме Visio с помощью C#*

Фигуры — это строительные блоки схем Visio. MS Visio поддерживает широкий спектр форм для создания диаграмм в различных областях. Следующие шаги показывают, как вставить фигуру в схему Visio.

- Создайте новую диаграмму или загрузите существующую, используя класс Diagram.
- Создайте страницу или получите нужную страницу в объекте Page;
- Добавьте master на диаграмму, используя метод Diagram.AddMaster(String fileName, Int masterID) ;
- Добавьте новую прямоугольную форму, используя метод Diagram.AddShape(pinX, pinY, width, height, masterName, PageIndex);
- Сохраните идентификатор формы, возвращенный методом Diagram.AddShape();
- Получите вновь добавленную фигуру в объекте Shape с помощью метода Page.Shapes.GetShape(long ID) ;
- Установите свойства фигуры, такие как текст, цвет и т. д;
- Сохраните диаграмму VSDX, используя метод Diagram.Save(Sring fileName, SaveFileFormat.VSDX).

### *Создание диаграммы Visio с помощью VSDX*

Прежде всего, давайте создадим с нуля пустую диаграмму VSDX. Ниже приведены шаги для этого:

- Создайте экземпляр класса Diagram;
- Используйте метод Diagram.Save(Sring fileName, SaveFileFormat.VSDX), чтобы сохранить файл как VSDX.

В следующем примере кода показано, как добавить фигуру на схему Visio с помощью C# [4].

На рис. 4 изображен код создания пустого файла vsdx.

```
using System.IO;
using System;
using Aspose.Diagram;

namespace Aspose.Diagram.Examples.CSharp.Diagrams
{
    1 reference
    public class CreateDiagram
    {
        1 reference
        public static void Run()
        {
            // ExStart:CreateDiagram
            // The path to the documents directory.
            string dataDir = RunExamples.GetDataDir_Diagrams();

            // Create directory if it is not already present.
            bool IsExists = System.IO.Directory.Exists(dataDir);
            if (!IsExists)
                System.IO.Directory.CreateDirectory(dataDir);
            // Initialize a new Visio
            Diagram diagram = new Diagram();
            dataDir = dataDir + "CreateDiagram_out.vsdx";
            // Save in the VSDX format
            diagram.Save(dataDir, SaveFileFormat.VSDX);
            // ExEnd:CreateDiagram
            Console.WriteLine("\nDiagram created successfully.\nFile saved at " + dataDir);
        }
    }
}
```

Рисунок 4 – Код создания файла vsdx

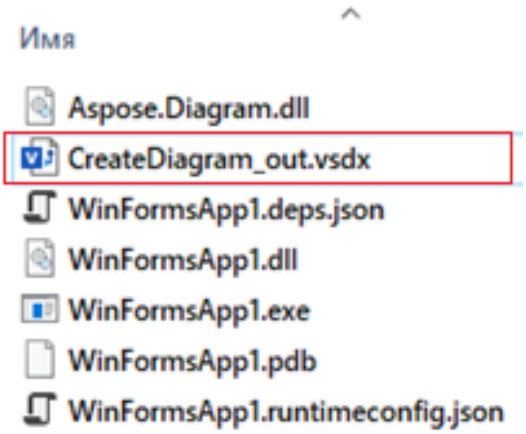



Рисунок 5 – Созданный файл vsdx

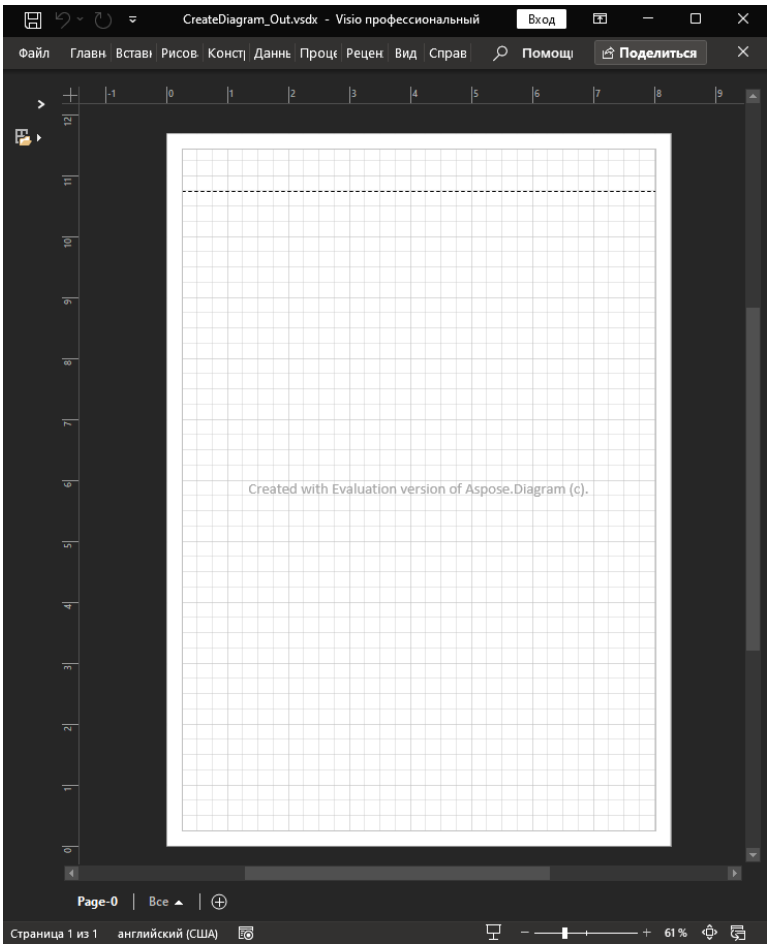


Рисунок 6 – Содержание файла vsdx.

Добавление текстовой фигуры на страницу Visio в C# [8].

В некоторых случаях вам также необходимо добавить текст к диаграммам Visio. Для этого вы можете выполнить следующие шаги.

- Создайте новую диаграмму или загрузите существующую, используя класс Diagram;
- Добавьте текст на определенную страницу, используя метод Diagram.Pages[0].AddText(PinX, PinY, Width, Height, “Test text”);
- Сохраните диаграмму VSDX, используя метод Diagram.Save(Sring fileName, SaveFileFormat.VSDX).

В следующем примере кода показано, как добавить текст на схему VSDX с помощью C#.

На рис. 6 изображен программный код создания фигур.

### *Работа со свойствами документа с помощью Aspose.Diagram*

Разработчики могут динамически управлять свойствами документа с помощью API Aspose.Diagram. Эта функция помогает разработчикам хранить полезную информацию вместе с файлом, например, когда файл был получен, обработан, с отметкой времени и т. д.

Aspose.Diagram for .NET непосредственно записывает информацию о API и номере версии в выходных документах.

Обратите внимание, что вы не можете поручить Aspose.Diagram for .NET изменить или удалить эту информацию из выходных документов.

```
1 reference
public static void Run()
{
    // ExStart:AddingNewShape
    // The path to the documents directory.
    string dataDir = RunExamples.GetDataDir_Shapes();

    // Load a diagram
    Diagram diagram = new Diagram(dataDir + "Drawing1.vsdx");
    // Get page by name
    Page page = diagram.Pages.GetPage("Page-2");

    // Add master with stencil file path and master name
    string masterName = "Rectangle";
    diagram.AddMaster(dataDir + "Basic Shapes.vss", masterName);

    // Page indexing starts from 0
    int PageIndex = 1;
    double width = 2, height = 2, pinX = 4.25, pinY = 4.5;
    // Add a new rectangle shape
    long rectangleId = diagram.AddShape(pinX, pinY, width, height, masterName, PageIndex);

    // Set shape properties
    Shape rectangle = page.Shapes.GetShape(rectangleId);
    rectangle.XForm.PinX.Value = 5;
    rectangle.XForm.PinY.Value = 5;
    rectangle.Type = TypeValue.Shape;
    rectangle.Text.Value.Add(new Txt("Aspose Diagram"));
    rectangle.TextStyle = diagram.StyleSheets[3];
    rectangle.Line.LineColor.Value = "#ff0000";
    rectangle.Line.LineWeight.Value = 0.03;
    rectangle.Line.Rounding.Value = 0.1;
    rectangle.Fill.FillBkgnd.Value = "#ff00ff";
    rectangle.Fill.FillForegnd.Value = "#ebf8df";

    /* refreshes shape's position, including XForm, TextXForm, connection and geom data
     when changing the shape's text and other's*/
    rectangle.RefreshData();
    diagram.Save(dataDir + "AddShape_out.vsdx", SaveFileFormat.VSDX);
    Console.WriteLine("Shape has been added.");
    // ExEnd:AddingNewShape
}
}
}
```

Рисунок 6 –Код создания фигур

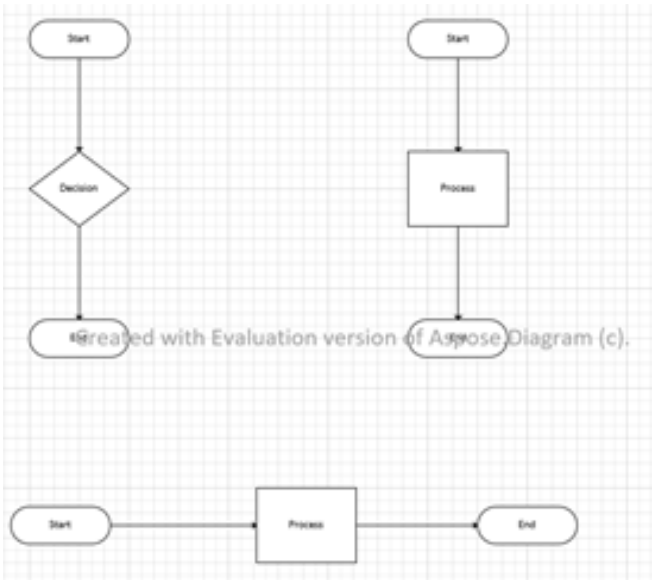



Рисунок 7 – Файл с фигурами

### *Доступ к свойствам документа*

Aspose.Diagram API поддерживают оба типа свойств документа, встроенные и настраиваемые. Aspose.Diagram далие Diagram class представляет файл Visio и, как и файл Visio.Diagram класс может содержать

несколько страниц, каждая из которых представлена Страница класс, тогда как коллекция страниц представлена Коллекция страниц учебный класс.

Использовать Diagram для доступа к свойствам документа файла, как описано ниже (рис. 8):

- Чтобы получить доступ к встроенным свойствам документа, используйте diagram.DocumentProps;
- Чтобы получить доступ к пользовательским свойствам документа, используйте diagram.DocumentProps.CustomProps.
- Как мы описали ранее в начале этого раздела, разработчики не могут добавлять или удалять встроенные свойства, поскольку эти свойства определяются системой, но можно добавлять или удалять настраиваемые свойства, поскольку они определяются пользователем.

### *Добавление пользовательских свойств*

- Aspose.Diagram API-интерфейсы раскрыли добавлять метод для CustomPropCollection класс для добавления настраиваемых свойств в коллекцию (рис. 9) [9].

```
1.  // The path to the documents directory.
2.  string dataDir = RunExamples.GetDataDir_Shapes();
3.
4.  // Load a Visio diagram
5.  Diagram diagram = new Diagram(dataDir + "Drawing1.vsdx");
6.
7.  //// Display Visio version and document modification time at different stages
8.  Console.WriteLine("Visio Instance Version : " + diagram.Version);
9.  Console.WriteLine("Full Build Number Created : " + diagram.DocumentProps.BuildNumberCreated);
10. Console.WriteLine("Full Build Number Edited : " + diagram.DocumentProps.BuildNumberEdited);
11. Console.WriteLine("Date Created : " + diagram.DocumentProps.TimeCreated);
12. Console.WriteLine("Date Last Edited : " + diagram.DocumentProps.TimeEdited);
13. Console.WriteLine("Date Last Printed : " + diagram.DocumentProps.TimePrinted);
14. Console.WriteLine("Date Last Saved : " + diagram.DocumentProps.TimeSaved);
15. Console.WriteLine("CustomProps Length " + diagram.DocumentProps.CustomProps.Count);
```

Рисунок 8 –Добавление или удаление пользовательских свойств документа

```
1.  // The path to the documents directory.
2.  string dataDir = RunExamples.GetDataDir_Shapes();
3.
4.  // Load a Visio diagram
5.  Diagram diagram = new Diagram(dataDir + "Drawing1.vsdx");
6.
7.  //// Get CustomProperties of diagram
8.  Aspose.Diagram.CustomPropCollection customProperties = diagram.DocumentProps.CustomProps;
9.  //Set property of CustomProp
10. Aspose.Diagram.CustomProp customProp = new Aspose.Diagram.CustomProp();
11. customProp.PropType = Aspose.Diagram.PropType.String;
12. customProp.CustomValue.ValueString = "Test";
13. //Add CustomProp to Collection
14. customProperties.Add(customProp);
```

Рисунок 9 – Добавление пользовательских свойств

### *Удаление пользовательских свойств*

Чтобы удалить пользовательские свойства с помощью Aspose.Diagram, вызовите CustomPropCollection. Удалить и передайте имя удаляемого свойства документа.

### *Добавление элемента SolutionXML в чертеж Visio*

SolutionXML[10] — это правильно сформированный XML, содержащийся в элементе SolutionXML, который предоставляет стандартизированные средства сохранения данных решения. Пользователи могут хранить

SolutionXML на уровне документа, где он сохраняется непосредственно в элементе VisioDocument. Как правило, это самый простой способ сохранить и получить SolutionXML с помощью Aspose.Diagram for .NET.

SolutionXML класс представляет элемент SolutionXML в чертежах Visio. Метод Add, предоставляемый SolutionXML класс позволяет добавить элемент SolutionXML (см. рис. 10).

### *Использование Aspose.Diagram for .NET с COM-элементами*

Информация в этом разделе относится к сценариям, в которых разработчикам требуется использовать Aspose.Diagram for .NET via COM-взаимодействие на любом поддерживаемом языке [11].

Aspose.Diagram for .NET выполняется под управлением .NET Framework, и это называется управляемым кодом. Код, написанный на всех языках, работающих за пределами .NET Framework, называется неуправляемым кодом. Взаимодействие между неуправляемым кодом и Aspose.Diagram происходит via средством .NET, называемым COM-взаимодействием.

Объекты Aspose.Diagram — это объекты .NET, но при использовании via COM-взаимодействия они отображаются как COM-объекты в вашем языке программирования. Поэтому лучше всего убедиться, что вы знаете, как создавать и использовать COM-объекты в вашем языке программирования, прежде чем вы начнете использовать Aspose.Diagram for .NET.

```
1.  // The path to the documents directory.
2.  string dataDir = RunExamples.GetDataDir_SolutionXML();
3.
4.  // Load source Visio diagram
5.  Diagram diagram = new Diagram(dataDir + "Drawing1.vsdx");
6.
7.  // Initialize SolutionXML object
8.  SolutionXML solXML = new SolutionXML();
9.  // Set name
10. solXML.Name = "Solution XML";
11. // Set xml value
12. solXML.XmlValue = "XML Value";
13. // Add SolutionXML element
14. diagram.SolutionXMLs.Add(solXML);
15.
16. // Save Visio diagram
17. diagram.Save(dataDir + "AddSolutionXMLElement_out.vsdx", SaveFileFormat.VSDX);
```

Рисунок 10 – Добавить пример программирования элемента SolutionX

- В мире COM мы различаем COM-сервер и COM-клиент. COM-сервер хранит COM-классы, а COM-клиент запрашивает у COM-сервера экземпляры классов, то есть COM-объекты;
- COM-клиент или просто клиентское приложение может что-то знать о содержимом COM-класса или вообще не знать о его методах и свойствах. Поэтому клиентское приложение может обнаружить структуру класса COM при компиляции/сборке или только во время выполнения. Процесс «открытия» известен как связывание, поэтому мы имеем раннее связывание, а также позднее связывание;
- вкратце COM-класс подобен черному ящику и для работы с ним нужна библиотека типов, в этом бинарном файле есть описание методов, свойств класса COM и любой язык высокого уровня, который поддерживает работу с COM-объектами, часто имеет синтаксическое выражение для добавления библиотеки типов, для примера это импорт в C++;
- библиотека типов используется для раннего связывания;
- COM-объект может раскрывать свои методы и свойства двумя способами: с помощью диспетчерский интерфейс (dispinterface) и в его виртуальная таблица (виртуальная таблица функций);
- внутри-интерфейса, каждый метод и свойство идентифицируются уникальным членом; этот член является идентификатором отправки функции (или DispID);
- виртуальная таблица это всего лишь набор указателей на функции, поддерживаемые интерфейсом COM-класса;
- объект, который предоставляет свои методы через оба интерфейса, поддерживает двойной интерфейс;

- преимущества есть у обоих типов переплета: раннее связывание обеспечивает повышенную производительность и проверку синтаксиса во время компиляции; позднее связывание наиболее выгодно, когда вы пишете клиентов, которые вы собираетесь использовать, совместим с будущими версиями вашего COM-класса; при позднем связывании информация из библиотеки типов не “зашита” в ваш клиент, поэтому вы можете быть более уверены, что ваш клиент сможет работать с будущими версиями класса COM без изменения кода; механизм позднего связывания имеет большое преимущество: если создатель COM-DLL решит выпустить новую версию с другим макетом интерфейса функций, любой код, вызывающий эти методы, не рухнет, если только эти методы больше не будут доступны; даже если виртуальная таблица отличается тем, что позднему связыванию удается обнаружить новые DISPID и вызвать соответствующие методы.

### *Вывод*

В предлагаемой статье были рассмотрены следующие вопросы:

1. Кейсы успешного применения УЯП: Примеры компаний, которые внедрили УЯП и достигли значительных улучшений в своих бизнес-процессах.
2. Сравнение с альтернативами: Анализ существующих альтернатив УЯП, их сильные и слабые стороны по сравнению с предложенной концепцией.
3. Влияние на командную работу: Как УЯП может изменить динамику командной работы, улучшить коммуникацию и координацию между различными отделами.
4. Технологические тренды: Обзор современных технологий (например, искусственный интеллект, облачные решения), которые могут быть интегрированы с УЯП для повышения его функциональности.
5. Проблемы внедрения: Обсуждение потенциальных трудностей и барьеров, с которыми могут столкнуться организации при внедрении УЯП, и возможные стратегии их преодоления.
6. Роль обратной связи: как регулярная обратная связь от пользователей может помочь в улучшении и адаптации УЯП к меняющимся потребностям бизнеса.
7. Метрики успеха: Определение ключевых показателей эффективности (KPI), которые помогут оценить успешность внедрения УЯП в организации.
8. Этические аспекты: Обсуждение этических вопросов, связанных с использованием УЯП, таких как защита данных и соблюдение конфиденциальности.
9. Глобальные тренды: Как глобализация и международные рынки влияют на потребность в универсальном языке проектирования.
10. Будущие исследования: Направления для дальнейших исследований и разработок в области УЯП, которые могут открыть новые возможности для бизнеса.

Исходя из проведенного анализа, можно сделать следующие основные выводы:

1. Просматривается внутренний набор языков представления объекта общий для всех компонентов офиса;
2. Фактически все средства автоматизации синтеза или модификации объектов представляют собой некоторый набор шаблонов, отличающихся различным включением в шаблон различного API (на ряду с большой частью API);
3. Разработать более эффективные средства автоматизации для всех компонентов офиса означает разработку более эффективных инструментариев разработки шаблонов, адаптированных на конкретный компонент офиса;
4. Предлагается путь разработки интеллектуальных паттернов.

В качестве вторичных выводов можно отметить, что разработка универсального языка проектирования на базе Microsoft Office Visio открывает перед нами множество значительных возможностей, которые способны значительно улучшить процессы проектирования и визуализации данных в различных областях. Потенциал данного подхода включает в себя следующие ключевые аспекты:

Ключевыми аспектами данной разработки являются:

1. Стандартизация коммуникации: Универсальный язык создаст общую основу для взаимодействия между участниками проекта, что улучшит понимание требований и решений.
2. Адаптивность и гибкость: Такой язык может быть адаптирован под конкретные потребности различных отраслей, будь то инженерия, архитектура или управление бизнес-процессами. Это обеспечит высокую степень универсальности и практическую применимость в широком круге задач.
3. Улучшение визуализации данных: С использованием возможностей Visio по созданию диаграмм и графиков, разработанный язык сможет акцентировать внимание на ключевых аспектах проектируемых систем, облегчая анализ и понимание сложных структур и процессов.
4. Интеграция с другими инструментами: Возможность интеграции на базе Visio с разными программными комплексами и платформами позволит пользователям использовать мощные аналитические инструменты и базы данных, тем самым улучшая качество проектной документации.

5. Повышение качества образования: Применение универсального языка в учебных заведениях может внести значительный вклад в подготовку специалистов, повысив их навыки и обеспечив более глубокое понимание проектирования и визуализации информации.

6. Стимулирование инноваций: Универсальный язык проектирования может послужить основой для новых идей и решений. Это создаст среду, способствующую инновациям, где специалисты смогут использовать язык как средство для разработки нестандартных и эффективных решений.

В итоге, создание универсального языка проектирования с использованием Microsoft Office Visio – это перспективная и актуальная инициатива, которая способна не только оптимизировать процессы проектирования, но и внести свой вклад в развитие технологий и подходов в различных сферах деятельности. Реализация данного проекта потребует комплексных усилий, а также постоянного обновления и адаптации языка к изменяющимся потребностям и требованиям времени.

Перспективой дальнейшей работы является следующие:

1. Разработка стандартов и протоколов: Создание общепринятых стандартов для УЯП, что обеспечит совместимость и унификацию подходов в различных отраслях.

2. Исследование пользовательского опыта: Проведение исследований для понимания потребностей пользователей и выявления лучших практик внедрения УЯП, что позволит адаптировать систему под реальные условия работы.

3. Обучение и сертификация: Разработка обучающих программ и сертификаций для специалистов, работающих с УЯП, что повысит квалификацию кадров и улучшит качество внедрения.

4. Создание экосистемы: Формирование сообщества разработчиков и пользователей УЯП для обмена знаниями, опытом и инновациями, что будет способствовать его развитию.

5. Адаптация к различным отраслям: Исследование специфических потребностей различных отраслей (например, здравоохранение, финансы, производство) для создания адаптированных решений на основе УЯП.

6. Мониторинг и оценка эффективности: Разработка методов мониторинга и оценки эффективности внедрения УЯП в организациях, что позволит выявлять успешные кейсы и делиться ими с другими.

7. Этические и правовые аспекты: Изучение и разработка рекомендаций по соблюдению этических норм и правовых требований при использовании УЯП, особенно в контексте защиты данных.

8. Глобальные инициативы: Участие в международных проектах и инициативах по внедрению УЯП, что позволит обмениваться опытом и находить решения для глобальных вызовов.

9. Инновационные подходы: Исследование новых подходов и методов в области проектирования и моделирования, которые могут дополнить и улучшить УЯП.

### *Литература*


1. Microsoft Visio - URL: https://ru.wikipedia.org/wiki/Microsoft_Visio;
2. Общие сведения о формате файлов Visio (VSDX) - URL: https://learn.microsoft.com/ru-ru/office/client-developer/visio/introduction-to-the-visio-file-formatvsdx (дата обращения: 18.11.2024);
3. Aspose.Diagram for .NET - URL: https://marketplace.visualstudio.com/items?itemName=AsposeMarketplace.AsposeDiagramforNET-10575;
4. Марка Прайса C# 10 и NET 6. Современная кроссплатформенная разработка - URL: https://habr.com/ru/companies/piter/articles/714396/ (дата обращения: 18.11.2024);
5. Ржевский К.В. Описание методов редактирования документа с помощью графических надстроек Microsoft Visio//Материалы VI Международной научно-технической конференции «Современные информационные технологии в образовании и научных исследованиях» (СИТОНИ2019). – Донецк: ДонНТУ, С. 328 – 335 (дата обращения: 18.11.2024);
6. Ржевский, К.В. Обзор возможностей использования встраивого компонента microsoft Visio/ К.В. Ржевский, А.В. Григорьев // Информатика, управляющие системы, математическое и компьютерное моделирование в рамках VI форума «Инновационные перспективы Донбасса» (ИУСМКМ – 2020): XI Международная научно-техническая конференция, 27-28 мая 2020, г. Донецк: / Донец. национал. техн. ун-т; редкол. Ю.К. Орлов и др. – Донецк: ДОННТУ, 2020. – С. 171-175 (дата обращения: 18.11.2024);
7. Культин Н. Microsoft Visual C# в примерах и задачах. — Санкт-Петербург "БХВ-Петербург", 2009. — 320 с. [Электронный ресурс] // Авидредерс – Режим доступа: http://avidreaders.ru/read-book/microsoftvisual-c-v-zadachah-i.html – Загл. с экрана(дата обращения: 18.11.2024);
8. Климов А. П. C#. Советы программистам. — СПб.: БХВ-Петербург, 2008. — 517 с. [Электронный ресурс] // Авидредерс – Режим доступа: http://avidreaders.ru/read-book/c-sovetyprogrammistam.html – Загл. с экрана (дата обращения: 18.11.2024);
9. Петров Ю. И., Шупикова Ю. В., Викулина Д. А., Макаров С. Н. Способы и средства автоматизации текстового процессора Microsoft Word // Перспективы развития информационных технологий.

2012. №8. URL: https://cyberleninka.ru/article/n/sposoby-i-sredstva-avtomatizatsii-tekstovogo-protsessora-microsoft-word (дата обращения: 18.11.2024);

10. Карышев Андрей Анатольевич, Афанасьев Владислав Романович Разработка web-сервиса для автоматизированной генерации документов на основе docx-шаблонов // Известия ТулГУ. Технические науки. 2017. №5. URL: https://cyberleninka.ru/article/n/razrabotka-web-servisa-dlya-avtomatizirovannoy-generatsii-dokumentov-na-osnove-docx- (дата обращения: 18.11.2024);

11. Министерство Образования и Науки Российской Федерации. Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Томский политехнический университет». Институт кибернетики. Технологии Microsoft В Теории И Практике Программирования: Сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, 25–26 марта 2015 г. — Томск, 2015.shablonov (дата обращения: 18.11.2024).

***Ржевский К.В., Григорьев А.В.*** *Анализ задачи разработки универсального языка проектирования. Статья посвящена вопросу расширенных возможностей создания API с возможностью интеллектуальных надстроек. В статье описан пример использования Aspose.Diagram в Visio для создания рабочих графических надстроек Microsoft Visio.*

***Ключевые слова:*** *интеллектуальные паттерны, онтологии, создание API, файл VSDX, библиотека Aspose.Diagram, программа Microsoft Visio.*

***Rzhevsky K.V., Grigoriev A.V.*** *Analysis of the task of developing a universal design language. The article is devoted to the issue of extended capabilities for creating APIs with the ability to create intelligent add-ons. The article describes an example of using Aspose.Diagram in Visio to create working graphical add-ons for Microsoft Visio.*

***Key words****: intelligent patterns, ontologies, creating APIs, VSDX file, Aspose.Diagram library, Microsoft Visio program.*

УДК 004.89

# Интеграция интеллектуальных систем для повышения эффективности Telegram-ботов

**Н. Т. Понамарёв[*1], А. В. Григорьев [*2]**

[*1] магистрант, Донецкий национальный технический университет,
nikitaponamarev6@gmail.com

[*2] к.т.н., доцент кафедры ПИ, Донецкий национальный технический университет,
grigorievalvl@mail.ru, SPIN-код: 8830-281

***Н.Т. Понамарёв, А.В. Григорьев. Интеграция интеллектуальных систем для повышения эффективности Telegram-ботов.*** *Статья посвящена анализу современных подходов к разработке Telegram-ботов, их применению в различных сферах и возможностям интеграции с интеллектуальными системами. Рассматриваются основные направления в развитии Telegram-ботов, включая использование обработки естественного языка (NLP), машинного обучения и мультимодальных взаимодействий. Особое внимание уделено сравнительному анализу способов разработки, таких как программирование с нуля, использование no-code и low-code платформ, а также гибридных решений. Приведены коммерческие примеры успешного внедрения Telegram-ботов в бизнес-процессы, включая автоматизацию поддержки клиентов, проведение транзакций, обработку заказов и рекомендации. Рассматриваются перспективы дальнейшего развития, такие как голосовые интерфейсы, эмоциональный интеллект ботов, интеграция с Интернетом вещей (IoT) и персонализация взаимодействия на основе данных о пользователях.*

***Ключевые слова:*** *Telegram-бот, искусственный интеллект, машинное обучение, автоматизация, no-code платформы, low-code платформы, гибридные решения, голосовые интерфейсы, персонализация, интеллектуальные системы, коммерческие примеры, конфиденциальность данных, этические аспекты.*

## *Введение*

Telegram-боты уже давно используются для различных целей: от автоматизации простых задач до предоставления интеллектуальной поддержки пользователям.

Нехватка ИИ в Telegram-ботах ограничивает их возможности и снижает привлекательность для пользователей. В условиях растущей конкуренции интеграция ИИ становится необходимым условием для повышения эффективности, улучшения пользовательского опыта и достижения бизнес-целей. Инвестиции в интеллектуальные технологии помогут ботам соответствовать современным ожиданиям и оставаться востребованными на рынке [1].

Интеграция методов искусственного интеллекта в технологию разработки Telegram-ботов способна значительно улучшить их функциональность, адаптивность и качество взаимодействия с пользователями.

Цель работы: рассмотреть популярные подходы к разработке Telegram-ботов и примеры их применения, а также проанализировать возможности использования интеллектуальных систем.

## *Основные направления в развитии Telegram-ботов*

Телеграм-боты продолжают развиваться, расширяя функциональность и предлагая новые возможности, такие как интеграция с искусственным интеллектом (ИИ) и мультиканальные решения. Эти тенденции делают их привлекательными для различных коммерческих и личных целей:

- Использование ИИ и обработки естественного языка (NLP)[2]: Системы машинного обучения, такие как обработка естественного языка, стали доступнее и часто интегрируются в ботов, улучшая их способность понимать и обрабатывать пользовательские запросы.

- Автоматизация бизнес-процессов: Боты часто используются для автоматизации заказов, бронирований, поддержки клиентов и маркетинговых кампаний.

- Рост популярности no-code и low-code платформ: No-code платформы позволяют создавать простые боты с минимальными усилиями, что способствует массовому распространению ботов для личного и делового использования.

AI-чат-боты применяют алгоритмы машинного обучения для повышения качества своих ответов. В процессе работы они обрабатывают данные, полученные во время взаимодействий с пользователями, и корректируют свои модели поведения. Для обучения таких систем используются различные подходы:

• Обучение с учителем: чат-бот обучается на заранее размеченных данных, где указаны примеры запросов и соответствующих ответов. На их основе он учится формировать корректные ответы.

• Обучение без учителя: бот анализирует большие объемы неструктурированных данных и самостоятельно находит закономерности. Этот метод позволяет чат-ботам совершенствоваться даже без прямых инструкций от разработчиков.

### *Основные способы разработки Telegram-ботов*

Разработка Telegram-ботов может осуществляться с помощью нескольких подходов: программирование с нуля, no-code платформы и гибридные low-code решения. Эти подходы позволяют адаптировать ботов под разные задачи и коммерческие цели.

### *Классический подход с использованием кода*

Этот способ предоставляет максимальную гибкость и позволяет создавать ботов с любыми функциями, используя библиотеки и API для взаимодействия с Telegram. Разработка с нуля подходит для тех, кто обладает навыками программирования и хочет максимального контроля [3].

Пример: Бот для интернет-магазина, написанный на Python, использующий библиотеку python-telegram-bot. Бот принимает заказы, отправляет пользователям уведомления о доставке и поддерживает базу данных с заказами.

Преимущества:

- Полный контроль над функциональностью и архитектурой.
- Поддержка всех функций Telegram API, таких как работа с файлами, обработка inline-запросов и webhook.
- Возможность интеграции с внешними системами, базами данных и API.

### *No-code платформы*

No-code платформы позволяют создавать ботов без написания кода, предоставляя простой интерфейс с визуальными элементами. Это идеальное решение для быстрого создания ботов с базовым функционалом, подходящее для малого бизнеса и личного использования.

Пример: Бот для кафе, созданный на платформе Chatfuel [4]. Бот принимает заказы, предоставляет меню, отвечает на частые вопросы и предлагает пользователям акции. Сценарий можно легко изменить с помощью редактора Chatfuel, добавляя сообщения и кнопки для взаимодействия с пользователями. Встроенные блоки дают возможность добавлять текстовые, графические и мультимедийные сообщения без программирования.

Преимущества:

- Простота и быстрота в настройке.
- Возможность добавления форм для сбора данных, создания цепочек сообщений и отслеживания аналитики.
- Не требуется опыт программирования, что позволяет быстро создать бота для бизнеса.

### *Low-code платформы и гибридные фреймворки*

Low-code платформы позволяют создавать более сложные сценарии и логику работы бота, сочетая визуальные элементы с возможностью написания кода. Эти решения предоставляют гибкость и мощные инструменты для работы с API, что позволяет создавать интеллектуальные и функциональные боты.

Пример: Бот для службы поддержки клиентов, разработанный с использованием Dialogflow[5]. Бот способен понимать естественный язык и выполнять задачи по категориям запросов. С помощью Dialogflow бот распознает и обрабатывает запросы пользователей, перенаправляет их в соответствующие службы поддержки или предоставляет предварительно подготовленные ответы.

Преимущества:

- Гибкость при разработке сложных сценариев.
- Возможность интеграции с API и базами данных.
- Обработка естественного языка для более «умного» взаимодействия с пользователями.

### *Интеллектуальные системы и их интеграция в структуру Telegram-ботов*

Одним из важнейших факторов в развитии современных Telegram-ботов стала интеграция с ИИ и обработкой естественного языка [6]. Это позволяет создавать интеллектуальных помощников, способных понимать запросы пользователя, распознавать речь и адаптироваться к конкретным ситуациям.

Возможности интеграции ИИ:

1) Обработка естественного языка (NLP): Системы, такие как Dialogflow от Google, Microsoft Bot Framework и Rasa, позволяют боту понимать естественный язык и адаптироваться к различным вопросам и задачам.

Пример: Бот для службы поддержки, разработанный на базе Dialogflow и интегрированный с Telegram, способен обрабатывать запросы на естественном языке, выявлять намерения и предлагать варианты решения или перенаправлять к оператору.

2) Машинное обучение и самообучение: Такие боты могут обучаться на основе взаимодействий с пользователями. Это позволяет улучшать ответы бота и адаптировать его поведение к реальным нуждам.

Пример: Чат-бот для туристического агентства, использующий модель машинного обучения для анализа отзывов клиентов, предложений и запросов. На основе этих данных бот может предлагать наиболее подходящие туры и персонализировать рекомендации.

3) Распознавание изображений: Современные боты также могут использовать компьютерное зрение для анализа изображений, распознавания текста и объектов.

Пример: Бот для магазина, который принимает фотографии товаров и предоставляет информацию о похожих продуктах, их наличии и стоимости.

Примеры использования интеллектуальных ботов:

- Боты для поддержки клиентов: Интеллектуальные боты помогают обрабатывать запросы пользователей, предоставлять справочную информацию, предлагать решения по типичным вопросам и перенаправлять сложные запросы к реальным операторам.
- Рекрутинговые боты: С помощью NLP и анализа данных такие боты могут проводить первичное собеседование, задавая вопросы кандидатам и анализируя их ответы.
- Боты для образовательных целей: ИИ позволяет таким ботам анализировать успеваемость студентов, рекомендовать учебные материалы и оценивать выполненные задания.

### *Сравнительный анализ способов разработки*

Для сравнения подходов к созданию Telegram-ботов можно выделить следующие критерии: гибкость, время разработки, требуемый уровень знаний, стоимость, поддержка интеграций и производительность.

Таблица 1 – Сравнительная таблица способов разработки

| Подход | Гибкость | Время разработки | Уровень знаний | Стоимость | Поддержка интеграций | Производительность |
|---|---|---|---|---|---|---|
| Кодовый подход | Высокая | Длительное | Высокий | Зависит от хостинга | Полная | Высокая |
| No-code платформы | Низкая | Быстрое | Низкий | Часто бесплатно, но есть платные тарифы | Ограниченная | Средняя |
| Low-code платформы | Средняя | Умеренное | Средний | Зависит от платформы | Хорошая | Производительность |
| Интеллектуальные решения | Очень высокая | Умеренное/Длительное | Высокий | Высокая | Полная | Высокая |

### *Коммерческие примеры реализации Telegram-ботов*

Чат-боты уже давно стали важным инструментом ведения бизнеса в современном мире, поскольку позволяют автоматизировать процессы, экономить человеческие ресурсы и средства компании [7]. Однако простыее чат-боты, основанные на сценарном подходе, часто становятся вредными для бизнеса, поскольку усложняют процесс поиска информации для клиента. Человеку должен пройти длинный алгоритм, прежде чем он сможет получить необходимую информацию. Поэтому в наши дни возникает потребность в интеллектуальных чат-ботах. Самый эффективный подход к проектированию строится на основе векторного представления текста и технологий word2vec и doc2vec. Существует множество чат-ботов, каждый из которых обеспечивает поддержку

определенной сферы. Однако универсального решения нет, либо оно дорогостоящее, поскольку требует переучивания модели нейронной сети, что довольно небыстрый и энергозатратный процесс.

Рассмотрим несколько успешных коммерческих примеров Telegram-ботов, которые помогли компаниям улучшить сервис, повысить лояльность и увеличить продажи.

### *AliExpress Telegram Bot*

Задача: Улучшить клиентский опыт, упростив отслеживание заказов и предоставление информации о скидках и акциях.

Решение: AliExpress создал бота, который уведомляет пользователей о статусе их заказов и предлагает эксклюзивные акции. Бот интегрирован с системой управления заказами, что позволяет в реальном времени предоставлять данные.

Результаты: Бот повысил вовлеченность клиентов и уменьшил нагрузку на службу поддержки.Перспективой данной работы является разработка технологии применения ФД для построения игр различных жанров, стилистики, целевой аудитории и т.д.

### *Бот «Тинькофф Банк» для поддержки клиентов*

Задача: Обеспечить круглосуточную поддержку клиентов, отвечать на вопросы и помогать в решении финансовых вопросов.

Решение: Банк создал бота, использующего ИИ и обработку естественного языка для ответа на типичные вопросы о банковских продуктах, услугах и условиях. Бот также предлагает клиентам опцию «живого чата» с оператором при необходимости.

Результаты: Бот помог снизить нагрузку на колл-центр и улучшить оперативность обслуживания клиентов, особенно в ночное время.

### *Бот «Райффайзенбанк» для поддержки клиентов и проведения транзакций*

Задача: Обеспечить удобное и быстрое обслуживание клиентов, помочь в управлении финансами через Telegram.

Решение: Райффайзенбанк разработал бот, который не только консультирует клиентов, но и позволяет выполнять простые операции (например, переводить средства) прямо через мессенджер.

Результаты: Бот увеличил лояльность клиентов и привлек новую аудиторию за счет удобного и быстрого обслуживания.

### *Бот «Яндекс.Еда» для заказа еды*

Задача: Автоматизировать заказы на доставку еды, поддерживать взаимодействие с клиентами.

Решение: Бот предоставляет клиентам меню, принимает заказы и сообщает об их статусе. Он интегрирован с системой доставки, что позволяет своевременно обновлять информацию.

Результаты: Бот улучшил удобство для клиентов и сократил время ожидания на оформление заказов.

### *Бот от агентства недвижимости «Циан»*

Задача: Помочь пользователям с поиском недвижимости и дать ответы на популярные вопросы.

Решение: Бот интегрирован с базой данных и системой рекомендаций, что позволяет находить объекты по заданным параметрам, уведомлять о новых предложениях и предоставлять клиентам информацию об актуальных объектах.

Результаты: Бот ускорил процесс поиска недвижимости для клиентов и облегчил задачу менеджерам агентства.

### *Бот для страховой компании «Росгосстрах»*

Задача: Упростить процесс покупки и продления полисов, снизить нагрузку на сотрудников.

Решение: Бот предоставляет клиентам информацию о полисах, рассчитывает стоимость и даже помогает оформлять заявки на покупку или продление полиса.

Результаты: Бот увеличил количество самообслуживания клиентов и сократил количество обращений в контакт-центр, что снизило затраты компании на обслуживание клиентов.

### *Перспективы развития Telegramm-ботов*

Интеграция искусственного интеллекта и новые технологии продолжают трансформировать Telegram-ботов, открывая множество перспектив:

#### *Голосовые интерфейсы и мультимодальные взаимодействия:*

1) Распознавание и синтез речи позволят пользователям взаимодействовать с ботами голосом. Это особенно полезно для улучшения доступности, например, для людей с ограниченными возможностями.
2) Пример: голосовые помощники для медицинских консультаций, которые понимают симптомы пациента на основе описаний.

#### *Совместное использование с Интернетом вещей (IoT):*

1) Боты могут управлять устройствами умного дома, автомобилями или промышленным оборудованием.
2) Пример: бот, интегрированный с системой умного дома, который включает свет, устанавливает температуру в помещении или открывает замки.

#### *Эмоциональный интеллект в ботах:*

1) Технологии распознавания эмоций и анализа тона сообщений (Sentiment Analysis) помогут ботам лучше адаптировать ответы.
2) Пример: бот психологической помощи, который подбирает ответы в зависимости от эмоционального состояния пользователя.

#### *Глубокая персонализация и контекстуальность:*

1) Использование истории взаимодействий и предпочтений для улучшения рекомендаций.
2) Пример: бот интернет-магазина, который предлагает товары на основе истории покупок и анализа поведения пользователя.

#### *Гибридные решения с использованием облачных и локальных вычислений:*

1) Для обеспечения конфиденциальности часть данных может обрабатываться локально, а сложные вычисления — в облаке.
2) Пример: корпоративный бот, который защищает данные клиентов, обрабатывая персональную информацию на месте.

### *Выводы*

В данной работе были рассмотрены наиболее популярные подходы к разработке Telegram-ботов, приведены коммерческие примеры их применения, а также были изучены возможности внедрения в них интеллектуальных систем с целью расширения функционала. Современные Telegram-боты обладают широкими возможностями, которые поддерживаются как простыми no-code решениями, так и интеллектуальными системами на базе ИИ. Выбор подхода зависит от специфики задач, ресурсов команды и нужного уровня автоматизации и гибкости. Интеграция интеллектуальных технологий, таких как NLP, машинное обучение и компьютерное зрение, делает ботов более мощными и позволяет им выполнять задачи, недоступные для обычных чат-ботов.

Коммерческие примеры показывают, что компании могут использовать ботов для множества задач: от обслуживания клиентов и обработки заказов до проведения финансовых операций и анализа предпочтений пользователей. Малый бизнес часто использует no-code и low-code платформы для быстрого создания ботов, которые помогают отвечать на вопросы, предоставлять информацию о товарах и услугах, принимать заказы и собирать отзывы. Средний и крупный бизнес выбирает интеллектуальные решения с обработкой естественного языка, машинным обучением и интеграцией с CRM и ERP-системами для повышения эффективности и персонализации обслуживания.

Перспективными задачами для дальнейшего исследования является разработка путей применения методов ИИ для повышения эффективности функционирования ботов и практическая программная реализации бота.

### *Литература*


1. Телеграм-боты с искусственным интеллектом: полное руководство по возможностям и внедрению в бизнес – URL: https://ridis.ru/blog/telegram-botyi-s-iskusstvennyim-intellektom-polnoe-rukovodstvo-po-vozmozhnostyam-i-vnedreniyu-v-biznes/ (дата обращения(10.11.2024)
2. 15 нейросетей в один Telegram-бот: история успеха и реализация помощника для создателей контента - URL: https://habr.com/ru/articles/690922/ (дата обращения: 10.11.2024)
3. Python telegram bot – URL: https://python-telegram-bot.org/ (дата обращения: 10.11.2024)
4. Как сделать своего чат-бота – URL: https://setters.education/blog/articles/kak-sdelat-svoego-chat-bota-obzor-treh-servisov (дата обращения: (11.11.2024)
5. Что умеет Dialogflow? – URL: https://habr.com/ru/articles/502688/ (дата обращения: 12.11.2024)
6. Создание чат-бота Telegram на базе искусственного интеллекта Gemini – URL: https://translated.turbopages.org/proxy_u/en-ru.ru.f3a7dcb1-673ce100-9d177855-74722d776562/https/www.geeksforgeeks.org/creating-a-telegram-chatbot-powered-by-gemini-ai/ (дата обращения 12.11.2024)
7. Чат-бот с использованием технологий нейронных сетей и методов обработки текста для повышения лояльности клиентов – URL: https://moitvivt.ru/ru/journal/pdf?id=1110 (дата обращения (13.11.2024)



***Н.Т. Понамарёв, А.В. Григорьев. Интеграция интеллектуальных систем для повышения эффективности Telegram-ботов.*** *Статья посвящена анализу современных подходов к разработке Telegram-ботов, их применению в различных сферах и возможностям интеграции с интеллектуальными системами. Рассматриваются основные направления в развитии Telegram-ботов, включая использование обработки естественного языка (NLP), машинного обучения и мультимодальных взаимодействий. Особое внимание уделено сравнительному анализу способов разработки, таких как программирование с нуля, использование no-code и low-code платформ, а также гибридных решений. Приведены коммерческие примеры успешного внедрения Telegram-ботов в бизнес-процессы, включая автоматизацию поддержки клиентов, проведение транзакций, обработку заказов и рекомендации. Рассматриваются перспективы дальнейшего развития, такие как голосовые интерфейсы, эмоциональный интеллект ботов, интеграция с Интернетом вещей (IoT) и персонализация взаимодействия на основе данных о пользователях.*



***Ключевые слова:*** *Telegram-бот, искусственный интеллект, машинное обучение, автоматизация, no-code платформы, low-code платформы, гибридные решения, голосовые интерфейсы, персонализация, интеллектуальные системы, коммерческие примеры, конфиденциальность данных, этические аспекты.*



***N.T. Ponamaryov, A.V. Grigoryev. Integration of Intelligent Systems to Enhance the Efficiency of Telegram Bots.*** *This article is dedicated to analyzing modern approaches to the development of Telegram bots, their applications across various domains, and the potential for integrating intelligent systems. It explores the main directions in the evolution of Telegram bots, including the use of natural language processing (NLP), machine learning, and multimodal interactions. Special attention is given to a comparative analysis of development methods, such as programming from scratch, the use of no-code and low-code platforms, and hybrid solutions. Commercial examples of successful integration of Telegram bots into business processes are provided, including customer support automation, transaction handling, order processing, and recommendations. The article also discusses prospects for further development, such as voice interfaces, bots with emotional intelligence, integration with the Internet of Things (IoT), and interaction personalization based on user data.*



***Keywords:*** *Telegram bot, artificial intelligence, machine learning, automation, no-code platforms, low-code platforms, hybrid solutions, voice interfaces, personalization, intelligent systems, commercial examples, data privacy, ethical aspects.*

УДК 681.5

# Виды логистических потоков предприятия

**А.И. Боровиков[*1], О.А. Криводубский[*2]**

[*1] аспирант, Донецкий национальный технический университет,
aleksey.borovikov.00@mail.ru
*2 д.т.н., проф. кафедры программной инженерии им. Л.П. Фельдмана, Донецкий национальный технический университет,
oleg.krivodubski.dn@gmail.com

***Боровиков А.И., Криводубский О.А. Виды логистических потоков предприятия.*** *В статье рассматривается управление ресурсами предприятия через логистический подход, основанный на работе с основными потоками: материальными, финансовыми и информационными. Приведены определения и классификации потоков, которые позволяют структурировать их и эффективно управлять ими. Уточняется значимость материальных потоков в логистических процессах, подчеркивается важность управления финансовыми потоками для обеспечения устойчивости предприятия и акцентируется роль информационных потоков в поддержке процессов принятия решений.*

***Ключевые слова:*** *логистика, материальные потоки, финансовые потоки, информационные потоки, управление ресурсами, классификация потоков, оптимизация процессов, система принятия решений, эффективность предприятия, конкурентоспособность*

## *Введение*

Деятельность любого предприятия основана на управлении ресурсами этого предприятия. Такие ресурсы (под ресурсами подразумевается материалов, компонент, определяющий деятельность и экономические показатели, определяющие их переработку) делятся на разные типы и также могут называться потоками. Современное предприятие ежедневно принимает огромное количество решений по управлению такими потоками, а скорость и качество принятия таких решений определяет успех предприятия или компании. Для повышения эффективности принятия решений требуется взвешенный подход и анализ ситуации. Здесь приходит на помощь логистика, позволяющая предприятию увеличить прибыль и сделать конкурентоспособными товары и услуги, используя материалопроводящие системы. Ключевой целью логистики является эффективное управление ресурсами и потоками предприятия [1].

Стоит отметить, что основное отличие логистического подхода от традиционного заключается в том, что управление осуществляется сквозным материальным потоком, а не потоками, разделёнными по функциональным областям логистики, или даже по логистическим операциям. Тем не менее, на практике логистический менеджмент часто сталкивается с необходимостью управлять разделёнными по логистическим процессам потоками различной природы: материальными, финансовыми, информационными [2].

Разделив потоки, необходимо собрать достаточное количество информации о каждом из них для эффективного анализа и управления. Таким образом, информация из вспомогательного инструмента превращается в один из важнейших ресурсов для эффективного функционирования предприятия. Однако, сбор информации часто оказывается огромной проблемой для компании, так как чем больше компания, тем больше информации ей нужно собирать, хранить и анализировать. Оптимизация информационных потоков позволит улучшить качество информации для дальнейшего анализа.

Современные предприятия функционируют в условиях высокой конкуренции и стремительно меняющихся внешних факторов, что требует от них умения быстро адаптироваться и принимать взвешенные управленческие решения. Одной из ключевых задач является организация внутренних процессов таким образом, чтобы обеспечить их максимальную эффективность. Для этого необходимо учитывать множество факторов, таких как доступность ресурсов, их рациональное распределение и координация действий между различными подразделениями предприятия.

Управление потоками в компании связано с решением большого числа взаимосвязанных задач, включая планирование, сбор данных и контроль их исполнения. Однако в условиях увеличения масштабов деятельности

компаний возрастает сложность таких процессов. Это приводит к необходимости разработки подходов, которые позволят систематизировать управление и минимизировать возможные риски. В современных условиях значительное внимание уделяется внедрению инновационных технологий, которые помогают автоматизировать процессы управления и повысить точность анализа. Однако сами по себе технологии не гарантируют успеха: требуется также глубокое понимание внутренней структуры предприятия и особенностей протекающих в нём процессов.

Таким образом, необходимость комплексного анализа управления ресурсами и потоками становится одной из актуальных задач, стоящих перед любым предприятием. В настоящей работе рассматривается важность этого подхода как основы для повышения эффективности управления и обеспечения устойчивости деятельности компании в долгосрочной перспективе.

### *Виды потоков*

Под потоком понимают направленное движение совокупности чего-либо условно однородного (например, продукции, информации, финансов, материалов, сырья и т.п.). Потоки представляют собой одни или множество объектов, воспринимаемых как единое целое, существующие, как процесс на определённом временном интервале и измеряемые в абсолютных единицах.

Первый основной вид потов - материальные. Основные потоки от неосновных отличаются своей значимостью в логистической цепочке. Материальный поток — находящиеся в состоянии движения материальные ресурсы, незавершённое производство и готовая продукция, к которым применяются логистические операции, связанные с их физическим перемещением в пространстве: погрузка, разгрузка, затаривание, перевозка, сортировка, консолидация, разукрупнение, и т.п. [3]. С их помощью осуществляется передача и обмен материальными ресурсами между различными объектами.

Материальные потоки - одни из главных видов потоков, которые играют огромную роль в современных производственных и экономических процессах. Они являются центральным элементом логистической системы, так как связаны непосредственно с основным продуктом или услугой, которые создают добавленную стоимость для компании.

Классификация материальных потоков позволяет систематизировать и описать их характеристики, а также провести анализ. Общая классификация материальных потоков:

1. по направлению перемещения:
   - входящие потоки – перемещение материальных ресурсов от поставщиков к организации;
   - исходящие потоки – перемещение материальных ресурсов от организации к клиентам;
   - внутренние потоки – перемещение материальных ресурсов внутри организации.
2. по характеру материала:
   - сырьевые потоки – перемещение сырья и материалов, которые не были подвергнуты обработке;
   - производственные потоки – перемещение материалов и полуфабрикатов внутри предприятия в процессе производства;
   - готовые продукты – перемещение готовых товаров от производителя к потребителю.
3. по методу перемещения:
   - потоки с прямым током – перемещение материальных ресурсов по прямой траектории, от поставщика к клиенту.
   - потоки с обратным током – перемещение материальных ресурсов по обратной траектории, например, при возврате товара клиентом;

Это лишь общая классификация материальных потоков, которая помогает организации структурировать и анализировать процессы перемещения материальных ресурсов и эффективно управлять ими [4].

Материальные потоки играют ключевую роль в обеспечении эффективной работы предприятия и его логистической системы. Их правильное управление позволяет оптимизировать затраты, сократить время выполнения заказов и повысить уровень обслуживания клиентов. Эффективное управление материальными потоками включает в себя планирование, организацию и контроль всех процессов, связанных с перемещением ресурсов. Это позволяет не только минимизировать издержки, но и улучшить качество продукции, так как на каждом этапе движения материалов можно выявлять и устранять возможные проблемы.

Важным аспектом управления материальными потоками является их классификация, которая помогает систематизировать процессы и выявить узкие места в логистической цепочке. Например, входящие потоки могут быть проанализированы для выявления наиболее надежных поставщиков, а исходящие потоки — для оптимизации маршрутов доставки. Внутренние потоки, в свою очередь, требуют внимания к организации производственных процессов, чтобы избежать простоев и повысить общую эффективность. Современные предприятия все чаще интегрируют управление материальными потоками с информационными потоками. Это

позволяет не только отслеживать физическое перемещение товаров, но и управлять данными о запасах, заказах и потребностях клиентов в реальном времени. Использование современных технологий, таких как системы управления ресурсами предприятия (ERP), помогает объединить все аспекты управления потоками в единую систему, что способствует улучшению координации между различными подразделениями.

Не менее важным является влияние материальных потоков на производственные процессы. Неправильная организация потоков может привести к простоям, снижению производительности и увеличению времени цикла производства. Поэтому важно учитывать такие факторы, как скорость потока и качество материалов. Оптимизация времени перемещения материалов между различными этапами производства позволяет значительно повысить общую эффективность.

Следующий основной вид потоков - финансовые. Единого определения финансовых потоков не существует, поэтому рассмотрим 2 определения. Глобальное определение финансового потока по Оксфордскому экономическому словарю -- обмен денежной стоимостью между двумя экономиками или между компонентами внутри экономики, включая торговлю, инвестиции и помощь. Определение финансового потока на предприятии сформулированное[5] В.В. Бочаровым -- объем денежных средств, который получает или выплачивает предприятие в течение отчётного или планируемого периода.

Однако несмотря на существование различных подходов, все авторы едины в определении сущностных характеристик финансового потока, рассматривая его в качестве:

1. движения (мобилизации и распределения) финансовых ресурсов;
2. платежа фискальной системе государства, служащего источником формирования централизованных и децентрализованных государственных фондов;
3. инструмента управления производственными процессами, поскольку контроль и своевременная корректировка направления движения финансовых потоков оказывают определяющее влияние на обеспечение отраслей и секторов экономики финансовыми ресурсами.

Понятие «финансовый поток» применительно к организациям (предприятиям) появилось сравнительно недавно в процессе развития финансовых отношений на всех стадиях современного общественного воспроизводства. При этом следует отметить, что в рыночной экономике эффективное управление финансовыми потоками организаций приобретает особое значение.

Целью управления финансовыми потоками является обеспечение финансовыми ресурсами финансово-хозяйственной деятельности организации, что позволит обеспечить необходимой финансовой устойчивостью предприятие на планируемый период. Выделим основные характеристики финансовых потоков, которые помогут улучшить анализ и управление ими:

1. финансовые потоки обусловливают изменение объёма, состава, размещения и использования финансовых ресурсов;
2. каждый финансовый поток имеет свой источник возникновения и направление движения;
3. понятие стоимости финансовых ресурсов основывается на их специфике в качестве «товара»; стоимостью финансовых ресурсов является минимальная альтернативная доходность вложения средств, сложившаяся на финансовом рынке на определённый период времени;
4. способность финансовых потоков генерировать экономический эффект, основной формой которого выступают «чистые потоки фондов»;
5. финансовый поток не всегда определяет реальное движение денежных средств и их эквивалентов;
6. изменение финансового состояния организации за определённый период является результатом воздействия финансовых потоков.

Таким образом, в основе финансовых потоков организации прежде всего лежит движение финансовых ресурсов. В экономической литературе и практике понятие «финансовые ресурсы» используется широко, однако его толкование неоднозначно. В энциклопедии «Политическая экономия» финансовые ресурсы рассматриваются как совокупность ресурсов на государственном уровне: «…Это составная часть экономических ресурсов, представляющая собой средства денежно-кредитной и бюджетной системы, которые используются для обеспечения бесперебойного функционирования и развития народного хозяйства...».

В экономической литературе рассматриваются около двух десятков различных признаков классификации финансовых потоков [6]. Основными из них, характеризующими их экономическую сущность, по нашему мнению, являются:

1. взаимосвязь финансовых потоков с бухгалтерским балансом (отчётом);
2. вид хозяйственной деятельности, генерирующий финансовые потоки;
3. направленность движения финансовых потоков;
4. достаточность и сбалансированность финансовых потоков.

Исходя из того, что показатели, отражающие наличие финансового потока, связаны с изменениями в статьях актива (имущества) и пассива (обязательств) баланса, можно поставить под сомнение традиционное

представление об источниках финансовых ресурсов как о средствах, отражённых в пассиве балансового отчёта. Разделение финансовых потоков на «активные», «пассивные» и «активно-пассивные» основывается именно на подходе к источникам финансовых ресурсов и их применению. «Активные» потоки приводят к изменениям только в составе имущества организации; «пассивные» — в источниках финансирования; а «активно-пассивные» — либо к росту, либо к уменьшению как имущества, так и источников его формирования. Эта классификация финансовых потоков играет ключевую роль, так как от грамотного управления их динамикой и направлением зависит финансовая устойчивость компании.

Информационные потоки являются заключающим видом основных потоков. Они представляют собой информацию, находящуюся в упорядоченном движении по заданным направлениям с фиксированными начальными, промежуточными и конечными точками.

Ежедневно каждое предприятие генерирует определённое количество данных, которые могут быть утеряны без надлежащей организации процессов, или же компания может настроить процессы так, чтобы информация сохранялась, накапливалась, структурировалась и впоследствии могла быть проанализирована. Информация, проходящая через такие процессы, формирует информационные потоки, и таких потоков может быть множество, в зависимости от структуры предприятия. Для того чтобы выявить факторы нарастания генерируемых и структурированных объёмов информации на предприятии, следует обобщить информацию. Обобщение поможет выделить классификацию и характеристики.

Информационные потоки могут классифицироваться в зависимости от принципов построения информационной системы, которая реализует эти потоки, и определяются видом индикации, однородностью, периодичностью, степенью взаимосвязей, объёмом и другими характеристиками передаваемой информации [7]. Классификация информационных потоков:

1. по отношению к логистической системе:
   - внутренние;
   - внешние;
   - горизонтальные;
   - вертикальные;
2. по назначению информации:
   - директивные;
   - нормативно-справочные;
   - учетно-аналитические;
   - вспомогательные;
3. по времени возникновения:
   - регулярные;
   - переодические;
4. по периодичности использования:
   - оперативные;
   - моментальные;
   - отложенные;
5. по степени открытости и уровню значимости:
   - открытые: коммерческие, простые;
   - закрытые: секретные, заказные;
6. по виду носителей информации:
   - на бумажных носителях;
   - на магнитных носителях;
   - электронные;
   - прочие;
7. по способу передачи данных:
   - курьером, почтой;
   - по телефону;
   - по радио, телевиденью;
   - электронной почтой;
   - по телекоммуникационным сетям.

Всю передающуюся информацию необходимо проанализировать, выделить самое главное – определить её характеристику. Динамической характеристикой системы, которой является предприятие, можно считать численность работающих на предприятии, а для информационных потоков - самих людей, поскольку именно они являются источниками восприятия и передачи информации.

Ещё одной важной характеристикой может быть время, поскольку оно отвечает за скорость передачи информации или актуальность. Одним из представлений о времени является представление времени в виде

спирали. Если рассматривать развивающуюся систему, то она также развивается по спирали. При прохождении каждого витка спирали в результате развития в системе происходит накопление информации. Эта информация накапливается в виде свободной информации. Спиралевидное развитие подразумевает, что на следующем витке развития повторяются некоторые черты, свойственные предыдущим виткам, но в целом все характеристики претерпевают качественные изменения.

### *Информация как экономический ресурс*

В течение длительного периода времени информация как экономический ресурс выносилась за пределы исследований в экономической теории. Только благодаря информатизации, увеличению значимости информационных факторов в экономике и появлению возможности использовать информацию как экономический ресурс, в последние десятилетия учёные начали уделять внимание изучению информации в рамках экономической деятельности. Это изменение стало возможным благодаря стремительному развитию технологий, которые кардинально изменили способы сбора, обработки и передачи информации. В условиях глобализации и цифровизации экономики информация приобрела особую ценность, став одним из ключевых факторов успеха для современных организаций.

Классическая экономическая теория исходила из предположения, что каждый хозяйствующий субъект обладает полной информацией о рыночной ситуации, на основе которой он принимает рациональные решения. Следовательно, информация не рассматривалась в классической теории в качестве экономического ресурса, признаком которого является ограниченность. Под информацией в данном подходе понимались различные сведения и знания о рынке. Однако данное предположение не учитывало реальную сложность и динамичность рыночной среды, где информация часто бывает неполной или асимметричной. Это приводит к необходимости переосмысления роли информации в принятии решений и управления рисками.

За последние два десятилетия роль информации в экономических отношениях существенно увеличилась. Современные технологии настолько сложны, что для их организации нужны колоссальные объёмы данных. Без информационных потоков не могут существовать бизнес-процессы в организации. Вся человеческая жизнь основана на обмене знаниями и опытом с другими людьми. Все большее значение приобретает информационное богатство. Информация порождает новые рыночные ниши, расширяет совокупное предложение и спрос, резко повышает ёмкость рынка. Оценка стоимости информации как экономического блага становится важной, но одновременно и сложной задачей для многих экономических субъектов.

Возвращаясь к логистическим потокам, стоит отметить, что сравнение и оценка действительного и планируемого состояний логистической системы даёт информацию, необходимую для выработки управленческих решений, ведущих к полезному результату процесса управления. В процессе управления выделяются следующие этапы: анализ ситуации, принятие решения, его реализация и контроль над исполнением. Из этого списка этапов управления наиболее ответственным является этап принятия решений. Причём решение, как правило, приходится принимать в условиях альтернативности и частичной неопределённости. Как следствие, частичная неопределённость переносится и на результаты логистических решений. Кроме того, на них накладываются недетерминированные воздействия внешней среды предприятия.

В концепции рассмотрения интегрального логистического потока как суммы стоимостей всех его трёх составляющих (материальной, финансовой и информационной) стоимостные вклады потоков различной природы в итоговый интегральный поток могут быть принципиально равноценными и «передавать» свою стоимость полностью или частично. Например, заказ на поставку (информационный поток) в рамках продажи превращается в реальную поставку (материальный поток), которая, в свою очередь, после оплаты трансформируется в поток денежных средств (финансовый поток). В соответствии с этой концепцией стоимостная оценка информационного потока равна стоимости поставки.

Таким образом, информация становится неотъемлемой частью всех бизнес-процессов. Она не только влияет на внутренние операции компаний, но также определяет их конкурентоспособность на рынке. В условиях быстро меняющейся деловой среды способность эффективно управлять информационными потоками становится критически важной для достижения успеха. Существуют различные подходы к обработке и анализу данных, которые помогают организациям адаптироваться к изменениям рынка и принимать обоснованные решения [8-9].

### *Заключение*

В ходе проведенного анализа рассмотрены основные виды потоков предприятия: материальные, финансовые и информационные. Определены их ключевые характеристики, сущностные особенности и предложена общая классификация.

Материальные потоки охватывают перемещение ресурсов, продукции и сырья, являясь центральным элементом логистической системы. Финансовые потоки связаны с движением денежных средств, обеспечивая финансовую устойчивость предприятия и являясь важным инструментом управления. Информационные потоки,

выступая связующим звеном, формируют основу для принятия решений, поддерживая взаимодействие между всеми элементами логистической системы. Особое внимание уделено информационным потокам, так как их эффективное управление оказывает значительное влияние на процессы анализа и оптимизации всех потоков предприятия. Проблемы сбора, хранения и анализа информации становятся ключевыми вызовами для компаний, особенно в условиях роста объёмов данных. В рамках концепции интегрального логистического потока установлено, что материальные, финансовые и информационные потоки взаимосвязаны, а их стоимостные оценки могут быть равноценными в зависимости от задач и этапов функционирования предприятия.

Итогом работы станет разработка специального математического программного обеспечения, направленного на поддержку принятия решений для управления потоками предприятия. Такое решение позволит автоматизировать анализ потоков, повысить оперативность и точность управленческих действий, что, в свою очередь, укрепит конкурентоспособность компании.

### *Литература*


1. Акаева, В. Р. Логистика: учебник / В. Р. Акаева. — Москва: 2022. — 327 с. — ISBN 978-5-406-10136-0.
2. Грейз Георгий Маркович, Кузменко Юлия Геннадьевна, Хатеев Игорь Валерьевич Анализ концепций взаимодействия основных логистических потоков // Российское предпринимательство. 2013. №5 (227). URL: https://cyberleninka.ru/article/n/analiz-kontseptsiy-vzaimodeystviya-osnovnyh-logisticheskih-potokov.
3. Материальный поток [Электронный ресурс] / GRANDARS : [Сайт]. – URL: https://www.grandars.ru/college/logistika/materialnyy-potok.html.
4. Материальный поток в логистике - https://wiki.fenix.help/logistika/materialnyj-potok. - удалена
5. Рудакова Т.А., Цыбенко Е.Н. ФИНАНСОВЫЕ ПОТОКИ ОРГАНИЗАЦИИ: ЦИФРОВОЕ СОПРОВОЖДЕНИЕ И ЭЛЕКТРОННОЕ УПРАВЛЕНИЕ // Управление современной организацией. 2023. №18. URL: http://journal.asu.ru/mmo/article/view/14672/12472.
6. Тютюкина Е. Б., Афашагов К. М. Финансовые потоки: сущность и признаки классификации // Финансы: теория и практика. 2007. №4. URL: https://cyberleninka.ru/article/n/finansovye-potoki-suschnost-i-priznaki-klassifikatsii.
7. Гайсарова А.А., Ветрова Н.М. Особенности процесса управления информационными потоками на предприятии в современных условиях // Экономика строительства и природопользования. 2019. №1 (70). URL: https://cyberleninka.ru/article/n/osobennosti-protsessa-upravleniya-informatsionnymi-potokami-na-predpriyatii-v-sovremennyh-usloviyah.
8. Грейз Георгий Маркович, Кузменко Юлия Геннадьевна, Хатеев Игорь Валерьевич Анализ концепций взаимодействия основных логистических потоков // Российское предпринимательство. 2013. №5 (227). URL: https://cyberleninka.ru/article/n/analiz-kontseptsiy-vzaimodeystviya-osnovnyh-logisticheskih-potokov.
9. Евсеева А. О., Закирова М. И. Информация как ресурс экономики // Инновационная наука. 2017. №2-1. URL: https://cyberleninka.ru/article/n/informatsiya-kak-resurs-ekonomiki.



***Боровиков А.И., Криводубский О.А. Виды логистических потоков предприятия.*** *В статье рассматривается управление ресурсами предприятия через логистический подход, основанный на работе с основными потоками: материальными, финансовыми и информационными. Приведены определения и классификации потоков, которые позволяют структурировать их и эффективно управлять ими. Уточняется значимость материальных потоков в логистических процессах, подчеркивается важность управления финансовыми потоками для обеспечения устойчивости предприятия и акцентируется роль информационных потоков в поддержке процессов принятия решений.*



***Ключевые слова:*** *логистика, материальные потоки, финансовые потоки, информационные потоки, управление ресурсами, классификация потоков, оптимизация процессов, система принятия решений, эффективность предприятия, конкурентоспособность*



***Borovikov A.I., Krivodubskiy O.A. Types of enterprise logistics flows.*** *The article examines enterprise resource management through a logistics approach based on working with the main flows: material, financial and information. Definitions and classifications of flows are provided that allow structuring them and effectively managing them. The significance of material flows in logistics processes is clarified, the importance of managing financial flows to ensure the sustainability of the enterprise is emphasized, and the role of information flows in supporting decision-making processes is emphasized.*



***Keywords:*** *logistics, material flows, financial flows, information flows, resource management, flow classification, process optimization, decision-making system, enterprise efficiency, competitiveness*

# СЕКЦИЯ 6. «ИНЖЕНЕРИЯ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ»

УДК 004.414.2

## Организация АЛУ с учетом архитектурных особенностей и оптимизаций

**Д.В. Николаенко[*1], Я.И. Выростков[*2], Л.П. Володько[*3]**

[*1] к.т.н, доцент, Донецкий национальный технический университет,
dv.nikolaenko@yandex.ru, SPIN-код: 9819-4763

[*2] магистрант, Донецкий национальный технический университет,
yarchikbroo129@gmail.com

[*3] к.т.н, доцент, Полесский государственный университет, г. Пинск, Республика Беларусь

***Николаенко Д.В., Выростков Я.И., Володько Л.П. Организация АЛУ с учетом архитектурных особенностей и оптимизаций.*** *В статье рассмотрены подходы к организации АЛУ с учетом его функциональных и архитектурных особенностей. Описаны основные виды операций, выполняемых в АЛУ, включая операции над двумя операндами и операции сдвига. Особое внимание уделено оптимизации структуры АЛУ: сравниваются варианты использования отдельных схем для каждой операции, универсальной схемы и гибридного подхода, при котором операции разделяются на группы. Рассматривается роль сдвигов в алгоритмах умножения и деления, а также предлагается подход с выделением отдельного модуля для выполнения всех операций сдвига. Обсуждаются компромиссы между быстродействием, сложностью схемы и экономией аппаратных ресурсов, приводящие к выбору оптимальной архитектуры.*

***Ключевые слова:*** *арифметико-логическое устройство, архитектура вычислительных устройств, оптимизация схемы, гибридный подход, быстродействие, упрощение архитектуры, аппаратные ресурсы.*

### *Введение*

Для изучения принципов построения компьютерной архитектуры, особенностей функционирования отдельных элементов недостаточно продемонстрировать студентам детальные рисунки и даже подробное описание. Следует потратить время для непосредственной работы с вычислительной машиной, оснащенной пультом управления и различными индикаторами. Для данных нужд можно использовать либо эталонную машину [1], либо модель.

60-летний опыт кафедры компьютерной инженерии преподавания дисциплин, связанных с изучением проектирования аппаратных элементов вычислительных систем, показывает, что оптимальным способом организации процесса донесения и усвоения информации является применение специально разработанной виртуальной лаборатории [2-4]. Ее использование в учебном процессе удовлетворяет, с одной стороны, потребности преподавателей в средствах донесения информации и, с другой стороны, помогает студентам воспринимать новый материал, выполнять комплекс лабораторных и курсовых работ.

Арифметико-логическое устройство (АЛУ) является ключевым компонентом ядра вычислительных систем, обеспечивающим выполнение базовых арифметических и логических операций. От его архитектуры и организации зависят быстродействие, сложность реализации и эффективность работы всего устройства. В современной практике проектирования АЛУ часто требуется искать баланс между производительностью, экономией аппаратных ресурсов и универсальностью [5, 6].

Настоящая статья посвящена анализу различных подходов к организации АЛУ. Рассматриваются их преимущества и недостатки, а также приводятся рекомендации по выбору наиболее эффективной структуры для различных задач**.**

### *Основные операции в АЛУ*

Арифметико-логическое устройство (АЛУ) служит для выполнения арифметических и логических преобразований над данными, или же операндами. Все выполняемые в арифметико-логическом устройстве операции являются логическими операциями (функциями), которые можно разделить на следующие группы [5]:

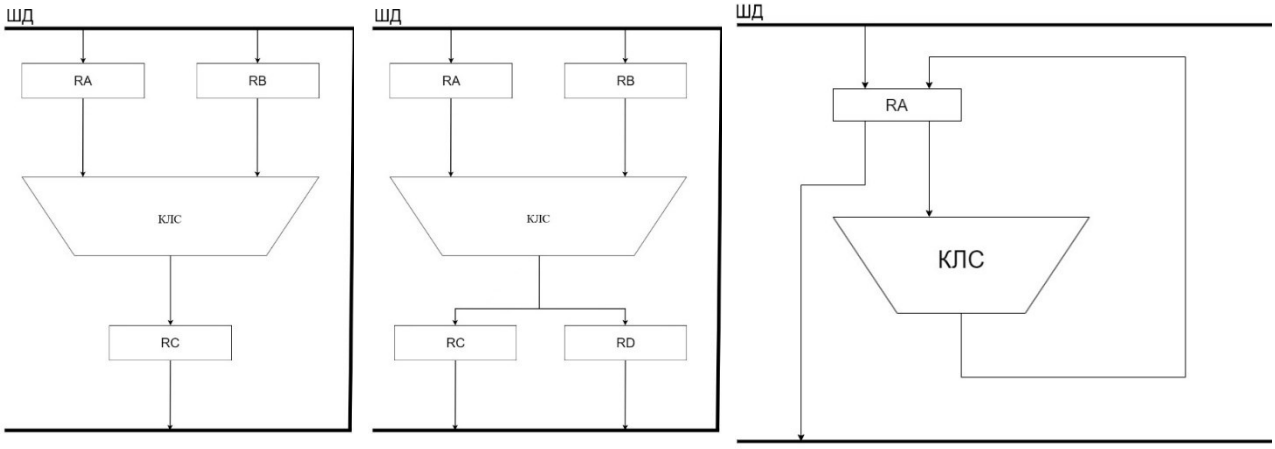
- операции двоичной арифметики для чисел с фиксированной точкой;
- операции двоичной арифметики для чисел с плавающей точкой;
- операции десятичной арифметики;
- операции специальной арифметики;
- логические операции;
- операции над алфавитно-цифровыми полями.

Современные компьютеры общего назначения обычно реализуют операции всех приведённых выше групп. К арифметическим операциям относятся сложение, вычитание, умножение и деление. Группу логических операций составляют операции дизъюнкция (логическое ИЛИ) и конъюнкция (логическое И) над операндами, сравнение кодов на равенство. АЛУ обладает набором операций, определяющим назначения устройства. Операции можно разделить на два вида:

- операции над двумя операндами – большинство арифметических и логических операций (сложение, вычитание, логические "И", "ИЛИ" и т.д.);

- операции над одним операндом – в основном сдвиги данных или инкремента/декремента.

Общая структура для операций над двумя операндами (рис. 1, а) содержит: регистры RA и RB для хранения исходных операндов; регистр результата RC и КЛС – комбинационно логическую схему, выполняющую определенную арифметическую или логическую операцию. Организация АЛУ для операции деления отличается наличием дополнительного регистра RD (рис. 1, б). Для операций с одним операндов достаточно использовать один регистр (рис. 1, в).




а) б) в)

Рисунок 1 – Структура АЛУ: а) для операций с двумя операндами; б) для операции деления; в) для операций с одним операндом

### *Подходы к реализации АЛУ*

В зависимости от архитектурных предпочтений, мы можем организовать выполнение операций в АЛУ одним из трех способов.

Способ 1 - Отдельные схемы для каждой операции. В этом случае для каждой операции из набора выделяется отдельный функциональный блок, например, сумматор, делитель и т.д.

*Преимущества:*

- лаконичность архитектуры каждого отдельного блока;
- высокая скорость выполнения операций;

*Недостатки:*

- увеличение количества элементов схемы;
- повышенная стоимость реализации и увеличение занимаемого пространства на плате;

Способ 2 - Единая универсальная схема. В этом случае все операции выполняются с использованием одной универсальной схемы.

*Преимущества:*

- значительное сокращение количества элементов;
- компактность схемы;

*Недостатки:*

- снижение быстродействия;
- более сложная алгоритмическая структура;
- повышенная нагрузка на управляющий автомат.

Способ 3 - Гибридный вариант. При его использовании операции распределяются по нескольким группам, каждая из которых реализуется в отдельном модуле. Например, логические операции реализуются в одном модуле, арифметические – в другом, умножение и деление – в третьем.

*Преимущества:*

- частичное устранение недостатков предыдущих подходов;
- сохранение высокой производительности;
- оптимизация структуры алгоритмов;
- умеренное количество элементов на схеме;

*Недостатки.*

Гибридный подход сочетает сильные стороны специализированных и универсальных схем, но также неизбежно теряет их ярко выраженные преимущества. Он не обеспечивает ни максимального быстродействия, свойственного специализированным схемам, ни значительной компактности и экономии ресурсов, характерных для универсального решения. В результате архитектура получается компромиссной, без явных недостатков ни в одном аспекте [7, 8].

### *Разработка методов для реализации моделей АЛУ*

Для реализации эмулятора АЛУ наиболее правильным кажется выбор объектно-ориентированного языка, т.к. все элементы можно описать как объекты, а затем настроить их соответствующее взаимодействие. Среди объектно-ориентированных языков довольно большой выбор – Java, C#, C++, Python и др. Каждый из них имеет ряд достоинств, а также устоявшиеся сферы применения. Так, C++ используют для высоконагруженных приложений и систем с высокими требованиями к производительности, C# применяют для разработки игр и приложений корпоративного сектора, Python обрел популярность в задачах машинного обучения и анализа данных. Одним из наиболее важных критериев является степень владения синтаксисом и стандартными библиотеками того или иного языка. По этому критерию в качестве языка программирования выбран C#. Среда программирования – традиционная Microsoft Visual Studio. C# предоставляет выбор платформы, на которой будет запускаться приложение. Среди других платформ выбрана .NET 8.0, в пользу которой говорит кроссплатформенность (персональный компьютер, веб-разработка, мобильные приложения) и долгосрочная поддержка платформы от производителя. Также в связке с C# базово поставляются несколько решений для проектирования пользовательского интерфейса – в частности, выбранное для текущего проекта – Windows Presentation Foundation. Выбор обусловлен актуальностью и гибким управлением размером элементов на форме.

В качестве примера операции над одним операндом приведена реализация метода Increment() (см. рис. 2).

```
Ссылок: 1
public void Increment()
{
    int TryOutput;
    if (!Int32.TryParse(ToStringBinFull(), System.Globalization.NumberStyles.BinaryNumber, null, out TryOutput))
        throw new Exception("ProgrammCounter: Недопустимые символы в строке счетчика команд!");
    TryOutput++;
    FromStringBinFull(Convert.ToString(TryOutput, 2).PadLeft(18, '0'));
}
```

Рисунок 2 – Реализация метода Increment()

В этом методе значение из памяти преобразуется в строковое, затем приводится к числу типа int, инкрементируется, обратно преобразуется в строковое, из которого перезаполняется память счетчика.

Схожая реализация выполнена и относительно метода Sum() класса EmulatorAccumulatorAdress. Оба слагаемых приводятся к целочисленному типу, выполняется сложение int, после чего результат помещается в память элемента (рис. 3).

```
Ссылок: 2
public void Sum(string Bin)
{
    if (Bin.Length > 32)
        throw new Exception("AdressAccumulator: Длина слагаемого превышает допустимую в аккумуляторе адреса!");
    if (Bin.Length < 32)
        Bin = Bin.PadLeft(32, '0');
    int iacum, iBin;
    if (!Int32.TryParse(ToStringBinFull(), System.Globalization.NumberStyles.BinaryNumber, null, out iacum))
        throw new Exception("AdressAccumulator: Недопустимые символы в строке аккумулятора!");
    if (!Int32.TryParse(Bin, System.Globalization.NumberStyles.BinaryNumber, null, out iBin))
        throw new Exception("AdressAccumulator: Недопустимые символы в строке регистра данных!");
    int rez = iacum + iBin;
    FromStringBin(Convert.ToString(rez, 2).PadLeft(32, '0'));
}
```

Рисунок 3 – Реализация метода Sum()

### *Разработка графического интерфейса эмулятора*

Модель АЛУ является одной из составляющих эмулятора процессорного устройства. Графический интерфейс эмулятора реализован по аналогии с отладчиками ассемблерного кода. Такие программы обычно разбиты на несколько областей, каждая из которых отображает текущую информацию о регистрах, памяти и т.д.

Главное окно эмулятора имеет вид, показанный на рис. 4. Окно разбито на 5 областей: меню, левая панель (область команд), правая панель (область состояния регистров и флагов), нижняя панель (область просмотра памяти).

Элементы меню позволяет выполнять некоторые сервисные действия. Например, подменю Файл содержит кнопки, позволяющие сохранять текущее состояние эмулятора в файл, и также загружать его в эмулятор. Отдельно можно сохранить и загрузить значение памяти, т.е. в отрыве от эмулятора можно написать исполняемую программу с использованием форматов команд процессорного устройства.

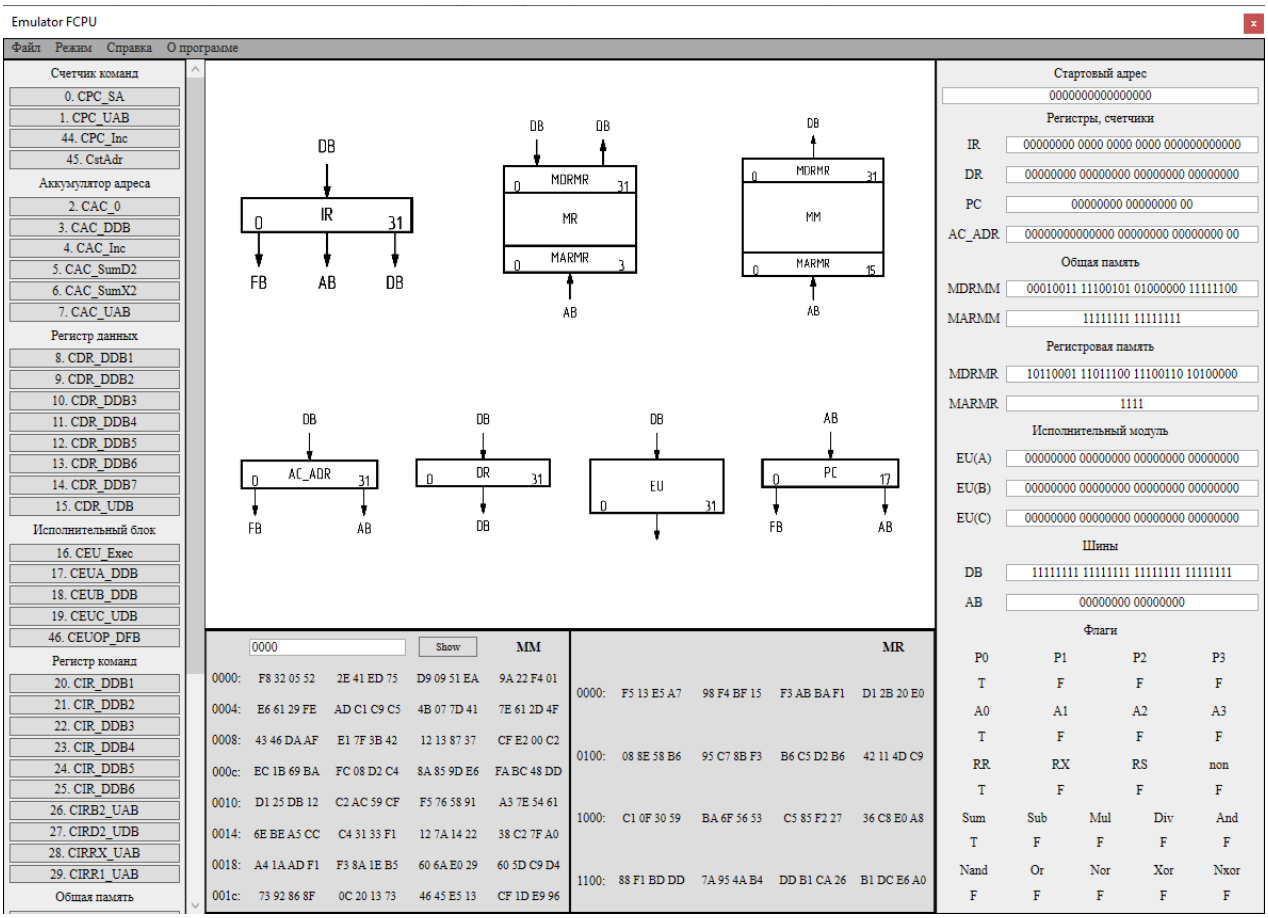



Рисунок 4 – Скриншот главного окна эмулятора

Подменю Режим позволяет выбрать ручной или автоматический режим. Автоматический режим запускает работу согласно граф-схеме алгоритма с некоторой задержкой, для отслеживания результатов. Его можно также назвать демонстрационным режимом. В этом режиме выполнение алгоритма можно поставить на паузу.

В ручном режиме предполагается самостоятельное исполнение микрокоманд путем нажатия на них в левой панели. На базе ручного режима можно построить выполнение лабораторной работы для отладки выведенного решения.

Подменю Справка позволяет дополнительно открыть окна со справочным материалом, которые содержат таблицу сигналов, флагов, полную функциональную схему и граф-схему алгоритма.

Пункт меню «О программе» выводит сообщение об авторе программы и ее назначении.

Левая панель – область команд, где можно применять определенную команду в ручном режиме.

### *Оптимизация работы сдвигов*

Сдвиги играют важную роль в вычислительных алгоритмах, особенно в умножении и делении, где они могут быть частью последовательности микрокоманд. Чтобы сократить сложность модулей умножения и деления, можно вынести сдвиги в отдельный специализированный модуль. В этом случае данные передаются в модуль сдвига, там обрабатываются и возвращаются с изменением в требуемом направлении.

Такой подход упрощает конструкции других модулей, делая их компактнее и более универсальными. Однако он вводит дополнительную задержку из-за необходимости трех шагов: передачи данных в модуль, выполнения сдвига и возврата результата. Это может немного сказаться на быстродействии, но позволяет существенно упростить общую архитектуру вычислительного устройства.

### *Выводы*

Проектирование АЛУ требует взвешенного подхода к выбору архитектуры, обеспечивающей баланс между производительностью, сложностью реализации и экономией аппаратных ресурсов.

Оптимизация работы АЛУ, включая вынесение операций сдвига в отдельный модуль, позволяет упростить архитектуру, однако сопровождается неизбежным снижением быстродействия. Итоговый выбор архитектуры должен основываться на конкретных требованиях системы, где приоритеты между производительностью, стоимостью и универсальностью будут определять оптимальное решение.

### *Литература*

1. Кириллов, В. В. Архитектура базовой ЭВМ. – СПб: СПбГУ ИТМО, 2010. – 144 с.

2. Мостовая, В. А. Разработка виртуальной лаборатории для изучения архитектуры процессора [Текст] / В. А. Мостовая, О. А. Авксентьева, Р. В. Мальчева // Материалы VI Международной научно-технической конференции «Современные информационные технологии в образовании и научных исследованиях» (СИТОНИ-2019). – Донецк: ДонНТУ, 2019. - С.375-379.

3. Мальчева, Р. В. Разработка виртуальной лаборатории для изучения и моделирования архитектур процессорных элементов [Текст] / Р. В. Мальчева, О. А. Авксеньева // Программная инженерия: методы и технологии разработки информационно- вычислительных систем (ПИИВС-2016): сборник научных трудов I научно-практической конференции. 16-17 ноября 2016 г. – Донецк: ДонНТУ, 2016. - С. 102-108.

4. Мальчева, Р. В. Моделирование внутренних операций процессорных элементов / Р. В. Мальчева, Т. В. Завадская // Информатика и кибернетика. - Донецк: ДонНТУ, 2016. - №3 (5). – С. 65-71.

5. Паттерсон, Д. Архитектура компьютера и проектирование компьютерных систем. / Д. Паттерсон, Дж. Хеннесси. - 4-е изд. – Санкт-Петербург: Питер, 2012. – 784 с.

6. Таненбаум, Э. Архитектура компьютера. / Э. Таненбаум, Т. Остин. - 6-е изд. – Санкт-Петербург: Питер, 2014. – 816 с.

7. Авксентьева, О. А. Разработка специализированного устройства на базе ПЛИС для реализации операции сложения и вычитания чисел с плавающей запятой / О. А. Авксентьева, Д. И. Выростков, Р. В. Мальчева // Информатика и кибернетика. – Донецк.: ДонНТУ. – 2020. – № 4(22). - С.66-72.

8. Мальчева, Р. В. FPGA-реализация векторно-матричных умножений / Р. В. Мальчева, А. И. Воронова, И. И. Дегтярева // Материалы XII Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ – 2021). - Донецк: ДонНТУ, 2021. - С.145-148.

***Николаенко Д.В., Выростков Я.И., Володько Л.П. Организация АЛУ с учетом архитектурных особенностей и оптимизаций.*** *В статье рассмотрены подходы к организации*

*АЛУ с учетом его функциональных и архитектурных особенностей. Описаны основные виды операций, выполняемых в АЛУ, включая операции над двумя операндами и операции сдвига. Особое внимание уделено оптимизации структуры АЛУ: сравниваются варианты использования отдельных схем для каждой операции, универсальной схемы и гибридного подхода, при котором операции разделяются на группы. Рассматривается роль сдвигов в алгоритмах умножения и деления, а также предлагается подход с выделением отдельного модуля для выполнения всех операций сдвига. Обсуждаются компромиссы между быстродействием, сложностью схемы и экономией аппаратных ресурсов, приводящие к выбору оптимальной архитектуры.*

***Ключевые слова:*** *арифметико-логическое устройство, архитектура вычислительных устройств, оптимизация схемы, гибридный подход, быстродействие, упрощение архитектуры, аппаратные ресурсы.*

***Nikolaenko Denis., Vyrostkov Yarosalv. ALU Organization Taking into Account Architectural Features and Optimizations.*** *The article considers approaches to organizing an arithmetic logic unit taking into account its functional and architectural features. The main types of operations performed in the ALU are described, including operations on two operands and shift operations. Particular attention is paid to optimizing the ALU structure: the options for using separate circuits for each operation, a universal circuit, and a hybrid approach in which operations are divided into groups are compared. The role of shifts in multiplication and division algorithms is considered, and an approach is proposed with the allocation of a separate module for performing all shift operations. Tradeoffs between performance, circuit complexity, and saving hardware resources are discussed, leading to the choice of the optimal architecture.*

***Keywords:*** *arithmetic logic unit, computing device architecture, circuit optimization, hybrid approach, performance, architecture simplification, hardware resources.*

УДК 004.8

# Основные элементы и составляющие искусственного интеллекта

**Воробьев А.Е.[*1], Корчевский А.Н. [*2], Воробьев К.А.*[3]**

[*1] д.т.н., профессор, Грозненский государственный нефтяной технический университет,
fogel_al@mail.ru, OrcID: 0000-0002-7324-428X, SPIN-код: 3457-6870
[* 2] к.т.н., доцент, Донецкий национальный технический университет,
korchevskyial@mail.ru, OrcID: 0000-0002-2247-7833, SPIN-код: 1293-7006
[*3] аспирант, Российский университет Дружбы Народов,
k.vorobyev98@mail.ru, SPIN-код: 8425-7290

***Воробьев А.Е., Корчевский А.Н., Воробьев К.А. Основные элементы и составляющие искусственного интеллекта.*** *Рассмотрены основные элементы и составляющие искусственного интеллекта. Представлена базовая основа для искусственного интеллекта в которую входят такие научные направления, как философия, математика, психология, экономика, лингвистика, нейронаука. Даны ключевые элементы искусственного интеллекта.*

***Ключевые слова:*** *искусственный интеллект, основа, дисциплины, элементы, составляющие.*

Многие современные научные дисциплины и направления создают базовую основу для искусственного интеллекта:

• Философия, базируемая на логике, а также основных методах рассуждения и разуме (как психо-физической системе).

• Математика – использующая формальное представление и доказательство, а также алгоритмы, вычисления, разрешимость и методы вероятности. Математика предлагает необходимые рабочие инструменты для обеспечения моделирования и решения возникающих задач оптимизации разнообразных процессов.

• Психология исследует феномены восприятия и двигательного контроля.

• Экономика – определяет формальную теорию рациональных решений.

• Лингвистика – обеспечивает необходимое представление знаний грамматики.

• Нейронаука – изучает, как работает человеческий мозг.

Кроме того, область искусственного интеллекта опирается на современную **информатику** (рис. 1). В частности, информатика предоставляет рабочие инструменты для последующего проектирования и построения необходимых алгоритмов.

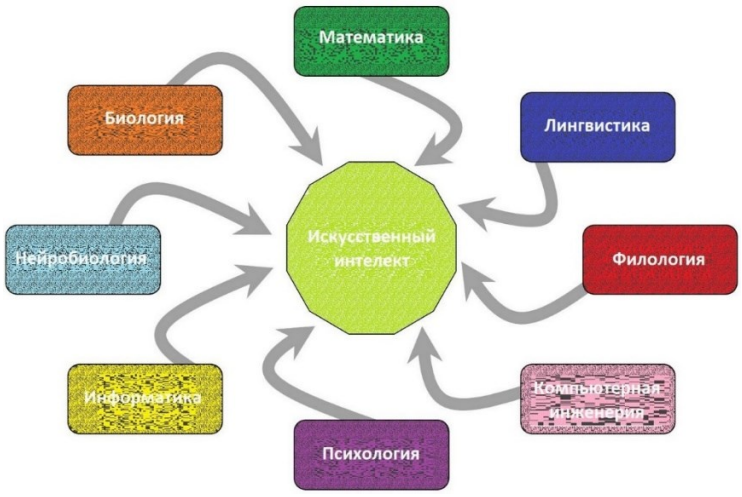



Рис. 1. Дисциплины, способствующие развитию искусственного интеллекта

Перечислим основные **ключевые элементы** искусственного интеллекта (табл. 1) и дадим им расшифровку [1-5].

Таблица 1 – Ключевые элементы искусственного интеллекта

| **Конструктивный элемент** | **Описание** |
|---|---|
| Теория | Тео́рия (греч. θεωρία «рассмотрение, исследование»):<br>— это упорядоченная и обоснованная система научных взглядов, суждений, положений, позволяющая адекватно объяснять факты, анализировать процессы, прогнозировать и регулировать их развитие;<br>— определенный уровень познания, на котором обобщаются и систематизируются имеющиеся знания о предмете исследования и формулируются основные понятия, категории, суждения и умозаключения |
| Нечеткая логика | Нечеткая логика — это форма многозначной логики, в которой истинностным значением переменных может быть любое действительное число от 0 до 1.<br>На основе этого понятия вводятся различные логические операции над нечёткими множествами и формулируется понятие лингвистической переменной, в качестве значений которой выступают преимущественно нечёткие множества |
| Механизм | Проявление действия причинно-следственных факторов |
| Машина | В сфере искусственного интеллекта слово «машина» может означать что угодно: от программного обеспечения, используемого в Интернете или в мобильном приложении, до робота-гуманоида |
| Интеллект | 1. Это способность думать и понимать, а не делать что-то инстинктивно или автоматически.<br>2. Это способность понимать и изучать какие-либо вещи, обстоятельства, особенности |
| Мышление | Это использование мозга для рассмотрения проблемы или создания какой-либо новой идеи |
| Данные обучения | Это набор данных, который используется для создания моделей, которые помогают обучать компьютер |
| Нейроны | Электрически возбудимая клетка, которая предназначена для приёма извне, обработки, хранения, передачи и вывода вовне информации с помощью электрических и химических сигналов |
| Персептрон | Перцептро́н (англ. perceptron от лат. perceptio — восприятие; нем. perzeptron) — математическая или компьютерная модель восприятия информации мозгом (кибернетическая модель мозга), предложенная Ф. Розенблаттом в 1957 г. и впервые реализованная в виде электронной машины «Марк-1» в 1960 г.<br>Перцептрон состоит из 3-х типов элементов, а именно: поступающие от датчиков сигналы передаются ассоциативным элементам, а затем — реагирующим элементам. Таким образом, перцептроны позволяют создать набор «ассоциаций» между входными стимулами и необходимой реакцией на выходе. В биологическом плане это соответствует преобразованию, например, зрительной информации в физиологический ответ от двигательных нейронов.<br>Согласно современной терминологии, перцептроны могут быть классифицированы как искусственные нейронные сети:<br>- с одним скрытым слоем;<br>- с пороговой передаточной функцией;<br>- с прямым распространением сигнала |
| Алгоритм | Система последовательных операций (в соответствии с определёнными правилами) для решения какой-нибудь задачи |
| Обучение на основе алгоритмов | Методы, используемые для создания соответствующих моделей, которые лучше всего представляет данные обучения |
| Экспертная система | Компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации |
| Модель | Математическая конструкция, помогающая прогнозировать новые значения или принимать решения |
| Классификация | Подразделение объектов по существенным признакам |

Термин «**интеллект**» относится к способности приобретать и применять различные навыки и знания для решения определенной проблемы [7]. Кроме того, интеллект также связан с использованием общих умственных способностей для решения, рассуждения и изучения различных ситуаций.

Необходимо отметить, что искусственный интеллект представляет собой комбинацию различных технологий, которые дают робототехнике возможность понимать, изучать, воспринимать или выполнять человеческую деятельность самостоятельно, автономно.

В настоящее время существуют различные инновации искусственного интеллекта, например, автомобили-роботы, которым не требуется, чтобы их контролировал водитель. Кроме того, технологии искусственного интеллекта включают в себя умные машины, которые способны обрабатывать большие объемы данных, которые человек выполнить, как правило, не может.

Обычно, интеллект интегрирован с различными когнитивными функциями (такими, как язык, внимание, планирование, память, восприятие и др.).

**Теория** – это попытка дать научное объяснение и, таким образом, получить основу для последующего изучения конкретного явления. Примерами являются теории разума, речи, зрения, мозга, интеллекта и т.д.

**Алгоритм искусственного интеллекта** — это набор инструкций или правил, которые система искусственного интеллекта использует для выполнения конкретной задачи или решения стоящей проблемы. При этом алгоритмы искусственного интеллекта можно разделить на 2 большие категории: символические (основанные на четких правилах) алгоритмы и числовые (статистические) алгоритмы.

Символьные алгоритмы используют правила и логические рассуждения для решения имеющихся проблем, а числовые алгоритмы используют математические и статистические методы для анализа и обработки собранных данных.

Алгоритмы искусственного интеллекта также можно классифицировать по типу обучения, которое они используют, например, обучение с учителем, без учителя или обучение с подкреплением.

Выбор того или иного алгоритма зависит от характера решаемой задачи и типа данных, доступных для машинного обучения.

**Нейрон** — это основная единица нервной системы живого организма, и считается, что посредством сети нейронов люди думают и хранят воспоминания. В человеческом мозге имеется около 100 трлн. электропроводящих клеток или нейронов [7], которые обеспечивают невероятную вычислительную мощность для быстрого и эффективного выполнения задач.

Нейрон состоит из трех отличительных частей (рис. 2): дендритов, сомов и аксона.

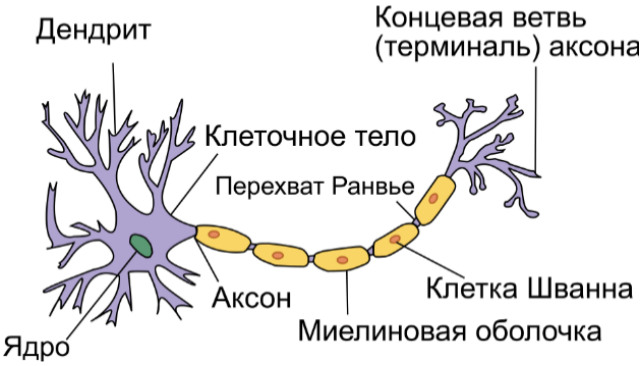



Рисунок 2 - Структура нейрона

Нейрон получает несколько сигналов от своих входных связей, вычисляет новый уровень активации и отправляет его в качестве выходного сигнала через выходные связи [8]. Входным сигналом могут быть необработанные данные или выходные данные других нейронов. Выходной сигнал может быть либо окончательным решением стоящей проблемы, либо входным сигналом для других нейронов.

Сигналы передаются от одного нейрона к другому посредством сложных электрохимических реакций. Химические вещества, выделяющиеся из синапсов, вызывают изменение электрического потенциала тела клетки. Когда потенциал достигает своего порога, возникает электрический импульс (потенциал действия), который отправляется вниз по аксону. Импульс распространяется и в конечном итоге достигает синапсов, заставляя их увеличивать или уменьшать свой потенциал.

Нейроны соединены связями, и каждая ссылка имеет связанный с ней числовой вес [6]. Нейронная сеть «обучается» путем многократной корректировки этих весов.

Для этого нейрон вычисляет взвешенную сумму входных сигналов и сравнивает результат с пороговым значением [8]. Если чистый вход меньше порога, то выход нейрона равен -1. Но если чистый входной сигнал

больше или равен порогу, то нейрон вновь активируется.

Таким образом, нейроны — это своего рода процессор сообщений, которые в дальнейшем в виде электрических импульсов поступают в дендриты от нескольких других нейронов. Затем сообщение обрабатывается сомой нейрона. Эта обработка «решает»: должен ли нейрон «срабатывать» или нет.

Следующей важной вехой стало открытие в 1958 г. психологом Фрэнком Розенблаттом **персептрона** [6]. Было предложено множество вариантов базовой сети персептрона, но наиболее популярной моделью был многослойный персептрон с прямой связью.

**Нейронная сеть** была разработана Маккалок и Уолтером Питтсом, которые предложили модель искусственных нейронных сетей, в которой постулировалось, что каждый нейрон находится в бинарном (т.е. в включенном и выключенном) состояниях. Однако, в дальнейшем эксперименты ясно показали, что бинарная модель нейронов неверна. Фактически, нейрон имеет сильно нелинейные характеристики и не должен рассматриваться только как простое устройство, лишь с двумя противоположными состояниями.

Эти сети состоят из слоев нейронов, обычно входного слоя, одного или нескольких средних или скрытых слоев и выходного слоя [6], каждый из которых полностью связан с другим слоем.

Использование многослойного персептрона с прямой связью было ограничено из-за отсутствия подходящего алгоритма обучения до тех пор, пока аспирант Пол Вербос в 1974 г., не представил обучение с «обратным распространением ошибки».

**«Глубокие» нейронные сети** — это те, которые имеют несколько «скрытых слоев» между входным и выходным слоем. Нейронные сети могут самостоятельно учиться, регулируя вес связей, имеющихся между каждым нейроном, чтобы оптимизировать пути продвижения сигнала через сеть для достижения определенного результата. Некоторые глубокие нейронные сети, используемые для распознавания изображений, могут иметь свыше сотни тысяч искусственных нейронов и еще больше связей между ними.

**Экспертная система** — это интерактивная и надежная компьютерная система принятия эффективных решений, которая использует определенные выявленные факты и закономерности эвристики, служащие для решения весьма непростых задач. Основная цель экспертной системы заключается в решении наиболее сложных задач в конкретной области.

**Нечеткая логика** — это теория нечетких множеств, которые калибруют существующую в решаемой задаче неопределенность. Нечеткая логика основана на идее, что все объекты имеют степени. Это концепция дескриптора частичной истины.

Нечеткая логика определяется как многозначная логическая форма, которая может иметь значения истинности переменных в любом действительном числе от 0 до 1.

Нечеткая логика отражает то, как люди думают. Она пытается смоделировать наше чувство слов, наше принятие тех или иных решений и наш здравый смысл. В реальной жизни пользователь может столкнуться с такой ситуацией, когда невозможно решить, истинно или ложно то или иное утверждение.

**Фрейм** — это структура данных с типичными знаниями о конкретном объекте или понятии, которая обеспечивает естественный способ структурированного и краткого представления знаний.

Понятие фрейма определяется совокупностью слотов (значение которых может быть символическим, числовым или логическим). Каждый слот описывает конкретный атрибут или операцию кадра.

Необходимо отметить, что **база данных** ImageNet, используемая для обучения классификаторов изображений, содержит более 14 млн. помеченных изображений, и организаторы поставили цель — 1000 изображений на каждую категорию изображений.

**Платформы глубокого обучения** используют нейронные сети со слоями абстракции, пытаясь имитировать работу человеческого мозга. Они предназначены для обработки многочисленных данных и создания шаблонов, на основе которых можно принимать конструктивные решения. В настоящее время они используются для выявления закономерностей и категоризации приложений, совместимых только с крупномасштабными наборами данных.

**Облако** — это термин, обозначающий доступ к большим данным и программным приложениям через сетевое соединение удаленных серверов, часто представляющие крупные центры обработки данных, использующих подключение к Интернету. В облаке могут работать практически все программные ресурсы: определенная программа или приложение, служба или даже другой смоделированный компьютер.

Примерами облачных систем являются OneDrive и Office 365.

**Облачные вычисления** — это система использования надежной сети удаленных серверов для хранения, управления и обработки данных. Она работает через сеть компьютеров, подключенных друг к другу и к пользователю через Интернет, а не через локальный сервер или отдельный персональный компьютер.

Система облачных вычислений, не распределяет копии одних и тех же файлов данных на отдельные клиентские устройства, а хранит свои критически важные данные на удаленных серверах.

**Интерфейс прикладного программного обеспечения** (API) — это набор подпрограмм, протоколов и инструментов для создания программных приложений. Цель внедрения API — помочь разработчикам создавать

интеллектуальные приложения, не имея прямых навыков или знаний в области искусственного интеллекта или науки о данных.

Например, цель Microsoft Azure Cognitive Services — помочь разработчикам создавать приложения, которые могут видеть, слышать, говорить, понимать и даже начинать логически рассуждать. Это позволяет организациям внедрять инновации и преобразования во всех существующих у них процессах и технологиях.

**Виртуальный агент** — это компьютерная программа, предназначенная для взаимодействия с людьми, которая работает в чат-ботах, а также при использовании некоторых приложений или посещении некоторых веб-сайтов.

Веб-приложения и мобильные приложения предоставляют чат-боты в качестве агентов по обслуживанию клиентов, которые взаимодействуют с людьми и отвечают на их запросы. Таким образом, виртуальные агенты также действуют как программное обеспечение, т.е. как услуга.

От контакт-центров до обслуживания различных бизнес-задач (таких, как ИТ-поддержка) виртуальные агенты выполняют не только базовые функции, но также могут обрабатывать сложные запросы, основанные на различных, довольно не простых, ситуациях.

**Виртуальный помощник** также действует как языковой помощник, который выбирает подсказки по выбору и предпочтениям пользователя. Например, виртуальный помощник IBM Watson понимает типичные запросы в службу поддержки клиентов, которые задаются несколькими способами.

**Биометрия** используется для идентификации, измерения и анализа физических аспектов структуры и формы тела. Это позволяет технологиям использовать более естественные формы взаимодействия между людьми и машинами, включая прикосновение, отпечатки пальцев, походку, речь, распознавание глаз и языка тела. Её часто используют как способ подтверждения личности или идентификации команды (например, махнув рукой Xbox или приложив палец к сканеру).

### *Литература*


1. Воробьев А.Е., Тчаро Х., Воробьев К.А. Цифровизация нефтяной промышленности: базовые подходы и обоснование "интеллектуальных" технологий // Вестник евразийской науки. 2018. Т. 10. № 2. С. 77.
2. Воробьев А.Е., Абишев А.А. Технология «умных скважин» // Вестник АИНГ (Казахстан). N 3 (39). 2016. С. 3-11.
3. Воробьев А.Е., Viktor Marenga. Концепция развития горнопромышленного «Умного города» // Горный вестник Узбекистана. № 4 (87). 2021. С. 106-109.
4. Воробьев А.Е., Тчаро Х., Воробьев К.А. Цифровизация нефтяной промышленности: «интеллектуальный» нефтепромысел // Вестник Евразийской науки №3 (май — июнь), Том 10. Науки о Земле. 2018. Идентификационный номер статьи в журнале: 77NZVN318
5. Воробьѐв А.Е., Торобеков Б.Т. Концепция «Умный город 4.0» // Известия Кыргызского государственного технического университета им. И. Раззакова. N 4 (60). 2021. С. 164-170.
6. Chandra Kambhampati. Artificial intelligence in medicine // Ann R Coll Surg Engl. 2014. Pp. 334–338. https://www.researchgate.net/publication/8379547. DOI: 10.1308/147870804290.
7. Jahanzaib Shabbir, Tarique Anwer. Artificial intelligence and its role in near future // Journal of latex class files. Vol. 14. N0. 8. 2015.
8. Michael Negnevitsky. Artificial Intelligence. A Guide to Intelligent Systems. 2004.



***Воробьев А.Е., Корчевский А.Н., Воробьев К.А. Основные элементы и составляющие искусственного интеллекта в системах цифровизации нефтяной промышленности.*** *Рассмотрены основные элементы и составляющие искусственного интеллекта. Представлена базовая основа для искусственного интеллекта в которую входят такие научные направления, как философия, математика, психология, экономика, лингвистика, нейронаука. Даны ключевые элементы искусственного интеллекта.*



***Ключевые слова:*** *искусственный интеллект, основа, дисциплины, элементы, составляющие.*



***Vorobyov A.E., Korchevsky A.N., Vorobyov K.A.*** *The main elements and components of artificial intelligence in the digitalization systems of the oil industry. The main elements and components of artificial intelligence are considered. The basic framework for artificial intelligence is presented, which includes such scientific fields as philosophy, mathematics, psychology, economics, linguistics, and neuroscience. The key elements of artificial intelligence are given.*



***Keywords:*** *artificial intelligence, foundation, disciplines, elements, components.*